

RANCANG BANGUN SISTEM ROTARY DRUM FILTER BERBASIS INTERNET OF THINGS (IOT)

Tugas Akhir

diajukan untuk memenuhi salah satu syarat

memperoleh gelar sarjana

dari Program Studi Teknologi Informasi (Kampus Kota Surabaya)

Fakultas Informatika

Universitas Telkom

1202202066

DAVID ALBUSTOMMY



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

Fakultas Informatika

Universitas Telkom

Surabaya

2024

LEMBAR PENGESAHAN

**RANCANG BANGUN SISTEM ROTARY DRUM FILTER BERBASIS
INTERNET OF THINGS (IOT)**

INTERNET OF THINGS (IOT) BASED ROTARY DRUM FILTER SYSTEM DESIGN

NIM : 1202202066

DAVID ALBUSTOMMY

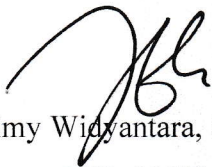
Tugas akhir ini telah diterima dan disahkan untuk memenuhi sebagian syarat memperoleh gelar pada Program Studi Sarjana Teknologi Informasi (Kampus Kota Surabaya)

Fakultas Informatika
Universitas Telkom

Surabaya, 13/08/2024

Menyetujui

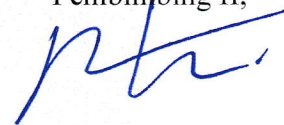
Pembimbing I,



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

NIP. 19790001

Pembimbing II,



Philip Tobianto Daely, S.T., M.Eng., Ph.D.

NIP.19940002

Ketua Program Studi
Sarjana Teknologi Informasi,



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

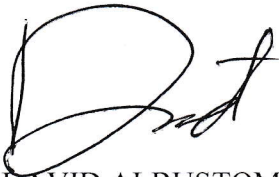
NIP: 23929009

LEMBAR PERNYATAAN

Dengan ini saya, DAVID ALBUSTOMMY, menyatakan sesungguhnya bahwa Tugas Akhir saya dengan judul RANCANG BANGUN SISTEM ROTARY DRUM FILTER BERBASIS INTERNET OF THINGS (IOT) 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, 13/08/2024

Yang Menyatakan



DAVID ALBUSTOMMY

RANCANG BANGUN SISTEM ROTARY DRUM FILTER BERBASIS INTERNET OF THINGS (IOT)

David Albustommy^{1,3}, Helmy Widyantara², Philip Tobianto Daely³

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

¹davidalbustommy@student.telkomuniversity.ac.id, ²helmywidyantara@telkomuniversity.ac.id,

³philipdaely@telkomuniversity.ac.id

Abstrak

Kualitas air memiliki peranan yang penting untuk kehidupan pada ikan. Khususnya pada kolam ikan nila di rooftop kampus Telkom University Surabaya yang perlu dipertimbangkan untuk mendukung keberhasilan budidaya dan pertumbuhannya. Namun, metode yang digunakan di rooftop kampus Telkom University Surabaya masih menggunakan manual untuk melakukan filtrasi kolam dan menjadikan kurang efektif. Oleh karena itu, dibuatlah sistem *rotary drum filter* dengan menggunakan logika fuzzy agar dapat menjadikan otomatis saat melakukan filtrasi. Dengan menambahkan monitoring dan kontroling menggunakan aplikasi blynk agar dapat memantau dan mengontrol filter dari manapun secara *real-time* agar lebih efisien. Penelitian sistem ini diharapkan mampu memberikan efisiensi filtrasi yang tinggi dalam menyaring partikel dan kotoran dari air. Hal ini akan meningkatkan kualitas air secara signifikan dan mendukung kelangsungan hidup ikan atau organisme lain yang dibudidayakan. Pada pengujian menunjukkan bahwa sistem telah berhasil mengolah data menggunakan metode fuzzy mamdani dan mendapatkan hasil akurasi dengan rata-rata akurat yaitu 85.7% dengan galat error 14.3%.

Kata kunci : Rancang Bangun, RDF Filter, *Internet of Things*, Logika Fuzzy, Kolam Ikan Nila.

Abstract

Water quality has an important role in the life of fish. Especially the tilapia fish pond on the rooftop of the Telkom University Surabaya campus which needs to be considered to support successful cultivation and growth. However, the method used on the rooftop of the Telkom University Surabaya campus still uses a manual to filter the pool, which makes it less effective. Therefore, a rotary drum filter system was created using fuzzy logic to make it automatic when carrying out filtration. By adding monitoring and control using the Blynk application you can monitor and control the filter from anywhere in real-time to make it more efficient. This system research is expected to provide high filtration efficiency in filtering particles and dirt from water. This will significantly improve water quality and support the survival of fish or other farmed organisms. The test shows that the system has succeeded in processing data using the fuzzy Mamdani method and obtained accuracy results with an average accuracy of 85.7% with an error of 14.3%.

Keywords: Design, RDF Filter, *Internet of Things*, Fuzzy Logic, Tilapia Fish Pond.

1. Pendahuluan

1.1 Latar Belakang

Kualitas air memiliki peranan yang penting untuk kehidupan pada ikan. Khususnya pada kolam ikan nila di rooftop kampus Telkom University Surabaya yang perlu dipertimbangkan untuk mendukung keberhasilan budidaya dan pertumbuhannya. Jika kualitas pada air kolam menurun, maka diperlukan pengurusan pada kolam untuk menjaga kelangsungan hidup pada ikan nila [1]. Penelitian ini menggunakan *Internet of Things* untuk melakukan monitoring air kolam secara *real-time* agar dapat lebih efisien [2]. Faktor yang perlu diperhatikan untuk kualitas air kolam yang baik meliputi tingkat keasaman (pH) air dan tingkat kejernihan air kolam. Dengan menggunakan sistem *Rotary Drum Filter* dalam melakukan filtrasi memungkinkan penyaringan dapat lebih efisien [3]. Pada penelitian ini menambahkan *fuzzy logic* dengan menggunakan metode mamdani agar dapat mengendalikan filter air kolam menjadi otomatis [4].

Pada budidaya ikan nila, kualitas air kolam memegang peran penting untuk perkembangan ikan nila. Kondisi pada air kolam yang optimal berada pada tingkat kekeruhan <300 NTU dan tingkat keasaman (pH) air kolam 7-8 [5].

Penelitian ini menerapkan monitoring untuk pemantauan kualitas air dan pH menggunakan aplikasi blynk. Dengan menggunakan aplikasi ini maka informasi terkait kualitas kekeruhan air dan kondisi pH dapat dilihat secara *real-time* [6]. Pada sistem penelitian dibuat menggunakan NodeMCU ESP32 sebagai mikrokontroler, Sensor turbidity digunakan untuk mengukur tingkat kekeruhan air kolam, sementara sensor pH digunakan untuk menentukan tingkat keasaman. Adapun komponen pendukung lainnya yaitu relay, water pump, motor dc,

motor driver, dan adaptor [7]. Penelitian ini mengintegrasikan teknologi *Internet of Things* dengan menggunakan NodeMCU ESP32 dan metode *fuzzy logic* untuk menentukan parameter air agar dapat dilakukan secara otomatis. Pada penelitian sistem ini diharapkan mampu memberikan efisiensi filtrasi yang tinggi dalam menyaring partikel dan kotoran dari air. Hal ini akan meningkatkan kualitas air secara signifikan dan mendukung kelangsungan hidup ikan atau organisme lain yang dibudidayakan.

1.2 Rumusan Masalah

Berdasarkan latar belakang penelitian, permasalahan yang harus diselesaikan adalah bagaimana cara membangun *Rotary Drum Filter* berbasis *Internet of Things* agar proses filtrasi ikan nila menjadi efisien.

1.3 Tujuan

Dapat melakukan proses filtrasi pada kolam ikan nila yang efisien dengan menggunakan *Rotary Drum Filter* berbasis *Internet of Things* dengan melihat kualitas pada air menggunakan aplikasi blynk untuk monitoring.

1.4 Batasan Masalah

Batasan masalah pada penelitian ini antara lain:

1. Penelitian ini hanya berfokus pada proses filtrasi di kolam ikan nila.
2. Sistem hanya dapat melakukan monitoring pada aplikasi blynk.
3. Sensor pH hanya dapat melakukan monitoring pH air pada kolam.
4. Penelitian ini berfokus pada kolam di rooftop kampus Telkom University Surabaya.

2. Studi Terkait

2.1 Penelitian Terdahulu

Penelitian yang telah dilakukan sebelumnya sangat penting untuk peneliti yang memiliki topik terkait dapat membantu penulis dalam pelaksanaan penelitian. Beberapa penelitian terkait dapat digunakan penulis sebagai bahan referensi.

Penelitian pertama, melakukan filtrasi untuk menjaga kejernihan air. Filtrasi dengan menggunakan sistem *Rotary Drum Filter* ini dibangun menggunakan beberapa komponen seperti sensor turbidity, ph dan suhu DS18B20 untuk mengetahui kualitas air. Dan juga menggunakan mikrokontroler Arduino UNO R3 Built-in IOT WiFi ESP8266 untuk memproses hasil nilai dari ketiga sensor itu [1].

Penelitian kedua, bertujuan untuk mengembangkan sistem pemantauan kualitas air yang dapat mengukur parameter fisik, kimia, dan mikrobiologis dari air secara real-time menggunakan teknologi IoT. Penelitian ini juga bertujuan untuk menciptakan sistem yang hemat biaya, sederhana, dan efisien dalam memantau kualitas air, serta memberikan notifikasi kepada pihak berwenang untuk tindakan lebih lanjut [2].

Penelitian ketiga, merancang sistem otomatisasi untuk pengendalian suhu dan pemantauan pH. Dalam penelitian ini, digunakan sensor DS18B20 untuk mengukur suhu dan pH. Mikrokontroler yang digunakan Nodemcu ESP8266. Pengujian pada alat ini melalui penjadwalan lampu pada Aquascape secara real-time berkala dan dapat juga melakukan penyesuaian lampu Aquascape selama 8 jam perhari [3].

Penelitian keempat, membahas tentang metode mamdani digunakan untuk mengendalikan sistem. Logika fuzzy memungkinkan sistem menangani ketidakpastian dan membuat keputusan yang lebih fleksibel dibandingkan dengan logika biner tradisional [4].

Penelitian kelima, bertujuan untuk memeriksa kualitas air dari segi pH dan kekeruhan dengan memanfaatkan teknologi *Internet of Things*. Metode ini mempercepat proses pengumpulan data, yang kemudian dikirimkan secara otomatis melalui aplikasi Blynk. Penelitian ini berhasil menunjukkan bahwa pengukuran kekeruhan dan pH air dilakukan secara otomatis dan akurat [6].

2.2 Dasar Teori

2.2.1 Ikan Nila

Ikan nila adalah jenis ikan yang memiliki potensi besar dan dapat bertahan dengan baik terhadap perubahan kondisi lingkungan perairan. Dalam budidaya ikan nila, penting untuk memperhatikan kejernihan air kolam guna menghindari kekeruhan [8].

2.2.2 *Internet of Things*

Internet of Things adalah sebuah konsep teknologi yang memungkinkan perangkat elektronik terhubung ke internet secara terus-menerus. Dengan menggunakan *Internet of Things*, perangkat akan dapat saling berinteraksi untuk mendapatkan informasi dari perangkat lainnya melalui jaringan. *Internet of Things* dapat berinteraksi antara mesin tanpa campur tangan pengguna dan berapapun jaraknya. Jaringan menjadi penghubung antara keduanya, sementara pengguna bertugas sebagai monitoring dan kontrol saat alatbekerja. Sistem dasar yang dibutuhkan adalah hardware, koneksi jaringan, cloud data center [9].

2.2.3 NodeMCU ESP32

NodeMCU ESP32 adalah mikrokontroler yang diluncurkan oleh *Espressif Systems* sebagai penerus dari ESP8266. Dibandingkan dengan mikrokontroler lainnya, ESP32 memiliki keunggulan seperti jumlah pin output yang lebih banyak, lebih banyak pin analog, kapasitas memori yang lebih besar, serta dilengkapi dengan *Bluetooth 4.0 Low Energy* dan WiFi, yang membuatnya ideal untuk aplikasi *Internet of Things* [10].

2.2.4 Sensor Turbidity

Sensor turbidity adalah perangkat yang digunakan untuk mengukur kualitas air kolam. Pada sensor ini, semakin tinggi tingkat kekeruhan air, maka tegangan yang ditampilkan pada sensor juga akan mengalami perubahan [11].

2.2.5 Sensor pH

Sensor pH adalah alat untuk melakukan pengukuran pada derajat keasaman atau kebasahan dari suatu larutan. Sensor ini sangat penting untuk melakukan pengukuran pada pH [12].

2.2.6 Water pump

Water pump merupakan alat yang mampu mengeluarkan air dengan cara yang lebih cepat dan efisien dibandingkan metode manual menggunakan ember. Jumlah air yang dipompa berpengaruh besar pada daya yang dibutuhkan untuk menjalankan pompa air, terlepas dari hasil akhirnya [13].

2.2.7 Motor DC

Motor DC adalah perangkat yang mengkonversi energi listrik menjadi energi kinetik atau gerakan, juga dikenal sebagai Motor Arus Searah. Motor ini menghasilkan putaran per menit (RPM) dan dapat berputar searah atau berlawanan arah jarum jam dengan membalik polaritas listrik. Prinsip kerjanya menggunakan fenomena elektromagnet, di mana arus listrik pada kumparan menyebabkan sisi utara dan selatan kumparan bergerak menuju kutub berlawanan pada magnet, menghasilkan gerakan hingga kumparan berhenti saat kutub bertemu [14].

2.2.8 *Fuzzy Logic*

Logika fuzzy adalah bentuk logika yang memiliki nilai yang tidak sepenuhnya benar atau salah. Tingkat kebenaran dan kesalahan ditentukan oleh derajat keanggotaan yang ada. Metode Mamdani, atau yang dikenal juga sebagai Metode Max-Min, adalah salah satu pendekatan penalaran dalam Logika Fuzzy [15].

2.2.9 MATLAB

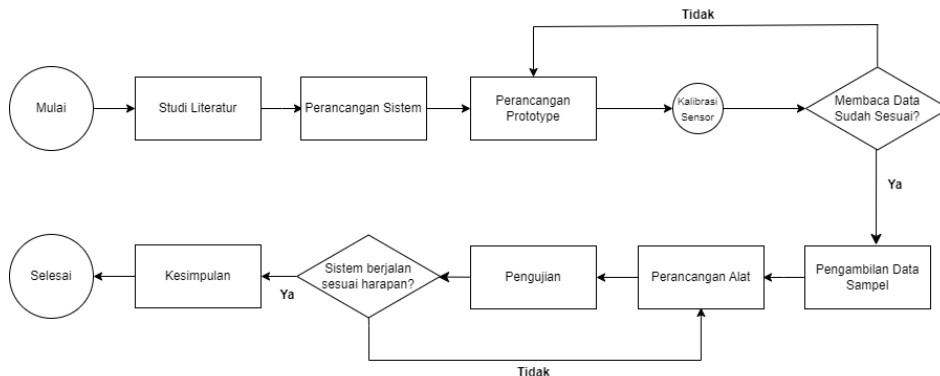
MATLAB merupakan bahasa pemrograman standar untuk matematika, teknik, dan ilmu terkait, yang menawarkan berbagai opsi untuk visualisasi dan pemrograman. MATLAB memudahkan pengkodean rumit dengan menyediakan fungsi GUI untuk memproses citra, menjadikannya alat yang andal untuk pengembang program [16].

2.2.10 Blynk

Blynk adalah aplikasi yang tersedia untuk iOS dan Android yang memungkinkan pengguna mengendalikan perangkat seperti Arduino dan Raspberry Pi melalui internet. Dirancang untuk *Internet of Things*, Blynk memungkinkan kontrol perangkat dari jarak jauh, menampilkan data sensor, menyimpan data, serta menjalankan berbagai fungsi canggih lainnya. Platform ini terdiri dari tiga komponen yaitu Blynk App, Blynk Server, dan Blynk Library [17].

3. Sistem yang Dibangun

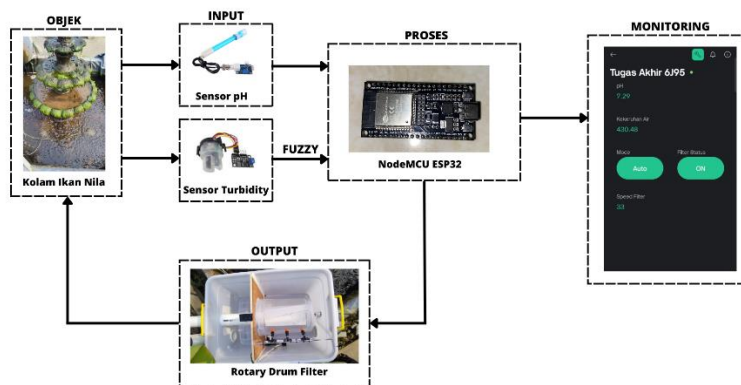
3.1 Prosedur Penelitian



Gambar 3. 1 Prosedur Penelitian

Pada prosedur penelitian ini dilakukan beberapa tahapan yang mencakup studi literature, perancangan sistem, perancangan prototype, implementasi prototype, pengambilan data sampel, implementasi sistem monitoring dan kendali, perancangan alat, pengujian, kesimpulan. Study literature dilakukan untuk mencari dan memahami jurnal, artikel, buku terkait pengolahan air kolam ikan nila, teknologi *Internet of Things*, dan penerapan *fuzzy logic*. Pada perancangan sistem terdapat penyusunan sketsa pada komponen yang digabungkan menjadi satu menggunakan diagram alur, pada diagram alur mencakup beberapa tahapan yaitu meliputi input, proses, pemantauan, dan output. Penelitian ini menggunakan kualitatif untuk mendapatkan data dengan melakukan kalibrasi sensor dan observasi tempat. Selanjutnya, pada perancangan prototype dilakukan dengan menggunakan NodeMCU ESP32, sensor turbidity, dan sensor pH. Setelah melakukan perancangan dan implementasi pada prototype, sensor turbidity dan pH akan mengirimkan data output sensor dan dilanjutkan untuk implementasi sistem monitoring dan kendali secara *real-time* melalui aplikasi blynk.

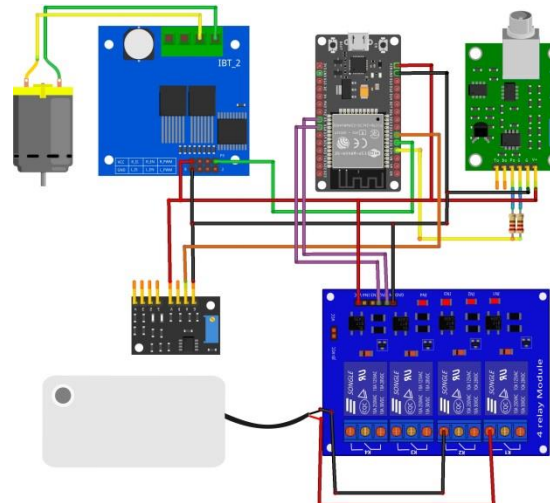
3.2 Metode yang digunakan



Gambar 3. 2 Alur Diagram

Diagram alur dalam penelitian ini dapat dilihat pada Gambar 3.2. Pada sistem ini dapat melakukan monitoring dan kontrol melalui aplikasi blynk. Sistem monitoring dilakukan agar dapat memantau kadar kekeruhan dan pH air pada kolam. Sedangkan sistem otomatis dilakukan untuk pengendalian *Rotary Drum Filter*. Terdapat beberapa sensor yang digunakan untuk penelitian pada air kolam, yaitu sensor turbidity dan sensor pH. Sensor turbidity digunakan untuk mengukur tingkat kekeruhan, sementara sensor pH mengukur kadar pH dalam air kolam nila. Kedua sensor ini terhubung dengan NodeMCU ESP32. Hasil dari data sensor akan diteruskan pada tahapan dimana seluruh proses dapat di monitoring dan pengambilan keputusan. Pada tahap proses *fuzzy logic* melalui data sensor turbidity yang di program pada NodeMCU ESP32 untuk mengontrol *Rotary Drum Filter* sesuai *rule based*. Pada hasil *fuzzy logic* dikirim melalui NodeMCU ESP32 ke aplikasi blynk untuk ditampilkan. Dalam tahap proses ini output *fuzzy logic* akan diteruskan pada *Rotary Drum Filter* sehingga pada alat *Rotary Drum Filter* dapat dijalankan dengan *rule based* yang telah ditentukan.

3.3 Perancangan Sistem

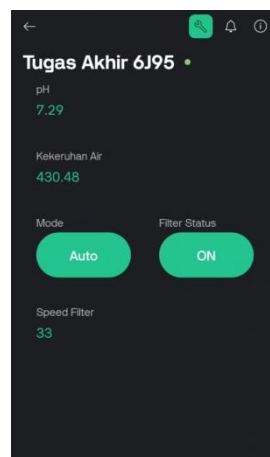


Gambar 3. 3 Perancangan Sistem

Pada sistem ini menggunakan NodeMCU ESP32 sebagai pusat untuk mengendalikan sensor turbidity dan sensor pH melalui pembacaan data. Data di dapat dari sensor turbidity melalui proses *fuzzy logic* dan akan ditampilkan secara *real-time*. Sensor turbidity akan mengukur tingkat kekeruhan air, sedangkan sensor pH akan mengukur tingkat keasaman air. Sistem ini terdapat dua komponen yaitu motor driver dan relay. Motor driver digunakan untuk mengatur kecepatan pada motor dc, sedangkan relay untuk mengatur pada waterpump. Pada motor dc menggerakkan *Rotary Drum Filter* sedangkan *waterpump* untuk menyemprotkan kotoran pada saringan *Rotary Drum Filter*.

3.4 Antarmuka Blynk

Terdapat antarmuka yang menampilkan informasi tentang pH air, kekeruhan air, fitur mode manual/auto, filter status, speed filter. Data pada nilai pH, kekeruhan air, mode, filter status, dan speed filter diambil secara *real-time*. Perancangan pada blynk dapat dilihat dari Gambar 3.4



Gambar 3. 4

3.4 Implementasi Prototype

Pada saat perancangan sistem telah dilakukan, maka selanjutnya melakukan perancangan prototype. Terdapat dua sensor pada perancangan prototype yang harus dikalibrasikan yaitu sensor pH untuk deteksi keasaman air. Untuk mikrokontroler yang akan digunakan yaitu NodeMCU ESP32.

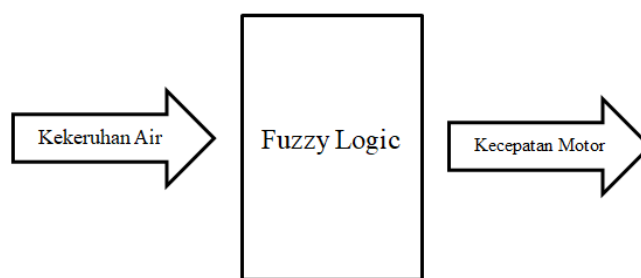
3.4.1 Kalibrasi Sensor pH air

Kalibrasi pada pH meter sangat penting agar mendapatkan akurasi yang baik untuk pengukuran. Pada proses kalibrasi harus dilakukan sebelum dan sesudah pengukuran. Kalibrasi dilakukan menggunakan dua jenis cairan buffer yaitu menggunakan pH 4 dan pH 7 [18].

Langkah pertama pada kalibrasi sensor pH air menyiapkan buffer pH 4 dan pH 7 sebagai pacuan kalibrasi. Masing-masing buffer pH dimasukkan pada wadah berukuran 220 ml dan di isi dengan air murni. Selanjutnya menghitung tegangan pH, yang mengubah tegangan dari 0-5V menjadi diartikan sebagai 14 skala pH yang berbeda. Jika voltase yang terbaca adalah 3.3V, maka voltase yang diperoleh untuk pH 4 dan pH 7. Lalu inputkan hasil dari kedua nilai voltase untuk menghitung nilai pH.

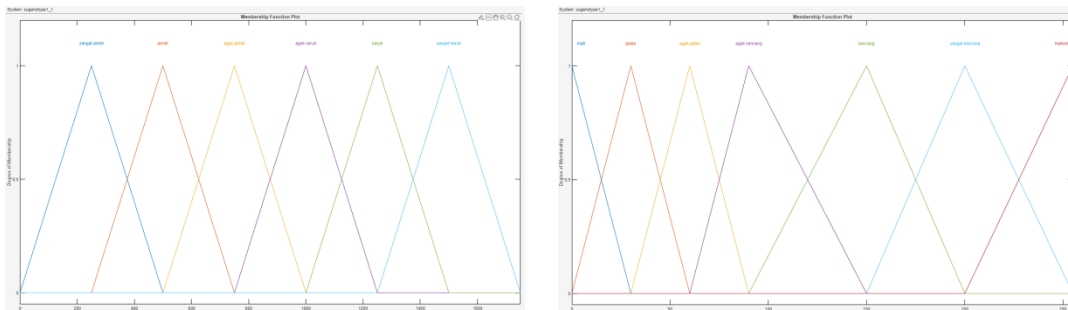
3.5 Implementasi Fuzzy

Pada tahap ini dilakukan implementasi fuzzy pada prototype. Dengan menggunakan metode *fuzzy logic*, maka motor DC akan memiliki kecepatan berputar sesuai dari hasil inputan pada parameter kekeruhan air. Penjelasan metode yang diterapkan berada pada Gambar 3.5.



Gambar 3. 5 Parameter Input dan Output

Pada penelitian ini terdapat variabel satu input dan satu output untuk digunakan variabel *fuzzy logic*. Terdapat input dari kekeruhan air dan output dari kecepatan motor DC terdapat pada Gambar 3.6.



Gambar 3. 6 Variabel Input dan Output

Nilai fungsi keanggotaan pada setiap variabel didapat melalui hasil pengambilan data awal sehingga nilai yang ditentukan berdasarkan data sebenarnya pada air kolam ikan nila dengan mengambil sampel air bersih, air keruh dan air sangat keruh dari kolam ikan nila di rooftop kampus Telkom University Surabaya. Pada pemilihan curva menggunakan trapezoidal pada setiap variabel digunakan agar sistem dapat dijalankan dengan stabil pada rentang yang sudah ditentukan.

1. Variabel Input Kekeruhan Air

Dapat dilihat pada gambar 3.5 bahwa fungsi keanggotaan kekeruhan air terbagi menjadi 6, terdiri dari sangat jernih, jernih, agak jernih, agak keruh, keruh, sangat keruh. Sangat jernih berkisar 0-500ntu, jernih berkisar 250-750ntu, agak jernih berkisar 500-1000ntu, agak keruh berkisar 750-1250ntu, keruh berkisar 1000-1500ntu, sangat keruh berkisar 1250-1750ntu.

2. Variabel Output Kecepatan Motor

Dapat dilihat pada gambar 3.5 bahwa fungsi keanggotaan kecepatan motor terbagi menjadi 7, terdiri dari mati, pelan, agak pelan, agak kencang, kencang, sangat kencang, maksimal. Mati berkisar 0-30pwm, pelan berkisar 10-60pwm, agak pelan berkisar 60-90pwm, agak kencang berkisar 60-150pwm, kencang berkisar 90-200pwm, sangat kencang berkisar 150-255pwm, maksimal berkisar 200-255pwm.

Metode ini terdapat tahapan-tahapannya yaitu:

1. Fuzzyfikasi

$$\mu_{Trapezoidal}(x) = \begin{cases} 0 & \text{jika } x \leq a \\ \frac{x-a}{b-a} & \text{jika } a < x < b \\ 1 & \text{jika } b < x \leq c \\ \frac{d-x}{d-c} & \text{jika } c < x < d \\ 0 & \text{jika } x > d \end{cases}$$

2. Penggunaan fungsi MIN memiliki implikasi tertentu, dan komposisi antar-rule dilakukan dengan menggunakan fungsi MAX.

$$\mu_{output}(z) = \max(\mu_{Rule1}(z), \mu_{Rule2}(z), \dots, \mu_{RuleN}(z))$$

3. Defuzzyfikasi

$$z^* = \frac{\int z \cdot \mu_{output}(z) dz}{\int \mu_{output}(z) dz}$$

4. Hasil dan Pembahasan

4.1 Hasil Pengambilan Data

Penelitian ini melakukan pengambilan data sebanyak lima kali perhari selama delapan hari. Setiap pengambilan data mencatat nilai sensor pH dan nilai kekeruhan. Pada sensor pH dilakukan kalibrasi menggunakan pH buffer, sedangkan kalibrasi sensor turbidity menggunakan larutan standar kekeruhan. Hasil dari pengambilan data terdapat empat puluh kali percobaan dengan mengambil data pada input nilai pH dan nilai kekeruhan dengan output dari kecepatan putaran pada aktuaktor yang dilihat dari nilai kekeruhan, hasil terkait pengambilan data terdapat pada Tabel 4.1.

Tabel 4. 1 Hasil Pengambilan Data

Percobaan ke-	Sensor pH	Nilai Kekeruhan	
		Input	Output
1	7.37	403.02	33
2	7.52	416.64	33
3	7.22	326.03	32
...
38	7.36	503.03	36
39	7.32	533.63	40
40	7.32	597.5	46

4.2 Implementasi Hardware

Pada *container box* terdapat putaran yang berasal dari motor dc dan penyemprotan yang berasal dari *waterpump* untuk melakukan pembersihan pada kotoran air kolam. Hardware lain yang digunakan untuk mengatur reaksi dan pengambilan data menggunakan NodeMCU ESP32, sensor turbidity, dan sensor pH. Penelitian ini melakukan pembersihan pada air kolam menggunakan reaksi pada *waterpump* dan motor dc. Dapat dilihat dari Gambar 4.1



Gambar 4. 1 Implementasi Hardware

4.3 Hasil Perbandingan Sistem dengan MATLAB

Dalam pengujian sistem menggunakan MATLAB, penguji menetapkan set fuzzy, parameter, dan derajat keanggotaan. Kemudian membuat tujuh rulebase yang mengacu pada penelitian terkait sensor turbidity. Pada rulebase digunakan acuan untuk output motor dc. Selanjutnya, melakukan pengujian hasil dari output NodeMCU ESP32 dengan output MATLAB. Hasil akurasi terdapat pada Tabel 4.2.

Tabel 4. 2 Hasil Perbandingan Sistem dengan MATLAB

No	Input Nilai Kekeruhan	Output		Akurasi
		ESP32	Matlab	
1	403.02	33	27,4	83%
2	416.64	33	28	85%
3	326.03	32	21,8	68%
...
38	503.03	36	30,6	85%
39	533.63	40	35,2	88%
40	597.5	46	42,4	92%
Rata - Rata				85.7%

Dapat dilihat dari table 4.2 bahwa nilai akurasi yang diperoleh dari rumus output motor DC yang dihasilkan oleh fuzzy nilai absolut menggunakan nilai NodeMCU ESP32 dikurangi nilai MATLAB dan dilanjutkan perhitungan akurasi dengan cara nilai absolut dibagi nilai MATLAB lalu dikali 100%. Untuk rata-rata pada nilai akurasi pada percobaan yaitu 85.7%. Selanjutnya dihitung dari galat error dengan cara perhitungan 100% dikurangi nilai akurasi.

5. Kesimpulan

Berdasarkan penelitian dan uji coba yang telah dilaksanakan pada kolam ikan nila di kampus Telkom University Surabaya, dapat disimpulkan bahwa sistem filtrasi pada kolam ikan nila sudah tepat untuk membantu melakukan pembersihan air kolam. Pada sistem yang dirancang, memudahkan untuk melakukan pemantauan dan kontroling pada filter air kolam. Penelitian ini dapat menjadikan lebih efektif dan efisien dengan hanya melakukan monitoring melalui aplikasi blynk. Pada pengujian menunjukkan bahwa sistem telah berhasil mengolah data menggunakan metode fuzzy mamdani pada motor DC dengan rule if-else pada kecepatan motor DC secara otomatis. Hasil dari output NodeMCU ESP32 dan MATLAB yang di ambil dari nilai kekeruhan mendapatkan rata-rata hasil akurasi sebesar 85.7% dengan galat error 14.3%.

5.1 Saran

Menurut temuan pada penelitian ini, terdapat aspek yang harus dikembangkan dan diperbaiki agar mendapatkan hasil yang optimal yaitu pada sensor pH dapat dikembangkan dengan melakukan penetralisir hasil pH menggunakan pH up dan pH down agar air kolam ikan nila dapat mencapai hasil yang diharapkan dan untuk perbaikan pada penelitian ini dengan melakukan pengecekan pada alat *Rotary Drum Filter* pada mikrokontroler agar dapat melakukan filtrasi lebih optimal.

Daftar Pustaka

- [1] A. Salim, "Jurnal Vocational Teknik Elektronika dan Informatika Sistem Monitoring Kualitas Air Pada Budidaya Bibit Ikan Nila Menggunakan Algoritma Decision Tree," vol. 11, no. 2, 2023, [Online]. Available: <http://ejournal.unp.ac.id/index.php/voteknika/>
- [2] V. Lakshmikantha, A. Hiriyannagowda, A. Manjunath, A. Patted, J. Basavaiah, and A. A. Anthony, "IoT based smart water quality monitoring system," *Global Transitions Proceedings*, vol. 2, no. 2, pp. 181–186, Nov. 2021, doi: 10.1016/j.glt.2021.08.062.
- [3] D. Ramdani, F. Mukti Wibowo, and Y. Adi Setyoko, "Journal of Informatics, Information System, Software Engineering and Applications Rancang Bangun Sistem Otomatisasi Suhu Dan Monitoring pH Air Aquascape Berbasis IoT (Internet Of Thing) Menggunakan Nodemcu Esp8266 Pada Aplikasi Telegram," vol. 3, no. 1, pp. 59–068, 2020, doi: 10.20895/INISTA.V2I2.
- [4] M. G. Husada and M. Z. Nurhidayat, "Fuzzy logic implementation in water quality monitoring and controlling system for fishwater cultivation."
- [5] R. Sukarno, M. Fajar Ramadhan, F. Andriansyah, Y. Adigutama, and H. Sampurno, "The Automatic Water Recirculation System for Clean Water and Energy Saving in Tilapia Farming," *Jurnal Konversi Energi dan Manufaktur*, vol. 9.
- [6] A. Taufik and A. Fadlil, "Sistem Monitoring pH dan Kekeruhan Kolam ikan Koi Berbasis Internet of Things Menggunakan Aplikasi Blynk," *Jurnal Teknologi Elektro*, vol. 14, no. 1, p. 56, Mar. 2023, doi: 10.22441/jte.2023.v14i1.010.
- [7] R. N. Rohmah, R. Jeprianto, and J. T. Elektro, "Monitoring dan Controlling Kadar pH Pada Air Kolam Ikan dengan Menggunakan Aplikasi Blynk Berbasis Esp Node Mcu," *Jurnal Teknik Elektro*, vol. 21, no. 2, 2021.
- [8] P. A. Indriati and H. Hafiludin, "Manajemen Kualitas Air Pada Pembenihan Ikan Nila (*Oreochromis niloticus*) Di Balai Benih Ikan Teja Timur Pamekasan," *Juvenil:Jurnal Ilmiah Kelautan dan Perikanan*, vol. 3, no. 2, pp. 27–31, Aug. 2022, doi: 10.21107/juvenil.v3i2.15812.
- [9] Nurul Hidayati Lusita Dewi. *Prototype smart home dengan modul nodemcu esp8266 berbasis internet of things (iot)*. Diss. Universitas Islam Majapahit Mojokerto, 2019.
- [10] W. Suriana, I. Gede, A. Setiawan, I. Made, and S. Graha, "Rancang Bangun Sistem Pengaman Kotak Dana Pusia berbasis Mikrokontroler NodeMCU ESP32 dan Aplikasi Telegram," 2021.
- [11] A. Noor *et al.*, "Aplikasi Pendeteksi Kualitas Air Menggunakan Turbidity Sensor dan Arduino Berbasis Web Mobile," *Jurnal CoreIT*, vol. 5, no. 1, 2019.
- [12] D. Aribowo and R. Pratama, "Penerapan Sensor pH pada Area Elektrolizer Di Pt. Sulfindo Adiusaha," *Jurnal PROSISKO*, vol. 5, no. 1, 2018.
- [13] A. Abdus Shofi *et al.*, "Rancang Bangun Water Pump Solar Energy Portable Perairan Sawah Untuk Membantu Petani Kabupaten Probolinggo", doi: 10.18196/mt.v4i.
- [14] S. T. Aprilyani, I. Irianto, and E. Sunarno, "Desain dan Komparasi Kontrol Kecepatan Motor DC," *Jurnal Ecotipe (Electronic, Control, Telecommunication, Information, and Power Engineering)*, vol. 7, no. 2, pp. 127–134, Oct. 2020, doi: 10.33019/jurnalecotipe.v7i2.1886.
- [15] Shoumi, Milyun Ni'ma, and Arie Rachmad Syulistyo. "Analisis Resiko Kanker Payudara (Breast Cancer) Menggunakan Fuzzy Inference System (Fis) Model Mamdani." *Informasi Interaktif* 6.1 (2021): 25-30.
- [16] A. Yasir and B. Solihin Hasugian, "Universitas Dharmawangsa Penggunaan Teknik Kompresi Jpeg Dalam Perancangan Kompresi Citra Digital Memakai Fungsi Gui Pada Matlab."
- [17] T. Sulistyorini, N. Sofi, and E. Sova, "Pemanfaatan Nodemcu Esp8266 Berbasis Android (Blynk) Sebagai Alat Alat Mematikan dan Menghidupkan Lampu," *JUIT*, vol. 1, no. 3, 2022.
- [18] R. Ardiyansyah and S. Abdullah, "Perancangan Sistem Pendeteksi Ph Air Hujan Berbasis Iot (Studi Kasus: Desa Gedepangrango Kabupaten Sukabumi)," *Jutekin (Jurnal Teknik Informatika)*, vol. 10, no. 1, Jun. 2022, doi: 10.51530/jutekin.v10i1.562.

Lampiran

Lampiran 1. Gambar Impelentasi Alat



Lampiran 2. Tabel Hasil Pengambilan Data

Percobaan ke-	Sensor pH	Nilai Kekeruhan	
		Input	Output
1	7.37	403.02	33
2	7.52	416.64	33
3	7.22	326.03	32
4	7.25	664.96	52
5	7.26	647.76	50
6	7.38	444.54	33
7	7.27	700.03	54
8	7.27	810.63	78
9	7.25	754.32	64
10	7.26	717.9	55
11	7.27	929.29	91
12	7.26	313.99	31
13	7.14	389.63	33
14	7.32	458.82	36
15	7.32	597.5	46
16	7.53	430.48	33
17	7.65	268.04	27
18	7.47	350.8	32
19	7.58	363.52	32
20	7.61	338.31	32
21	7.38	754.32	64
22	7.29	430.48	33
23	7.39	326.03	32
24	7.57	389.63	33

25	7.63	376.46	33
26	7.23	458.83	33
27	7.24	416.64	33
28	7.27	503.03	36
29	7.3	518.22	38
30	7.29	581.19	44
31	7.36	488.07	33
32	7.31	389.63	33
33	7.32	849.29	83
34	7.36	565.11	43
35	7.32	473.34	33
36	7.32	549.26	41
37	7.31	376.46	33
38	7.36	503.03	36
39	7.32	533.63	40
40	7.32	597.5	46

Lampiran 3. Tabel Perbandingan Sistem dengan MATLAB

No	Input Nilai Kekuruhan	Output		Akurasi
		ESP32	Matlab	
1	403.02	33	27,4	83%
2	416.64	33	28	85%
3	326.03	32	21,8	68%
4	664.96	52	48,9	94%
5	647,76	50	47,2	94%
6	444.54	33	29	88%
7	700.03	54	52,8	98%
8	810.63	78	78	100%
9	754,32	64	62	97%
10	717.9	55	55	100%
11	929.29	91	91,9	99%
12	313,99	31	20,4	66%
13	389,63	33	26,7	81%
14	458,82	36	29,4	82%
15	597.5	46	42,4	92%
16	430.48	33	28,6	87%
17	268.04	27	13,4	50%
18	350,8	32	24,1	75%
19	363.52	32	25	78%
20	338.31	32	23	72%
21	754,32	64	62	97%
22	430.48	33	28,6	87%
23	326,03	32	21,8	68%
24	389.63	33	26,7	81%

25	376.46	33	25,9	79%
26	458,83	33	29,4	89%
27	416,64	33	28	85%
28	503,03	36	30,6	85%
29	518,22	38	33	87%
30	581.19	44	40,7	93%
31	488,07	33	29,9	91%
32	389,63	33	26,7	81%
33	849.29	83	83,6	99%
34	565.11	43	39	91%
35	473,34	33	29,7	90%
36	549.26	41	37,2	91%
37	376.46	33	25,9	79%
38	503.03	36	30,6	85%
39	533.63	40	35,2	88%
40	597.5	46	42,4	92%
Rata - Rata				85.7%

Lampiran 4. Kodingan

```
#define BLYNK_TEMPLATE_ID "TMPL6YeAxPbta"
#define BLYNK_TEMPLATE_NAME "Tugas Akhir"

#define BLYNK_FIRMWARE_VERSION    "0.1.0"
#define BLYNK_PRINT Serial
// #define BLYNK_DEBUG
#define APP_DEBUG
#define USE_ESP32_DEV_MODULE
#define pwmChannel 1
#define turbidity_offset -0
#define btsPin 18
#define btsPinn 19
#define btsEn 15
#define relayPin1 27
#define relayPin2 26
#define turbidityPin 32
#define phPin 34

#define intervalSend 1000 //mengirim ke blynk setiap 1000ms (1 detik)
#define sample 50
#define PH4 2.06 //tegangan ketika baca PH4 (untuk kalibrasi PH pada PH4)
#define PH7 1.46 //tegangan ketika baca PH7 (untuk kalibrasi PH pada PH7)

#include "BlynkEdgent.h"
#include <Fuzzy.h>

Fuzzy *fuzzy = new Fuzzy();

// Inisialisasi Fuzzy Set Kekерuhan
FuzzySet *sangat_jernih = new FuzzySet(0, 250, 250, 500);
FuzzySet *jernih = new FuzzySet(250, 500, 500, 750);
FuzzySet *agak_jernih = new FuzzySet(500, 750, 750, 1000);
FuzzySet *agak_keruh = new FuzzySet(750, 1000, 1000, 1250);
FuzzySet *keruh = new FuzzySet(1000, 1250, 1250, 1500);
```



```

FuzzySet *sangat_keruh = new FuzzySet(1250, 1500, 1500, 1750);

// Inisialisasi Fuzzy Set Motor
FuzzySet *mati = new FuzzySet(0, 0, 0, 30);
FuzzySet *pelan = new FuzzySet(10, 30, 30, 60);
FuzzySet *agak_pelan = new FuzzySet(30, 60, 60, 90);
FuzzySet *agak_kencang = new FuzzySet(60, 90, 90, 150);
FuzzySet *kencang = new FuzzySet(90, 150, 150, 200);
FuzzySet *sangat_kencang = new FuzzySet(150, 200, 200, 255);
FuzzySet *maksimal = new FuzzySet(200, 255, 255, 255);

unsigned long lstRead;
unsigned long fuzzyStartTime;
bool filterOn = false;
int pwm = 0;
int mode = 0;
bool fuzzyActive = false;

BLYNK_WRITE(V4)
{
  mode = param.asInt();
  Serial.print("V4 value is: ");
  Serial.println(mode);

  if (mode == 1) {
    fuzzyActive = true;
    fuzzyStartTime = millis();
  } else {
    fuzzyActive = false;
  }
}

BLYNK_WRITE(V2)
{
  filterOn = param.asInt();
  Serial.print("V2 value is: ");
  Serial.println(filterOn);
}

void setup()
{
  Serial.begin(115200);
  pinMode(turbidityPin, INPUT);
  pinMode(phPin, INPUT);
  pinMode(btsPin, OUTPUT);
  pinMode(btsPinn, OUTPUT);
  pinMode(btsEn, OUTPUT);
  pinMode(relayPin1, OUTPUT);
  pinMode(relayPin2, OUTPUT);
  digitalWrite(btsEn, HIGH);
  digitalWrite(btsPinn, LOW);

  ledcSetup(pwmChannel, 5000, 8);
  ledcAttachPin(btsPin, pwmChannel);
  delay(100);

  BlynkEdgent.begin();
  FuzzySet();
  fuzzyRule();
}

```

```

void FuzzySet() {
    // Inisialisasi Fuzzy Input Keketuhan Air
    FuzzyInput *turbidity = new FuzzyInput(1);
    turbidity->addFuzzySet(sangat_jernih);
    turbidity->addFuzzySet(jernih);
    turbidity->addFuzzySet(agak_jernih);
    turbidity->addFuzzySet(agak_keruh);
    turbidity->addFuzzySet(keruh);
    turbidity->addFuzzySet(sangat_keruh);
    // Memasukan Data dari Fuzzy Input ke Dalam Logika Fuzzy
    fuzzy->addFuzzyInput(turbidity);

    FuzzyOutput *filter = new FuzzyOutput(1);
    filter->addFuzzySet(mati);
    filter->addFuzzySet(pelan);
    filter->addFuzzySet(agak_pelan);
    filter->addFuzzySet(agak_kencang);
    filter->addFuzzySet(kencang);
    filter->addFuzzySet(sangat_kencang);
    filter->addFuzzySet(maksimal);
    // Memasukan Data dari Fuzzy Output ke Dalam Logika Fuzzy
    fuzzy->addFuzzyOutput(filter);
}

void fuzzyRule() {
    // Rule untuk kondisi sangat jernih
    FuzzyRuleAntecedent *ifSangatJernih = new FuzzyRuleAntecedent();
    ifSangatJernih->joinSingle(sangat_jernih);
    FuzzyRuleConsequent *thenMati = new FuzzyRuleConsequent();
    thenMati->addOutput(mati);
    fuzzy->addFuzzyRule(new FuzzyRule(1, ifSangatJernih, thenMati));

    // Rule untuk kondisi jernih
    FuzzyRuleAntecedent *ifJernih = new FuzzyRuleAntecedent();
    ifJernih->joinSingle(jernih);
    FuzzyRuleConsequent *thenPelan = new FuzzyRuleConsequent();
    thenPelan->addOutput(pelan);
    fuzzy->addFuzzyRule(new FuzzyRule(2, ifJernih, thenPelan));

    // Rule untuk kondisi agak jernih
    FuzzyRuleAntecedent *ifAgakJernih = new FuzzyRuleAntecedent();
    ifAgakJernih->joinSingle(agak_jernih);
    FuzzyRuleConsequent *thenAgakPelan = new FuzzyRuleConsequent();
    thenAgakPelan->addOutput(agak_pelan);
    fuzzy->addFuzzyRule(new FuzzyRule(3, ifAgakJernih, thenAgakPelan));

    // Rule untuk kondisi agak keruh
    FuzzyRuleAntecedent *ifAgakKeruh = new FuzzyRuleAntecedent();
    ifAgakKeruh->joinSingle(agak_keruh);
    FuzzyRuleConsequent *thenAgakKencang = new FuzzyRuleConsequent();
    thenAgakKencang->addOutput(agak_kencang);
    fuzzy->addFuzzyRule(new FuzzyRule(4, ifAgakKeruh, thenAgakKencang));

    // Rule untuk kondisi keruh
    FuzzyRuleAntecedent *ifKeruh = new FuzzyRuleAntecedent();
    ifKeruh->joinSingle(keruh);
    FuzzyRuleConsequent *thenKencang = new FuzzyRuleConsequent();
    thenKencang->addOutput(kencang);
    fuzzy->addFuzzyRule(new FuzzyRule(5, ifKeruh, thenKencang));
}

```

```

// Rule untuk kondisi sangat keruh
FuzzyRuleAntecedent *ifSangatKeruh = new FuzzyRuleAntecedent();
ifSangatKeruh->joinSingle(sangat_keruh);
FuzzyRuleConsequent *thenSangatKencang = new FuzzyRuleConsequent();
thenSangatKencang->addOutput(sangat_kencang);
fuzzy->addFuzzyRule(new FuzzyRule(6, ifSangatKeruh, thenSangatKencang));
}

void loop() {
  BlynkEdgent.run();
  float kekeruhan = readTurbidity();
  float ph = readPH();

  if (mode == 1) { // mode auto, apabila mode auto, speed akan otomatis disesuaikan
    if (fuzzyActive && (millis() - fuzzyStartTime < 60000)) { // Active fuzzy logic for 60 seconds
      fuzzy->setInput(1, kekeruhan);
      fuzzy->fuzzify();
      pwm = fuzzy->defuzzify(1);

      if (pwm == 0) filterOn = false;
      else filterOn = true;
      Blynk.virtualWrite(V2, filterOn);
    } else {
      fuzzyActive = false;
      pwm = 0;
      filterOn = false;
    }
  } else {
    pwm = 120; // manual mode, set fixed speed
  }

  if (filterOn) {
    digitalWrite(relayPin1, LOW);
    digitalWrite(relayPin2, LOW);
  } else {
    digitalWrite(relayPin1, HIGH);
    digitalWrite(relayPin2, HIGH);
    pwm = 0;
  }

  ledcWrite(pwmChannel, pwm);

  if (millis() - lstRead >= intervalSend) {
    lstRead = millis();

    Blynk.virtualWrite(V0, ph);
    Blynk.virtualWrite(V1, kekeruhan);
    Blynk.virtualWrite(V3, pwm);

    Serial.println("====Turbidity =====");
    Serial.print(kekeruhan, 3);
    Serial.print("\n");
  }
}

float readTurbidity() {
  float ntu;
  float volt = 0;

```

```
for (int i = 0; i < sample; i++) {
    volt += ((float)analogRead(turbidityPin) / 4095) * 3.3;
}
volt = volt / sample;
volt = round_to_dp(volt, 2);

ntu = -1120.4 * sq(volt) + 5742.3 * volt - 4353.8;
if (ntu < 0) ntu = 0;
if (ntu > 3000) ntu = 3000;
ntu = (3000 - ntu) - turbidity_offset;
return ntu;
}

float round_to_dp(float in_value, int decimal_place) {
    float multiplier = powf(10.0f, decimal_place);
    in_value = roundf(in_value * multiplier) / multiplier;
    return in_value;
}

float readPH() {
    unsigned long tempValuePH = 0; // pendefinisian nilai pertama
    for (int a = 0; a < sample; a++) { // pengambilan nilai sesuai dengan nilai sample
        tempValuePH += analogRead(phPin);
    }

    float adcPH = tempValuePH / sample; // merata-rata nilai sample
    float voltPH = 5.0 / 4095.0 * adcPH; // menghitung nilai tegangan sensor

    float PH_Val = (PH4 - PH7) / 3; // Rumus untuk mencari tegangan dari hasil kalibrasi PH
    float PO = 7.00 + ((PH7 - voltPH) / PH_Val); // menyesuaikan hasil pembacaan dan hasil kalibrasi

    Serial.println("===== PH =====");
    Serial.print(adcPH);
    Serial.print("\t");
    Serial.print(voltPH, 3);
    Serial.print("\t");
    Serial.print(PO, 3);
    Serial.print("\n");

    return PO; // mengeluarkan nilai PH yang telah dikalibrasi
}
```