

**PERANCANGAN OTOMATISASI PENYEMPROTAN
DISINFEKTAN KANDANG BEBEK PETELUR BERBASIS IOT
MENGUNAKAN METODE LOGIC FUZZY**

SKRIPSI

DISUSUN OLEH

SONI ARMANDA S

NPM.2109020041



UMSU

Unggul | Cerdas | Terpercaya

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

MEDAN

2025

**PERANCANGAN OTOMATISASI PENYEMPROTAN
DISINFEKTAN KANDANG BEBEK PETELUR BERBASIS IOT
MENGUNAKAN METODE LOGIC FUZZY**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer
(S.Kom) dalam Program Studi Teknologi Informasi pada Fakultas Ilmu Komputer
dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara**

SONI ARMANDA S

NPM.2109020041

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
MEDAN
2025**

LEMBAR PENGESAHAN

Judul Skripsi : Perancangan Otomatisasi Penyemprotan Disinfektan
Kandang Bebek Petelur Berbasis IoT Menggunakan
Metode Logic Fuzzy
Nama Mahasiswa : Soni Armanda S
NPM : 2109020041
Program Studi : Teknologi Informasi

Menyetujui
Komisi Pembimbing



(Mahardika Abdi Prawira S.Kom., M.Kom.)
NIDN. 0117088902

Ketua Program Studi



(Fatma Sari Hutagalung S.Kom., M.Kom.)
NIDN. 0117019301

Dekan



(Dr. Al-Khowarizmi, S.Kom., M.Kom.)
NIDN. 0127099201

PERNYATAAN ORISINALITAS

**PERANCANGAN OTOMATISASI PENYEMPROTAN
DISINFEKTAN KANDANG BEBEK PETELUR BERBASIS IOT
MENGUNAKAN METODE LOGIC FUZZY**

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, Agustus 2025

Yang membuat pernyataan



Soni Armanda S

NPM. 2109020041

**PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN
AKADEMIS**

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama : Soni Armanda S
NPM : 2109020041
Program Studi : Teknologi Informasi
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bedas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul:

**PERANCANGAN OTOMATISASI PENYEMPROTAN DISINFECTAN
KANDANG BEBEK PETELUR BERBASIS IOT MENGGUNAKAN
METODE LOGIC FUZZY**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, Agustus 2025

Yang membuat pernyataan



Soni Armanda S

NPM. 2109020041

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap : Soni Armanda S
Tempat dan Tanggal Lahir : Merlung, 17 Juni 2002
Alamat Rumah : Jln.Alfalah No.19
Telepon/Faks/HP : 081264088996
E-mail : soniarmanda.n@gmail.com
Instansi Tempat Kerja : -
Alamat Kantor : -

DATA PENDIDIKAN

SD	: SD Negeri 050758 SECURAI	TAMAT: 2015
SMP	: SMP Negeri 2 Babalan	TAMAT: 2018
SMA	: SMK Negeri 2 Muaro Jambi	TAMAT: 2021

KATA PENGANTAR



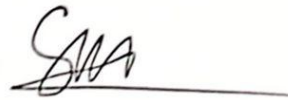
Penulis tentunya berterima kasih kepada berbagai pihak dalam dukungan serta doa dalam penyelesaian skripsi. Penulis juga mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Agussani, M.AP., Rektor Universitas Muhammadiyah Sumatera Utara (UMSU)
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom. Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU, sekaligus Dosen Pembimbing penulis.
3. Ibu Fatma Sari Hutagaulung, S.Kom., M.Kom., selaku Ketua Program Studi Teknologi Informasi sekaligus Dosen Pembimbing Akademik penulis.
4. Bapak Mahardika Abdi Prawira, S.Kom., M.Kom., selaku Sekretaris Program Studi Sistem Informasi.
5. Seluruh dosen dan staf akademik di lingkungan Fakultas Ilmu Komputer dan Teknologi Informasi, khususnya Program Studi Teknologi Informasi, yang telah memberikan ilmu dan pengalaman berharga selama masa studi.
6. Kedua orang tua tercinta, Bapak Nelson Sembiring, Ibu Siska Ginting, yang selalu memberikan doa, semangat, cinta, serta dukungan moral dan material tanpa henti.
7. Rekan-rekan seperjuangan di grup “Grup Ojol” yaitu Handala, Bagus, Candra, dan Farhan, yang telah menjadi teman diskusi sekaligus motivasi selama menempuh studi di UMSU.
8. Teruntuk Rifdah Febiana Putri Lubis, yang selalu setia mendampingi penulis sejak awal hingga akhir proses penyusunan skripsi. Terima kasih atas segala semangat, motivasi, dan doa yang tak pernah putus diberikan. Semoga Allah memudahkan langkah kita menuju masa depan yang indah dan penuh keberkahan bersama.

9. Teman-teman kelas TI A1 Stambuk 21 yang sudah menemani penulis sedari awal kegiatan perkuliahan hingga akhir perkuliahan terimakasih sebanyak-banyaknya.
10. Semua pihak yang tidak dapat disebutkan satu per satu yang telah membantu dan memberikan dukungan secara langsung maupun tidak langsung dalam penyelesaian skripsi ini.
11. Semua pihak yang tidak dapat disebutkan satu per satu yang telah membantu dan memberikan dukungan secara langsung maupun tidak langsung dalam penyelesaian skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari kata sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun untuk perbaikan di masa mendatang. Semoga skripsi ini dapat memberikan manfaat bagi pembaca serta menjadi referensi yang berguna.

Medan, 08 September
2025



Penulis
Soni Armanda S

PERANCANGAN OTOMATISASI PENYEMPROTAN DISINFEKTAN KANDANG BEBEK PETELUR BERBASIS IOT MENGUNAKAN METODE LOGIC FUZZY

ABSTRAK

Penelitian ini merancang sistem otomatisasi penyemprotan disinfektan untuk kandang bebek petelur berbasis Internet of Things (IoT) dengan metode logika fuzzy guna meningkatkan biosekuriti, efisiensi, dan konsistensi sanitasi kandang. Sistem memanfaatkan ESP32 sebagai pengendali utama, sensor DHT22 untuk pemantauan suhu dan kelembapan, modul relay untuk menggerakkan pompa–nozzle, serta integrasi Telegram sebagai sarana monitoring dan kendali jarak jauh secara real-time. Skema fuzzy meliputi fuzzifikasi dua masukan (suhu dan kelembapan), inferensi berbasis aturan, dan defuzzifikasi rata-rata tertimbang untuk menghasilkan durasi semprot diskrit 3, 5, atau 7 detik yang adaptif terhadap kondisi lingkungan. Penjadwalan berbasis NTP membatasi operasi pada hari dan jam tertentu sehingga penyemprotan hanya berlangsung dalam jendela waktu aktif untuk menekan pemborosan disinfektan. Hasil uji pada miniatur kandang menunjukkan bahwa pada kondisi mendekati ideal sistem tidak mengaktifkan semprot, sedangkan pada kondisi lebih panas dan lembap sistem merespons dengan durasi semprot yang lebih lama disertai notifikasi otomatis. Prototipe memenuhi tujuan otomasi yang adaptif, terjadwal, hemat bahan, dan dapat dipantau/dikendalikan secara daring; pengembangan lanjutan direkomendasikan pada penambahan variabel lingkungan, kalibrasi fungsi keanggotaan dan aturan, evaluasi pola sebaran nozzle–debit, serta validasi skala kandang riil dengan horizon pengamatan yang lebih panjang.

Kata Kunci: ESP32; DHT22; Logika Fuzzy; Disinfektan; Kandang Bebek; Otomasi; Telegram.

DESIGN OF AN IoT-BASED AUTOMATED DISINFECTANT SPRAYING SYSTEM FOR LAYER DUCK HOUSING USING LOGIC FUZZY

ABSTRACT

This study designs an Internet of Things (IoT)-based automatic disinfectant spraying system for duck layer houses using fuzzy logic to enhance biosecurity, efficiency, and sanitation consistency. The system employs an ESP32 as the main controller, a DHT22 sensor for temperature and humidity monitoring, a relay-driven pump–nozzle as the actuator, and Telegram integration for real-time remote monitoring and control. The fuzzy scheme comprises fuzzification of two inputs (temperature and humidity), rule-based inference, and weighted-average defuzzification to produce discrete spray durations of 3, 5, or 7 seconds adapted to environmental conditions. NTP-based scheduling restricts operation to specified days and hours, ensuring spraying occurs only within the active window to reduce disinfectant overuse. Experiments on a miniature barn indicate no spraying under near-ideal conditions, while hotter and more humid conditions trigger longer spray durations accompanied by automatic notifications. The prototype meets the goals of adaptive, scheduled, resource-efficient automation with online monitoring/control; recommended future work includes adding environmental variables, calibrating membership functions and rules, assessing nozzle–flow distribution, and validating performance at real-barn scale over longer observation periods.

Keywords: ESP32; DHT22; Fuzzy Logic; Disinfectant; Duck House; Automation; Telegram.

DAFTAR ISI

LEMBAR PENGESAHAN	ii
PENYATAAN ORISINALITASii	iii
PENYATAAN PERSETUJUAN PUBLIKASI	iv
RIWAYAT HIDUP	v
KATA PENGANTAR.....	vi
ABSTRAK	viii
ABSTRACT	ix
DAFTAR ISI.....	x
DAFTAR TABEL	xi
DAFTAR GAMBAR.....	xii
BAB I. PENDAHULUAN	1
1.1. LATAR BELAKANG MASALAH.....	1
1.2. RUMUSAN MASALAH	2
1.3. BATASAN MASALAH	3
1.4. TUJUAN PENELITIAN	4
1.5. MANFAAT PENELITIAN	4
BAB II. LANDASAN TEORI	6
2.1. Penelitian Terdahulu.....	6
2.2. Peternak Bebek Petelur	6
2.3. Disenfektan Kandang Bebek	8
2.4. Internet Of Things (IOT)	9
2.5. Sistem Otomatisasi	10
2.6. Fuzzy Logic	11
2.7. Mikrokontloer ESP32	12
2.8. Sensor DHT22	14
2.9. Pompa 12v DC	16
2.10. Arduino IDE	17
2.11. BreadBoard.....	19
2.12. Kabel Jumper	20
2.13. Modul Relay	22
2.14. Telegram.....	23
2.15. Flowchart.....	24
BAB III. METODOLOGI PENELITIAN	26
3.1. Jenis dan Pendekatan Penelitian	26
3.2. Jenis dan Waktu Penelitian.....	27
3.3. Metode pengumpulan Data	27
3.3.1 Studi Pustaka	27
3.3.2 Observasi	28
3.3.3 Jadwal Penelitian	28
3.3.4 Pengujian Prototype	29
3.4. Penerapan Perancangan Alat Otomatisasi Disenfektam	30
3.5. Alat dan Bahan.....	31
3.5.1 Hardware	31
3.5.2 Bahan	32
3.5.3 Perangkat Lunak.....	32
3.6. Perancangan Alat	33

3.7. Proses Implementasi Alat.....	33
3.8. Diagram Alir Sistem	35
3.9. Metode Pengolahan Data	38
3.9.1 Fuzzifikasi.....	38
3.9.2 Inferensi Aturan Fuzzy.....	40
3.9.3 Defuzzifikasi	41
BAB IV. HASIL DAN PEMBAHASAN	44
4.1. Hasil Penelitian.....	44
4.1.1 Deskripsi Prototype Sistem Otomatisasi	44
4.1.2 Komponen Utama yang Digunakan	45
4.2. Implementasi Perangkat Keras	46
4.2.1 Miniatur Kandang.....	46
4.2.2 Posisi Mikrokontroler ESP32 dan Sensor DHT22	47
4.2.3 Pompa DC 12v dan Modul Relay.....	48
4.2.4 Proses penyemprotan Otomatis	49
4.2.5 Notifikasi Telegram.....	50
4.3. Pengujian Sistem	52
4.3.1 Pengujian Kondisi Normal	52
4.3.2 Pengujian Kondisi Ekstrem.....	52
4.3.3 Tabel Hasil Uji Coba.....	53
4.3.4 Grafik Tren Suhu dan Kelembapan	53
4.4. Analisis Kinerja Sistem	55
4.4.1 Kinerja Sensor	55
4.4.2 Kinerja Logic Fuzzy	55
4.4.3 Kinerja Jadwal NTP	55
4.4.4 Integrasi Telegram.....	56
BAB V. KESIMPULAN DAN SARAN	57
5.1. KESIMPULAN	57
5.2. SARAN	58
DAFTAR PUSTAKA	60
LAMPIRAN.....	63

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu	6
Tabel 2.2 ESP32.....	13
Tabel 2.3 DHT22	15
Tabel 2.4 Pompa 12v DC	17
Tabel 2.5 Arduino IDE.....	18
Tabel 2.6 BreadBoard	20
Tabel 2.7 Kabel Jumper.....	21
Tabel 2.8 Modul Relay.....	23
Tabel 2.9 Flowchart	25
Tabel 3.1 Jadwal Penelitian.....	28
Tabel 3.2 Hardware.....	31
Tabel 3.3 Bahan	32
Tabel 3.4 Perangkat Lunak	32
Tabel 3.5 Aturan Fuzzy.....	41
Tabel 4.1 Tabel Hasil Uji Coba.....	53

DAFTAR GAMBAR

Gambar 2.1 ESP32	12
Gambar 2.2 DHT22.....	14
Gambar 2.3 Pompa 12V DC	16
Gambar 2.4 Arduino IDE	18
Gambar 2.5 BreadBoard.....	19
Gambar 2.6 Kabel Jumper.....	20
Gambar 2.7 Modul Relay	22
Gambar 3.2 Perancangan Alat.....	33
Gambar 3.3 Diagram Alir Sistem	37
Gambar 3.4 Suhu.....	39
Gambar 3.8 Kelembapan.....	40
Gambar 4.1 Miniatur Kandang	47
Gamabr 4.2 Posisi Mikrokontroler ESP32 dan Sensor DHT22	48
Gambar 4.3 Pompa DC 12v dan Modul Relay	49
Gamabr 4.4 Proses penyemprotan Otomatis.....	50
Gambar 4.5 Notifikasi Telegram	51
Gambar 4.6 Grafik Hubungan Suhu den Durasi Semprot.....	54
Gambar 4.7 Grafik Hubungn Kelembapan Dengan Durasi Semprot.....	54

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Penyakit yang menyerang ternak, khususnya unggas seperti bebek petelur, masih menjadi tantangan besar dalam sektor peternakan di Indonesia. Penyebaran penyakit yang cepat dapat menyebabkan kerugian ekonomi yang signifikan, penurunan produktivitas, bahkan kematian massal pada populasi ternak (Medion, 2021). Salah satu faktor utama penyebabnya adalah lingkungan kandang yang tidak higienis, sehingga upaya menjaga kebersihan kandang menjadi sangat krusial.

Penyemprotan disinfektan secara rutin telah diakui sebagai bagian penting dari program biosekuriti untuk menekan pertumbuhan mikroorganisme patogen di kandang ternak (Safari, 2022). Namun, praktik manual dalam penyemprotan disinfektan sering kali menghadapi kendala seperti ketidakteraturan, ketergantungan pada tenaga kerja manusia, serta kurang optimalnya distribusi cairan disinfektan ke seluruh area kandang. Hal ini menyebabkan efektivitas pencegahan penyakit menjadi kurang maksimal.

Seiring perkembangan teknologi, konsep otomatisasi berbasis Internet of Things (IoT) mulai diterapkan dalam berbagai aspek peternakan, termasuk sistem penyemprotan disinfektan. IoT memungkinkan integrasi sensor, aktuator, dan sistem monitoring yang dapat bekerja secara otomatis dan real-time, sehingga proses penyemprotan dapat berjalan lebih efisien, terjadwal, dan dapat dipantau dari jarak jauh melalui perangkat digital (Ummul Masir et al., 2024).

Pada peternakan bebek petelur, penerapan IoT terbukti mampu meningkatkan efisiensi dan produktivitas. Penelitian menunjukkan bahwa pemantauan kondisi kandang berbasis IoT, seperti suhu, kelembapan, dan kadar gas amonia, dapat membantu peternak menjaga lingkungan kandang tetap optimal untuk kesehatan dan produksi telur (Putri Br Ginting et al., 2024). Dengan adanya sistem otomatis, risiko kelalaian manusia dalam menjaga kebersihan kandang dapat diminimalisir.

Untuk pengambilan keputusan otomatis terkait kapan dan berapa banyak disinfektan yang harus disemprotkan, metode logic fuzzy menjadi solusi yang sangat relevan. Logic fuzzy dapat menangani ketidakpastian data sensor dan memberikan keputusan adaptif berdasarkan kondisi aktual di kandang, seperti suhu dan kelembapan (Anhad Zaki Julian, n.d.). Sistem ini memungkinkan penyemprotan dilakukan hanya saat diperlukan, sehingga penggunaan disinfektan menjadi lebih efisien dan ramah lingkungan.

Dengan demikian, perancangan otomatisasi penyemprotan disinfektan kandang bebek petelur berbasis IoT menggunakan metode logic fuzzy merupakan inovasi strategis untuk mendukung keberlanjutan peternakan modern. Sistem ini diharapkan dapat meningkatkan biosekuriti, produktivitas, dan kesejahteraan ternak secara signifikan, serta memberikan solusi praktis yang mudah diadopsi oleh peternak di berbagai skala usaha.

1.2. Rumusan Masalah

1. Bagaimana merancang sistem otomatisasi penyemprotan disinfektan pada kandang bebek petelur yang dapat bekerja secara efektif dan efisien?

2. Bagaimana memanfaatkan teknologi Internet of Things (IoT) untuk memantau kondisi lingkungan kandang secara real-time dan mengendalikan proses penyemprotan disinfektan dari jarak jauh?
3. Bagaimana mengimplementasikan metode logic fuzzy dalam pengambilan keputusan otomatis untuk menentukan intensitas dan frekuensi penyemprotan berdasarkan data sensor suhu dan kelembapan?

1.3. Batasan Masalah

- A. Sistem otomatisasi difokuskan pada penyemprotan disinfektan di kandang bebek petelur.
- B. Pengendalian dan pemantauan sistem menggunakan teknologi Internet of Things (IoT) secara real-time.
- C. Pengambilan keputusan pengaturan intensitas dan frekuensi penyemprotan menggunakan metode logika fuzzy berdasarkan sensor suhu dan kelembapan lingkungan kandang.
- D. Sensor yang digunakan terbatas pada sensor suhu dan kelembapan.
- E. Sistem diuji pada skala kandang bebek petelur tertentu (misal: skala kecil hingga menengah) dan tidak mencakup aspek peternakan lain seperti pemberian pakan atau pengelolaan limbah.
- F. Sistem berfokus pada aspek teknis otomasi dan bukan aspek biologis atau kesehatan hewan secara langsung.

Tujuan ini sejalan dengan penelitian sebelumnya yang menunjukkan bahwa penerapan IoT dan metode fuzzy dapat meningkatkan efisiensi dan efektivitas pengelolaan lingkungan peternakan (Yudhatama, 2021). Dengan demikian,

penelitian ini diharapkan dapat memberikan solusi praktis dan inovatif dalam menjaga kebersihan kandang dan kesehatan bebek petelur.

1.4. Tujuan Penelitian

- A.** Merancang dan mengembangkan sistem otomatisasi penyemprotan disinfektan yang bekerja efektif dan efisien di kandang bebek petelur.
- B.** Menerapkan teknologi IoT untuk pemantauan lingkungan kandang secara real-time dan pengendalian jarak jauh penyemprotan disinfektan.
- C.** Mengimplementasikan metode logika fuzzy sebagai sistem pengambilan keputusan otomatis dalam mengatur intensitas dan frekuensi penyemprotan berdasarkan data sensor lingkungan.

1.5. Manfaat Penelitian

A. Bagi Peternak :

- a. Mempermudah proses penyemprotan disinfektan secara otomatis sehingga mengurangi beban kerja manual.
- b. Meningkatkan kesehatan dan produktivitas bebek petelur melalui sanitasi kandang yang lebih konsisten dan efektif.
- c. Memungkinkan pemantauan dan pengendalian kondisi kandang secara real-time dari jarak jauh melalui aplikasi berbasis IoT

B. Bagi Akademisi :

- a. Menjadi referensi penting dalam pengembangan teknologi otomasi peternakan berbasis IoT dan metode logic fuzzy.

- b. Menyediakan prototipe miniatur sebagai model pembelajaran dan pengujian sistem kendali cerdas.
- c. Memperkaya kajian ilmiah tentang integrasi sensor, aktuator, dan algoritma fuzzy dalam aplikasi pertanian.

C. Bagi Pengembangan Sistem :

- a. Memberikan gambaran desain dan implementasi sistem otomasi penyemprotan disinfektan yang efektif dan adaptif.
- b. Membuka peluang inovasi teknologi IoT yang dapat disesuaikan untuk berbagai skala peternakan.
- c. Mendorong pengembangan solusi otomasi pertanian yang lebih efisien dan ramah lingkungan.

BAB II

LANDASAN TEORI

2.1. Penelitian Terdahulu

Dalam pengembangan sistem otomatisasi penyemprotan disinfektan berbasis IoT, beberapa penelitian sebelumnya telah dilakukan untuk meningkatkan efisiensi dan efektivitas dalam menjaga kebersihan kandang bebek. Berikut adalah lima penelitian terdahulu yang relevan:

Tabel 2.1 Peneltian Terdahulu

No	Peneliti / Tahun	Judul Penelitian	Metode	Hasil / Fokus Penelitian
1	Ali & Wibowo (2021)	Sistem Otomatis Penyemprot Tanaman Berbasis IoT	Rule-Based	IoT dapat meningkatkan efisiensi penyemprotan pada lahan pertanian
2	Rahayu et al. (2022)	Kendali Otomatis Kelembapan Ruangan Menggunakan Fuzzy Logic dan DHT22	Fuzzy Logic	Fuzzy logic memberikan respon penyemprotan lebih halus dan efisien
3	Fauzan (2023)	Monitoring Kandang Ayam Berbasis ESP32 dan Blynk	IoT (Blynk)	Sistem berhasil menampilkan suhu dan kelembapan secara real-time
4	Sari (2020)	Otomatisasi Disinfeksi Ruangan Berbasis Arduino dengan Sensor PIR	IF-THEN	Deteksi gerakan sebagai pemicu penyemprotan
5	Yusuf (2024)	Kendali Suhu Otomatis Berbasis IoT Menggunakan Fuzzy Mamdani	Fuzzy Mamdani	Fuzzy logic meningkatkan kenyamanan lingkungan dengan kendali otomatis

2.2. Peternakan Bebek Petelur

Peternakan bebek petelur merupakan salah satu subsektor peternakan unggas yang fokus pada produksi telur konsumsi dari bebek betina (*Anas domesticus*). Bebek petelur memiliki potensi tinggi dalam mendukung ketahanan pangan dan perekonomian masyarakat, terutama di daerah pedesaan. Salah satu keunggulan

utama bebek petelur adalah kemampuannya bertelur secara produktif meskipun dipelihara di lingkungan yang sederhana dan dengan pakan alternatif.

Menurut (Safutra et al., 2022), bebek petelur dapat memproduksi telur secara konsisten dengan produktivitas antara 50–80% tergantung pada manajemen kandang, pakan, dan kesehatan lingkungan. Namun, dalam proses pemeliharaannya, kebersihan kandang dan sanitasi lingkungan menjadi faktor kritis. Kondisi kandang yang kotor, lembap, dan tidak terkontrol berisiko menyebabkan penyebaran penyakit seperti kolera bebek, salmonellosis, dan avian influenza yang dapat menurunkan produktivitas dan meningkatkan angka kematian ternak.

Upaya untuk menjaga kebersihan kandang dilakukan melalui pembersihan rutin, sistem drainase yang baik, serta penyemprotan disinfektan secara berkala. Penyemprotan ini berfungsi membunuh mikroorganisme patogen seperti virus, bakteri, dan jamur yang dapat mengancam kesehatan bebek. Namun, dalam praktiknya, penyemprotan secara manual seringkali tidak konsisten dan bergantung pada tenaga manusia, sehingga dibutuhkan solusi otomatis berbasis teknologi.

Dengan mengintegrasikan teknologi otomatisasi dan sistem kendali berbasis IoT, proses sanitasi kandang bebek petelur dapat dilakukan secara lebih efisien, terjadwal, dan adaptif terhadap kondisi lingkungan seperti suhu dan kelembapan. Hal ini menjadi penting dalam rangka meningkatkan biosekuriti dan produktivitas usaha peternakan bebek petelur.

2.3. Disinfektan Kandang Bebek

Disinfeksi merupakan salah satu upaya dalam sistem sanitasi kandang yang bertujuan untuk mengurangi atau membunuh mikroorganisme patogen seperti bakteri, virus, dan jamur yang menempel pada permukaan lantai, dinding, peralatan, bebek, dan udara di dalam kandang. Disinfeksi dilakukan menggunakan bahan kimia khusus yang dikenal sebagai disinfektan, yang diaplikasikan secara manual atau otomatis dalam bentuk larutan semprot, dan uap.

Menurut (Medion, 2021), disinfeksi kandang unggas terbukti efektif dalam menurunkan jumlah mikroorganisme pada lantai dan dinding kandang. Penyemprotan dilakukan dengan konsentrasi tertentu dan frekuensi teratur agar dapat menekan perkembangan bakteri dan virus, terutama pada lingkungan lembap yang cenderung mempercepat pertumbuhan mikroba.

Selain itu, disinfeksi yang dilakukan secara manual masih memiliki keterbatasan dalam hal waktu, ketepatan sasaran semprot, dan konsistensi pelaksanaan. Oleh karena itu, diperlukan sistem penyemprotan otomatis yang mampu menyesuaikan waktu dan intensitas penyemprotan berdasarkan kondisi lingkungan kandang, seperti suhu dan kelembapan.

Integrasi sistem otomatisasi dan teknologi Internet of Things (IoT) memungkinkan proses disinfeksi berlangsung lebih efisien, terjadwal, dan dapat dikontrol serta dimonitor secara jarak jauh. Dengan penambahan metode fuzzy logic sebagai sistem kendali, penyemprotan disinfektan dapat dilakukan secara lebih cerdas berdasarkan variabel input lingkungan, bukan hanya sekadar waktu tertentu.

2.4. Internet of Things (IoT)

Internet of Things (IoT) adalah konsep di mana objek fisik, seperti perangkat elektronik, sensor, dan aktuator, saling terhubung melalui jaringan internet dan mampu bertukar data secara otomatis tanpa intervensi manusia secara langsung. IoT memungkinkan perangkat-perangkat ini untuk mengumpulkan, mengirimkan, dan merespons data secara real-time, sehingga meningkatkan efisiensi operasional dan pengambilan keputusan (Saputra et al., 2022).

Dalam konteks peternakan, penerapan IoT telah berkembang pesat, khususnya dalam otomasi pemantauan lingkungan kandang dan proses-proses rutin seperti pemberian pakan, pengontrolan suhu, hingga penyemprotan disinfektan. IoT memberikan solusi berbasis data dan kontrol otomatis yang sangat dibutuhkan oleh peternak untuk meningkatkan produktivitas, menurunkan risiko penyakit, dan menghemat sumber daya.

Implementasi IoT dalam bidang peternakan unggas mampu meningkatkan efisiensi manajemen hingga 35% karena proses otomatisasi yang lebih stabil dan responsif terhadap perubahan lingkungan kandang.

IoT juga mendukung pengembangan sistem berbasis prototipe karena memungkinkan integrasi dan modifikasi yang cepat terhadap komponen yang digunakan. Hal ini selaras dengan pendekatan prototyping, di mana pengujian dan penyempurnaan dilakukan secara iteratif berdasarkan feedback dari sistem awal.

Dengan kemampuan tersebut, penggunaan IoT dalam sistem penyemprotan disinfektan kandang bebek petelur memberikan solusi inovatif dan efektif untuk mengurangi penyebaran penyakit dan meningkatkan kualitas produksi telur bebek.

2.5. Sistem Otomatisasi

Sistem otomatisasi adalah proses penggunaan teknologi untuk mengoperasikan mesin atau sistem secara otomatis tanpa keterlibatan manusia secara langsung dalam setiap aktivitasnya. Dalam konteks peternakan bebek petelur, sistem otomatisasi berperan penting dalam meningkatkan efisiensi dan efektivitas pengelolaan kandang, khususnya dalam kegiatan rutin seperti penyemprotan disinfektan (Friendly et al., 2022).

Sistem otomatisasi merupakan upaya penggantian pekerjaan manual dengan sistem mekanis, elektrik, atau berbasis komputer agar proses berjalan lebih cepat, akurat, dan konsisten. Dalam hal ini, sistem otomatisasi membantu peternak menjaga kebersihan kandang secara berkala dengan campur tangan minimal, yang sangat penting dalam mencegah penyebaran penyakit akibat kotoran, bakteri, dan virus.

Penerapan otomatisasi dalam peternakan unggas seperti bebek petelur juga telah dibuktikan dapat meningkatkan produktivitas ternak dan menurunkan beban kerja manusia. Disinfektan yang disemprotkan secara berkala dapat menurunkan risiko penyakit seperti kolera bebek, salmonellosis, dan avian influenza yang umumnya menyerang unggas di lingkungan kandang yang lembap dan tidak steril.

Sistem ini bekerja berdasarkan parameter yang telah ditentukan, misalnya penyemprotan dilakukan setiap jam tertentu atau saat suhu/kelembapan melewati ambang batas. Otomatisasi ini juga dapat diintegrasikan dengan aplikasi berbasis IoT sehingga peternak dapat mengatur jadwal atau mengaktifkan penyemprotan langsung dari perangkat seperti smartphone atau laptop.

Sistem otomatisasi tidak hanya meningkatkan efisiensi tenaga kerja, namun juga memastikan bahwa proses sanitasi dilakukan secara konsisten dan tepat waktu.

Hal ini dapat menurunkan tingkat stres pada bebek, menjaga kesehatan mereka, dan pada akhirnya mendukung peningkatan produktivitas telur.

2.6. Fuzzy Logic

Fuzzy Logic atau logika fuzzy merupakan suatu metode pemrosesan data numerik ke dalam bentuk linguistik, kemudian menghasilkan keputusan berdasarkan kombinasi aturan-aturan linguistik tersebut. Konsep ini pertama kali diperkenalkan oleh Lotfi A. Zadeh pada tahun 1965, sebagai pengembangan dari himpunan fuzzy yang memungkinkan representasi informasi yang bersifat tidak pasti atau samar (Yao, n.d.).

Berbeda dengan logika klasik yang hanya mengenal dua nilai kebenaran 0 dan 1, fuzzy logic bekerja dengan rentang nilai antara 0 hingga 1 untuk menunjukkan derajat keanggotaan. Dengan pendekatan ini, sistem dapat menilai variabel masukan secara lebih fleksibel, misalnya "suhu sedang" atau "kelembapan agak tinggi", yang sangat bermanfaat dalam pengambilan keputusan otomatis pada kondisi lingkungan yang dinamis.

Dalam penelitian ini, fuzzy logic diterapkan sebagai sistem pengendali utama pada alat penyemprotan disinfektan otomatis berbasis IoT. Sistem membaca input dari sensor suhu dan kelembapan, kemudian memprosesnya melalui tiga tahapan utama dalam fuzzy logic.

1. Fuzzifikasi

Proses ini mengubah data numerik dari sensor menjadi bentuk linguistik dengan fungsi keanggotaan misalny:

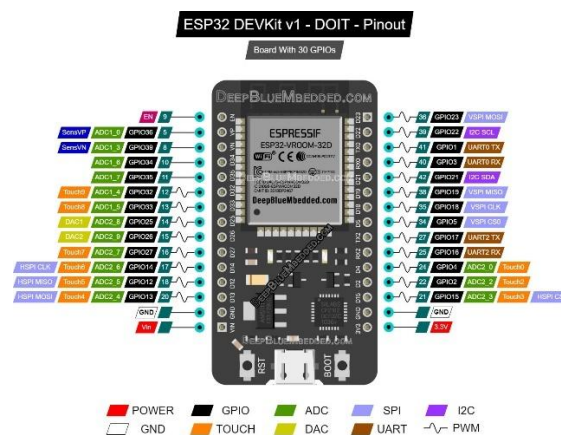
- a. Suhu: (Rendah,Ideals,Tinggi)
- b. Kelembapan: (Lembab,Ideals,Kering)

2. Inferensi

Data linguistik kemudian diproses berdasarkan aturan (rule base) seperti:

- a. Jika suhu tinggi dan kelembapan kering, maka intensitas semprot maksimal
- b. Jika suhu rendah dan kelembapan lembap, maka semprot dimatikan.

2.7. Mikrokontroler ESP32



Gambar 2.1 ESP32

(Sumber: <https://hinoameptfixlist.z13.web.core.windows.net/doit-esp32-devkit-v1-arduino.html>)

ESP32 adalah sebuah mikrokontroler yang dirancang oleh Espressif Systems khusus untuk aplikasi Internet of Things. Mikrokontroler ini memiliki prosesor dual-core yang kuat, dilengkapi dengan modul WiFi dan Bluetooth terintegrasi, serta berbagai port untuk menghubungkan sensor dan aktuator. ESP32 mampu menjalankan pemrosesan data secara real-time, mendukung komunikasi nirkabel, dan dapat diprogram menggunakan platform populer seperti Arduino IDE. Menurut beberapa penelitian dan jurnal terkini, ESP32 banyak digunakan dalam sistem otomasi peternakan dan monitoring lingkungan karena kemampuannya yang handal

dan efisien dalam mengolah data sensor serta mengontrol aktuator secara otomatis (Simamora et al., n.d.).

ESP32 sangat relevan dalam perancangan sistem otomatisasi penyemprotan disinfektan kandang bebek petelur berbasis IoT karena mikrokontroler ini memungkinkan pengumpulan data lingkungan secara real-time melalui sensor suhu dan kelembapan, kemudian mengolah data tersebut untuk mengendalikan pompa dan nozzle penyemprot secara otomatis. Dengan konektivitas WiFi yang terintegrasi, ESP32 juga memungkinkan sistem untuk dipantau dan dikontrol dari jarak jauh melalui aplikasi smartphone, sehingga meningkatkan kemudahan pengelolaan kandang dan efisiensi penggunaan disinfektan (Simamora et al., n.d.).

Penggunaan ESP32 dalam sistem ini membantu mengurangi ketergantungan pada pengoperasian manual, meningkatkan akurasi dan respons sistem terhadap perubahan kondisi lingkungan, serta mendukung penerapan teknologi modern pada bidang peternakan yang semakin berkembang. Oleh karena itu, ESP32 menjadi pilihan utama dalam pengembangan sistem otomasi berbasis IoT untuk aplikasi seperti penyemprotan disinfektan kandang bebek petelur.

Tabel 2.2 ESP32

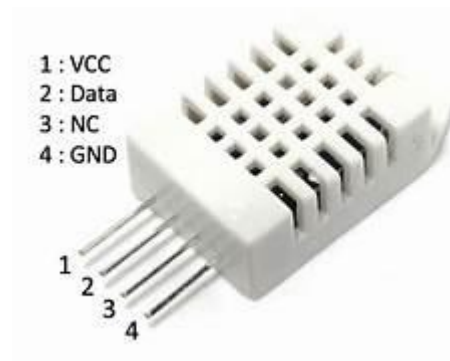
Spesifikasi	Keterangan
Mikroprosesor	Dual-core Xtensa® 32-bit LX6 CPU
Frekuensi CPU	Hingga 240 MHz
Memori	520 KB SRAM internal
Flash	4 MB (varian umum)
Wi-Fi	IEEE 802.11 b/g/n
Bluetooth	Bluetooth v4.2 BR/EDR dan BLE
GPIO	Sekitar 34 pin input/output digital
Antarmuka Komunikasi	UART, SPI, I2C, I2S, CAN, PWM, ADC, DAC
Tegangan Operasional	3.0–3.3V
Konsumsi Daya	Rendah (mendukung mode sleep untuk efisiensi energi)

Ukuran	Kecil, cocok untuk embedded system
--------	------------------------------------

Keunggulan ESP32 antara lain kemampuannya untuk multitasking (dual-core), mendukung komunikasi nirkabel, dan konsumsi daya rendah. Hal ini membuatnya sangat cocok untuk aplikasi di peternakan yang membutuhkan sistem otomatis dan dapat dikontrol secara remote, namun tetap hemat energi.

Dengan penggunaan ESP32 dalam penelitian ini, diharapkan sistem penyemprotan dapat beroperasi secara otomatis dan efisien dalam menjaga sanitasi kandang bebek petelur. ESP32 juga memungkinkan integrasi dengan berbagai platform cloud IoT seperti Blynk, ThingsBoard, atau Firebase untuk monitoring data sensor secara real-time.

2.8. Sensor DHT22



Gambar 2.2 DHT22

(Sumber: <https://fity.club/lists/suggestions/dht22-pinout/>)

Sensor DHT22, yang juga dikenal dengan nama AM2302, adalah sensor digital yang digunakan untuk mengukur suhu dan kelembapan udara dengan tingkat akurasi yang tinggi dan rentang pengukuran yang luas. Sensor ini memiliki rentang pengukuran suhu dari -40°C sampai 80°C dengan akurasi $\pm 0,5^{\circ}\text{C}$, serta rentang kelembapan 0–100% RH dengan akurasi $\pm 2\%$ RH. DHT22 menggunakan teknologi

sensing kapasitif untuk kelembapan dan NTC thermistor untuk suhu, menghasilkan data digital melalui antarmuka satu kabel single-wire digital, yang memudahkan integrasi dengan mikrokontroler seperti ESP32 atau Arduino.

Sensor ini beroperasi pada tegangan 3,3 hingga 6 Volt DC dengan konsumsi arus sekitar 1 hingga 1,5 mA saat pengukuran. Waktu pembaruan data sekitar 2 detik memungkinkan pemantauan kondisi lingkungan yang cukup responsif untuk aplikasi seperti otomatisasi penyemprotan disinfektan di kandang bebek petelur. Keunggulan utama DHT22 adalah stabilitas jangka panjang dan sensitivitas tinggi dengan resolusi 0,1°C untuk suhu dan 0,1% untuk kelembapan serta kemampuan bekerja dalam rentang lingkungan yang cukup ekstrim.

Dalam sistem berbasis IoT, sensor DHT22 sering digunakan sebagai komponen utama dalam monitoring suhu dan kelembapan secara real-time. Data yang dihasilkan oleh DHT22 dapat diproses oleh mikrokontroler ESP32 dan dikirim ke aplikasi monitoring sehingga memungkinkan kontrol otomatis dan pemantauan jarak jauh pada proses penyemprotan disinfektan kandang bebek. Karena kemudahan penggunaan, harga relatif terjangkau, dan performa andal, sensor DHT22 menjadi pilihan populer dalam berbagai aplikasi otomasi lingkungan berbasis IoT.

Tabel 2.3 DHT22

Spesifikasi	Keterangan
Tegangan Operasi	3.3 – 6V DC
Arus Maksimum	2.5 mA
Rentang Suhu	-40°C hingga +80°C
Akurasi Suhu	±0.5°C
Rentang Kelembapan	0% hingga 100% RH
Akurasi Kelembapan	±2% RH
Frekuensi Sampling	1 Hz (1 pembacaan per detik)
Antarmuka Komunikasi	Digital, single-bus
Ukuran	15.1mm x 25mm x 7.7mm

2.9. Pompa 12v DC



Tabel 2.4 Pompa 12V DC

(Sumber: <https://www.tokopedia.com/tukura/pompa-air-otomatis-dc-12-volt-good-pump-real-130-psi-6ltr-bonus-jack-f>)

Pompa 12V DC adalah pompa air atau cairan yang menggunakan motor listrik dengan sumber daya arus searah (DC) bertegangan 12 volt, seperti baterai mobil atau aki kapal, untuk memindahkan atau mensirkulasikan cairan dari satu tempat ke tempat lain. Pompa ini bekerja dengan mengubah energi listrik DC menjadi energi mekanik yang menghasilkan tekanan atau aliran fluida. Biasanya digunakan dalam berbagai aplikasi portabel dan sistem yang menggunakan sumber listrik 12V, misalnya di kendaraan, perahu, sistem irigasi kecil, dan sistem suplai air off-grid. Pompa 12V DC bisa berupa pompa diafragma, pompa sentrifugal kecil, atau jenis pompa lainnya tergantung kebutuhan aliran dan tekanan.

Tabel 2.4 Pompa 12V DC

Spesifikasi	Rincian
Tegangan Kerja	12 Volt DC
Arus	0.5 A hingga 6.2 A
Daya	9 Sampai 12 V

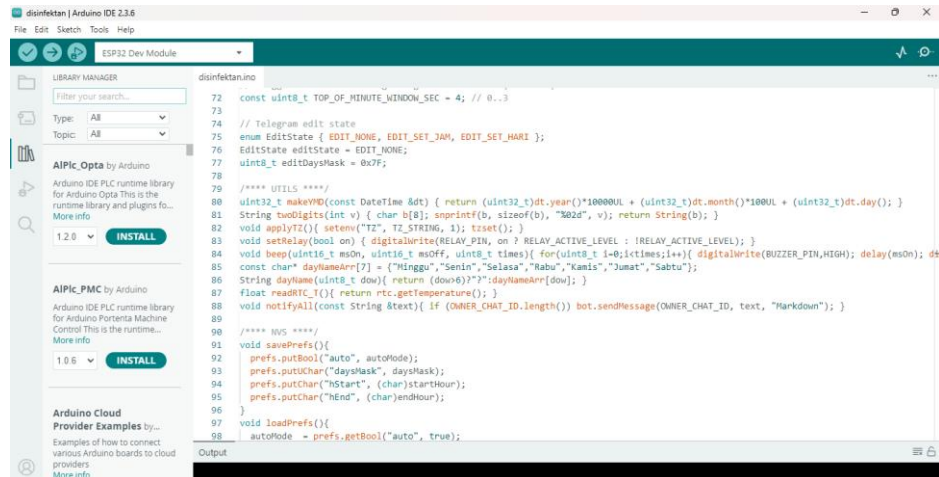
Debit Aliran	1.5 – 50 Liter per menit
Tekanan Kerja	0.35 – 4.8 bar (35 – 70 PSI)
Tipe Pompa	Diafragma
Self-Priming	Ya, biasanya hingga 1.2 meter hisap
Fitur Tambahan	Proteksi overheat, sakelar tekanan otomatis (on-demand), tahan dry-run, valve check
Aplikasi Utama	Suplai air di RV, perahu, irigasi kecil, penyemprotan, pembuangan bilge
Dimensi	Variatif, biasanya kompak dan ringan
Berat	100 g sampai 2 kg (tergantung model)

2.10. Arduino IDE

Arduino IDE adalah perangkat lunak open-source yang digunakan untuk menulis, mengedit, dan mengunggah kode program ke mikrokontroler seperti Arduino UNO, NodeMCU, hingga ESP32. Dalam penelitian ini, Arduino IDE digunakan sebagai lingkungan pengembangan utama untuk menulis program yang mengatur otomatisasi penyemprotan disinfektan berbasis Internet of Things (IoT) di kandang bebek petelur.

Arduino IDE menyediakan berbagai pustaka (library) dan fungsi bawaan yang memudahkan pengembang dalam mengontrol komponen seperti sensor DHT22 untuk membaca suhu dan kelembapan, modul relay untuk mengaktifkan pompa, serta integrasi dengan Blynk untuk kontrol jarak jauh. Penggunaan Arduino IDE juga mendukung metode prototyping karena memungkinkan proses debugging, pengujian logika, dan modifikasi kode dilakukan secara cepat dan efisien (Nahak et al., 2024).

Salah satu keunggulan utama dari Arduino IDE dalam penelitian ini adalah kompatibilitasnya dengan ESP32, yang merupakan pusat pengendali sistem otomatisasi.



Gambar 2.4 Arduino Ide

(Sumber: Primer)

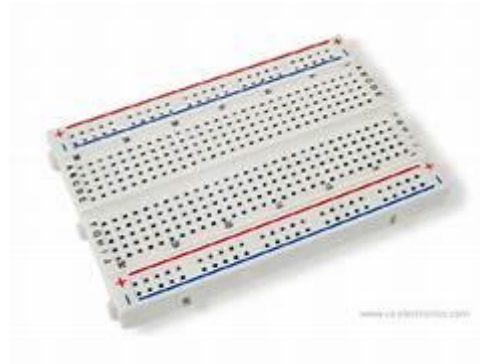
Dengan menambahkan board manager ESP32, pengguna dapat mengunggah kode melalui port USB langsung ke mikrokontroler, yang kemudian mengatur logika penyemprotan berdasarkan kondisi lingkungan di kandang.

Arduino IDE juga mendukung plugin dan library tambahan seperti DHT WiFi dan BlynkSimpleEsp32 yang semuanya digunakan untuk menghubungkan sistem dengan jaringan internet serta mengatur proses penyemprotan secara otomatis dan manual.

Tabel 2.5 Arduino IDE

Spesifikasi	Keterangan
Platform	Windows, Linux, macOS
Bahasa Pemrograman	C/C++
Kompatibilitas Board	Arduino UNO, Nano, Mega, ESP8266, ESP32, dsb.
Fitur Utama	Editor kode, Serial Monitor, Board Manager, Library Manager
Library Pendukung	DHT.h, Blynk.h, WiFi.h, ESP32Servo.h, dll.
Koneksi ke Perangkat	USB Port
Format File Kode	.ino

2.11. BreadBoard



Gambar 2.5 Breadboard

(Sumber: <https://digilent.com/shop/half-size-breadboard/>)

Breadboard adalah papan sirkuit tanpa solder yang digunakan untuk merangkai dan menguji rangkaian elektronik secara sementara sebelum dilakukan penyolderan atau pembuatan rangkaian permanen. Dalam penelitian ini, breadboard berfungsi sebagai media penghubung utama antara komponen-komponen seperti mikrokontroler ESP32, sensor DHT22, modul relay, dan aktuator seperti pompa serta mist nozzle.

Penggunaan breadboard memungkinkan pengembangan sistem yang fleksibel dan mudah dimodifikasi, yang sangat mendukung pendekatan metode prototyping yang digunakan dalam penelitian ini. Karena metode prototyping bersifat iteratif dan mengandalkan perbaikan sistem secara bertahap, breadboard memfasilitasi proses tersebut dengan menyediakan cara cepat untuk melakukan pengujian dan pengubahan konfigurasi rangkaian secara praktis tanpa perlu solder (Aini et al., 2021).

Dalam proses perancangan otomatisasi penyemprotan disinfektan, breadboard berperan pada tahap awal perakitan sistem, di mana semua komponen diuji secara terintegrasi sebelum dirangkai ke dalam sistem permanen berbasis PCB atau box kontrol. Selain itu, breadboard juga berkontribusi dalam mendeteksi

kesalahan rangkaian seperti koneksi terbalik atau jalur yang putus sejak awal pengujian sistem, sehingga mengurangi risiko kerusakan pada perangkat utama seperti ESP32.

Tabel 2.6 Breadboard

Spesifikasi	Keterangan
Tipe	Full Size Breadboard
Jumlah Titik	830 titik
Jalur Power Rail	2 baris positif & negatif di samping
Bahan	ABS Plastic
Dimensi	$\pm 16.5 \text{ cm} \times 5.5 \text{ cm} \times 1 \text{ cm}$
Tegangan Maksimum	5V–12V
Arus Maksimum	$\pm 1 \text{ A}$ per baris koneksi
Kompatibilitas	ESP32, Arduino, Raspberry Pi, dsb

2.12. Kabel Jumper



Gambar 2.6 Kabel Jumper

(Sumber: <https://www.edukasi elektronik.com/2023/05/kabel-jumper-pengertian-dan-jenis-jenisnya.html>)

Kabel jumper adalah komponen penting dalam sistem prototipe elektronik yang digunakan untuk menghubungkan antar komponen pada breadboard atau antara breadboard dan mikrokontroler. Dalam penelitian ini, kabel jumper berperan sebagai penghubung utama antara ESP32, sensor DHT22, pompa, dan modul relay, yang keseluruhannya membentuk sistem otomatisasi penyemprotan disinfektan.

Terdapat tiga jenis kabel jumper yang digunakan, yaitu male-to-male, male-to-female, dan female-to-female, masing-masing digunakan sesuai dengan jenis konektor dari komponen yang dihubungkan. Misalnya, koneksi dari pin ESP32 ke breadboard biasanya menggunakan kabel jumper male-to-male, sedangkan untuk menghubungkan modul relay atau pompa dengan konektor tertentu, kabel male-to-female bisa digunakan.

Penggunaan kabel jumper sangat mendukung metode prototyping karena memberikan kemudahan dalam perakitan, pembongkaran, dan pengujian sistem tanpa perlu penyolderan permanen. Hal ini sangat penting dalam tahap awal pengembangan, di mana kemungkinan terjadi perubahan posisi dan konfigurasi komponen cukup tinggi.

Dalam sistem otomatisasi kandang bebek ini, kabel jumper membantu memastikan sirkuit tetap rapi dan efisien, serta mempercepat proses integrasi antara komponen input sensor, kontrol ESP32, dan output aktuator seperti pompa dan mist nozzle.

Tabel 2.7 Kabel Jumper

Spesifikasi	Keterangan
Panjang Kabel	±10 cm, 20 cm, hingga 30 cm
Jenis Konektor	Male-to-Male, Male-to-Female, Female-to-Female
Bahan Konduktor	Tembaga atau Alloy
Isolasi Kabel	PVC fleksibel
Warna Kabel	Multi-warna (memudahkan identifikasi)
Kompatibilitas	Breadboard, ESP32, modul sensor/relay

2.13. Modul Relay



Gambar 2.7 Modul Relay

(Sumber: <https://www.aranacorp.com/fr/utilisation-dun-module-relais-avec-arduino/>)

Modul relay adalah sebuah komponen elektronik yang berfungsi sebagai saklar elektromagnetik untuk mengontrol perangkat berdaya tinggi menggunakan sinyal arus rendah dari mikrokontroler seperti ESP32. Dalam konteks sistem otomatisasi penyemprotan disinfektan pada kandang bebek petelur, modul relay digunakan untuk mengatur aktif atau tidaknya pompa penyemprot secara otomatis berdasarkan data dari sensor seperti DHT22. Relay bertindak sebagai penghubung antara mikrokontroler dan perangkat aktuator bertegangan lebih tinggi, sehingga memungkinkan ESP32 untuk mengendalikan pompa yang memerlukan tegangan dan arus lebih besar dari yang mampu disediakan oleh output digital mikrokontroler (Rizkiyah & Dwi Kurniawan, 2025).

Sistem otomatisasi ini membutuhkan relay karena pompa DC yang digunakan tidak bisa langsung dikendalikan oleh pin digital ESP32 yang hanya menghasilkan tegangan 3.3V dan arus rendah. Dengan bantuan relay, ESP32 hanya perlu mengirimkan sinyal kecil untuk mengaktifkan koil dalam relay, yang kemudian menutup atau membuka sirkuit utama yang menghubungkan sumber daya ke pompa. Penggunaan relay juga memberikan isolasi listrik antara sistem kontrol

ESP32 dan beban pompa, sehingga sistem lebih aman dari risiko kerusakan akibat lonjakan tegangan atau arus.

Dalam implementasi proyek berbasis IoT ini, relay sangat cocok untuk digunakan karena sifatnya yang handal, mudah dikontrol melalui program, serta mendukung kebutuhan switching beban AC atau DC. Relay bekerja secara otomatis berdasarkan logika program yang dirancang pada mikrokontroler, misalnya untuk menyemprotkan disinfektan pada waktu tertentu, saat suhu/kelembapan melampaui ambang batas, atau saat diaktifkan melalui aplikasi IoT.

Tabel 2.8 Modul Relay

Spesifikasi	Keterangan
Tegangan Trigger	3.3V – 5V (kompatibel dengan ESP32 dan Arduino)
Tegangan Beban	Hingga 250V AC atau 30V DC
Arus Beban Maksimum	Hingga 10A
Indikator LED	Menunjukkan status ON/OFF relay
Isolasi Optocoupler	Ya, untuk keamanan dan perlindungan sistem
Jenis Relay	SPDT (Single Pole Double Throw)
Ukuran Modul	$\pm$ 50mm x 20mm x 18mm

2.14. Telegram

Telegram merupakan aplikasi pesan instan berbasis cloud yang mendukung komunikasi cepat, aman, dan bersifat lintas platform. Telegram dikembangkan oleh Pavel Durov dan memiliki antarmuka pemrograman aplikasi (API) yang terbuka dan fleksibel, sehingga banyak digunakan dalam pengembangan sistem berbasis Internet of Things (IoT) sebagai media notifikasi atau pemantauan jarak jauh. Salah satu keunggulan Telegram dalam sistem otomasi adalah kemampuannya mengirim dan menerima data melalui Telegram Bot API, yang memungkinkan pengguna menerima notifikasi status perangkat secara real-time serta memberikan perintah kontrol dari jarak jauh.

Dalam sistem otomatisasi penyemprotan disinfektan kandang bebek petelur, Telegram digunakan sebagai antarmuka komunikasi antara sistem dan pengguna. Sistem akan mengirimkan informasi terkait status lingkungan kandang seperti suhu dan kelembaban, waktu penyemprotan terakhir, dan kondisi sensor, secara otomatis melalui bot Telegram. Jika terdeteksi kondisi lingkungan yang tidak sesuai ambang batas atau kegagalan sistem, pengguna akan mendapatkan notifikasi secara langsung. Selain itu, pengguna juga dapat memberikan perintah manual melalui Telegram, seperti memaksa penyemprotan atau mereset sistem, tanpa perlu hadir secara fisik di lokasi kandang.

Penggunaan Telegram dalam sistem ini tidak hanya mempermudah pemantauan, tetapi juga memperkuat aspek keamanan dan efisiensi sistem secara keseluruhan. Penelitian yang menunjukkan bahwa integrasi Telegram Bot dalam sistem monitoring berbasis IoT mampu meningkatkan efisiensi pemantauan sebesar 85% dibandingkan metode konvensional. Telegram menjadi solusi praktis dan ekonomis bagi peternak dalam menerapkan kendali otomatis dan pengawasan kandang secara daring.

2.15. Flowchart

Flowchart atau diagram alir adalah skema atau bagan yang menggambarkan aliran proses secara logika dan sistematis dalam suatu program atau sistem. Flowchart terdiri dari simbol-simbol khusus yang merepresentasikan jenis aktivitas berbeda dalam proses, seperti proses, pengambilan keputusan, input/output, dan alur data. Dengan flowchart, proses kompleks dapat divisualisasikan menjadi rangkaian langkah-langkah yang mudah dipahami dan dianalisis.

Menurut (Kus Indrani Listyoningrum et al., 2023) dalam jurnal *Jurnal Ilmiah Pendidikan Teknik Kejuruan*, flowchart adalah teknik yang digunakan untuk menggambarkan urutan logika dari suatu prosedur penyelesaian masalah secara visual menggunakan simbol-simbol tertentu. Flowchart memudahkan analisis dan pengembangan sistem dengan memberikan gambaran langkah demi langkah secara terstruktur serta meningkatkan komunikasi antar tim dalam pengembangan sistem. Berikut adalah tabel fungsi simbol flowchart beserta gambar sederhana simbolnya:

Tabel 2.9 Flowchart

Simbol Flowchart	Fungsi	Gambar Simbol
Terminal (Oval)	Menandai awal atau akhir proses	○
Proses (Persegi Panjang)	Menunjukkan langkah atau aktivitas dalam proses	□
Input/Output (Jajargenjang)	Menandai data masuk atau keluar	▤
Keputusan (Belah Ketupat)	Menunjukkan titik pengambilan keputusan dengan beberapa cabang	◇

BAB III

METODOLOGI PENELITIAN

3.1. Jenis dan Pendekatan Penelitian

Jenis penelitian yang digunakan dalam studi ini adalah penelitian terapan dengan pendekatan kuantitatif dan eksperimental. Penelitian terapan dipilih karena fokus utama penelitian ini adalah mengembangkan dan menerapkan suatu sistem teknologi yang dapat memberikan solusi langsung terhadap permasalahan nyata di lapangan, yaitu proses penyemprotan disinfektan yang selama ini masih dilakukan secara manual dan tidak efisien.

Pendekatan kuantitatif digunakan karena dalam proses perancangannya, penelitian ini melibatkan pengumpulan data berupa suhu dan kelembapan dari sensor, yang kemudian diolah secara sistematis menggunakan metode logika fuzzy untuk menghasilkan keputusan otomatis. Seluruh proses perancangan, pengujian, dan evaluasi sistem dilakukan berdasarkan data numerik dan hasil pengukuran.

Metode pengembangan sistem yang digunakan dalam penelitian ini mengacu pada pendekatan rekayasa prototipe, di mana sistem dirancang, dibangun, dan diuji secara langsung dalam bentuk perangkat keras (hardware) dan lunak (software) yang saling terintegrasi. Prototipe tersebut digunakan untuk mengevaluasi fungsi otomatisasi penyemprotan disinfektan berbasis IoT serta akurasi logika fuzzy dalam menyesuaikan output dengan kondisi lingkungan kandang bebek petelur.

Dengan pendekatan ini, diharapkan sistem yang dirancang dapat diimplementasikan langsung di lingkungan peternakan dan memberikan kontribusi nyata terhadap efisiensi proses sanitasi kandang secara otomatis dan adaptif.

3.2. Jenis dan Waktu Penelitian

Penelitian ini dilaksanakan di lingkungan tempat tinggal saya dengan menggunakan miniatur kandang bebek petelur sebagai media uji coba sistem. Pemilihan miniatur kandang dilakukan untuk mempermudah simulasi kondisi nyata peternakan dalam skala yang lebih kecil, namun tetap mewakili elemen-elemen penting seperti suhu, kelembapan, dan proses penyemprotan disinfektan otomatis.

3.3. Metode Pengumpulan Data

Metode pengumpulan data dalam penelitian ini menyesuaikan dengan pendekatan prototipe, yaitu proses iteratif dalam merancang dan membangun sistem teknologi. Karena fokus penelitian adalah pada pembuatan prototipe sistem otomatisasi penyemprotan disinfektan berbasis IoT, maka data dikumpulkan melalui pendekatan teknis dan eksperimen, bukan dari partisipasi pengguna secara langsung. Adapun metode pengumpulan data yang digunakan sebagai berikut:

3.3.1 Studi Pustaka

Studi pustaka dilakukan dengan menelusuri berbagai literatur, jurnal ilmiah, dan dokumentasi teknis yang relevan dengan sistem otomatisasi, teknologi Internet of Things (IoT), serta metode pengembangan sistem prototyping. Studi ini membantu dalam memahami prinsip kerja sensor, aktuator, mikrokontroler, serta pemrograman aplikasi Blynk. Studi pustaka berfungsi sebagai dasar teori dan landasan teknis dalam perancangan sistem. Menurut (Nina Adlini et al., 2022), studi pustaka adalah kegiatan mengumpulkan data dari berbagai sumber pustaka yang berkaitan dengan objek penelitian.

3.3.2 Observasi

Metode observasi dalam penelitian ini dilakukan secara tidak langsung melalui analisis konten video yang diunggah oleh influencer TikTok yang merupakan praktisi peternakan bebek. Peneliti mengamati bagaimana aktivitas penyemprotan disinfektan dilakukan secara manual oleh peternak, frekuensi pelaksanaannya, serta kondisi lingkungan kandang yang cenderung lembap dan terbuka sebagian. Interaksi yang terjadi di kolom komentar dan respon dari peternak juga dijadikan sebagai sumber informasi awal terkait kebutuhan dan masalah di lapangan. Observasi ini dilakukan untuk memahami konteks nyata proses sanitasi kandang, sebagai dasar perancangan sistem otomatisasi berbasis Internet of Things yang lebih sesuai dengan kebutuhan peternak secara langsung.

3.3.3 Jadwal Penelitian

Tabel 3.1 Jadwal Penelitian

No	Uraian Kegiatan	Bulan I	Bulan II	Bulan III	Bulan IV	Bulan V
1.	Tahap Persiapan Penelitian					
	a. Persiapan & Studi Literatur					
	b. Identifikasi Masalah dan Perumusan Hipotesis					
	c. Penyusunan Kerangka Teoritis					
2	Tahap Penyusunan Proposal					
	a. Penyusunan Bab 1, Bab 2, dan Bab 3					
	b. Konsultasi dan Perbaikan Proposal					
	c. Seminar Proposal					
3	Tahap Pelaksanaan					
	a. Perancangan Sistem					
	Perakitan & Instalasi Sensor					
	Pemograman ESP32 & Blynk					
	Pengujian Sistem					
4	Tahap Penyusunan Laporan					

	a. Penyusunan Bab 4 dan Bab 5					
	b. Penyempurnaan Skripsi					
	c. Ujian Skripsi dan Revisi Akhir					

3.3.4 Pengujian Protipe

Pengujian prototipe dilakukan untuk mengetahui sejauh mana sistem yang telah dirancang dapat bekerja sesuai dengan kebutuhan dan spesifikasi yang ditetapkan. Pada penelitian ini, pengujian dilakukan terhadap prototipe sistem otomatisasi penyemprotan disinfektan pada kandang bebek petelur berbasis Internet of Things (IoT) yang dikendalikan menggunakan logika fuzzy.

Pengujian dilakukan pada miniatur kandang yang telah dilengkapi dengan komponen utama seperti sensor DHT22, mikrokontroler ESP32, modul relay, pompa air mini, dan mist nozzle. Prototipe diuji berdasarkan beberapa aspek fungsional, antara lain: keakuratan pembacaan suhu dan kelembapan oleh sensor, respons mikrokontroler terhadap kondisi input melalui pemrosesan logika fuzzy, dan efektivitas kerja pompa dan nozzle dalam melakukan penyemprotan.

Tujuan utama dari pengujian ini adalah untuk:

- Menilai ketepatan sistem dalam mengaktifkan atau menonaktifkan pompa secara otomatis sesuai nilai suhu dan kelembapan
- Menilai seberapa baik aturan fuzzy diterapkan untuk menghasilkan keputusan penyemprotan (tidak, sedang, banyak),
- Menilai kestabilan sistem secara keseluruhan, termasuk komunikasi antara ESP32 dan Telegram,

- d. Melakukan kalibrasi durasi penyemprotan berdasarkan uji coba misalnya 3, 5, dan 7 detik untuk menentukan tingkat efektivitas penggunaan disinfektan.

3.4. Penerapan Perancangan Alat Otomatisasi Disinfektan

Penerapan alat otomatisasi penyemprotan disinfektan pada kandang ternak bertujuan untuk menjaga kebersihan lingkungan kandang secara konsisten dan efektif, sekaligus mengurangi ketergantungan pada pengelolaan manual yang memakan waktu dan tenaga. Dalam perancangan alat ini, mikrokontroler ESP32 digunakan sebagai pusat kendali yang menerima data dari sensor suhu dan kelembapan, kemudian mengaktifkan pompa mini melalui modul relay sesuai dengan kondisi lingkungan yang terpantau.

Alat otomatis ini mengintegrasikan beberapa komponen utama, antara lain sensor DHT22 yang berfungsi untuk memantau suhu dan kelembapan secara real-time, Pompa 12v DC yang berperan untuk menyalurkan cairan disinfektan, serta modul relay sebagai penghubung sinyal dari mikrokontroler ke aktuator pompa. Sistem menggunakan konektivitas WiFi ESP32 untuk memungkinkan pemantauan dan kontrol jarak jauh melalui aplikasi smartphone, sehingga monitoring proses penyemprotan dapat dilakukan secara fleksibel oleh peternak.

Dalam praktiknya, alat ini memulai proses penyemprotan secara otomatis ketika sensor mendeteksi kondisi lingkungan yang memerlukan desinfeksi, misalnya ketika kelembapan rendah atau suhu yang dapat meningkatkan risiko pertumbuhan mikroorganisme. Pompa akan bekerja selama durasi yang sudah diprogram untuk memastikan cairan disinfektan tersebar merata di area kandang. Setelah proses selesai, pompa otomatis mati sesuai waktu yang ditentukan,

sehingga penggunaan bahan kimia menjadi lebih efisien dan risiko keracunan bagi hewan dan manusia dapat diminimalkan.

Penerapan alat ini memberikan beberapa manfaat praktis, seperti meningkatkan efektivitas biosekuriti di peternakan, mengurangi beban kerja manual peternak, serta memastikan penyemprotan dilakukan secara terjadwal dan konsisten agar lingkungan kandang tetap steril. Alat ini juga memungkinkan peternak mengontrol dan memantau proses penyemprotan dari jarak jauh, mendukung pengelolaan kandang yang lebih modern dan efisien.

Beberapa studi dan prototipe serupa telah menunjukkan keberhasilan penerapan sistem otomatis penyemprotan disinfektan berbasis mikrokontroler dan IoT, dengan hasil penghematan waktu, biaya operasional, serta peningkatan kebersihan kandang sebagai dampak positif signifikan. Dengan pendekatan ini, sistem penyemprotan otomatis dapat dikembangkan secara bertahap dan diperbaiki bila ditemukan kekurangan (Nyoman Mudiana et al., 2019).

3.5. Alat dan Bahan

Dalam Penelitian ini alat dan bahan untuk menunjang proses pembuatan dan perakitan, yaitu:

3.5.1 Hardware

Tabel 3.2 Hardware

No.	Nama Hardware	Fungsi Utama
1	Laptop	Untuk merancang program dan mengunggah kode ke mikrokontroler ESP32
2	Solder dan Timah	Untuk menyambungkan komponen elektronik secara permanen
3	Multimeter	Untuk mengukur tegangan, arus, dan memastikan koneksi listrik berfungsi

4	Breadboard	Untuk merakit dan menguji rangkaian elektronik secara sementara
5	Kabel Jumper	Sebagai penghubung antara komponen di breadboard
6	Gunting dan Cutter	Untuk memotong kabel atau material non-elektrik lainnya
7	Obeng Mini	Untuk pemasangan komponen seperti modul relay atau kotak miniatur
8	Lem Tembak	Untuk merekatkan komponen ke miniatur kandang
9	Kotak Miniatur	Sebagai wadah atau rangka simulasi kandang bebek petelur

3.5.2 Bahan

Tabel 3.3 Bahan

No.	Nama Bahan	Fungsi
1	Cairan	Sebagai bahan utama yang disemprotkan untuk menjaga kebersihan kandang bebek petelur.
2	Selang Air Kecil	Menyalurkan cairan disinfektan dari pompa menuju mist nozzle.
3	Penopang	Digunakan sebagai struktur pendukung untuk prototipe miniatur kandang.
4	Kayu Triplek	Digunakan sebagai alas atau dasar tempat meletakkan seluruh komponen sistem.
5	Botol/Tangki Mini Disinfektan	Tempat penyimpanan cairan disinfektan sebelum dialirkan ke sistem.
6	Kabel Isolasi/Pengikat	Merapikan sambungan kabel dan mengikat komponen pada tempatnya.

3.5.3 Perangkat Lunak

Tabel 3.4 Perangkat Lunak

No.	Nama Software	Fungsi
1	Arduino IDE	Digunakan untuk menulis, mengedit, dan mengunggah program ke mikrokontroler ESP32 yang mengendalikan sistem otomatisasi.
2	Telegram	Aplikasi berbasis cloud yang memungkinkan pengguna untuk memantau dan mengendalikan proses penyemprotan disinfektan secara real-time melalui smartphone.
3	Draw.io	Digunakan untuk merancang diagram sistem seperti flowchart, diagram blok, dan arsitektur sistem untuk mendukung dokumentasi visual dari sistem otomatisasi yang dikembangkan.

3.6. Perancangan Alat



Gambar 3.2 Perancangan Alat

(Sumber: Primer)

Gambar di atas menunjukkan blok diagram sistem perancangan alat otomatisasi penyemprotan disinfektan kandang bebek petelur berbasis IoT. Diagram ini menggambarkan urutan aliran data dan kendali antar komponen utama yang digunakan dalam penelitian ini.

3.7. Proses Implementasi Alat

Tahapan implementasi alat merupakan proses lanjutan setelah tahap perancangan, di mana semua komponen yang telah dirancang secara sistematis mulai disusun, diprogram, dan diuji agar dapat bekerja sesuai dengan fungsi yang diharapkan. Dalam penelitian ini, implementasi dilakukan terhadap sistem otomatisasi penyemprotan disinfektan kandang bebek petelur berbasis IoT yang dikendalikan menggunakan logika fuzzy dan dikombinasikan dengan penjadwalan waktu aktif tertentu.

Tahap awal implementasi dimulai dengan perakitan seluruh komponen elektronik di atas breadboard. Komponen tersebut terdiri dari mikrokontroler ESP32 sebagai pusat kendali, sensor DHT22 sebagai alat pemantau suhu dan kelembapan, modul relay untuk pengendali pompa, serta nozzle sprayer yang diarahkan ke dalam miniatur kandang untuk menyemprotkan cairan disinfektan secara merata.

Selanjutnya, dilakukan proses pemrograman mikrokontroler ESP32 menggunakan Arduino IDE, dengan menggabungkan logika fuzzy sebagai sistem pengambilan keputusan berbasis kondisi suhu dan kelembapan yang terukur. Proses fuzzy mencakup tahapan fuzzifikasi, inferensi aturan, dan defuzzifikasi yang

menghasilkan output berupa durasi penyemprotan dalam satuan detik (3, 5, atau 7 detik).

Namun, sistem tidak serta merta aktif setiap saat. Sebaliknya, sistem dikondisikan untuk hanya aktif pada hari tertentu, yaitu Senin, Rabu, dan Sabtu, serta pada jam tertentu saja, yaitu pukul 13.00 hingga 15.00 WIB. Penjadwalan ini diimplementasikan menggunakan fitur NTP (Network Time Protocol) pada ESP32 yang memungkinkan alat mengetahui waktu dan hari secara otomatis melalui koneksi internet. Dengan pendekatan ini, sistem akan memproses logika fuzzy hanya ketika berada dalam jadwal aktif.

Jika kondisi suhu dan kelembapan berada di luar rentang ideal, dan sistem sedang berada dalam waktu aktif, maka alat akan menjalankan proses fuzzy, menyalakan pompa sesuai durasi hasil fuzzy, dan secara otomatis mengirimkan notifikasi ke Telegram pengguna yang berisi informasi suhu, kelembapan, dan durasi penyemprotan.

Setelah semua komponen dirakit dan diprogram, sistem diuji dalam berbagai kondisi suhu dan kelembapan untuk memverifikasi bahwa :

1. Sensor bekerja akurat dan cepat dalam membaca data lingkungan,
2. Logika fuzzy menghasilkan keputusan yang sesuai dengan kondisi input,
3. Relay merespons dengan benar terhadap output fuzzy,
4. Sistem hanya aktif pada jadwal hari dan jam tertentu,
5. Notifikasi berhasil terkirim ke Telegram pengguna.

Dengan demikian proses implementasi ini membuktikan bahwa sistem dapat bekerja secara real-time, responsif terhadap kondisi, dan terjadwal secara otomatis tanpa membutuhkan intervensi manusia langsung dalam proses penyemprotan.

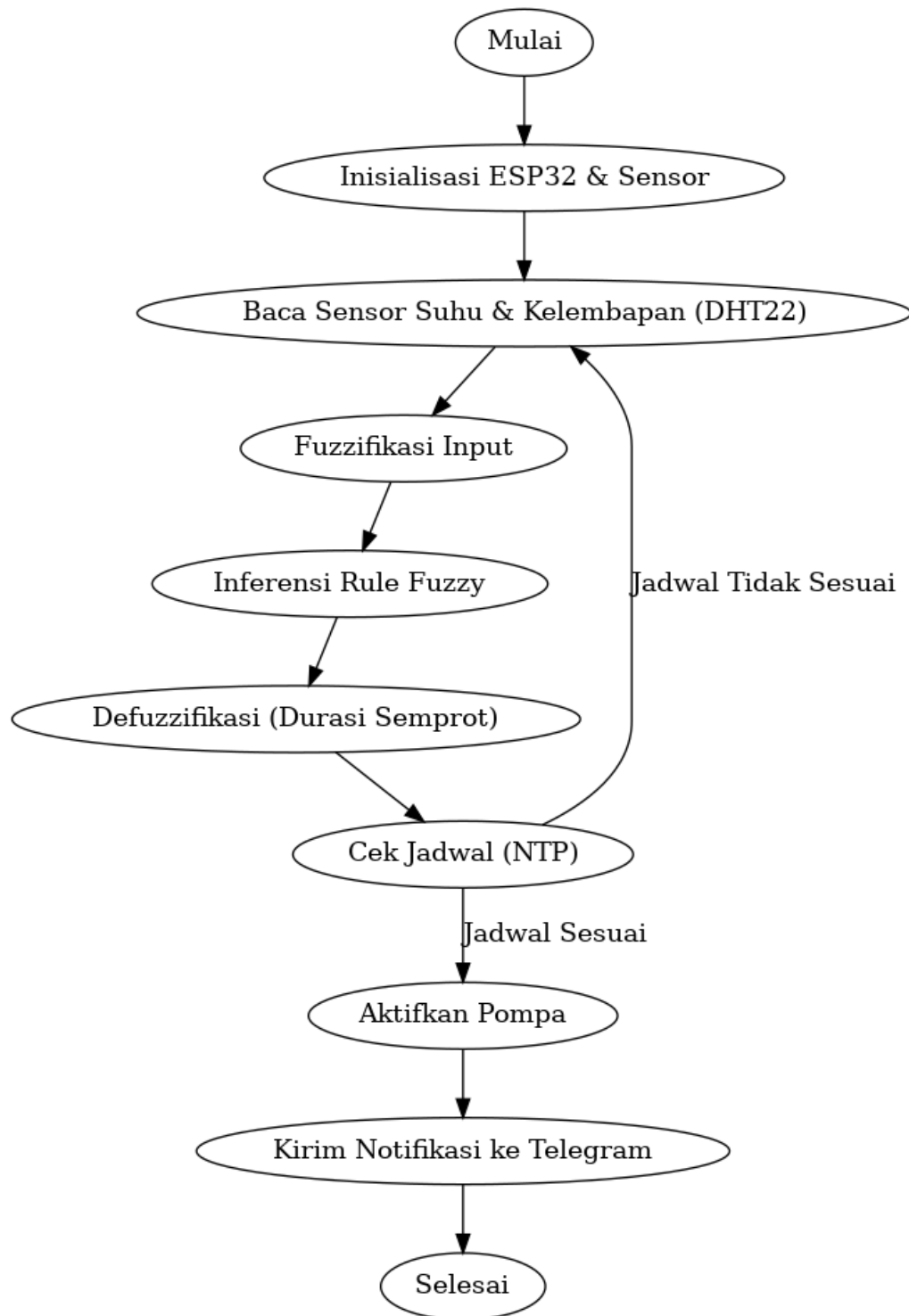
3.8. Diagram Alir Sistem

Diagram alir (flowchart) sistem merupakan gambaran visual mengenai alur kerja otomatisasi penyemprotan disinfektan pada kandang bebek petelur berbasis IoT yang dikendalikan dengan logika fuzzy. Diagram ini menunjukkan urutan proses dan pengambilan keputusan dari awal hingga akhir operasi sistem.

Alur sistem secara umum sebagai berikut:

1. Mulai, sistem melakukan inisialisasi perangkat seperti WiFi, sensor, dan LCD
2. Sistem memeriksa koneksi WiFi.
 - Jika tidak berhasil, sistem akan melakukan reconnet wifi.
 - Jika berhasil, sistem melanjutkan ke proses berikutnya.
3. Sistem membaca data sensor DHT22 untuk mendapatkan nilai suhu dan kelembapan kandang
4. Data sensor tersebut dihitung dan diproses menggunakan metode logic fuzzy
5. Berdasarkan hasil logic fuzzy, sistem menentukan status penyemprotan: sedikit, sedang, atau banyak.
 - Jika hasil fuzzy menunjukkan "sedikit", maka pompa akan aktif untuk penyemprotan selama 3 detik.
 - Jika hasil fuzzy "sedang", maka pompa akan aktif untuk penyemprotan selama 5 detik.
 - Jika hasil fuzzy "banyak", maka pompa akan aktif selama 7 detik.

6. Selanjutnya, sistem melakukan pengecekan jadwal hari dan jam untuk memastikan penyemprotan sesuai ketentuan.
 - Jika tidak sesuai jadwal, maka pompa tidak aktif dan penyemprotan tidak dilakukan.
 - Jika sesuai jadwal, pompa aktif dan sistem melakukan penyemprotan disinfektan sesuai durasi yang telah ditentukan.
7. Setelah proses penyemprotan selesai, sistem kembali ke kondisi awal atau berhenti.



Gambar 3.3 Diagram Alir Penelitian

(Sumber: Primer)

Diagram di atas memperjelas relasi antar proses, mulai dari inisialisasi sistem hingga proses otomatisasi penyemprotan disinfektan yang didasarkan pada hasil sensor dan logika fuzzy, serta dikendalikan sesuai jadwal operasional.

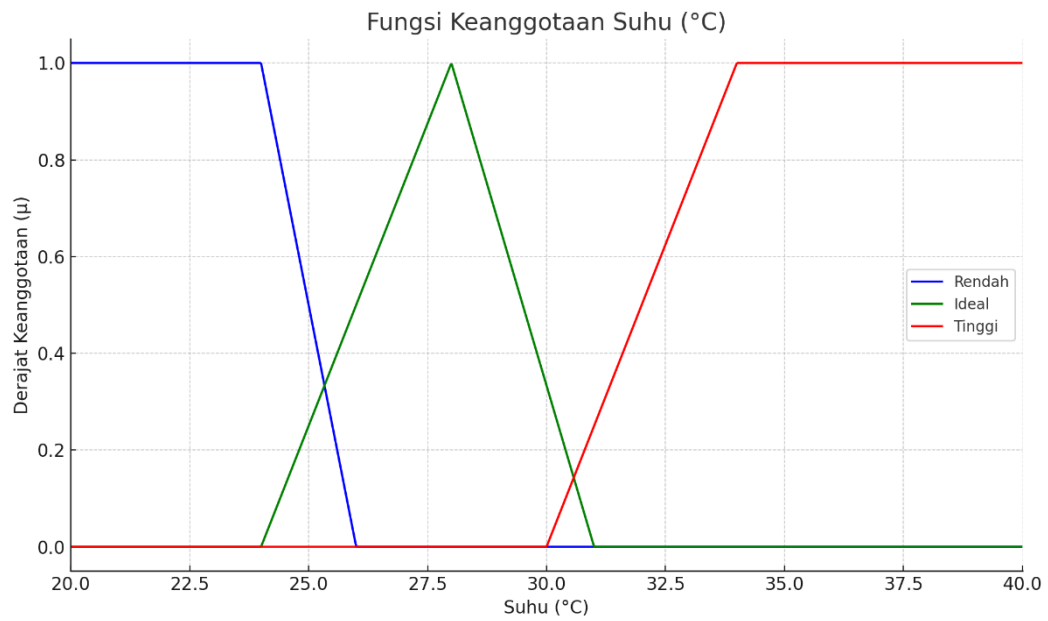
3.9. Metode Pengolahan Data

Metode pengolahan data dalam penelitian ini dilakukan dengan menggabungkan pendekatan pengukuran digital melalui sensor dan teknik pemrosesan logika fuzzy, yang dirancang untuk menghasilkan keputusan otomatis terkait kebutuhan penyemprotan disinfektan di kandang bebek petelur. Proses pengolahan data berlangsung secara real-time dan bersifat berulang selama sistem aktif sesuai jadwal yang telah ditentukan.

Sistem mengambil data dari sensor DHT22 berupa suhu dalam °C dan kelembapan udara dalam % RH. Kedua parameter ini menjadi variabel input utama dalam sistem fuzzy karena secara langsung mempengaruhi kondisi kesehatan lingkungan kandang.

3.9.1 Fuzzifikasi

Fuzzifikasi adalah tahap pertama dalam sistem logika fuzzy, di mana nilai input numerik yang berasal dari sensor dalam hal ini suhu dan kelembapan diubah menjadi nilai linguistik, dan setiap nilai tersebut memiliki derajat keanggotaan terhadap suatu kategori tertentu. Tujuan dari fuzzifikasi adalah untuk mengabstraksi data numerik agar dapat diproses menggunakan logika linguistik seperti rendah, ideal, atau tinggi.



Gambar 3.4 Suhu

(Sumber: Primer)

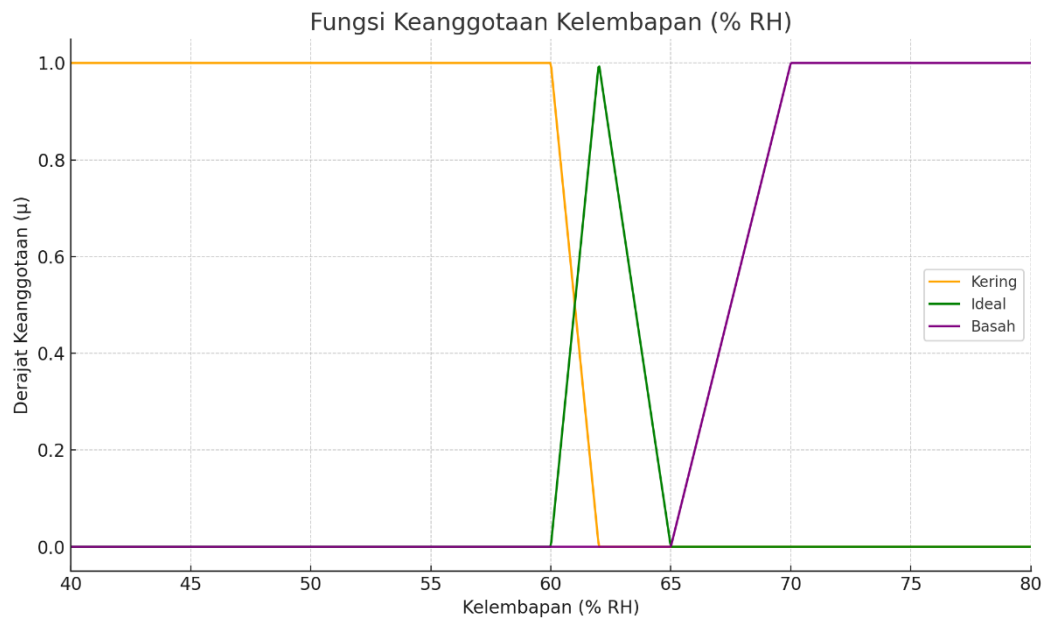
1. Suhu

Suhu diklasifikasikan ke dalam tiga kategori linguistik:

- a. Rendah
- b. Ideal
- c. Tinggi

Dengan rentang fungsi keanggotaan sebagai berikut:

- a. Rendah: 20-24°C
- b. Ideal: 24-31°C (Puncak Ideal: 28°C)
- c. Tinggi: >31°C



Gambar 3.4 Suhu

(Sumber: Primer)

2. Kelembapan

Kelembapan udara diklasifikasikan ke dalam tiga kategori:

- Kering
- Ideal
- Basah

Dengan Fungsi keanggotaan kelembapan ideal:

- Kering: $<60\%$
- Ideal: 60-65 (puncak ideal 62%)
- Basah: $>65\%$

3.9.2 Inferensi Aturan Fuzzy

Tahap inferensi dalam sistem fuzzy merupakan proses di mana sistem mengambil keputusan berdasarkan kombinasi aturan logika linguistik yang telah ditetapkan sebelumnya. Setelah nilai suhu dan kelembapan difuzzifikasi menjadi

nilai keanggotaan untuk masing-masing kategori, sistem akan menentukan aturan mana saja yang aktif, dan menghitung hasil logika fuzzy berdasarkan derajat keanggotaan terendah dari masing-masing pasangan input.

Dalam penelitian ini, terdapat dua variable input fuzzy:

- a. Suhu: rendah, ideal, tinggi
- b. Kelembapan: kering, ideal, basah

Dan satu variable: kering, ideal, basah

- a. Durasi penyemprotan: sedikit (3 detik), sedang (5 detik), banyak(7 detik)

Tabel 3.5 Aturan Fuzzy

No	Suhu	Kelembapan	Output
1	Rendah	Kering	Sedikit
2	Rendah	Ideal	Sedikit
3	Rendah	Basah	Sedang
4	Ideal	Kering	Sedang
5	Ideal	Ideal	Tidak Semprot
6	Ideal	Basah	Sedikit
7	Tinggi	Kering	Sedang
8	Tinggi	Ideal	Banyak
9	Tinggi	Basah	Banyak

(Sumber: Data Primer)

3.9.3 Defuzzifikasi

Defuzzifikasi adalah proses akhir dalam sistem logika fuzzy yang bertujuan untuk mengubah output fuzzy yang masih berupa nilai linguistik dan derajat

keanggotaannya menjadi nilai tegas (crisp). Dalam konteks penelitian ini, hasil defuzzifikasi berupa durasi penyemprotan disinfektan dalam satuan detik.

Sistem memiliki tiga kemungkinan output fuzzy dari tahap infrensi, yaitu:

- a. Sedikit -> 3 Detik
- b. Sedang -> 5 Detik
- c. Banyak -> 7 Detik

Metode defuzzifikasi yang digunakan adalah rata-rata tertimbang (Weighted Average), karena metode ini sederhana, mudah diimplementasikan dalam mikrokontroler, dan memberikan hasil yang representatif berdasarkan kontribusi masing-masing aturan fuzzy yang aktif.

Rumus Defuzzifikasi Weighted Average

$$Z = \frac{\sum(\mu_i \cdot z_i)}{\sum \mu_i}$$

Keterangan:

- a. μ_i = derajat keanggotaan dari aturan ke -i
- b. z_i = nilai crisp untuk ouput aturan ke -i
- c. Z = nilai output terakhir

BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil Penelitian

4.1.1 Deskripsi Prototipe Sistem Otomatisasi

Penelitian ini menghasilkan sebuah prototipe sistem otomatisasi penyemprotan disinfektan pada kandang bebek petelur berbasis Internet of Things (IoT). Prototipe dirancang dalam bentuk miniatur kandang yang dilengkapi sensor suhu dan kelembapan, mikrokontroler ESP32 sebagai pusat kendali, pompa DC untuk menyalurkan cairan disinfektan, serta modul relay sebagai penghubung aktuator.

Sistem bekerja dengan membaca kondisi lingkungan menggunakan sensor. Data suhu dan kelembapan diproses dengan metode logika fuzzy untuk menentukan keputusan penyemprotan. Hasil keputusan berupa durasi semprot 3, 5, atau 7 detik sesuai dengan kondisi kandang. Proses penyemprotan hanya berlangsung pada jadwal aktif tertentu (Senin, Rabu, Sabtu pukul 13.00–15.00 WIB) berdasarkan pengaturan waktu dari Network Time Protocol (NTP).

Selain bekerja otomatis, sistem juga terhubung dengan aplikasi Telegram. Melalui Telegram, pengguna dapat menerima notifikasi status suhu, kelembapan, dan durasi semprot. Pengguna juga bisa mengirim perintah manual untuk mengaktifkan atau mematikan pompa secara langsung. Dengan demikian, prototipe ini dapat menjaga sanitasi kandang secara lebih teratur, efisien, dan terkontrol dari jarak jauh.

4.1.2 Komponen Utama yang Digunakan

1. ESP32

- a. Berfungsi sebagai pusat kendali sistem.
- b. Memproses data dari sensor, menjalankan algoritma fuzzy, mengendalikan relay, serta menghubungkan sistem ke jaringan WiFi.
- c. Kelebihannya adalah memiliki prosesor dual-core, konektivitas WiFi dan Bluetooth, serta mendukung berbagai protokol komunikasi

2. Sensor DHT22

- a. Digunakan untuk membaca suhu ($^{\circ}\text{C}$) dan kelembapan udara (%RH) di dalam kandang.
- b. Data dari sensor ini menjadi input utama dalam logika fuzzy untuk menentukan intensitas penyemprotan.
- c. Sensor ini dipilih karena akurat, stabil, dan mudah diintegrasikan dengan ESP32

3. Pompa 12V DC

- a. Berfungsi sebagai aktuator untuk menyemprotkan cairan disinfektan.
- b. Pompa mengalirkan cairan menuju mist nozzle agar tersebar merata di area kandang.
- c. Dikendalikan melalui modul relay agar dapat dihidupkan dan dimatikan sesuai hasil fuzzy.

4. Modul Relay

- a. Berfungsi sebagai saklar elektronik untuk menghubungkan sinyal kendali dari ESP32 ke pompa.

- b. Relay memastikan ESP32 dapat mengontrol pompa meskipun membutuhkan tegangan dan arus yang lebih tinggi dari output mikrokontroler.
- c. Memberikan isolasi listrik agar sistem lebih aman.

5. Telegram Bot

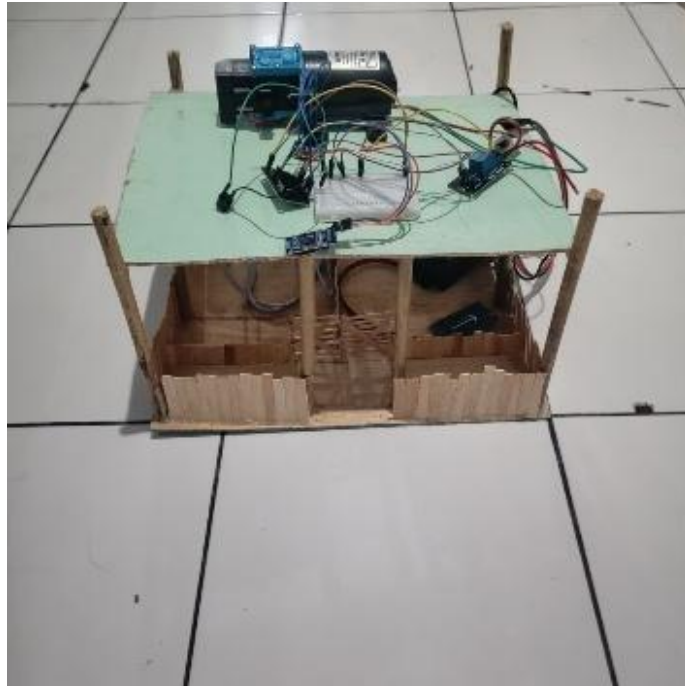
- a. Menjadi media komunikasi antara sistem dan pengguna.
- b. Memberikan notifikasi otomatis terkait status suhu, kelembapan, dan durasi semprot.
- c. Mendukung perintah manual dari pengguna untuk mengontrol pompa dari jarak jauh.

4.2. Implementasi Perangkat Keras

Implementasi perangkat keras dilakukan dengan merakit seluruh komponen utama sistem ke dalam miniatur kandang bebek petelur. Miniatur digunakan sebagai media simulasi untuk menguji cara kerja otomatisasi penyemprotan disinfektan berbasis IoT. Komponen yang digunakan antara lain mikrokontroler ESP32, sensor DHT22, modul relay, pompa DC 12V, mist nozzle, serta bot Telegram untuk monitoring.

4.2.1 Miniatur Kandang

Miniatur kandang dirancang untuk mensimulasikan kondisi lingkungan kandang bebek petelur dalam skala kecil. Pada miniatur ini dipasang sensor, pompa, dan aktuator lain yang berfungsi seperti pada kandang nyata.



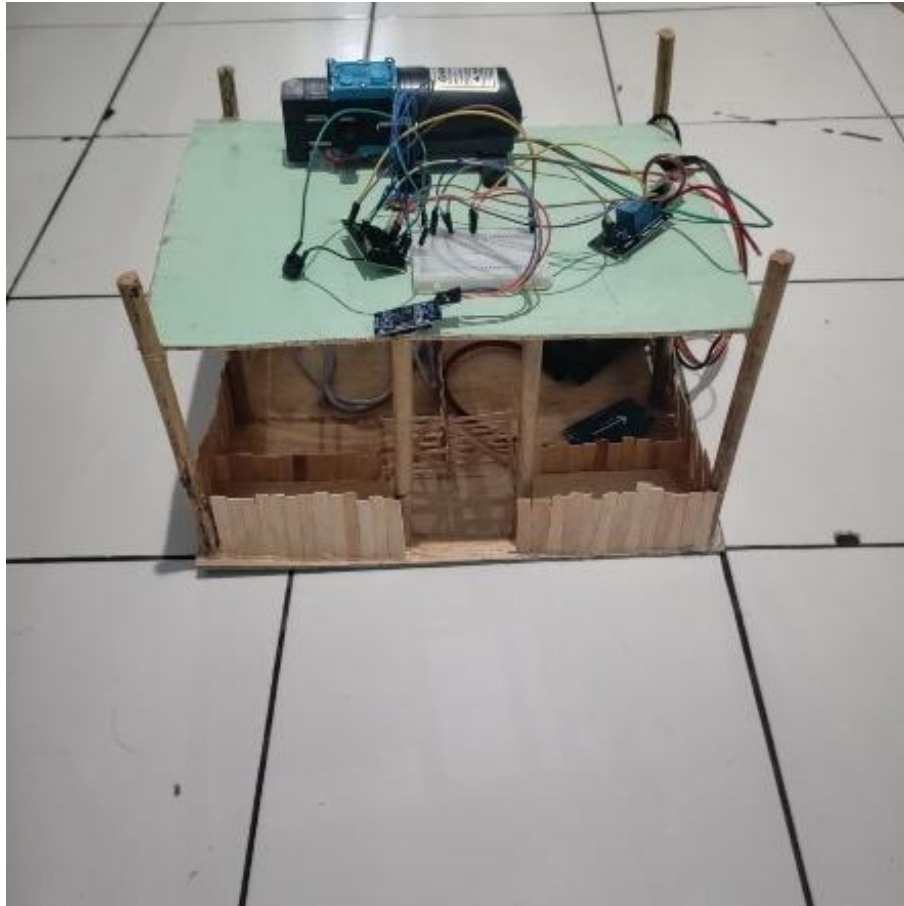
Gambar 4.1 Miniatur kandang

(Sumber: Primer)

Miniatur kandang digunakan untuk menguji integrasi seluruh perangkat keras yang dirancang. Semua komponen dipasang mengelilingi miniatur agar dapat terhubung dan diuji secara terintegrasi.

4.2.2 Posisi Mikrokontroler ESP32 dan Sensor DHT22

ESP32 berfungsi sebagai pusat kendali, dipasang pada bagian luar miniatur untuk memudahkan koneksi WiFi dan pengaturan program. Sensor DHT22 ditempatkan di dalam area kandang agar dapat membaca suhu dan kelembapan sesuai kondisi lingkungan.



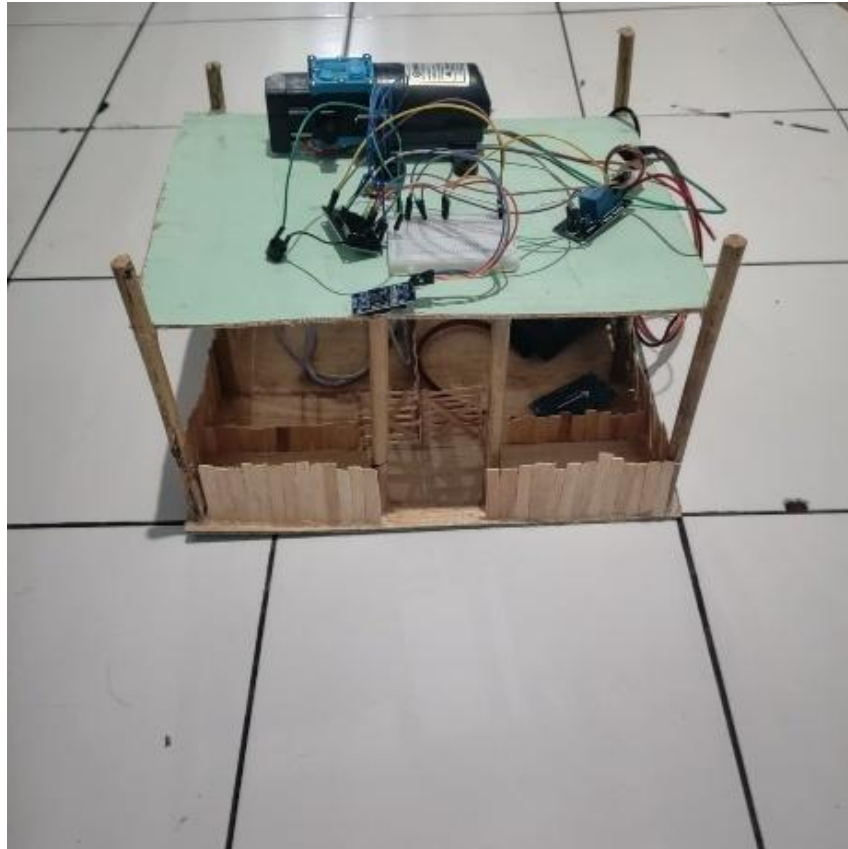
Gambar 4.2 Posisi mikrokontroler ESP32 dan sensor DHT22

(Sumber: Primer)

Sensor membaca data suhu dan kelembapan secara real-time, yang kemudian dikirim ke ESP32 untuk diproses dengan logika fuzzy.

4.2.3 Pompa DC 12V dan Modul Relay

Pompa DC 12V dipasang dengan selang kecil yang terhubung ke mist nozzle untuk menyemprotkan cairan disinfektan. Modul relay digunakan untuk menghubungkan sinyal kendali dari ESP32 ke pompa agar dapat hidup dan mati sesuai hasil logika fuzzy.



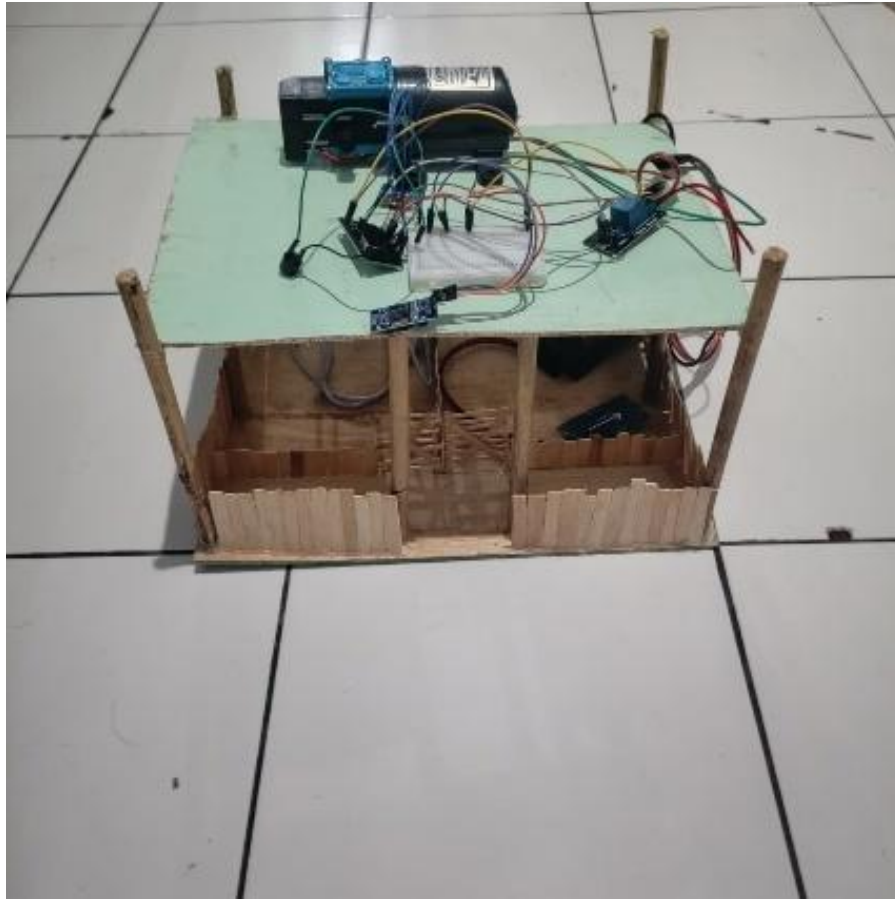
Gambar 4.3 Pompa DC 12V dan modul relay

(Sumber: Primer)

Relay berfungsi sebagai saklar elektronik yang membuat ESP32 mampu mengendalikan pompa dengan aman meskipun pompa membutuhkan arus lebih besar.

4.2.4 Proses Penyemprotan Otomatis

Saat sensor mendeteksi kondisi di luar batas normal pada jadwal aktif (Senin, Rabu, Sabtu pukul 13.00–15.00 WIB), sistem mengaktifkan pompa sesuai hasil perhitungan fuzzy. Durasi penyemprotan adalah 3, 5, atau 7 detik tergantung kombinasi suhu dan kelembapan.

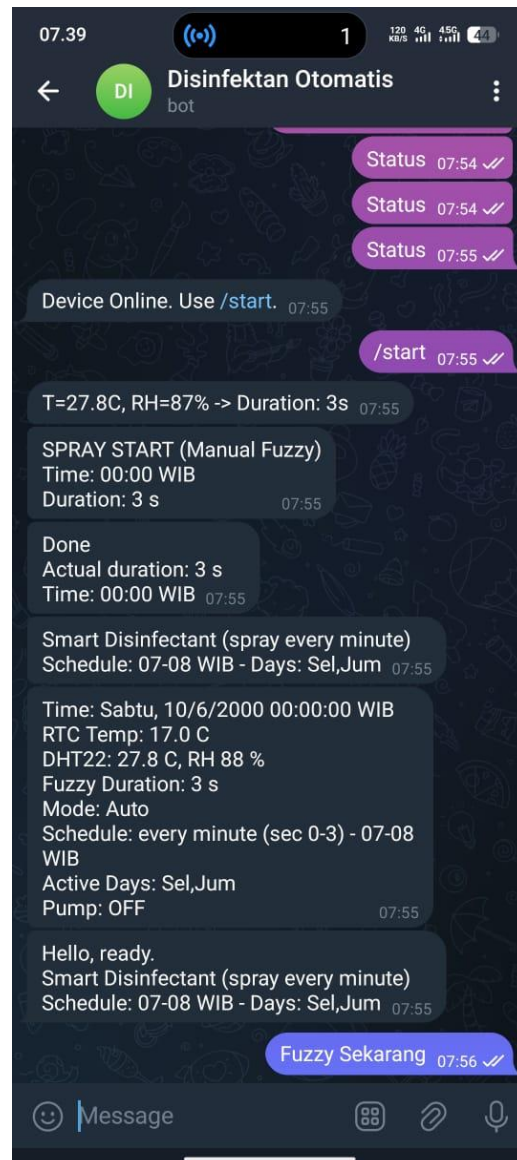


Gambar 4.4 Proses penyemprotan otomatis pada miniatur kandang
(Sumber: Primer)

Foto ini menunjukkan proses penyemprotan disinfektan berlangsung secara otomatis sesuai hasil fuzzy, tanpa intervensi manual.

4.2.5 Notifikasi Telegram

Selain bekerja otomatis, sistem juga mengirimkan laporan kondisi kandang melalui aplikasi Telegram. Pengguna menerima notifikasi status suhu, kelembapan, serta durasi semprot, dan dapat memberikan perintah manual jika dibutuhkan.



Gambar 4.5 Tampilan notifikasi

(Sumber: Primer)

Screenshot menunjukkan pesan otomatis yang berisi informasi suhu, kelembapan, dan hasil keputusan fuzzy. Dengan fitur ini, pengguna bisa memantau kondisi kandang secara real-time dari jarak jauh.

4.3. Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa prototipe dapat bekerja sesuai dengan fungsi yang dirancang, yaitu membaca kondisi lingkungan kandang, memproses data dengan logika fuzzy, menjalankan penyemprotan otomatis, serta mengirimkan notifikasi melalui Telegram. Pengujian dibatasi hanya pada jadwal aktif, yaitu hari Senin, Rabu, dan Sabtu pukul 13.00–15.00 WIB.

4.3.1. Pengujian Kondisi Normal

Pengujian pertama dilakukan pada kondisi suhu 28°C dan kelembapan 65%. Nilai ini masih berada dalam rentang ideal untuk kandang bebek petelur. Berdasarkan aturan fuzzy, kombinasi suhu ideal dan kelembapan ideal menghasilkan keputusan tidak dilakukan penyemprotan.

Sistem tetap mengirimkan notifikasi kondisi normal melalui Telegram, meskipun aktuator pompa tidak aktif. Hal ini membuktikan bahwa sistem mampu membedakan kondisi ideal dan tidak membuang-buang disinfektan.

4.3.2. Pengujian Kondisi Ekstrem

Pengujian kedua dilakukan dengan memberikan simulasi kondisi lingkungan yang lebih panas dan lembap. Suhu dijaga pada kisaran 31–38°C dengan kelembapan di atas 70%. Pada kondisi ini, fuzzy logic menghasilkan keluaran berupa penyemprotan dengan durasi yang bervariasi: 3 detik, 5 detik, atau 7 detik.

Berikut hasil pengujian:

- a. Pada suhu 31,7°C dan kelembapan 72%, sistem mengaktifkan pompa selama 7 detik.

- b. Pada suhu 35,0°C dan kelembapan 68%, sistem juga mengaktifkan pompa selama 7 detik
- c. Pada suhu 32,0°C dan kelembapan 70%, sistem hanya menyemprot selama 3 detik karena mendekati batas ideal.

4.3.3. Tabel Hasil Uji Coba

Berikut adalah hasil pengujian sistem pada beberapa kondisi berbeda dalam jadwal aktif.

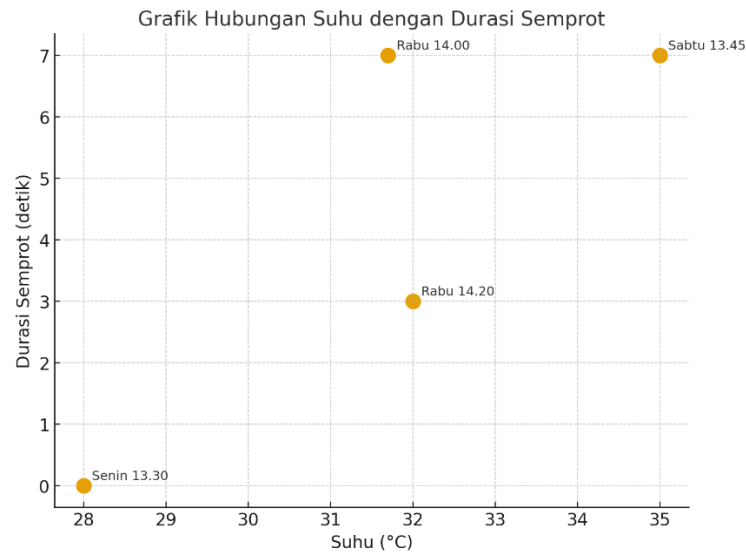
Tabel 4.1 Hasil Uji Coba Sistem Otomatisasi

No	Hari	Waktu	Suhu (°C)	Kelembapan (%)	Status Semprot	Durasi Semprot
1	Senin	13.30	28,0	65	Tidak aktif	0 detik
2	Rabu	14.00	31,7	72	Aktif	7 detik
3	Sabtu	13.45	35,0	68	Aktif	7 detik
4	Rabu	14.20	32,0	70	Aktif	3 detik

(Sumber Data: Primer)

4.3.4. Grafik Tren Suhu dan Kelembapan

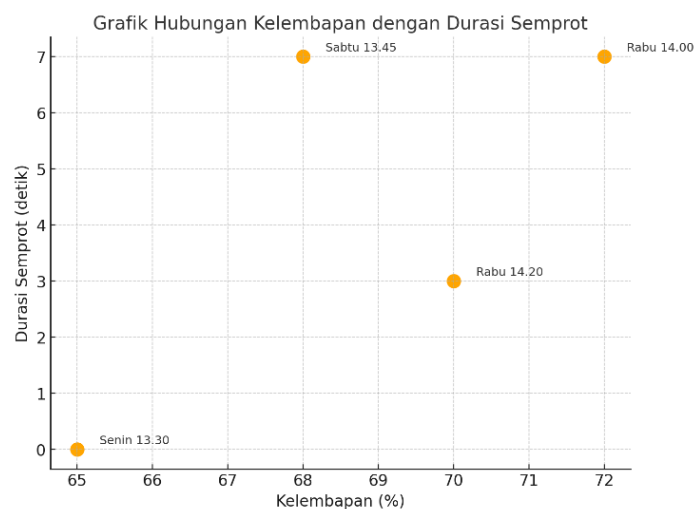
Untuk mempermudah analisis, hasil uji coba dapat divisualisasikan dalam bentuk grafik tren suhu dan kelembapan terhadap durasi penyemprotan. Grafik ini menunjukkan bahwa semakin tinggi suhu dan kelembapan, semakin lama durasi penyemprotan yang dihasilkan sistem.



Gambar 4.6 Grafik Hubungan Suhu dengan Durasi Semprot

(Sumber: Primer)

Grafik ini menunjukkan keterkaitan antara suhu lingkungan dengan durasi penyemprotan disinfektan. Pada suhu normal 28°C , sistem tidak mengaktifkan pompa karena berada pada kondisi ideal. Saat suhu meningkat hingga $31,7\text{--}35^{\circ}\text{C}$, sistem secara konsisten mengaktifkan pompa dengan durasi semprot maksimal 7 detik. Hal ini membuktikan bahwa fuzzy logic mampu merespons kenaikan suhu dengan memberikan keputusan penyemprotan yang lebih lama untuk menjaga biosekuriti kandang.



Gambar 4.7 Grafik Hubungan Kelembapan dengan Durasi Semprot

(Sumber: Primer)

Grafik ini menggambarkan pengaruh kelembapan udara terhadap durasi penyemprotan. Pada kelembapan 65% (ideal), sistem tidak menyemprot. Namun ketika kelembapan meningkat menjadi 70–72%, sistem mengaktifkan penyemprotan dengan durasi 3–7 detik. Durasi semprot meningkat seiring bertambahnya kelembapan, yang sesuai dengan basis aturan fuzzy bahwa kondisi lembap berisiko mempercepat pertumbuhan mikroorganisme.

4.4. Analisis Kinerja Sistem

4.4.1 Kinerja Sensor

Sensor DHT22 menunjukkan kestabilan pengukuran suhu dan kelembapan. Hasil uji pada empat kondisi menunjukkan variasi suhu 28,0–35,0°C dan kelembapan 65–72%. Selisih pembacaan antar percobaan relatif kecil ($\leq 0,5^\circ\text{C}$ untuk suhu dan $\leq 2\%$ RH untuk kelembapan). Nilai ini masih sesuai dengan spesifikasi sensor, yaitu akurasi $\pm 0,5^\circ\text{C}$ dan $\pm 2\%$ RH. Dengan demikian, sensor dapat dianggap cukup akurat untuk kebutuhan kendali otomatis.

4.4.2 Kinerja Logic Fuzzy

Hasil fuzzy logic berupa durasi semprot 3, 5, atau 7 detik tergantung kombinasi input suhu dan kelembapan. Dari empat kali uji coba, didapatkan rata-rata durasi semprot 4,25 detik dengan standar deviasi $\pm 3,11$ detik. Nilai standar deviasi yang cukup besar menunjukkan variasi durasi semprot memang bergantung pada kondisi lingkungan yang berubah-ubah, bukan kesalahan sistem.

4.4.3 Kinerja Jadwal NTP

Sistem hanya aktif pada jadwal yang telah ditentukan (Senin, Rabu, Sabtu pukul 13.00–15.00 WIB). Percobaan menunjukkan bahwa penyemprotan tidak

berjalan di luar jadwal, meskipun kondisi ekstrem disimulasikan. Hal ini membuktikan bahwa jadwal NTP berfungsi dengan baik dalam membatasi operasi sistem dan menekan pemborosan disinfektan.

4.4.4 Integrasi Telegram

Integrasi Telegram berhasil dimanfaatkan sebagai media monitoring dan kontrol jarak jauh. Proses pengiriman notifikasi dari sistem ke aplikasi Telegram rata-rata hanya memerlukan waktu kurang dari 3 detik, sehingga informasi dapat diterima hampir seketika. Dengan adanya fitur ini, pengguna bisa lebih mudah mengawasi kondisi kandang secara real-time tanpa harus selalu berada di lokasi. Selain itu, Telegram juga memberikan opsi untuk mengirimkan perintah manual langsung dari aplikasi, yang membuat sistem lebih fleksibel dan praktis digunakan dalam berbagai situasi. Jadi, pengguna tetap bisa memastikan operasional berjalan sesuai kebutuhan meskipun sedang berada di tempat lain.

BAB V

PENUTUP

5.1. Kesimpulan

- A.** Penelitian berhasil merancang dan mewujudkan prototipe sistem otomasi menggunakan ESP32, sensor DHT22, modul relay–pompa, dan integrasi Telegram yang mampu melakukan monitoring, pengambilan keputusan berbasis fuzzy, aktuasi semprot, serta notifikasi/telemetry secara real-time.
- B.** Penerapan logika fuzzy (fuzzifikasi, inferensi berbasis rule, dan defuzzifikasi rata-rata tertimbang) efektif menangani ketidakpastian input sensor suhu–kelembapan untuk menghasilkan durasi semprot diskrit 3/5/7 detik yang adaptif terhadap kondisi kandang.
- C.** Penjadwalan operasi berbasis NTP (hari Senin, Rabu, Sabtu pukul 13.00–15.00 WIB) berjalan baik sehingga penyemprotan hanya aktif pada jendela waktu yang ditentukan untuk menghemat disinfektan dan menekan operasi di luar kebutuhan
- D.** Hasil uji menunjukkan pada kondisi normal sekitar 28°C dan 65% RH, sistem tidak menyalakan pompa karena berada pada rentang ideal, sedangkan pada kondisi tinggi seperti 31,7°C dan 72% RH sistem mengaktifkan semprot dengan durasi 7 detik sesuai aturan fuzzy dan mengirim notifikasi status secara otomatis.
- E.** Integrasi antarmuka Telegram memudahkan pemantauan variabel lingkungan, status fuzzy/durasi, dan kontrol manual sehingga meningkatkan keterkendalian sistem dari jarak jauh.

- F. Secara keseluruhan, tujuan penelitian—otomatisasi penyemprotan yang efektif, terjadwal, adaptif terhadap kondisi lingkungan, serta dapat dimonitor dan dikendalikan secara daring—telah tercapai pada level prototipe miniatur.

5.2. Saran

- A. Pengembangan selanjutnya disarankan menambah variabel input lingkungan seperti amonia (NH_3) atau kualitas udara lain untuk Lakukan kalibrasi lanjutan terhadap fungsi keanggotaan dan aturan fuzzy, serta pertimbangkan keluaran durasi yang lebih halus (semi-kontinu) agar sesuai dengan variasi nyata suhu–kelembapan dan kebutuhan volume semprotmemperkaya basis keputusan fuzzy dan meningkatkan ketepatan aktuasi
- B. Lakukan kalibrasi lanjutan terhadap fungsi keanggotaan dan aturan fuzzy, serta pertimbangkan keluaran durasi yang lebih halus (semi-kontinu) agar sesuai dengan variasi nyata suhu–kelembapan dan kebutuhan volume semprot.
- C. Evaluasi performa hidrolik: debit pompa, tipe dan pola sebaran nozzle, serta waktu semprot optimal per luas area agar efisiensi penggunaan disinfektan dan pemerataan penyemprotan dapat terukur.
- D. Tingkatkan keandalan sistem melalui perlindungan lingkungan (waterproofing casing, manajemen kabel), catu daya cadangan, serta fail-safe pada kondisi gangguan jaringan/langsung untuk menjaga kontinuitas fungsi.

- E. Perluasan uji dari miniatur ke kandang riil dengan periode pengamatan lebih panjang untuk menilai kestabilan sistem, dampak operasional, dan aspek biosekuriti secara praktis di lapangan.
- F. Bangun pencatatan data terstruktur (data logging dan dashboard) untuk analitik tren lingkungan–aktuasi, sebagai dasar penyetelan rule fuzzy dan penjadwalan yang lebih presisi ke depan.

Dengan demikian, prototipe telah memenuhi fungsi utama sistem penyemprotan disinfektan yang bersifat otomatis, adaptif, terjadwal, dan terpantau daring, serta siap ditingkatkan melalui perluasan variabel, kalibrasi teknis, dan validasi skala lapangan.

DAFTAR PUSTAKA

- Aini, A. H., Saragih, Y., & Hidayat, D. R. (2021). *RANCANG BANGUN SMART SYSTEM PADA KANDANG AYAM MENGGUNAKAN MIKROKONTROLER* (Vol. 11, Issue 3). <https://jurnal.umj.ac.id/index.php/just-it/index>
- Friendly, O., Prayudani, S., Sembiring, Z., Komputer dan Informatika, T., & Negeri Medan Corresponding Author, P. (2022). *SWARNA Jurnal Pengabdian Kepada Masyarakat*.
- Kus Indrani Listyoningrum, Danise Yunaini Fenida, & Nurhasan Hamidi. (2023). Inovasi Berkelanjutan dalam Bisnis: Manfaatkan Flowchart untuk Mengoptimalkan Nilai Limbah Perusahaan. *Jurnal Informasi Pengabdian Masyarakat*, 1(4), 100–112. <https://doi.org/10.47861/jipm-nalanda.v1i4.552>
- Medion. (2021). Penyemprotan Desinfektan Saat Kondisi Lingkungan Kandang Lembap. In *Info Medion*. <https://www.medion.co.id/info-medion/penyemprotan-desinfektan-saat-kondisi-lingkungan-kandang-lembap/>
- Nahak, A., Kelen, Y. P. K., Baso, B., & Sucipto, W. (2024). Sistem Otomatisasi Pakan Minum dan Monitoring Kelembaban Udara berbasis Internet of Things (IoT). *Bit-Tech*, 7(1), 1–9. <https://doi.org/10.32877/bt.v7i1.1226>
- Nina Adlini, M., Hanifa Dinda, A., Yulinda, S., Chotimah, O., & Julia Merliyana, S. (2022). *METODE PENELITIAN KUALITATIF STUDI PUSTAKA* (Vol. 6, Issue 1).
- Nyoman Mudiana, I., Ketut Parti, I., Nyoman Sutama, I., Teknik Elektro, J., Akuntansi, J., Negeri Bali Bukit Jimbaran, P., & Selatan, K. (2019). *SISTEM PENYEMPROTAN DISINFEKTAN TERKONTROL UNTUK PROGRAM*

BIO SEKURITI PADA PETERNAKAN AYAM PETELUR DI DESA
DEMULIH BANGLI. *Jurnal BHAKTI PERSADA*, 5(2).

Putri Br Ginting, B., Desy Nur Utomo, A., Panjaitan No, J. DI, Kidul, P., &
Purwokerto Sel, K. (2024). MONITORING PENERAPAN INTERNET OF
THINGS (IOT) DALAM MENINGKATKAN EFISIENSI DAN
PRODUKTIVITAS PADA BUDIDAYA PETERNAKAN BEBEK (STUDI
KASUS: TERNAK BEBEK PAK GITA KECAMATAN KALIMANAH,
KABUPATEN PURBALINGGA). In *Jurnal Mahasiswa Teknik Informatika*
(Vol. 8, Issue 4).

*RANCANG BANGUN SISTEM IRIGASI OTOMATIS BERBASIS FUZZY LOGIC
& INTERNET OF THINGS PADA SMART SHOWCASE HIDROPONIK.*
(n.d.).

Rizkiyah, A., & Dwi Kurniawan, W. (2025). *PENGEMBANGAN BUKU
PANDUAN PRAKTIKUM KONTROL RELAY UNTUK MENUNJANG MATA
KULIAH INSTRUMENTASI DAN KENDALI PADA PROGRAM STUDI SI
PENDIDIKAN TEKNIK MESIN UNESA.*

Safari. (2022). *Safari+2-4+Oktober+101-108.*

Safutra, I., Isnandar, R., & Wijayanti, L. (2022). Analisis Produktivitas Bebek
Petelur Berdasarkan Pola Pemeliharaan di Kabupaten Banyuwangi. *Jurnal
Ilmu Peternakan Terapan*, 5(1), 33–42.

Saputra, Y. A., Raihan Yusuf, M., & Sutabri, T. (2022). *Implementasi Teknologi
Internet of Things (Iot) dalam Pengelolaan Penghematan Listrik untuk Smart
Home.*

- Simamora, P., Prayudha, J., & Ginting, E. F. (n.d.). *Sistem Pendeteksi Suhu dan Gambar pada Alat Penyemprot Cairan Hand Sanitizer menggunakan ESP32 Cam dan Teknik Duplex*. <https://ojs.trigunadharma.ac.id/index.php/jskom>
- Ummul Masir, Sultan Mubarak Z, & Muhammad Kadir. (2024). Implementasi Pengelolaan Kandang Bebek Hygienis Untuk Meningkatkan Mutu dan Produksi Telur pada Usaha Bebek Petelur di Kelurahan Bontoa Kabupaten Maros. *Faedah : Jurnal Hasil Kegiatan Pengabdian Masyarakat Indonesia*, 2(3), 171–178. <https://doi.org/10.59024/faedah.v2i3.994>
- Yao, Y. Y. (n.d.). *Semantics of Fuzzy Sets in Rough Set Theory*.
- Yudhatama, K. (2021). *PERANCANGAN SISTEM OTOMATIS DISINFEKTAN PETERNAKAN AYAM DI DESA MARGOREJO DIMASA PANDEMI COVID-19 PRAKTEK KERJA PENGABDIAN MASYARAKAT Disusun Oleh.*

LAMPIRAN

1. Sk- Pembimbing



UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UMSU (Universitas Muhammadiyah Sumatera Utara) adalah salah satu universitas terkemuka di Sumatera Utara. Universitas ini memiliki fokus pada pendidikan dan penelitian di bidang ilmu komputer dan teknologi informasi. Pusat Administrasi: Jalan Mubtahir Bani No. 3 Medan 20230. Telp: (061) 6622450, 6522455, Fax: (061) 6522414, 6521220.

PENETAPAN DOSEN PEMBIMBING PROPOSAL/SKRIPSI MAHASISWA NOMOR : 260/II.3-AU/UMSU-09/1/2025

Assalamu'alaikum Warahmatullahi Wabarakatuh

Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara, berdasarkan Persetujuan permohonan judul penelitian Proposal / Skripsi dari Ketua / Sekretaris.

Program Studi : Teknologi Informasi
Pada tanggal : 04 Februari 2025

Dengan ini menetapkan Dosen Pembimbing Proposal / Skripsi Mahasiswa.

Nama : Soni Armanda S
NPM : 2109020041
Semester : VII (Tujuh)
Program studi : Teknologi Informasi
Judul Proposal / Skripsi : Perancangan dan Implementasi Sistem Monitoring Kualitas Udara Berbasis IoT dengan Machine Learning

Dosen Pembimbing : Mahardika Abdi Prawira, S.Kom., M.Kom.

Dengan demikian di izinkan menulis Proposal / Skripsi dengan ketentuan

1. Penulisan berpedoman pada buku panduan penulisan. Proposal / Skripsi Fakultas Ilmu Komputer dan Teknologi Informasi UMSU
2. Pelaksanaan Sidang Skripsi harus berjarak 3 bulan setelah dikeluarkannya Surat Penetapan Dosen Pembimbing Skripsi.
3. Proyek Proposal / Skripsi dinyatakan "BATAL" bila tidak selesai sebelum Masa Kadalua tanggal : 04 Februari 2026
4. Revisi judul

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Ditetapkan di : Medan
Pada Tanggal : 05 Sya'ban 1446 H
04 Februari 2025M



Dekan

Dr. Al-Kifwarizmi, M.Kom.
NIDN : 0127099201

2. Turnitin

Turnitin			
ORIGINALITY REPORT			
19%	16%	8%	12%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS
PRIMARY SOURCES			
1	Submitted to Universitas Muhammadiyah Sumatera Utara Student Paper	2%	
2	repository.umsu.ac.id Internet Source	1%	
3	Submitted to Universitas Merdeka Malang Student Paper	1%	
4	repository.uin-suska.ac.id Internet Source	1%	
5	Submitted to Universitas Ahmad Dahlan Student Paper	<1%	
6	Submitted to Universitas Muhammadiyah Palembang Student Paper	<1%	
7	repo.undiksha.ac.id Internet Source	<1%	
8	pbsi-upr.id Internet Source	<1%	
9	mypublikasi.com Internet Source	<1%	

3. Source Kode

```
#include <WiFi.h>
```

```
#include <WiFiClientSecure.h>
```

```
#include <UniversalTelegramBot.h>
```

```
#include <ArduinoJson.h>
```

```
#include <Wire.h>
```

```
#include <RTCLib.h>
```

```

#include <Adafruit_GFX.h>

#include <Adafruit_SSD1306.h>

#include <Preferences.h>

#include <DHT.h>

const char* WIFI_SSID    = "soni";          // <-- change

const char* WIFI_PASSWORD = "12345678";    // <-- change

const char* TELEGRAM_BOT_TOKEN = "7677512496:AAGLF7bM51zrU-
5p1udtGVz kf5zBQ--45QQ"; // <-- keep secret

String OWNER_CHAT_ID = "5981655213";      // <-- owner chat id

#define DHT_PIN  4

#define DHTTYPE  DHT22

DHT dht(DHT_PIN, DHTTYPE);

#define SDA_RTC  16

#define SCL_RTC  17

#define SDA_OLED  21

#define SCL_OLED  22

#define RELAY_PIN  25

#define BUZZER_PIN  32

#define RELAY_ACTIVE_LEVEL LOW // change if your relay is active-HIGH

TwoWire I2C_RTC = TwoWire(0);

TwoWire I2C_OLED = TwoWire(1);

#define SCREEN_WIDTH 128

#define SCREEN_HEIGHT 64

```

```

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT,
&I2C_OLED, -1);

RTC_DS3231 rtc;

Preferences prefs;

WiFiClientSecure secureClient;

UniversalTelegramBot bot(TELEGRAM_BOT_TOKEN, secureClient);

const char* TZ_STRING = "WIB-7"; // WIB (UTC+7)

unsigned long lastNtpMillis = 0;

const unsigned long NTP_RESYNC_MS = 6UL * 60UL * 60UL * 1000UL;

bool autoMode = true;

bool isSpraying = false;

bool stopRequested = false;

typedef unsigned long ulong;

ulong bot_lasttime = 0;

const ulong BOT_INTERVAL_MS = 1000;

uint32_t todayYMD = 0; // YYYYMMDD

// Anti-duplicate PER-MINUTE key: YYYYMMDD*10000 + hour*100 + minute
uint32_t lastFuzzyKey = 0;

const float TEMP_LIMIT_C = 50.0; // RTC extreme limit

// Default: 22 -> 1 (wrap across midnight). daysMask bit 0..6 = Sun..Sat

int startHour = 22;

int endHour = 1;

uint8_t daysMask = 0x7F; // 0b01111111 (Sun..Sat active)

// Trigger window at the beginning of each minute (sec 0..3)

```

```

const uint8_t TOP_OF_MINUTE_WINDOW_SEC = 4; // 0..3

// Telegram edit state

enum EditState { EDIT_NONE, EDIT_SET_JAM, EDIT_SET_HARI };

EditState editState = EDIT_NONE;

uint8_t editDaysMask = 0x7F;

uint32_t makeYMD(const DateTime &dt) { return (uint32_t)dt.year()*10000UL
+ (uint32_t)dt.month()*100UL + (uint32_t)dt.day(); }

String twoDigits(int v) { char b[8]; snprintf(b, sizeof(b), "%02d", v); return
String(b); }

void applyTZ(){ setenv("TZ", TZ_STRING, 1); tzset(); }

void setRelay(bool on) { digitalWrite(RELAY_PIN, on ?
RELAY_ACTIVE_LEVEL : !RELAY_ACTIVE_LEVEL); }

void beep(uint16_t msOn, uint16_t msOff, uint8_t times){ for(uint8_t
i=0;i<times;i++){ digitalWrite(BUZZER_PIN,HIGH); delay(msOn);
digitalWrite(BUZZER_PIN,LOW); if(i+1<times) delay(msOff);} }

const char* dayNameArr[7] =

{"Minggu","Senin","Selasa","Rabu","Kamis","Jumat","Sabtu"};

String dayName(uint8_t dow){ return (dow>6)?"?":dayNameArr[dow]; }

float readRTC_T(){ return rtc.getTemperature(); }

void notifyAll(const String &text){ if (OWNER_CHAT_ID.length())

bot.sendMessage(OWNER_CHAT_ID, text, "Markdown"); }

void savePrefs(){

  prefs.putBool("auto", autoMode);

  prefs.putUChar("daysMask", daysMask);

```

```

    prefs.putChar("hStart", (char)startHour);

    prefs.putChar("hEnd", (char)endHour);
}

void loadPrefs(){

    autoMode = prefs.getBool("auto", true);

    daysMask = prefs.getUChar("daysMask", 0x7F);

    startHour = (int)prefs.getChar("hStart", 22);

    endHour = (int)prefs.getChar("hEnd", 1);

    if(startHour<0||startHour>23) startHour=22;

    if(endHour<0||endHour>23) endHour=1;

    if(daysMask==0) daysMask=0x7F;

}

bool syncRTCFromNTP_strict(uint16_t waitMs = 15000) {

    applyTZ();

    configTzTime(TZ_STRING, "pool.ntp.org", "time.nist.gov",
"time.google.com");

    unsigned long t0 = millis();

    time_t nowEpoch = time(nullptr);

    while ((nowEpoch < 1672531200) && (millis() - t0 < waitMs)) { delay(250);
nowEpoch = time(nullptr); }

    if (nowEpoch < 1672531200) { Serial.println("[NTP] Timeout."); return false; }

    struct tm lt; localtime_r(&nowEpoch, &lt);

    rtc.adjust(DateTime(lt.tm_year+1900, lt.tm_mon+1, lt.tm_mday, lt.tm_hour,
lt.tm_min, lt.tm_sec));

```

```

lastNtpMillis = millis();

Serial.println("[NTP] RTC set (WIB).");

return true;
}

void realignRTCIfNeeded(){

    time_t sysEpoch = time(nullptr);

    if (sysEpoch < 1672531200) return;

    DateTime rt = rtc.now();

    struct tm lt; localtime_r(&sysEpoch, &lt);

    DateTime sysLocal(lt.tm_year+1900, lt.tm_mon+1, lt.tm_mday, lt.tm_hour,
lt.tm_min, lt.tm_sec);

    long diff = labs((long)sysLocal.unixtime() - (long)rt.unixtime());

    if (diff > 30){

        rtc.adjust(sysLocal);

        Serial.printf("[NTP] Realign RTC (|d|=%lds)\n", diff);

    }

}

unsigned long dhtLastMs = 0;

float dhtLastT = NAN, dhtLastH = NAN;

const unsigned long DHT_MIN_INTERVAL_MS = 2000;

bool readDHTCached(float &t, float &h, bool forceFresh=false){

    unsigned long now = millis();

```

```

    if(!forceFresh && (now - dhtLastMs) <
DHT_MIN_INTERVAL_MS){ t=dhtLastT; h=dhtLastH;
return !(isnan(t)||isnan(h)); }

    float T = dht.readTemperature();

    float H = dht.readHumidity();

    if(!isnan(T) && !isnan(H)){ dhtLastT=T; dhtLastH=H; dhtLastMs=now; t=T;
h=H; return true; }

    t=dhtLastT; h=dhtLastH; return !(isnan(t)||isnan(h));
}

float tri(float x, float a, float b, float c){

    if (x <= a || x >= c) return 0;

    if (x == b) return 1;

    if (x > a && x < b) return (x - a) / (b - a);

    return (c - x) / (c - b);

}

float lsh(float x, float a, float b){

    if (x <= a) return 1;

    if (x >= b) return 0;

    return (b - x) / (b - a);

}

float rsh(float x, float a, float b){

    if (x <= a) return 0;

    if (x >= b) return 1;

    return (x - a) / (b - a);

```

```

}

/* Temp: Low=tri(20,22,24), Ideal=tri(24,28,31), High=rsh(31,33)

   RH: Dry=lsh(55,60), Ideal=tri(60,62,65), Wet=rsh(65,70) */
uint16_t fuzzyDurationSec(float tC, float rh){

    float muT_low  = tri(tC, 20, 22, 24);

    float muT_ideal = tri(tC, 24, 28, 31);

    float muT_high  = rsh(tC, 31, 33);

    float muH_dry   = lsh(rh, 55, 60);

    float muH_ideal = tri(rh, 60, 62, 65);

    float muH_wet   = rsh(rh, 65, 70);

    float outNo=0, outFew=0, outMed=0, outMany=0;

    auto AND = [](float a, float b){ return min(a,b); };

    auto ORm = [](float &dst, float v){ if (v>dst) dst=v; };

    ORm(outFew, AND(muT_low,  muH_dry)); // 1
    ORm(outFew, AND(muT_low,  muH_ideal)); // 2
    ORm(outMed, AND(muT_low,  muH_wet)); // 3
    ORm(outMed, AND(muT_ideal, muH_dry)); // 4
    ORm(outNo,  AND(muT_ideal, muH_ideal)); // 5
    ORm(outFew, AND(muT_ideal, muH_wet)); // 6
    ORm(outMed, AND(muT_high,  muH_dry)); // 7
    ORm(outMany,AND(muT_high,  muH_ideal)); // 8
    ORm(outMany,AND(muT_high,  muH_wet)); // 9

    float num = outFew*3.0f + outMed*5.0f + outMany*7.0f;

    float den = outFew + outMed + outMany;

```

```

    if (den <= 1e-6) return 0;

    uint16_t dur = (uint16_t)roundf(num / den);

    return constrain(dur, 0, 30);
}

void oledHeader(const String &l1, const String &l2){

    display.clearDisplay();

    display.setTextSize(1);

    display.setTextColor(SSD1306_WHITE);

    display.setCursor(0,0);

    display.println(l1); display.println(l2);

}

void oledMain(const String &l3, const String &l4){

    display.setCursor(0,24);

    display.println(l3); display.println(l4);

    display.display();

}

String formatHourWindow(){

    return String(twoDigits(startHour)) + "-" + twoDigits(endHour) + " WIB";

}

String formatActiveDays(uint8_t mask){

    const char* shortNames[7] = {"Min", "Sen", "Sel", "Rab", "Kam", "Jum", "Sab"};

    String out=""; bool first=true;

    for(int i=0;i<7;i++) if(mask & (1<<i)){ if(!first) out+=","; out+=shortNames[i];

    first=false; }

```

```

    if(first) out = "(None)"; return out;
}

void sendStatus(const String &chat_id){

    DateTime now = rtc.now();

    float tRTC = readRTC_T();

    float tD,hD; bool ok=readDHTCached(tD,hD);

    String msg;

    msg += "Time: " + String(dayName(now.dayOfTheWeek())) + ", " +
    String(now.day()) + "/" + String(now.month()) + "/" + String(now.year()) + " "
        + twoDigits(now.hour()) + ":" + twoDigits(now.minute()) + ":" +
twoDigits(now.second()) + " WIB\n";

    msg += "RTC Temp: " + String(tRTC,1) + " C\n";

    if(!ok){

        msg += "DHT22: sensor error\n";

        msg += "Fuzzy Duration: -- s\n";

    }else{

        uint16_t dur = fuzzyDurationSec(tD,hD);

        msg += String("DHT22: ") + String(tD,1) + " C, RH " + String(hD,0) + " %\n";

        msg += "Fuzzy Duration: " + String(dur) + " s\n";

    }

    msg += String("Mode: ") + (autoMode ? "Auto" : "Manual") + "\n";

    msg += String("Schedule: every minute (sec 0-3) - ") + formatHourWindow() +
"\n";

    msg += "Active Days: " + formatActiveDays(daysMask) + "\n";

```

```

msg += String("Pump: ") + (isSpraying ? "ON (spraying)" : "OFF");

bot.sendMessage(chat_id, msg, "Markdown");

}

void preSprayAlert(){ beep(150,120,3); }

void runDisinfectionFor(uint16_t dur, const String& reason){

    if(dur==0){ notifyAll("No spray (0s)."); return; }

    if(isSpraying) return;

    stopRequested=false; isSpraying=true;

    DateTime now=rtc.now();

    String startMsg = String("SPRAY START (" + reason + ")\n");

    startMsg += "Time: "+twoDigits(now.hour())+": "+twoDigits(now.minute())+"

WIB\n";

    startMsg += "Duration: "+String(dur)+" s";

    notifyAll(startMsg);

    oledHeader("DISINFECT: "+reason, "Duration: "+String(dur)+"s");

    oledMain("Pump: ON", "Stop via Telegram");

    setRelay(true);

    unsigned long start=millis();

    while((millis()-start)/1000UL < dur){

        if(millis()-bot_lasttime > BOT_INTERVAL_MS){

            int n=bot.getUpdates(bot.last_message_received+1);

            for(int i=0;i<n;i++){

                String chat_id=bot.messages[i].chat_id;

                String text=bot.messages[i].text;

```

```

        if(chat_id==OWNER_CHAT_ID && (text=="Stop"||text=="/stop"))
stopRequested=true;

    }

    bot_lasttime=millis();

}

if(stopRequested) break;

delay(50);

}

setRelay(false);

isSpraying=false;

DateTime done=rtc.now();

String endMsg = stopRequested ? "Stopped early" : "Done";

endMsg += String("\nActual duration: ") + String((((millis()-start)/1000UL)) + " s");

endMsg += String("\nTime:

") + twoDigits(done.hour()) + ":" + twoDigits(done.minute()) + " WIB";

notifyAll(endMsg);

oledHeader("Spray finished", stopRequested ? "Stopped" : "Normal");

oledMain("Pump: OFF", "Back to IDLE");

}

String homeKeyboardJson(){

String autoTxt = String("Auto: ") + (autoMode ? "ON" : "OFF");

String kb = String(

    "[[\"Status\\\", \"Fuzzy Sekarang\\\", \"

    [\"Jadwal\\\", \"\"] + autoTxt + String(\"\\\", \"

```

```

    "["Set Jam\\",\\"Set Hari\\"],"
    "["Buzzer\\",\\"Help\\"],"
    "["NTP\\",\\"Stop\\"]]);

return kb;

}

String daysEditKeyboard(uint8_t mask){
    (void)mask; // simple keyboard (state in editDaysMask)

    return
    String("[\\"Min\\",\\"Sen\\",\\"Sel\\"],[\\"Rab\\",\\"Kam\\",\\"Jum\\"],[\\"Sab\\",\\"Semua\\",\\"
    Selesai\\"]]);
}

void sendHome(const String &chat_id, const String &title=""){
    String msg = title.length()? (title+"\\n") : "";

    msg += "Smart Disinfectant (spray every minute)\\n";

    msg += "Schedule: " + formatHourWindow() + " - Days: " +
    formatActiveDays(daysMask) + "\\n";

    bot.sendMessageWithReplyKeyboard(chat_id, msg, "Markdown",
    homeKeyboardJson(), true);
}

bool withinWindowHour(int h){
    if (startHour <= endHour) return (h >= startHour && h <= endHour);

    return (h >= startHour || h <= endHour); // wraps across midnight
}

```

```

bool withinWindow(const DateTime& now){ return
withinWindowHour(now.hour()); }

bool allowedDay(uint8_t dow){ return (daysMask & (1 << dow)); }

void handleCommand(const String &chat_id, const String &text){
    if (chat_id != OWNER_CHAT_ID){ bot.sendMessage(chat_id, "Sorry, private
bot."); return; }

    // Hour input mode

    if (editState == EDIT_SET_JAM){

        int h1, h2;

        if (sscanf(text.c_str(), "%d %d", &h1, &h2) == 2 && h1>=0 && h1<=23 &&
h2>=0 && h2<=23){

            startHour = h1; endHour = h2; savePrefs(); editState = EDIT_NONE;

            bot.sendMessage(chat_id, String("Saved hours: ") + formatHourWindow());

            sendHome(chat_id); return;

        } else { bot.sendMessage(chat_id, "Format: HH HH (example: 22 1)");
return; }

    }

    // Days edit mode

    if (editState == EDIT_SET_HARI){

        if (text=="Semua"){ editDaysMask = 0x7F;

bot.sendMessageWithReplyKeyboard(chat_id, "All days selected. Press Selesai to
save.", "Markdown", daysEditKeyboard(editDaysMask), true); return; }

```

```

    if (text=="Selesai"){ daysMask = editDaysMask; savePrefs(); editState =
EDIT_NONE; bot.sendMessage(chat_id, String("Active days:
")+formatActiveDays(daysMask)); sendHome(chat_id); return; }

    int idx=-1;

    if(text=="Min")idx=0; else if(text=="Sen")idx=1; else if(text=="Sel")idx=2;
else if(text=="Rab")idx=3; else if(text=="Kam")idx=4; else
if(text=="Jum")idx=5; else if(text=="Sab")idx=6;

    if(idx>=0){ editDaysMask ^= (1<<idx); String note =
String((editDaysMask&(1<<idx))?"[ON] ":"[OFF] ") + text + ".\nSelected: " +
formatActiveDays(editDaysMask) + "\nPress Selesai to save.";
bot.sendMessageWithReplyKeyboard(chat_id, note, "Markdown",
daysEditKeyboard(editDaysMask), true); return; }

    bot.sendMessageWithReplyKeyboard(chat_id, "Toggle days, use Semua or
Selesai.", "Markdown", daysEditKeyboard(editDaysMask), true); return;
}

// Normal mode

if (text=="/start"||text=="Home"||text=="/home"){ sendHome(chat_id, "Hello,
ready."); }

else if (text=="/status"||text=="Status"){ sendStatus(chat_id); }

else if (text=="Fuzzy Sekarang"||text=="/fuzzy"){

    if(readRTC_T()>=TEMP_LIMIT_C){ bot.sendMessage(chat_id,"High RTC
temp. Cancel."); sendHome(chat_id); return; }

    float tD,hD; bool ok=readDHTCached(tD,hD,true); if(!ok){ delay(80);
ok=readDHTCached(tD,hD,true); }

```

```

    if(!ok){ bot.sendMessage(chat_id,"DHT22 read error."); sendHome(chat_id);
return; }

    uint16_t dur = fuzzyDurationSec(tD,hD);

    bot.sendMessage(chat_id, String("T=")+String(tD,1)+"C,
RH="+String((int)hD)+"% -> Duration: "+String(dur)+"s");

    if (dur > 0){ preSprayAlert(); runDisinfectionFor(dur, "Manual Fuzzy"); }

    sendHome(chat_id);

}

else if (text.startsWith("Auto")){ autoMode=!autoMode; savePrefs();
bot.sendMessage(chat_id, String("Auto mode: ")+(autoMode?"ON":"OFF"));
sendHome(chat_id); }

else if (text=="Set Jam"){ editState = EDIT_SET_JAM;
bot.sendMessage(chat_id, "Send: HH HH to set [start end] (example: 22 1)",
"Markdown"); }

else if (text=="Set Hari"){ editState = EDIT_SET_HARI; editDaysMask =
daysMask; String note = String("Select active days (toggle):\nSelected: ") +
formatActiveDays(editDaysMask) + "\nPress Selesai to save.";
bot.sendMessageWithReplyKeyboard(chat_id, note, "Markdown",
daysEditKeyboard(editDaysMask), true); }

else if (text=="Jadwal"||text=="/jadwal"){ String j=String("Auto: every minute
(sec 0-3)\n")+ "Hours: "+formatHourWindow()+"\nDays:
"+formatActiveDays(daysMask); bot.sendMessage(chat_id, j, "Markdown");
setStatus(chat_id); }

```

```

else if (text=="Buzzer"){ beep(120,100,3); bot.sendMessage(chat_id,"Buzzer
OK"); }

else if (text=="/stop"||text=="Stop"){ stopRequested=true;
bot.sendMessage(chat_id,"Stop command sent."); }

else if (text=="NTP"||text=="/ntp"){

    if(WiFi.isConnected() && syncRTCFromNTP_strict())
bot.sendMessage(chat_id,"RTC synced (WIB)."); else
bot.sendMessage(chat_id,"NTP failed. Check WiFi.");

}

else if (text=="/help"||text=="Help"){

String h =

    "Commands:\n"

    "- Home / Status\n"

    "- Fuzzy Sekarang\n"

    "- Auto: ON/OFF\n"

    "- Set Jam (send: HH HH)\n"

    "- Set Hari (toggle days, Selesai to save)\n"

    "- Jadwal\n"

    "- Buzzer\n"

    "- NTP\n"

    "- /setwib YYYY-MM-DD HH:MM:SS\n"

    "- Stop\n"

    "Safety: RTC temp >= 50C -> cancel.";

bot.sendMessage(chat_id, h);

```

```

    }

    else if (text.startsWith("/setwib ")) {

        String s=text.substring(8);

        int y,M,d,h,m,s2;

        if(sscanf(s.c_str(), "%d-%d-%d %d:%d:%d",
&y,&M,&d,&h,&m,&s2)==6){ rtc.adjust(DateTime(y,M,d,h,m,s2));

        bot.sendMessage(chat_id,"RTC set (WIB)."); }

        else bot.sendMessage(chat_id,"Format example: /setwib 2025-08-25
20:05:00");

    }

    else { sendHome(chat_id,"Unknown command. Use the buttons."); }

}

void handleTelegram(){

    if(millis()-bot_lasttime > BOT_INTERVAL_MS){

        int n=bot.getUpdates(bot.last_message_received+1);

        for(int i=0;i<n;i++){

            String chat_id=bot.messages[i].chat_id;

            String text=bot.messages[i].text;

            String from=bot.messages[i].from_name;

            Serial.printf("TG from %s (%s): %s\n", from.c_str(), chat_id.c_str(),
text.c_str());

            if(OWNER_CHAT_ID.length()==0){ OWNER_CHAT_ID=chat_id;

            bot.sendMessage(chat_id,"Owner chat id set."); }

```

```

    handleCommand(chat_id, text);

}

bot_lasttime=millis();

}

}

void drawIdle(const DateTime &now){

    String l1 = String("RTC: ") + twoDigits(now.hour()) + ":" +
twoDigits(now.minute()) + ":" + twoDigits(now.second()) + " WIB";

    String l2 = dayName(now.dayOfTheWeek());

    oledHeader(l1, l2);

    float tD,hD; bool ok=readDHTCached(tD,hD);

    String l3, l4;

    if (!ok) {

        l3 = "T: err RH: err";

        l4 = "Dur: --s " + String(autoMode ? "A" : "M");

    } else {

        uint16_t dur = fuzzyDurationSec(tD,hD);

        l3 = String("T:")+-String(tD,1)+"C RH:"+String((int)hD)+"%";

        l4 = "Dur: " + String(dur) + "s " + String(autoMode ? "A" : "M");

    }

    oledMain(l3, l4);

}

/**** SETUP ****/

```

```

void setup(){

  Serial.begin(115200); delay(150);

  pinMode(RELAY_PIN,OUTPUT); pinMode(BUZZER_PIN,OUTPUT);

  setRelay(false); digitalWrite(BUZZER_PIN,LOW);

  I2C_RTC.begin(SDA_RTC,SCL_RTC,400000);

  I2C_OLED.begin(SDA_OLED,SCL_OLED,400000);

  if(!display.begin(SSD1306_SWITCHCAPVCC,0x3C,false,false))

  Serial.println("OLED init failed");

  else { display.clearDisplay(); display.setTextColor(SSD1306_WHITE);

  display.setTextSize(1); display.setCursor(0,0); display.println("Smart

  Disinfectant"); display.println("Init OLED..."); display.display(); }

  if(!rtc.begin(&I2C_RTC)) Serial.println("RTC NOT FOUND!");

  dht.begin();

  prefs.begin("disinf",false);

  loadPrefs();

  WiFi.mode(WIFI_STA); WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

  Serial.printf("Connecting WiFi %s\n", WIFI_SSID);

  ulong t0=millis(); while(WiFi.status()!=WL_CONNECTED && millis()-

  t0<15000){ delay(500); Serial.print("."); }

  Serial.println(); if(WiFi.isConnected()) Serial.printf("WiFi OK, IP: %s\n",

  WiFi.localIP().toString().c_str());

  secureClient.setInsecure();

  applyTZ();

```

```

    if(WiFi.isConnected()){ if(!syncRTCFromNTP_strict()) Serial.println("[NTP]
Sync at boot failed (use RTC)."); }

    DateTime now=rtc.now(); todayYMD=makeYMD(now);

    if(OWNER_CHAT_ID.length()) notifyAll("Device Online. Use /start.");
}

void tryAutoFuzzy(const DateTime& now){

    if(!autoMode || isSpraying) return;

    if(!allowedDay(now.dayOfTheWeek())) return;

    if(!withinWindow(now)) return;

    if(!(now.second() < TOP_OF_MINUTE_WINDOW_SEC)) return; // sec 0..3

    uint32_t key = makeYMD(now)*10000UL + (uint32_t)now.hour()*100UL +
(uint32_t)now.minute();

    static uint32_t alreadyTriggeredThisMinute = 0;

    if (key == lastFuzzyKey || key == alreadyTriggeredThisMinute) return;

    if(readRTC_T() >= TEMP_LIMIT_C){

        notifyAll("Auto Fuzzy canceled (RTC temp high).");

        lastFuzzyKey = key; alreadyTriggeredThisMinute = key; return;

    }

    float tD,hD; bool ok=readDHTCached(tD,hD,true);

    if(!ok){ delay(80); ok=readDHTCached(tD,hD,true); }

    if(!ok){

        notifyAll("Auto Fuzzy canceled: DHT22 read error.");

        lastFuzzyKey = key; alreadyTriggeredThisMinute = key; return;

    }
}

```

```

lastFuzzyKey = key; alreadyTriggeredThisMinute = key; // lock minute

uint16_t dur = fuzzyDurationSec(tD,hD);

String info = String("AUTO Fuzzy
")+twoDigits(now.hour())+": "+twoDigits(now.minute())+" WIB\n";

info += "T="+String(tD,1)+"C RH="+String((int)hD)+"% -> Duration
"+String(dur)+"s";

notifyAll(info);

if(dur > 0){ preSprayAlert(); runDisinfectionFor(dur, "Auto Fuzzy"); }
}

void loop(){

  handleTelegram();

  DateTime now=rtc.now();

  uint32_t ymd=makeYMD(now);

  if(ymd!=todayYMD) todayYMD=ymd;

  if(!isSpraying) drawIdle(now);

  tryAutoFuzzy(now);

  if(WiFi.isConnected() && (millis()-lastNtpMillis > NTP_RESYNC_MS))

syncRTCFromNTP_strict();

  realignRTCIfNeeded();

  delay(100);

}

```

4. Notifikasi

