



Implementasi Kontrol PID dan Logika Fuzzy untuk Optimasi Regulasi Suhu Kandang *Day Old Chick* (DOC) Berbasis *Internet of Things*

Tugas Akhir

Oleh:

Gabriel Tambunan (4242001041)

Risky Soleman (4242011039)

**Program Studi Teknologi Rekayasa Elektronika
Jurusan Teknik Elektro
Politeknik Negeri Batam
2026**

Lembar Pengesahan

Tugas Akhir disusun untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Terapan Teknik (S.Tr.T)

Disusun oleh:

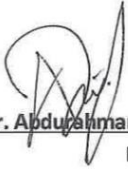
Gabriel Tambunan (4242001041)

Risky Soleman (4242011039)

Tanggal Sidang: 27 Juli 2026

Disetujui oleh:

Penguji I


Dr. Abdulrahman Dwijotomo. S.ST., M.SC
NIK: 122257

Pembimbing



Illa Aryeni S.T, M.T
NIK: 122255

Penguji II


Nanta Fakhri Prebianto S.ST., M.SC
NIK: 117182

[Implementasi Kontrol PID dan Logika Fuzzy untuk Optimasi Regulasi Suhu Kandang *Day Old Chick* (DOC) Berbasis *Internet of Things*]

Abstrak

Penelitian ini bertujuan untuk menjaga kestabilan suhu kandang ayam *Day Old Chicken* (DOC) melalui implementasi serta analisis performa dari metode kontrol PID dan Logika Fuzzy pada sistem optimasi suhu kandang berbasis *Internet of Things* (IoT). Sistem dirancang menggunakan mikrokontroler ESP32 sebagai pusat kontrol, sensor DHT22 sebagai pembaca suhu, AC *light* dimmer sebagai *aktuator*, dan *heating lamp* sebagai sumber panas. Metode PID menggunakan parameter K_p , K_i , dan K_d untuk menentukan kestabilan keluaran, sedangkan metode Logika Fuzzy menggunakan fungsi keanggotaan *COLD*, *WARM*, dan *HOT* untuk menentukan kondisi *ON/OFF heating lamp*. Implementasi PID terintegrasi dengan *Bot Telegram* untuk *Monitoring* dan pengontrolan sistem, sedangkan metode Logika Fuzzy terhubung dengan *Cloud API* dan *Google Sheets* untuk penyimpanan serta pembacaan data secara *real-time*. Kalibrasi sensor dilakukan dengan membandingkan pembacaan suhu sensor DHT22 dan termometer *wood GEA* yang menghasilkan selisih pengukuran sebesar $0,1^{\circ}\text{C}$ - $0,8^{\circ}\text{C}$. Pengujian kontrol PID menunjukkan *Setpoint* $33,5^{\circ}\text{C}$ merupakan titik operasi yang paling stabil dan presisi dibanding *setpoint* yang lain dengan hasil *Overshoot* terendah (2,39%) dan *steady state error* yang rendah 0%. Pada pengujian dengan metode Fuzzy, hasil menunjukkan suhu berada pada rentang yang optimal yaitu $32,8^{\circ}\text{C}$ - $35,0^{\circ}\text{C}$ dengan suhu rata-rata $34,04^{\circ}\text{C}$. Berdasarkan hasil penelitian, implementasi PID dan Logika Fuzzy mampu meningkatkan kestabilan suhu kandang secara otomatis, responsif, dan efisien untuk mendukung kesehatan serta pertumbuhan ayam DOC pada fase awal pemeliharaan di lingkungan kandang tertutup.

Kata kunci: DOC, ESP32, IoT, Logika Fuzzy, PID, Suhu Kandang

[Implementation of PID Control and Fuzzy Logic for Optimization of Temperature Regulation of Day-Old Chick (DOC) Cage Based on Internet of Things]

Abstract

This research aims to maintain the temperature stability of a Day-Old Chicken (DOC) coop through the implementation and performance analysis of PID control and Fuzzy Logic methods in an IoT-based coop temperature optimization system. The system was designed using an ESP32 microcontroller as the central controller, a DHT22 sensor as the temperature reader, an AC light dimmer as the actuator, and a heating lamp as the heat source. The PID method uses the K_p, K_i, and K_d parameters to determine output stability, while the Fuzzy Logic method uses COLD, WARM, and HOT membership functions to determine the ON/OFF condition of the heating lamp. The PID implementation is integrated with a Telegram Bot for system monitoring and control, while the Fuzzy Logic method is connected to a Cloud API and Google Sheets for real-time data storage and retrieval. Sensor calibration was performed by comparing the temperature readings of the DHT22 sensor with a GEA wood thermometer, yielding a measurement difference of 0.1°C–0.8°C. PID control testing showed that the 33.5°C setpoint was the most stable and precise operating point compared to the other setpoints, with the lowest Overshoot (2.39%) and a Steady-state error of 0%. In testing with the Fuzzy method, the results showed that the temperature remained within the optimal range of 32.8°C–35.0°C, with an average temperature of 34.04°C. Based on the research results, the implementation of PID and Fuzzy Logic is able to improve coop temperature stability automatically, responsively, and efficiently to support the health and growth of DOC chickens during the early rearing phase in an enclosed coop environment.

Keywords: Coop temperature, DOC, ESP32, Fuzzy Logic, IoT, PID

Daftar Isi

Lembar Pengesahan	i
Abstrak	ii
<i>Abstract</i>	iii
Daftar Isi	iv
Daftar Gambar	viii
Daftar Tabel	ix
Bab 1. Pendahuluan	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	4
1.4 Manfaat Penelitian	4
1.5 Batasan Masalah	4
1.6 <i>Work Breakdown Structure</i>	5
Bab 2. Tinjauan Pustaka	6
2.1 Kajian Pustaka	6
2.2 Deskripsi Teori Dan Komponen	7
2.2.1 Sistem Kontrol PID	7
2.2.2 Komponen Kontrol PID	8
2.2.3 Metode <i>Tuning</i> PID	8
2.2.3.1 Metode Ziegler-Nichols	8
2.2.4 Parameter Performansi Sistem Kontrol PID	10
2.2.4.1 <i>Overshoot</i>	10
2.2.4.2 <i>Rise-time (Tr)</i>	11
2.2.4.3 <i>Settling time (Ts)</i>	11
2.2.4.4 <i>Steady-state error (SSE)</i>	11
2.2.5 Fuzzy Logic Controller	12
2.2.6 Parameter Kinerja Sistem Kontrol Logika Fuzzy	13

2.2.6.1 <i>Overshoot</i>	13
2.2.6.2 Rise Time	14
2.2.6.3 Settling Time.....	16
2.2.6.4 <i>Steady-state error</i> (SSE).....	17
2.2.7 <i>Internet Of Things</i>	18
2.2.8 Arduino IDE.....	19
2.2.9 ESP 32	19
2.2.10 DHT 22	20
2.2.11 AC <i>Light Dimmer Module</i>	20
Bab 3. Metodologi Penelitian	22
3.1 Alur Penelitian	22
3.2 Blok Diagram Sistem.....	23
3.2.1 Blok Diagram Sistem Kontrol PID.....	23
3.2.2. Blok Diagram Sistem Kontrol Logika Fuzzy	25
3.3 Perancangan Perangkat Keras.....	26
3.3.1 Perancangan Mekanikal	26
3.3.2 Perancangan Elektrikal	27
3.4 Perancangan Perangkat Lunak	29
3.4.1 Perancangan Sistem Kontrol PID	29
3.4.2 Perancangan Sistem Logika Fuzzy	31
3.5 Perancangan Metode Penelitian	33
3.5.1 Perancangan Kontrol PID.....	33
3.5.2 Perancangan Logika Fuzzy	35
3.5.2.1 Fungsi Keanggotaan <i>Input</i>	36
3.5.2.2 Basis Aturan Fuzzy	38
3.5.2.3 Metode Inferensi Fuzzy Berbasis Pembobotan Derajat Keanggotaan.....	39
3.6 Alat Dan Bahan.....	41
3.7 Jadwal Pelaksanaan.....	44

Bab 4. Hasil Dan Pembahasan	47
4.1 Hasil Pengujian	47
4.1.1 Pengujian Sensor	47
4.1.2 Pengujian Parameter PID.....	48
4.1.2.1 Pengujian <i>Autotune</i> Ziegler-Nichols	48
4.1.2.2 Grafik Osilasi Suhu dan <i>Output</i> Dimmer	49
1) Respons Suhu pada <i>Setpoint</i> 32°C.....	50
2) Respons Suhu pada <i>Setpoint</i> 33,5°C.....	52
3) Respons Suhu pada <i>Setpoint</i> 35°C.....	53
4.1.2.3 Analisis Perbandingan Ketiga <i>Setpoint</i>	55
4.1.2.4 Pengujian Keandalan Sistem.....	57
1) PENGUJIAN SESI 1 ($\pm$ 8 JAM)	57
2) PENGUJIAN SESI 2 ($\pm$ 13 JAM)	58
3) PENGUJIAN SESI 3 ($\pm$ 3 JAM)	59
4.1.2.5 Kesimpulan Pengujian Keandalan Sistem PID.....	60
4.1.3 Pengujian Sistem Kontrol Logika Fuzzy.....	61
4.1.3.1 Konfigurasi Pengujian	61
4.1.3.2 Kondisi dan Konfigurasi Pengujian	62
4.1.3.3 Hasil Pengujian Sistem Kontrol Logika Fuzzy	62
4.1.3.4 Analisis Respons Suhu Terhadap <i>Setpoint</i>	65
4.1.3.5 Distribusi Keluaran Daya <i>AC Light Dimmer</i>	67
4.1.3.6 Distribusi Derajat Keanggotaan Fuzzy	68
4.1.3.7 Analisis Statistik Hasil Pengujian Pada Setiap Sesi	70
4.1.3.8 Analisis <i>Rise-Time</i>	71
4.1.3.9 Analisis <i>Settling-time</i>	73
4.1.3.10 Analisis <i>Steady-state error</i> (SSE).....	74
4.1.3.11 Analisis <i>Overshoot</i>	75
4.1.4 Pengujian Kedua Metode Kontrol Pada Anak Ayam	77
4.1.5 Perbandingan Langsung PID (<i>Setpoint</i> 35 °C) vs Logika Fuzzy	78

4.2 Pembahasan	81
Bab 5. Penutup	83
5.1 Simpulan.....	83
5.2 Saran	84
Daftar Pustaka	85
Lampiran	90
Lampiran 1	90
Lampiran 2	90
Lampiran 3	106

Daftar Gambar

Gambar 2. 1 <i>Internet of Things</i>	19
Gambar 2. 2 Arduino IDE.....	19
Gambar 2. 3 ESP 32	20
Gambar 2. 4 DHT 22	20
Gambar 2. 5 RobotDyn AC <i>Light Dimmer</i>	21
Gambar 3. 1 Alur Penelitian	23
Gambar 3. 2 Blok Diagram Sistem Kontrol PID.....	24
Gambar 3. 3 Blok Diagram Sistem Kontrol Logika Fuzzy	25
Gambar 3. 4 Desain kandang.....	26
Gambar 3. 5 Tampak depan	27
Gambar 3. 6 Desain <i>Elektrical</i>	28
Gambar 3. 7 <i>Flowchart</i> Perancangan Sistem Kontrol PID	30
Gambar 3. 8 Menu telegram bot PID	31
Gambar 3. 9 <i>Flowchart</i> Perancangan Perangkat Lunak logika fuzzy	32
Gambar 3. 10 Diagram alir PID	33
Gambar 3. 11 <i>Flowchart</i> PID.....	34
Gambar 3. 12 Diagram Alir Logika Fuzzy	35
Gambar 4. 1 Grafik osilasi suhu dan output dimmer selama proses <i>autotune</i>	49
Gambar 4. 2 Grafik Respons Suhu terhadap <i>Setpoint</i> 32°C.....	51
Gambar 4. 3 Grafik Respons Suhu terhadap <i>Setpoint</i> 33,5°C.....	52
Gambar 4. 4 Grafik Respons Suhu terhadap <i>Setpoint</i> 35°C.....	54
Gambar 4. 5 Grafik respons suhu terhadap waktu — Sesi 1 ($\pm$ 8 jam)	58
Gambar 4. 6 Grafik respons suhu terhadap waktu — Sesi 2 ($\pm$ 13 jam)	59
Gambar 4. 7 Grafik respons suhu terhadap waktu — Sesi 3 ($\pm$ 3 jam)	60
Gambar 4. 8 Grafik Respons Suhu terhadap Waktu Sesi 1	63
Gambar 4. 9 Grafik Respons Suhu terhadap Waktu Sesi 2	63
Gambar 4. 10 Grafik Respons Suhu terhadap Waktu Sesi 3	64
Gambar 4. 11 Respons Suhu Sistem Kontrol Logika Fuzzy Selama 24 Jam.....	66
Gambar 4. 12 Hubungan Suhu Aktual terhadap Keluaran	68
Gambar 4. 13 Distribusi Derajat Keanggotaan <i>COLD</i> , <i>WARM</i> , dan <i>HOT</i> terhadap Suhu Aktual	70
Gambar 4. 14 Penentuan T10, T90, dan Rise-Time pada Tiga Sesi Pengujian	72
Gambar 4. 15 Error Suhu terhadap Setpoint 35 °C Selama 24 Jam	74
Gambar 4. 16 Overshoot Sistem Logika Fuzzy Tiap Sesi Terhadap Setpoint 35°C	76
Gambar 4. 17 Respon Anak Ayam Ketika Suhu Belum Stabil	77
Gambar 4. 18 Respon Anak Ayam Ketika Suhu Telah Stabil.....	78
Gambar 4. 19 Perbandingan respons suhu aktual terhadap waktu untuk kontrol PID dan Logika Fuzzy pada <i>setpoint</i> 35°C.	78

Daftar Tabel

Tabel 1. 1 <i>Work Breakdown Structure</i>	5
Tabel 2. 1 Rumus <i>Tuning Ziegler-Nichols</i> Metode <i>Closed-loop</i> [32],[38].....	10
Tabel 3. 1 Konfigurasi PIN.....	28
Tabel 3. 2 Aturan Fuzzy Berdasarkan Kondisi Suhu	39
Tabel 3. 3 Alat dan bahan	42
Tabel 3. 4 Jadwal penelitian	45
Tabel 4. 1 Pengujian sensor.....	47
Tabel 4. 2 Hasil perhitungan <i>autotune</i> Ziegler-Nichols.....	49
Tabel 4. 3 Parameter Respons Transien pada <i>Setpoint</i> 32°C.....	51
Tabel 4. 4 Parameter Respons Transien pada <i>Setpoint</i> 33.5°C.....	53
Tabel 4. 5 Parameter Respons Transien pada <i>Setpoint</i> 35°C.....	54
Tabel 4. 6 Perbandingan Parameter Respons Transien pada Ketiga <i>Setpoint</i>	55
Tabel 4. 7 Hasil Statistik Suhu dan Keluaran Daya Setiap Sesi.....	64
Tabel 4. 8 Statistik Keseluruhan Pengujian.....	66
Tabel 4. 9 Distribusi Keluaran Daya AC Light Dimmer	67
Tabel 4. 10 Distribusi Derajat Keanggotaan Fuzzy.....	69
Tabel 4. 11 Perbandingan Statistik Tiga Sesi	71
Tabel 4. 12 Hasil Perhitungan Rise-time Sistem Logika Fuzzy	72
Tabel 4. 13 Hasil Perhitungan <i>Settling-time</i> Sistem Logika Fuzzy	73
Tabel 4. 14 Hasil Perhitungan <i>Steady-state error</i> (SSE)	74
Tabel 4. 15 Hasil Perhitungan <i>Overshoot</i> Sistem Logika Fuzzy.....	76
Tabel 4. 16 Perbandingan Parameter Kinerja Kontrol PID vs Logika Fuzzy.....	79

Bab 1. Pendahuluan

1.1 Latar Belakang

Sektor peternakan unggas di Indonesia memiliki peranan strategis dalam menopang ketahanan pangan nasional, khususnya sebagai penyedia utama sumber protein hewani bagi masyarakat. Seiring dengan pertumbuhan jumlah penduduk dan meningkatnya kesadaran gizi, permintaan terhadap produk unggas, terutama ayam pedaging, terus mengalami peningkatan. Dalam rantai produksi perunggasan, fase pemeliharaan anak ayam pada masa awal kehidupan atau *Day Old Chick* (DOC) merupakan tahap paling kritis yang sangat menentukan keberhasilan pemeliharaan pada fase selanjutnya. DOC memiliki nilai ekonomi yang tinggi, namun pada saat yang sama memiliki tingkat kerentanan yang sangat tinggi terhadap perubahan lingkungan karena sistem fisiologis dan mekanisme pengaturan suhu tubuhnya belum berkembang secara sempurna [1], [12].

Secara biologis, DOC masih tergolong poikilotermik, yaitu belum mampu mengatur suhu tubuhnya secara mandiri sehingga sangat bergantung pada suhu lingkungan di sekitarnya. Ketidakstabilan suhu pada masa brooding dapat menyebabkan DOC mengalami stres termal, baik akibat suhu yang terlalu rendah maupun terlalu tinggi. Idealnya, suhu yang dibutuhkan agar ayam DOC 0 – 7 hari dapat berada pada fase perkembangan yang nyaman yaitu pada rentang 32-35°C [43]. Suhu yang berada di bawah kisaran ideal dapat mengakibatkan hipotermia, penurunan metabolisme, kegagalan penyerapan nutrisi dari kuning telur, serta meningkatnya angka mortalitas [12], [29]. Oleh karena itu, kestabilan suhu kandang menjadi faktor utama dalam menjaga kesehatan dan kelangsungan hidup DOC pada usia 0–7 hari.

Permasalahan pengendalian suhu kandang DOC semakin kompleks apabila dikaitkan dengan karakteristik iklim tropis Indonesia yang memiliki fluktuasi suhu lingkungan cukup tinggi, terutama pada malam hari dan masa peralihan musim. Perubahan suhu lingkungan yang cepat dan tidak menentu sering kali menyebabkan suhu kandang sulit dikontrol secara manual. Metode konvensional yang umum digunakan oleh peternak rakyat, yaitu pengaturan lampu pemanas secara manual atau sistem *on/off*, memiliki keterbatasan karena hanya bekerja secara diskrit tanpa mempertimbangkan besarnya deviasi suhu terhadap nilai *setpoint* [2], [4]. Akibatnya, suhu kandang cenderung berosilasi dan tidak stabil, sehingga berpotensi membahayakan DOC.

Perkembangan teknologi sistem kontrol menawarkan solusi yang lebih presisi dalam mengatasi permasalahan tersebut. Salah satu metode yang banyak digunakan adalah kontrol *Proportional Integral Derivative* (PID), yang bekerja berdasarkan perhitungan kesalahan antara suhu aktual dan suhu referensi (*setpoint*). Kontrol PID mampu memberikan respon yang cepat dan presisi serta menekan fluktuasi suhu apabila parameter pengendaliannya ditentukan dengan tepat [4], [6], [13]. Namun demikian, penerapan PID pada sistem biologis seperti kandang DOC yang bersifat *non-linear* dan dipengaruhi gangguan lingkungan eksternal masih memerlukan pengujian lebih lanjut untuk memastikan keandalannya.

Sebagai alternatif, Logika Fuzzy dikembangkan untuk menangani sistem yang memiliki ketidakpastian dan *non-linearitas* tanpa memerlukan model matematis yang kompleks. Logika Fuzzy meniru cara berpikir manusia melalui aturan linguistik berbentuk *If-Then* sehingga mampu menentukan tingkat intensitas pemanas secara adaptif berdasarkan kondisi suhu yang terdeteksi [5], [7], [24]. Fleksibilitas Logika Fuzzy menjadikannya potensial untuk diterapkan pada pengendalian suhu kandang DOC yang dipengaruhi oleh banyak variabel eksternal.

Berdasarkan perbedaan karakteristik kedua metode tersebut, diperlukan suatu kajian komparatif untuk mengetahui metode kontrol yang paling optimal dalam menjaga kestabilan suhu kandang DOC. Penelitian ini berfokus pada implementasi dan pengujian kontrol PID dan Logika Fuzzy secara terpisah pada sistem pemanas kandang DOC dengan sumber panas berupa lampu pijar, tanpa penggunaan sistem pendingin tambahan. Selain itu, sistem dikombinasikan dengan teknologi *Internet of Things* (IoT) berbasis mikrokontroler ESP32 untuk memungkinkan proses *Monitoring* suhu secara *real-time* melalui aplikasi Telegram [26], [28]. Dengan adanya sistem *Monitoring* jarak jauh, peternak dapat melakukan pengawasan kondisi kandang secara kontinu dan responsif.

Selain faktor suhu, keberhasilan pemeliharaan DOC juga sangat dipengaruhi oleh kemampuan sistem dalam merespons perubahan lingkungan secara cepat dan berkelanjutan. Keterlambatan respon sistem pemanas terhadap penurunan suhu lingkungan, khususnya pada malam hari, dapat menyebabkan DOC mengalami stres dingin dalam waktu singkat. Kondisi ini sering kali tidak disadari secara langsung oleh peternak apabila tidak tersedia sistem *Monitoring* yang kontinu dan *real-time*. Oleh karena itu, integrasi sistem kontrol otomatis dengan sistem *Monitoring* berbasis IoT menjadi kebutuhan yang semakin relevan dalam manajemen kandang DOC modern.

Penggunaan lampu pijar sebagai sumber panas dalam penelitian ini dipilih karena masih banyak digunakan oleh peternak skala kecil dan menengah di Indonesia. Meskipun sederhana, karakteristik panas dari lampu pijar yang bersifat tidak instan dan dipengaruhi oleh tegangan listrik serta kondisi lingkungan menjadikannya menantang untuk dikontrol secara presisi. Hal ini menuntut adanya algoritma kontrol yang mampu mengatur daya pemanas secara adaptif agar suhu kandang tetap berada pada rentang ideal tanpa menyebabkan pemborosan energi atau lonjakan suhu yang berbahaya bagi DOC.

Berdasarkan permasalahan tersebut, penelitian ini berfokus pada implementasi kontrol PID dan Logika Fuzzy untuk mengatur suhu kandang DOC secara otomatis. Pada metode Logika Fuzzy, suhu aktual hasil pembacaan DHT22 diproses melalui fungsi keanggotaan dan aturan Fuzzy untuk menghasilkan nilai keluaran yang digunakan dalam mengatur daya *heating lamp* melalui AC light dimmer. Sementara itu, metode PID menghasilkan keluaran pengendalian berdasarkan parameter proporsional, integral, dan derivatif. Kedua metode kemudian diuji pada *setpoint* 35°C untuk mengetahui kemampuan masing-masing metode dalam mempertahankan kestabilan suhu serta untuk memperoleh perbandingan kinerja berdasarkan parameter respons sistem kontrol.

Dengan demikian, penelitian berjudul “Implementasi Kontrol PID dan Logika Fuzzy untuk Optimasi Regulasi Suhu Kandang *Day Old Chick* (DOC) Berbasis *Internet of Things*” ini diharapkan mampu memberikan kontribusi ilmiah berupa analisis yang lebih komprehensif terhadap kinerja kedua metode kontrol dalam menjaga kestabilan suhu kandang DOC. Hasil penelitian ini diharapkan tidak hanya menjadi pembandingan teknis antara kontrol PID dan Logika Fuzzy, tetapi juga menjadi rujukan praktis bagi peternak dan peneliti dalam menerapkan sistem otomasi pemanas kandang DOC yang efisien, adaptif, dan mudah diimplementasikan di lapangan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, rumusan masalah yang dapat diambil pada penelitian ini adalah sebagai berikut:

1. Bagaimana merancang sistem yang mampu mengontrol suhu kandang DOC sesuai dengan kriteria suhu yang ditentukan?
2. Bagaimana merancang sistem yang mampu memonitoring suhu kandang DOC secara *real-time* berbasis IoT?

3. Bagaimana perbandingan kinerja kontrol PID dan Logika Fuzzy dalam menjaga kestabilan suhu kandang DOC?

1.3 Tujuan Penelitian

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Merancang sistem pengontrol dan *Monitoring* suhu kandang DOC berbasis mikrokontroler ESP32.
2. Mengimplementasikan metode kontrol PID dan Logika Fuzzy pada sistem pengendalian suhu kandang DOC.
3. Menganalisis dan membandingkan kinerja kontrol PID dan Logika Fuzzy dalam menjaga kestabilan suhu kandang DOC.

1.4 Manfaat Penelitian

Adapun manfaat yang telah diperoleh dalam penelitian ini adalah sebagai berikut:

1. Memberikan solusi teknologi pengendalian suhu kandang DOC yang lebih stabil dan presisi.
2. Membantu peternak dalam meningkatkan efisiensi waktu dan tenaga melalui sistem otomatisasi dan *Monitoring* jarak jauh.
3. Menjadi referensi ilmiah bagi penelitian selanjutnya di bidang sistem kontrol dan teknologi peternakan unggas.

1.5 Batasan Masalah

Berdasarkan uraian di atas dapat ditarik beberapa hal yang perlu diperhatikan dalam merumuskan batasan masalah, antara lain sebagai berikut;

1. Sensor suhu yang digunakan adalah DHT22.
2. Mikrokontroler yang digunakan adalah ESP32.
3. Kontrol *Autotune* dan *Monitoring* data suhu secara *real-time* untuk metode PID adalah melalui Telegram.
4. *Monitoring* data suhu secara *real-time* untuk metode Logika Fuzzy adalah melalui *Google Sheets*.
5. Objek penelitian adalah anak ayam (*DOC*) berumur 0–7 hari.
6. Jumlah sampel yang digunakan dalam penelitian ini adalah 8 ekor DOC.

1.6 Work Breakdown Structure

Work Breakdown Structure (WBS) adalah cara untuk membagi sebuah proyek besar menjadi beberapa bagian sehingga dapat dengan mudah diselesaikan, Proyek dipecah dimulai dari tujuan utama, lalu menjadi pekerjaan besar, kemudian diuraikan lagi menjadi tugas - tugas detail. Dengan WBS, tim bisa lebih jelas melihat apa saja yang harus dilakukan, siapa yang bertanggung jawab, serta lebih mudah mengatur waktu dan biaya agar proyek berjalan terarah dan tidak ada pekerjaan yang terlewat. Tabel 1.1 di bawah ini merupakan pembagian tugas dan tanggung jawab pada penelitian ini.

Tabel 1. 1 *Work Breakdown Structure*

No	Nama	Tugas dan Tanggung Jawab dalam Tim
1	Gabriel Tambunan (4242001041)	1. Studi literatur pada topik Logika fuzzy. 2. Meredesain kandang ayam DOC yang telah ada. 3. Membuat code program Logika Fuzzy. 4. Mengimplementasikan Code program ke MCU. 5. Melakukan Pengujian dan Dokumentasi sesuai dengan Topik logika Fuzzy. 6. Menganalisis hasil pengujian.
2	Risky Soleman (4242011039)	1. Studi literatur pada topik kontrol PID. 2. Merebuild kandang penelitian ayam DOC. 3. Merancang dan membangun elektrikal perangkat penelitian. 4. Menginstal perangkat sistem ke kandang ayam DOC 5. Membuat code program kontrol PID. 6. Mengimplementasikan Code program ke MCU. 7. Melakukan Pengujian dan Dokumentasi sesuai dengan topik kontrol PID. 8. Menganalisis hasil pengujian.

Bab 2. Tinjauan Pustaka

2.1 Kajian Pustaka

Penelitian terkait pengendalian dan *Monitoring* suhu kandang ayam, khususnya pada fase DOC, telah banyak dilakukan dengan berbagai pendekatan metode kontrol. Fattah, dkk. (2023) menegaskan bahwa pengaturan suhu dan kelembaban yang stabil pada kandang ayam sangat berpengaruh terhadap performa broiler, terutama pada fase awal pemeliharaan [1]. Ketidakstabilan suhu pada masa DOC dapat menimbulkan stres fisiologis yang berdampak pada penurunan produktivitas dan meningkatnya angka kematian [12].

Metode kontrol PID menjadi salah satu pendekatan yang paling umum digunakan dalam pengontrolan suhu. Shafiudin dan Kholis (2018) menerapkan kontrol PID pada sistem inkubator telur dan menunjukkan bahwa metode ini mampu menjaga suhu dengan tingkat presisi yang lebih baik dibandingkan kontrol *on/off* [2]. Hal serupa ditunjukkan oleh Caesario, dkk. (2023) yang mengimplementasikan PID *controller* pada kandang ayam broiler dan memperoleh hasil kestabilan suhu yang lebih baik serta fluktuasi yang lebih kecil [4]. Penelitian Hidayah, dkk. (2024) juga memperlihatkan bahwa penggunaan mikrokontroler ESP32 dengan kontrol PID mampu menghasilkan respon suhu yang cepat dan akurat pada sistem pemanas [6].

Di sisi lain, Logika Fuzzy dikembangkan sebagai metode kontrol alternatif untuk sistem yang bersifat *non-linear* dan memiliki ketidakpastian tinggi. Aborisade dan Oladipo (2014) mengkombinasikan metode Fuzzy-PID pada sistem kandang ayam dan menunjukkan peningkatan kestabilan suhu dibandingkan metode konvensional [3]. Wijaya dan Hartati (2024) menerapkan Logika Fuzzy pada sistem pendingin kandang ayam *closed house* dan berhasil meningkatkan kenyamanan termal secara signifikan [5]. Penelitian Masyhuda, dkk. (2024) serta Trinaldi, dkk. (2024) juga membuktikan bahwa Logika Fuzzy Mamdani dan Tsukamoto efektif dalam menentukan intensitas pemanas berdasarkan kondisi suhu aktual tanpa memerlukan model matematis yang kompleks [7], [9].

Perkembangan teknologi IoT turut mendorong integrasi sistem *Monitoring* jarak jauh pada kandang ayam. Pasnur, dkk. (2019) serta Imam dan Abdillah (2024) mengembangkan sistem *Monitoring* suhu kandang ayam berbasis IoT yang memungkinkan pemantauan kondisi kandang secara *real-time* [26], [28].

Hadyanto dan Amrullah (2022) menegaskan bahwa *Monitoring* suhu yang responsif sangat penting untuk mencegah keterlambatan penanganan ketika suhu kandang berada di luar batas ideal [29].

Meskipun berbagai penelitian telah membahas pengontrolan suhu kandang ayam menggunakan PID maupun Logika Fuzzy, sebagian besar penelitian masih berfokus pada satu metode kontrol atau mengombinasikan keduanya dengan sistem pendingin tambahan. Oleh karena itu, masih terdapat celah penelitian terkait perbandingan kinerja kontrol PID dan Logika Fuzzy secara terpisah pada sistem pemanas kandang DOC yang hanya menggunakan lampu pijar sebagai sumber panas utama. Selain aspek kontrol suhu, beberapa penelitian juga menekankan pentingnya keandalan sensor dan akurasi akuisisi data dalam sistem pengontrolan suhu. Pratama, dkk. (2024) serta Sumantri, dkk. (2024) menunjukkan bahwa pemilihan sensor suhu yang tepat dan sistem akuisisi data yang stabil sangat berpengaruh terhadap kinerja kontrol PID maupun Logika Fuzzy. Kesalahan pembacaan sensor dapat menyebabkan respon kontrol yang tidak sesuai, sehingga berdampak langsung pada kestabilan suhu kandang.

Dengan mempertimbangkan aspek biologis DOC, karakteristik sistem pemanas, metode kontrol, serta integrasi IoT, penelitian ini diharapkan dapat mengisi celah penelitian yang ada dengan menyajikan analisis komparatif yang komprehensif mengenai karakteristik respon, kestabilan suhu, dan efektivitas kontrol PID dan Logika Fuzzy dalam menjaga kondisi termal optimal bagi DOC.

2.2 Deskripsi Teori Dan Komponen

2.2.1 Sistem Kontrol PID

Kontrol PID (*Proportional-Integral-Derivative*) merupakan salah satu jenis pengontrol yang paling banyak digunakan dalam sistem kontrol industri maupun penelitian. Kontrol ini bekerja dengan menghasilkan sinyal kontrol berdasarkan tiga aksi, yaitu proporsional, integral, dan derivatif terhadap sinyal kesalahan (*error*) antara nilai *setpoint* dan nilai aktual keluaran sistem (32).

2.2.2 Komponen Kontrol PID

Setiap komponen kontrol PID memiliki peran yang berbeda terhadap respons sistem. Pemahaman terhadap pengaruh masing-masing parameter menjadi dasar penting dalam proses penalaan (*tuning*) kontroler [34].

a) Aksi Proporsional (K_p)

Aksi proporsional menghasilkan sinyal kontrol yang sebanding dengan nilai kesalahan saat ini. Peningkatan nilai K_p akan mempercepat *rise-time* dan mengurangi *Steady-state error*, namun dapat menyebabkan *Overshoot* yang lebih besar dan potensi ketidakstabilan sistem [32].

b) Aksi Integral (K_i)

Aksi integral bekerja berdasarkan akumulasi kesalahan dari waktu ke waktu. Komponen ini efektif untuk mengeliminasi *Steady-state error* secara permanen. Namun, nilai K_i yang terlalu besar dapat menyebabkan respons sistem berosilasi dan memperpanjang *Settling time* [33].

c) Aksi Derivatif (K_d)

Aksi derivatif bekerja berdasarkan laju perubahan sinyal kesalahan, sehingga bersifat antisipatif terhadap perubahan yang akan datang. Komponen ini membantu mengurangi *Overshoot* dan mempercepat *settling time*.

Pada sistem termal, nilai K_d umumnya kecil atau bahkan dapat diabaikan karena respons sistem yang lambat [35].

2.2.3 Metode *Tuning* PID

Penalaan (*tuning*) kontrol PID merupakan proses penentuan nilai parameter K_p , K_i , dan K_d yang optimal agar sistem menghasilkan respons yang diinginkan. Terdapat berbagai metode *tuning* yang dapat digunakan salah satunya yaitu menggunakan metode Ziegler-Nichols [35].

2.2.3.1 Metode Ziegler-Nichols

Metode Ziegler-Nichols pertama kali diperkenalkan oleh J.G. Ziegler dan N.B. Nichols pada tahun 1942 melalui makalah berjudul "*Optimum Settings for Automatic Controllers*" yang dipublikasikan dalam *Transactions of the ASME*.

Metode ini merupakan pendekatan sistematis dan terukur yang dirancang untuk menghasilkan respons dengan decay ratio sebesar 0,25 (*quarter decay*), artinya amplitudo osilasi berikutnya berkurang menjadi seperempat dari sebelumnya sebagai kompromi antara kecepatan respons dan stabilitas sistem [31].

Terdapat dua varian metode Ziegler-Nichols, yaitu metode *open-loop* (kurva reaksi) dan metode *closed-loop* (osilasi tertutup). Pada penelitian ini digunakan metode *closed-loop* dengan prosedur sebagai berikut[36]:

- Menonaktifkan aksi integral dan derivatif dengan mengatur $K_i = 0$ dan $K_d = 0$.
- Menaikkan nilai K_p secara bertahap dari nol hingga sistem menghasilkan osilasi yang stabil (tidak teredam dan tidak meledak). Nilai K_p pada kondisi ini dicatat sebagai K_u (*Ultimate Gain*).
- Mengukur periode osilasi pada kondisi kritis tersebut sebagai T_u (*Ultimate Period*) dalam satuan detik.
- Menghitung parameter PID menggunakan tabel rumus Ziegler-Nichols dengan memasukkan nilai K_u dan T_u yang telah diperoleh.

Rumus perhitungan parameter PID menggunakan metode Ziegler-Nichols *closed-loop* untuk tipe kontrol PID adalah sebagai berikut :

$K_p = 0,6 \times K_u$ $T_i = T_u / 2 \rightarrow K_i = \frac{K_p}{T_i}$ $T_d = T_u / 8 \rightarrow K_d = \frac{K_p}{8} \times T_d$

Tabel 2. 1 Rumus *Tuning* Ziegler-Nichols Metode *Closed-loop* [32],[38]

Tipe	Kp	Ti	Ki	Td	Kd
P	0,5 Ku	∞	0	0	0
PI	0,45 Ku	Tu / 1,2	Kp / Ti	0	0
PID	0,6 Ku	Tu / 2	Kp / Ti	Tu / 8	Kp × Td

Metode Ziegler-Nichols terbukti efektif dan sudah digunakan secara luas di industri sejak tahun 1942. Namun, metode ini cenderung menghasilkan respons yang agresif dengan *Overshoot* sekitar 25%, sehingga seringkali masih diperlukan *fine-tuning* lanjutan setelah penerapan awal [33].

2.2.4 Parameter Performansi Sistem Kontrol PID

Dalam evaluasi kinerja sistem kontrol, terdapat beberapa parameter performansi yang umum digunakan untuk mengukur kualitas respons sistem terhadap sinyal masukan berupa fungsi tangga (*step input*). Parameter-parameter ini menjadi acuan utama dalam membandingkan hasil *tuning* metode yang berbeda, yaitu *Overshoot*, *rise-time* (*Tr*), *Settling time* (*Ts*), dan *Steady-state error* (*SSE*) [32].

2.2.4.1 *Overshoot*

Overshoot adalah kondisi di mana keluaran sistem melampaui nilai *setpoint* sebelum akhirnya stabil. *Overshoot* dinyatakan dalam persentase terhadap nilai *setpoint* dan dihitung menggunakan persamaan sebagai berikut [32] :

$$Mp = (T_{max} - T_{sp}) / T_{sp} \times 100\%$$

Pada rumus diatas, *Tmax* adalah suhu maksimum yang dicapai sistem (°C) dan *Tsp* adalah suhu *setpoint* (°C). *Overshoot* yang besar menandakan sistem terlalu agresif dalam merespons perubahan *setpoint*.

2.2.4.2 Rise-time (T_r)

Rise-time (T_r) didefinisikan sebagai waktu yang dibutuhkan keluaran sistem untuk naik dari 10% menjadi 90% dari perubahan suhu antara kondisi awal menuju *setpoint*. *Rise-time* merupakan indikator kecepatan respons sistem; semakin kecil nilai T_r , semakin cepat sistem merespons perubahan *setpoint*. Persamaan untuk menghitung *rise-time* secara umum adalah:

$$T_r = t_{90\%} - t_{10\%}$$

Pada rumus diatas, $t_{90\%}$ adalah waktu saat keluaran mencapai 90% dari perubahan menuju *setpoint* dan $t_{10\%}$ adalah waktu saat keluaran mencapai 10% dari perubahan menuju *setpoint*. Peningkatan nilai K_p pada kontrol PID umumnya mempercepat *rise-time*, namun berpotensi meningkatkan *Overshoot* [34].

2.2.4.3 Settling time (T_s)

Settling time (T_s) adalah waktu yang dibutuhkan keluaran sistem untuk masuk dan tetap berada dalam batas toleransi tertentu di sekitar nilai *setpoint* secara permanen.

Batas toleransi yang umum digunakan adalah $\pm 2\%$ atau $\pm 5\%$ dari nilai *setpoint*, tergantung pada standar yang diterapkan[32]. Secara matematis, *Settling time* dengan kriteria $\pm 2\%$ dinyatakan sebagai berikut:

T_s = waktu terkecil t sedemikian sehingga $|T(t) - T_{sp}| \leq 0,02 \times T_{sp}$, untuk semua $t \geq T_s$, dengan $T_{sp} = 35^\circ\text{C}$ sehingga pita toleransi adalah $34,3^\circ\text{C} \leq T(t) \leq 35,7^\circ\text{C}$

Settling time dipengaruhi oleh karakteristik osilasi dan redaman sistem. Nilai K_d yang tepat pada kontrol PID dapat membantu mempercepat *Settling time* dengan cara meredam osilasi yang terjadi setelah respons awal. Nilai *Settling time* yang besar menandakan bahwa sistem membutuhkan waktu lama untuk mencapai kondisi stabil setelah terjadi gangguan atau perubahan *setpoint* [33].

2.2.4.4 Steady-state error (SSE)

Steady-state error (SSE) adalah perbedaan yang tersisa antara nilai *setpoint* dan nilai keluaran sistem setelah sistem mencapai kondisi tunak (*steady-state*).

SSE mencerminkan ketidakmampuan sistem untuk mencapai nilai referensi yang diinginkan secara tepat [32]. SSE dihitung menggunakan persamaan sebagai berikut :

$$SSE = |SP - y_{ss}|$$

Pada rumus diatas, SP adalah nilai *setpoint* dan y_{ss} adalah nilai keluaran sistem pada kondisi *steady-state*. Aksi integral (Ki) pada kontrol PID berperan utama dalam mengeliminasi SSE, karena komponen integral terus mengakumulasi *error* hingga selisih antara *setpoint* dan keluaran sistem menjadi nol. Tanpa aksi integral, sistem dengan gangguan beban (*disturbance*) atau dengan karakteristik tertentu akan selalu memiliki SSE yang tidak dapat dihilangkan sepenuhnya[33].

Keempat parameter performansi tersebut saling berkaitan dan seringkali menunjukkan *trade-OFF* satu sama lain. Sebagai contoh, peningkatan kecepatan respons (penurunan *rise-time*) umumnya disertai dengan peningkatan *Overshoot*. Oleh karena itu, proses *tuning* PID pada dasarnya adalah upaya untuk mencapai keseimbangan yang optimal di antara keempat parameter ini sesuai dengan spesifikasi dan kebutuhan sistem yang dikontrol [32]).

2.2.5 Fuzzy Logic Controller

Fuzzy Logic Controller merupakan metode sistem kontrol yang menggunakan konsep derajat keanggotaan untuk merepresentasikan kondisi suatu variabel dalam bentuk linguistik. Berbeda dengan sistem kontrol konvensional yang menggunakan batas nilai secara tegas, *fuzzy logic* memungkinkan suatu nilai *input* memiliki derajat keanggotaan antara 0 hingga 1 pada suatu himpunan tertentu. Pendekatan tersebut memungkinkan sistem mengambil keputusan berdasarkan kondisi yang tidak sepenuhnya bersifat biner.

Pada penelitian ini, Fuzzy Logic Controller digunakan untuk mengatur temperatur kandang ayam *DOC* berdasarkan suhu aktual yang diperoleh dari sensor DHT22. Nilai suhu yang dibaca oleh sensor digunakan sebagai *input* sistem fuzzy dan diklasifikasikan ke dalam tiga himpunan, yaitu *COLD*, *WARM*, dan *HOT*. Hasil proses fuzzy kemudian digunakan untuk menentukan besarnya daya yang diberikan kepada *heating lamp* melalui *AC Light Dimmer*.

Penggunaan *AC Light Dimmer* menyebabkan keluaran sistem tidak lagi terbatas pada kondisi ON dan OFF, tetapi dapat berupa tingkat daya antara 0% hingga 100%. Dengan demikian, ketika suhu kandang masih rendah, sistem dapat memberikan daya pemanasan yang lebih besar.

Ketika suhu mendekati *setpoint*, daya pemanasan dapat dikurangi secara bertahap. Sementara itu, ketika suhu telah mencapai kondisi panas, daya pemanasan dapat dikurangi hingga 0%.

Pada penelitian ini, proses pengambilan keputusan fuzzy terdiri atas tahap fuzzifikasi, evaluasi aturan, dan defuzzifikasi. Tahap fuzzifikasi mengubah nilai suhu aktual menjadi derajat keanggotaan pada himpunan *COLD*, *WARM*, dan *HOT*. Selanjutnya, aturan fuzzy digunakan untuk menentukan tingkat daya pemanasan berdasarkan kondisi suhu. Hasil evaluasi aturan kemudian menghasilkan nilai keluaran yang digunakan sebagai referensi persentase daya *AC Light Dimmer*.

2.2.6 Parameter Kinerja Sistem Kontrol Logika Fuzzy

Parameter kinerja sistem kontrol digunakan untuk mengetahui kemampuan suatu sistem dalam memberikan respons terhadap perubahan nilai masukan dan mempertahankan keluaran pada kondisi yang diharapkan. Pada penelitian ini, parameter kinerja digunakan untuk mengevaluasi respons kontrol PID dan Logika Fuzzy dalam mengendalikan suhu kandang DOC terhadap nilai *setpoint* yang telah ditentukan. Penggunaan parameter kinerja yang sama pada kedua metode bertujuan agar perbandingan dapat dilakukan berdasarkan karakteristik respons sistem yang setara.

Parameter kinerja yang digunakan dalam penelitian ini meliputi *Overshoot*, *rise time*, *settling time*, dan *Steady-state error* (SSE). Keempat parameter tersebut digunakan untuk mengetahui kecepatan respons sistem dalam mencapai kondisi target, besarnya penyimpangan suhu setelah melewati *setpoint*, waktu yang dibutuhkan sistem untuk mencapai kondisi stabil, serta besarnya kesalahan pada kondisi tunak.

2.2.6.1 Overshoot

Overshoot merupakan kondisi ketika keluaran sistem melewati nilai *setpoint* selama proses menuju kondisi stabil. Parameter ini digunakan untuk mengetahui seberapa besar respons sistem melampaui nilai suhu yang ditargetkan. Pada sistem pengendalian suhu kandang DOC, *Overshoot* perlu diperhatikan karena kenaikan suhu yang terlalu tinggi di atas *setpoint* dapat menyebabkan kondisi termal yang tidak sesuai dengan kebutuhan DOC.

Pada penelitian ini, *Overshoot* dihitung berdasarkan nilai suhu maksimum yang terjadi selama respons terhadap *setpoint*. Karena nilai acuan pengujian adalah 35 °C, persentase *Overshoot* dihitung terhadap nilai *setpoint*, bukan terhadap nilai suhu steady-state.

Persamaan yang digunakan adalah:

$$M_p = (T_{max} - T_{sp}) / T_{sp} \times 100\%$$

Keterangan:

M_p = Persentase *Overshoot* (%)

T_{max} = Suhu maksimum yang dicapai sistem (°C)

T_{sp} = Suhu set point (°C)

Apabila suhu maksimum sistem melebihi *setpoint*, maka nilai *Overshoot* dihitung menggunakan persamaan tersebut. Sebaliknya, apabila selama pengujian suhu maksimum tidak melebihi *setpoint*, maka sistem dinyatakan tidak mengalami *Overshoot* dan nilai *Overshoot* ditetapkan sebesar 0%.

Sebagai contoh, apabila *setpoint* ditetapkan sebesar 35 °C dan suhu maksimum yang tercatat adalah 35,40 °C, maka:

$$M_p = (35.40 - 35) / 35 \times 100\% \\ M_p = 1.143\%$$

Dengan demikian, sistem mengalami *Overshoot* sebesar 1,143%. Nilai *Overshoot* yang semakin kecil menunjukkan bahwa sistem mampu mendekati suhu target dengan penyimpangan maksimum yang semakin rendah.

2.2.6.2 Rise Time

Rise-time merupakan salah satu parameter yang digunakan untuk mengetahui kecepatan respons suatu sistem kontrol dalam mencapai daerah kerja yang diharapkan setelah terjadi perubahan kondisi awal. Parameter ini menunjukkan waktu yang diperlukan oleh respons sistem untuk bergerak dari 10% hingga 90% dari perubahan nilai yang terjadi antara kondisi awal dan nilai tujuan. Semakin kecil nilai *rise-time*, semakin cepat sistem memberikan respons terhadap perubahan yang terjadi.

Pada sistem pengendalian suhu kandang DOC, *rise-time* digunakan untuk mengetahui seberapa cepat suhu aktual bergerak dari kondisi awal menuju *setpoint* yang telah ditentukan. Karena *setpoint* pengujian ditetapkan sebesar 35 °C, nilai 10% dan 90% tidak ditentukan dari nilai suhu absolut, melainkan dari perubahan suhu antara kondisi awal pengujian dan *setpoint*.

Pendekatan tersebut digunakan agar perhitungan *rise-time* dapat diterapkan secara konsisten pada sistem PID maupun Logika Fuzzy. Nilai suhu pada batas 10% dan 90% dapat ditentukan menggunakan persamaan berikut:

$$\begin{aligned}T_{10} &= T_{awal} + 0,1 (SP - T_{awal}) \\T_{90} &= T_{awal} + 0,9 (SP - T_{awal})\end{aligned}$$

dengan:

T_{10} = suhu pada saat respons mencapai 10%,

T_{90} = suhu pada saat respons mencapai 90%,

T_{awal} = suhu awal pengujian,

SP = nilai *setpoint*,

dan 0,1 serta 0,9 masing-masing menunjukkan 10% dan 90% dari perubahan suhu menuju *setpoint*.

Setelah nilai T_{10} dan T_{90} diperoleh, *rise-time* dihitung berdasarkan selisih waktu ketika respons mencapai kedua batas tersebut. Persamaannya adalah:

$$T_r = t_{90} - t_{10}$$

dengan:

T_r = *rise-time*,

t_{90} = waktu ketika suhu mencapai T_{90} ,

t_{10} = waktu ketika suhu mencapai T_{10} .

Sebagai contoh, apabila suhu awal pengujian adalah 33,69 °C dan *setpoint* sebesar 35 °C, maka perubahan suhu yang harus dicapai adalah:

$$\Delta T = 35 - 33.69 = 1.31^{\circ}\text{C}$$

Nilai 10% dan 90% dari perubahan tersebut adalah:

$$T_{10} = 33.69 + (0.1 \times 1.31) = 33.821^{\circ}\text{C}$$

$$T_{90} = 33.69 + (0.9 \times 1.31) = 34.869^{\circ}\text{C}$$

Selanjutnya, waktu ketika suhu aktual pertama kali mencapai 33,821 °C ditetapkan sebagai t_{10} , sedangkan waktu ketika suhu aktual pertama kali mencapai 34,869 °C ditetapkan sebagai t_{90} . Selisih kedua waktu tersebut merupakan *rise-time* sistem.

Metode perhitungan ini digunakan baik pada pengujian *PID* maupun Logika Fuzzy sehingga parameter *rise-time* kedua metode memiliki dasar perhitungan yang sama. Dengan demikian, nilai *rise-time* dapat digunakan sebagai salah satu parameter dalam melakukan komparasi kecepatan respons antara kontrol *PID* dan Logika Fuzzy pada *setpoint* 35 °C.

2.2.6.3 Settling Time

Settling-time merupakan waktu yang diperlukan sistem untuk mencapai dan mempertahankan kondisi stabil di sekitar nilai *setpoint*. Parameter ini berbeda dengan *rise-time* karena tidak hanya memperhatikan kapan suhu mencapai nilai tertentu, tetapi juga memperhatikan kemampuan sistem untuk tetap berada dalam daerah toleransi setelah mencapai daerah tersebut.

Pada penelitian ini, daerah toleransi *settling-time* ditentukan sebesar $\pm 2\%$ dari nilai *setpoint*. Penggunaan *setpoint* sebagai acuan dilakukan agar kriteria kestabilan yang digunakan pada kontrol *PID* dan Logika Fuzzy tetap sama. Dengan *setpoint* sebesar 35 °C, batas toleransi dapat dihitung sebagai:

$$\begin{aligned}\Delta T_s &= 0.02 \times T_{sp} \\ \Delta T_s &= 0.02 \times 35 \\ \Delta T_s &= 0.7^\circ\text{C}\end{aligned}$$

Dengan demikian, batas bawah daerah kestabilan adalah:

$$\begin{aligned}T_{lower} &= T_{sp} - 0.02T_{sp} \\ T_{lower} &= 35 - 0.7 \\ T_{lower} &= 34.3^\circ\text{C}\end{aligned}$$

Sedangkan batas atas adalah:

$$\begin{aligned}T_{upper} &= T_{sp} + 0.02T_{sp} \\ T_{upper} &= 35 + 0.7 \\ T_{upper} &= 35.7^\circ\text{C}\end{aligned}$$

Dengan demikian, sistem dinyatakan berada dalam daerah kestabilan apabila:

$$34.3^\circ\text{C} \leq T(t) \leq 35.7^\circ\text{C}$$

Settling-time ditentukan dari waktu ketika respons sistem memasuki daerah toleransi tersebut dan selanjutnya tetap berada di dalam daerah toleransi hingga akhir pengamatan. Dengan kriteria ini, suatu respons tidak langsung dianggap stabil hanya karena sesaat melewati 35 °C. Apabila setelah masuk ke daerah toleransi suhu kembali keluar dari batas 34,3–35,7 °C, maka waktu tersebut belum dapat ditetapkan sebagai. Penghitungan dilanjutkan sampai ditemukan waktu ketika respons telah berada di dalam daerah toleransi dan tidak lagi keluar dari daerah tersebut.

Pada sistem pengendalian suhu kandang DOC, *settling-time* digunakan untuk mengetahui kemampuan pengendali dalam meredam perubahan suhu dan mempertahankan suhu di sekitar *setpoint*. Nilai *settling-time* yang lebih kecil menunjukkan bahwa sistem lebih cepat mencapai kondisi stabil, sedangkan nilai yang lebih besar menunjukkan bahwa sistem membutuhkan waktu lebih lama untuk mencapai kondisi stabil.

2.2.6.4 Steady-state error (SSE)

Steady-state error (SSE) merupakan penyimpangan antara nilai keluaran sistem pada kondisi tunak dengan nilai *setpoint*. Parameter ini menunjukkan seberapa dekat suhu aktual dapat dipertahankan terhadap suhu target setelah respons transien sistem berakhir. Semakin kecil nilai SSE, semakin kecil penyimpangan suhu terhadap *setpoint* pada kondisi tunak.

Pada penelitian ini, SSE digunakan untuk mengetahui kemampuan kontrol PID dan Logika Fuzzy dalam mempertahankan suhu kandang di sekitar *setpoint* 35°C setelah sistem mencapai kondisi stabil. Nilai SSE diperoleh berdasarkan perbedaan antara nilai suhu pada kondisi tunak dan nilai *setpoint*. Parameter ini digunakan bersama dengan *Overshoot*, *rise time*, dan *Settling time* untuk memberikan gambaran yang lebih menyeluruh mengenai kinerja kedua metode pengendalian. Secara sederhana, SSE dapat dihitung menggunakan persamaan:

$$SSE = |T_{sp} - T_{ss}|$$

Keterangan:

SSE = *Steady-state error* (°C)

T_{sp} = suhu *setpoint* (°C)

T_{ss} = suhu rata-rata sistem pada kondisi tunak (°C).

Nilai T_{ss} diperoleh dari rata-rata sejumlah data suhu yang berada pada kondisi stabil yang telah ditentukan berdasarkan hasil pengujian. Secara umum, nilai tersebut dapat dihitung menggunakan:

$$T_{ss} = 1/N \sum (i=1 \text{ to } N) T_i$$

Keterangan:

N = jumlah data pada kondisi *steady-state*,

T_i = nilai suhu pada data ke-i pada kondisi *steady-state*.

Sebagai contoh, apabila suhu *steady-state* sistem berada pada rata-rata 34,82 °C dan *setpoint* yang digunakan adalah 35 °C, maka:

$$SSE = |35 - 34.82|$$

$$SSE = 0.18^\circ\text{C}$$

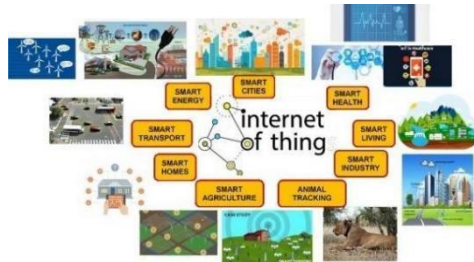
Nilai SSE yang semakin kecil menunjukkan bahwa suhu pada kondisi tunak semakin mendekati *setpoint*. Sebaliknya, nilai SSE yang semakin besar menunjukkan adanya penyimpangan suhu yang lebih besar dari nilai target.

Dalam penelitian ini, SSE digunakan sebagai salah satu parameter utama untuk membandingkan ketepatan kontrol PID dan Logika Fuzzy dalam mempertahankan suhu kandang DOC pada *setpoint* 35 °C.

2.2.7 Internet Of Things

Menurut Casagras, seorang koordinator dan *Support Action for Global RFID-Related Activities and Standardisation* atau Koordinator dan Aksi Dukungan untuk Kegiatan dan Standardisasi Global yang terkait dengan RFID mendefinisikan *Internet of Things* atau IOT adalah sebuah infrastruktur jaringan global, dimana, mereka menggabungkan benda (berupa fisik dan virtual) melalui kemampuan eksploitasi, rekaman serta komunikasi.

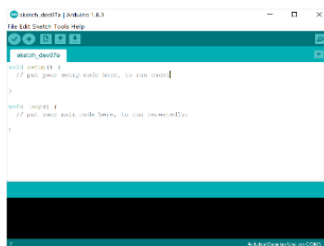
Infrastruktur tersebut terdiri daripada Jaringan yang telah ada, serta internet seperti sekarang ini dengan jaringannya yang sudah dikembangkan [39].



Gambar 2. 1 Internet of Things

2.2.8 Arduino IDE

Arduino IDE adalah *software* yang digunakan untuk membuat sketch pemrograman atau dengan kata lain arduino IDE sebagai media untuk pemrograman pada *board* yang ingin diprogram. Arduino IDE ini berguna untuk mengedit, membuat, meng-upload ke *board* yang ditentukan, dan meng-*coding* program tertentu. Arduino IDE dibuat dari bahasa pemrograman JAVA, yang dilengkapi dengan library C/C++(*wiring*), yang membuat operasi *input/output* lebih mudah. [42].



Gambar 2. 2 Arduino IDE

2.2.9 ESP 32

ESP32 adalah modul mikrokontroler terintegrasi yang memiliki fitur lengkap dan kinerja tinggi. Modul ini merupakan pengembangan dari modul WiFi ESP8266.

Terdapat dua CPU yang masing-masing memiliki tingkat kecepatan 80 MHz dan 160 MHz. Selain itu, ESP32 juga memiliki banyak perangkat tambahan seperti ADC, DAC, I2C, I2S, SPI, dan UART untuk berbagai macam aplikasi [40].



Gambar 2. 3 ESP 32

2.2.10 DHT 22

DHT22, atau dikenal juga dengan nama AM2302, adalah sensor digital yang digunakan untuk mengukur suhu dan kelembapan udara. Sensor ini mampu memberikan data yang cukup akurat dan stabil, membuatnya sangat cocok untuk digunakan dalam berbagai sistem *Monitoring* berbasis IoT [41].



Gambar 2. 4 DHT 22

2.2.11 AC Light Dimmer Module

Modul Peredup Lampu AC RobotDyn adalah pengontrol pemutus fasa berbasis TRIAC saluran tunggal yang dirancang untuk mengatur daya AC ke beban resistif dan beberapa beban induktif.

Modul ini memungkinkan mikrokontroler seperti Arduino, ESP8266, ESP32, atau STM32 untuk menyesuaikan kecerahan lampu pijar atau lampu LED yang dapat diredupkan secara halus, atau mengontrol kecepatan motor AC tertentu [44].

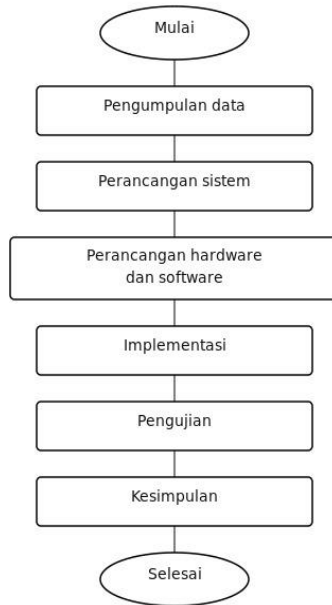


Gambar 2. 5 RobotDyn AC *Light* Dimmer

Bab 3. Metodologi Penelitian

3.1 Alur Penelitian

Dalam penelitian ini, hal pertama yang dilakukan ialah merancang kandang ayam DOC, kemudian dilanjutkan dengan merancang sistem kontrol suhu menggunakan pengontrol PID dan logika fuzzy. metode yang digunakan adalah pendekatan eksperimen percobaan dan perbaikan secara langsung untuk melihat kinerja antara mikrokontroler, sensor, AC *light* dimmer dan aplikasi yang dibuat dalam implementasinya pada Implementasi kontroler PID dan logika fuzzy untuk optimalisasi regulasi suhu pada kandang *day old chicks* (DOC). Dalam melakukan penelitian ini, diperlukan beberapa tahapan. Diawali dengan pengumpulan informasi yakni dengan meneliti studi sebelumnya. Tinjauan pustaka dilakukan dengan mengumpulkan data dan informasi dari hasil studi sebelumnya yang relevan dengan penelitian ini. Selanjutnya melakukan perancangan sistem Hardware dan software dengan menganalisis komponen hardware dan software yang tepat dan mengacu pada kebutuhan hasil analisis untuk memastikan bahwa perancangan dapat berjalan tepat dan sesuai dengan tujuan. Tahapan berikutnya, ialah pembuatan komponen hardware dan software yang mampu membangun koneksi diantara keduanya. Setelah itu, melakukan pengujian hasil dari perancangan sistem hardware dan software yang dilakukan dengan menempatkan instrumen penelitian pada berbagai kondisi. Terakhir, kesimpulan dilakukan setelah proses perancangan, pembuatan dan pengujian instrumen berhasil diselesaikan.

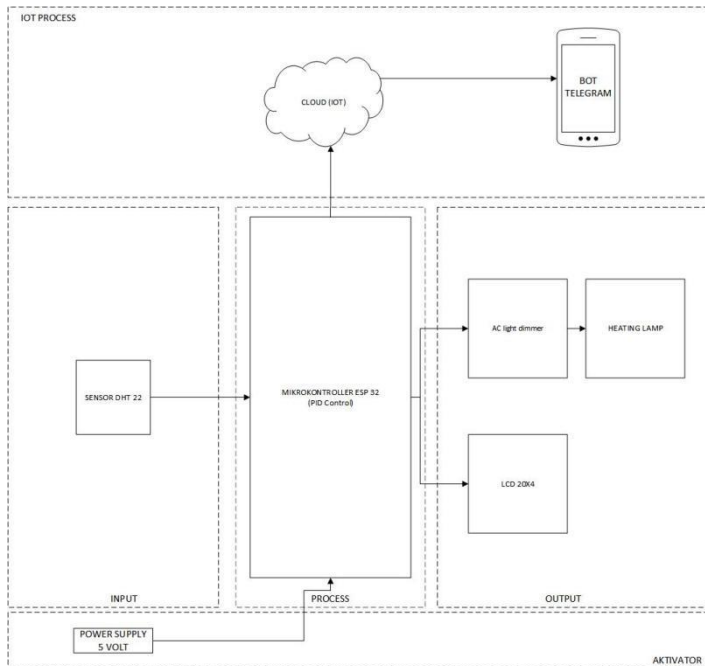


Gambar 3. 1 Alur Penelitian

3.2 Blok Diagram Sistem

3.2.1 Blok Diagram Sistem Kontrol PID

Dalam perancangan suatu sistem optimalisasi suhu kandang ayam DOC yang menggunakan kontrol PID, dibuat lah suatu gambaran bagaimana penerapan sistem tersebut berjalan. Sistem yang dirancang ini memerlukan beberapa komponen yang meliputi bagian *input*, *output*, Mikrokontroller. Blok diagram sistem dari rancangan sistem ini dapat dilihat pada gambar 3.2.



Gambar 3. 2 Blok Diagram Sistem Kontrol PID

Pada desain blok diagram menunjukkan bahwa sistem memiliki beberapa bagian bidang kerja yaitu *input*, *proses*, *output*, *aktivator* dan *proses iot*. Untuk sumber tegangan, menggunakan *power supply* yang memiliki daya 5 volt untuk mensuplai kebutuhan daya ke perangkat mikrokontroler yang merupakan sumber dari inti bagian sistem perancangan. Bagian *input* terdiri dari sensor suhu dan kelembaban yang menggunakan jenis sensor DHT-22.

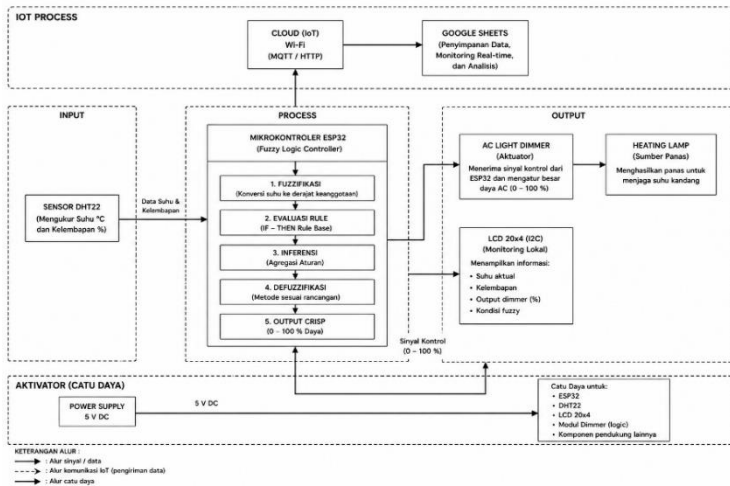
Untuk bagian *proses*, dikontrol oleh mikrokontroler model ESP-32 sebagai pusat pemrosesan data yang menghubungkan antara *input*, *output* dan *IOT* melalui bot telegram pada sistem. Proses melibatkan perintah kalkulasi PID, pengambilan nilai data actual dari DHT22 (*input*), eksekusi AC *light* dimmer (*output*) pada *heating lamp* dan tampilan status terkini sistem pada LCD dan pada bot telegram yang terhubung.

Pada bagian *output*, terdiri dari AC *light* dimmer 1 channel, yang terhubung dengan heater lamp. Pada bagian *output* juga terdapat LCD yang dapat menampilkan informasi tentang suhu dan nilai parameter PID.

Pada bagian IoT, alat kontrol suhu terhubung dengan bot telegram yang dapat digunakan untuk melihat informasi terkait suhu terkini. Pada bot telegram juga memungkinkan untuk melakukan kontrol pada beberapa segmen seperti kalibrasi suhu dan set *autotune* (ziegler-nichols).

3.2.2. Blok Diagram Sistem Kontrol Logika Fuzzy

Berdasarkan Gambar 3.3, sistem kontrol Logika Fuzzy terdiri dari bagian input, proses, output, IoT process, dan aktivator. Sensor DHT22 berfungsi sebagai input untuk membaca suhu aktual di dalam kandang ayam DOC. Data suhu tersebut dikirimkan ke mikrokontroler ESP32 yang berfungsi sebagai Fuzzy Logic Controller untuk melakukan proses fuzzifikasi, evaluasi aturan, dan menentukan kondisi suhu serta keputusan kontrol. Berikut blok diagram sistem kontrol logika fuzzy ;



Gambar 3. 3 Blok Diagram Sistem Kontrol Logika Fuzzy

Berdasarkan gambar 3.3, Hasil keputusan dari ESP32 diteruskan ke AC *light* dimmer untuk mengendalikan intensitas *heating* lamp. Nilai suhu, kondisi fuzzy, dan status *heating lamp* juga ditampilkan melalui LCD 20x4. Selain itu, data hasil pembacaan dikirimkan melalui koneksi WiFi ke Google Sheets sehingga suhu dan kondisi sistem dapat dipantau secara *real-time*.

Sistem memperoleh catu daya dari *power supply* 5V DC yang terhubung ke ESP32. Seluruh proses berlangsung secara berulang, sehingga perubahan suhu yang dibaca oleh DHT22 dapat diproses kembali oleh Logika Fuzzy dan menghasilkan keputusan kontrol berikutnya. Dengan demikian, sistem dapat melakukan pengendalian dan monitoring suhu kandang ayam DOC secara otomatis.

3.3 Perancangan Perangkat Keras

3.3.1 Perancangan Mekanikal

Sistem ini memiliki rancangan mekanik yang terdiri dari sistem elektrikal dan desain kandang untuk ayam DOC. Gambar desain keseluruhan bisa dilihat di gambar 3.4.



Gambar 3. 4 Desain kandang



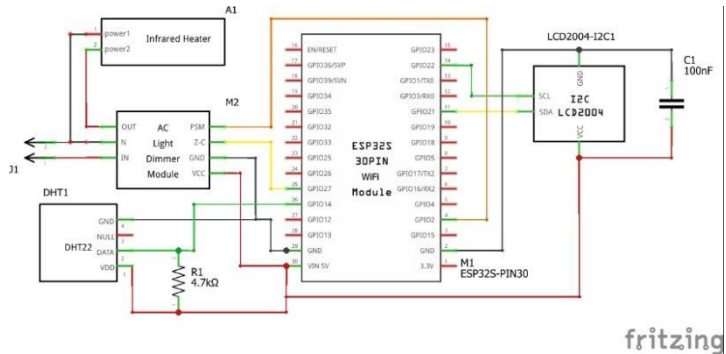
Gambar 3. 5 Tampak depan

Kandang ayam doc terbuat dari kayu jati belanda dengan ukuran total yaitu panjang 60cm, lebar 40cm dan tinggi 70cm. Kandang ini memiliki 1 pintu dan 5 ventilasi. Pada bagian kiri kandang terdapat 1 ventilasi, bagian kanan kandang 1 ventilasi, bagian depan 1 ventilasi dan bagian belakang 2 ventilasi.

Posisi alat diletakkan pada posisi yang seefisien mungkin untuk mengatur posisi sensor yang pas dikarenakan sifat sensor yaitu *single-wire* yang rentan terhadap hambatan sinyal.

3.3.2 Perancangan Elektrikal

Dalam Perancangan sistem optimalisasi suhu ini terdapat desain elektrikal yang merupakan hubungan antara satu komponen dengan komponen lainnya dengan tujuan membangun sistem yang diinginkan.



Gambar 3. 6 Desain *Elektrical*

Tabel 3. 1 Konfigurasi PIN

Tabel Konfigurasi PIN			
ESP 32	LCD 20x4 with I2C	DHT 22	AC Dimmer PWM
3.3V	-	VCC	-
5V	VCC	-	VCC
GND	GND	GND	GND
D14	-	Data	-
D2	-	-	-
D22	SCL	-	-
D21	SDA	-	-
D27	-	-	Z-C
D2	-	-	PSM

Seperti yang terlihat pada gambar 3.6 Mikrokontroler ESP 32 merupakan komponen utama yang terhubung dengan komponen *input* dan *output* serta sebagai penggerak untuk *output* setelah mendapatkan sinyal *input* yang didapat dari sensor. Sensor yang dipakai adalah sensor DHT 22 dan digunakan sebagai parameter untuk membaca nilai actual suhu.

Untuk menggerakkan *output* sistem, AC *light* dimmer 1 channel digunakan sebagai penggerak intensitas cahaya untuk komponen *output* yaitu *heating lamp*.

Pada Tabel 3.1 menunjukkan konfigurasi pin yang menghubungkan mikrokontroler ESP32 sebagai pusat kontrol dengan tiga komponen pendukung, yaitu LCD 20x4 dengan modul I2C, sensor DHT22, dan AC *light* dimmer 1 channel. Konfigurasi ini terbagi menjadi dua jenis koneksi, yaitu jalur catu daya dan jalur sinyal data/kontrol. Pada jalur catu daya, pin 3.3V ESP32 dihubungkan ke VCC sensor DHT22, sedangkan pin 5V ESP32 dihubungkan ke VCC LCD dan AC *light* dimmer. Ketiga komponen tersebut juga disatukan pada jalur ground (GND) ESP32 yang terhubung ke GND LCD, GND DHT22, dan GND pada AC *light* dimmer.

Pada jalur sinyal data dan kontrol, pin D14 ESP32 dihubungkan ke pin Data pada sensor DHT22. Kemudian untuk jalur sinyal ke AC *light* dimmer, pin D27 ESP32 dihubungkan ke port Z-C dan pin D2 dihubungkan ke port PSM. Sementara itu, komunikasi antara ESP32 dan LCD 20x4 menggunakan protokol I2C, di mana pin D22 berfungsi sebagai SCL (clock) dan pin D21 berfungsi sebagai SDA (data).

Untuk mengurangi *noise* pada sensor, pada jalur pin data DHT22 dan pin 3.3v dipasang resistor senilai 4.7k Ω . Komponen kapasitor 104 atau senilai 0,1 μ F juga dipasang diantara pin VCC LCD dan GND LCD dengan tujuan menghindari tampilan *error* pada LCD yang berasal dari *noise* tegangan.

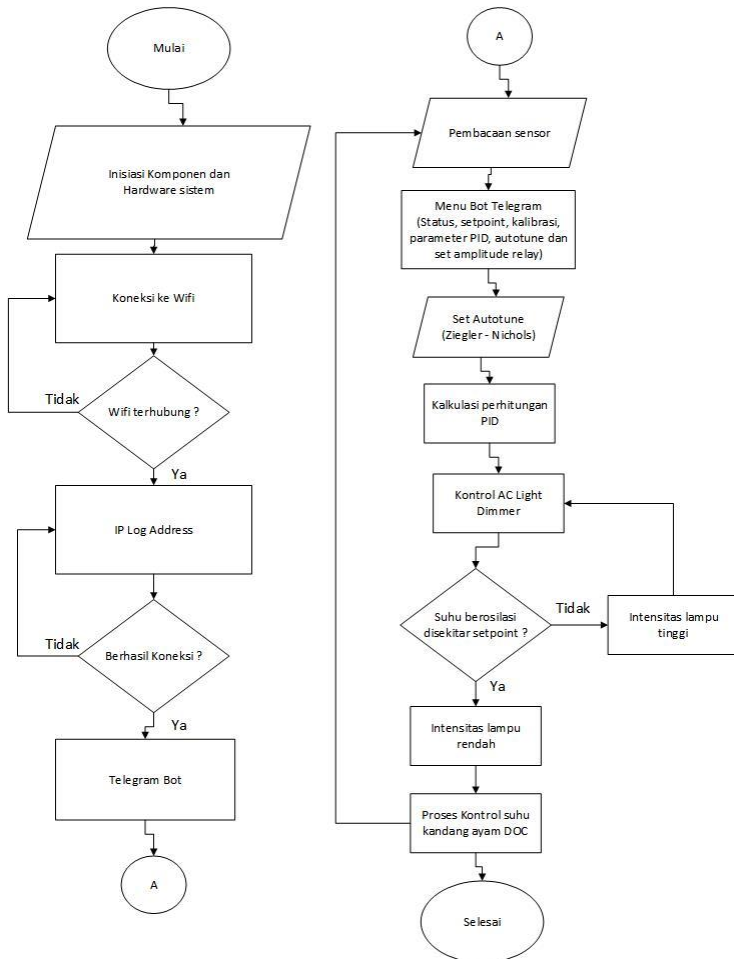
Dengan konfigurasi pin tersebut, ESP32 dapat menerima data suhu dari sensor DHT22, menampilkannya melalui LCD, serta mengendalikan aktuator *heating lamp* melalui AC *light* dimmer sesuai dengan hasil pengolahan sistem kontrol (PID/Fuzzy Logic) yang diterapkan.

3.4 Perancangan Perangkat Lunak

3.4.1 Perancangan Sistem Kontrol PID

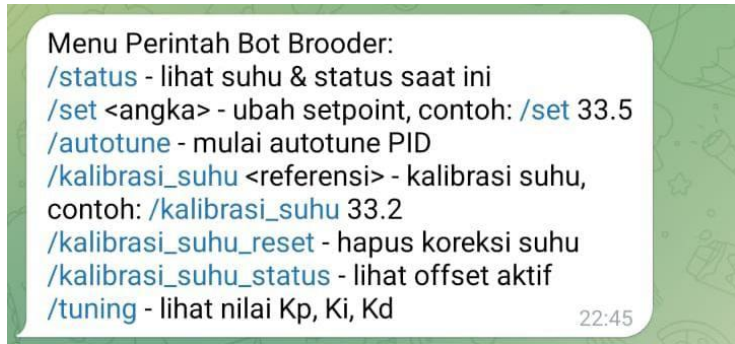
Pada perancangan perangkat lunak sistem ini terdapat logika kontrol otomatis yang menggunakan kontrol PID untuk mengatur *point value* yaitu suhu pada *setpoint* yang diinginkan. Proses ini dimulai dengan inisialisasi komponen perangkat keras dan perangkat lunak.

Sistem memeriksa koneksi Wi-Fi dan kesesuaian IP Address untuk memastikan mikrokontroler ESP32 dapat berkomunikasi melalui server yang terhubung yaitu telegram bot. *Flowchart* perangkat lunak dapat dilihat pada Gambar 3.7.



Gambar 3. 7 Flowchart Perancangan Sistem Kontrol PID

Selain menampilkan status kontrol pid, pada menu telegram bot terdapat beberapa segment penting yang dapat diubah yaitu kalibrasi suhu, *setpoint*, dan set *Autotune* (Ziegler - Nichols). Setiap perintah diawali dengan tanda garis miring (/). Desain menu telegram bot nya dapat dilihat di gambar 3.8.



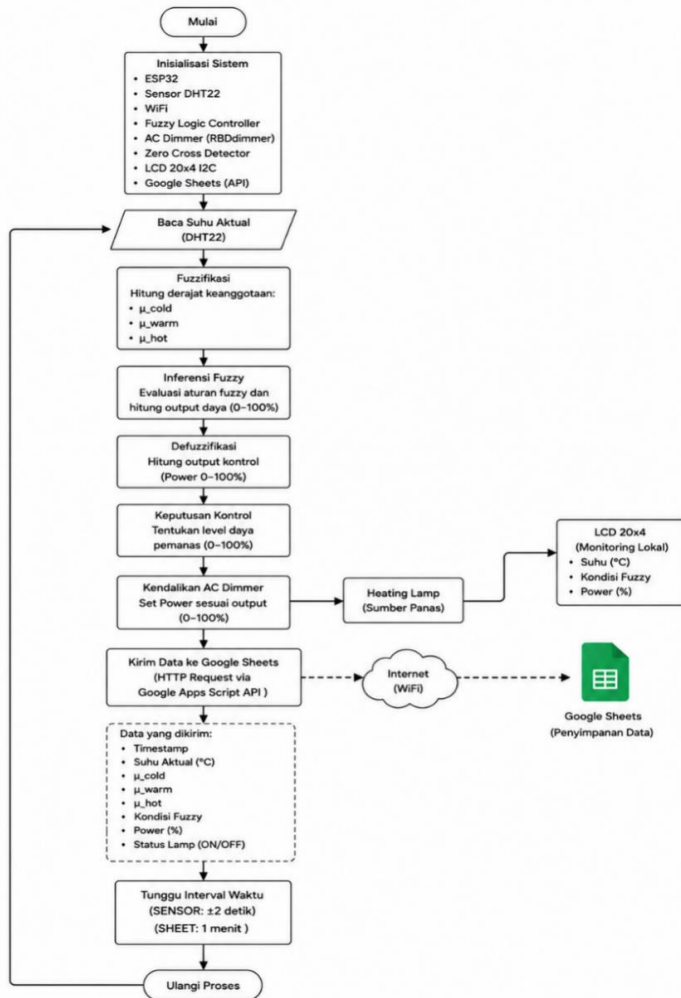
Gambar 3. 8 Menu telegram bot PID

3.4.2 Perancangan Sistem Logika Fuzzy

Perancangan perangkat lunak sistem Fuzzy Logic dilakukan untuk menghasilkan pengendalian daya pemanasan berdasarkan suhu aktual yang diperoleh dari sensor DHT22. ESP32 digunakan sebagai pusat pengolahan seluruh proses, mulai dari pembacaan sensor, *fuzzifikasi*, *inferensi*, *defuzzifikasi*, hingga pengiriman keluaran kepada AC *light* dimmer. Dengan rancangan tersebut, proses pengendalian dapat dilakukan secara berulang berdasarkan kondisi suhu terbaru yang diperoleh dari sensor.

Proses pengendalian dimulai dengan pembacaan suhu aktual dari sensor DHT22. Nilai suhu tersebut kemudian diproses melalui fungsi keanggotaan untuk menentukan derajat keanggotaan terhadap tiga kondisi linguistik, yaitu *COLD*, *WARM*, dan *HOT*. Hasil proses tersebut menghasilkan tiga nilai derajat keanggotaan, yaitu μ_{COLD} , μ_{WARM} , dan μ_{HOT} . Ketiga nilai tersebut digunakan sebagai dasar dalam menentukan aturan yang aktif pada proses inferensi.

Pada tahap inferensi, nilai derajat keanggotaan diproses berdasarkan aturan Fuzzy yang telah dirancang. Setiap aturan memberikan kontribusi terhadap keluaran berdasarkan kondisi suhu yang terdeteksi. Hasil inferensi kemudian digabungkan dan diteruskan ke tahap *defuzzifikasi* untuk memperoleh satu nilai keluaran numerik.



Gambar 3. 9 Flowchart Perancangan Perangkat Lunak logika fuzzy

Nilai hasil defuzzifikasi digunakan sebagai dasar untuk menentukan tingkat daya yang diberikan kepada AC light dimmer.

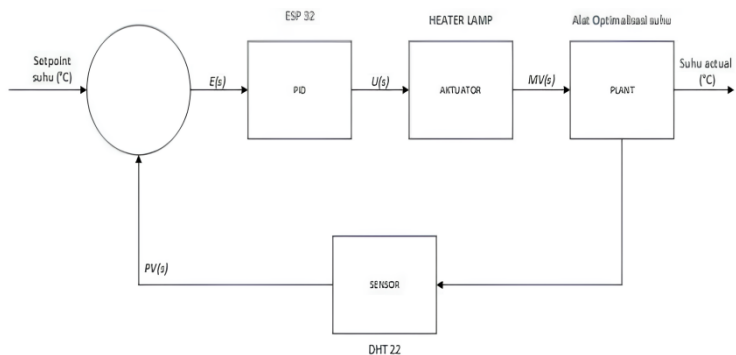
Dengan mekanisme tersebut, daya *heating lamp* dapat disesuaikan terhadap kondisi suhu aktual. Ketika suhu masih berada jauh di bawah *setpoint*, sistem dapat memberikan daya pemanasan yang lebih besar. Ketika suhu mendekati *setpoint*, daya dapat dikurangi sehingga kenaikan suhu dapat dikontrolkan secara lebih bertahap.

Selain digunakan untuk pengendalian, hasil proses Fuzzy Logic juga dicatat sebagai data pengujian. Data tersebut meliputi waktu pengambilan data, suhu aktual, μ_{COLD} , μ_{WARM} , μ_{HOT} , kondisi Fuzzy Logic, dan nilai keluaran daya. Data tersebut digunakan untuk menganalisis hubungan antara perubahan suhu, derajat keanggotaan, dan perubahan daya selama proses pengujian.

3.5 Perancangan Metode Penelitian

3.5.1 Perancangan Kontrol PID

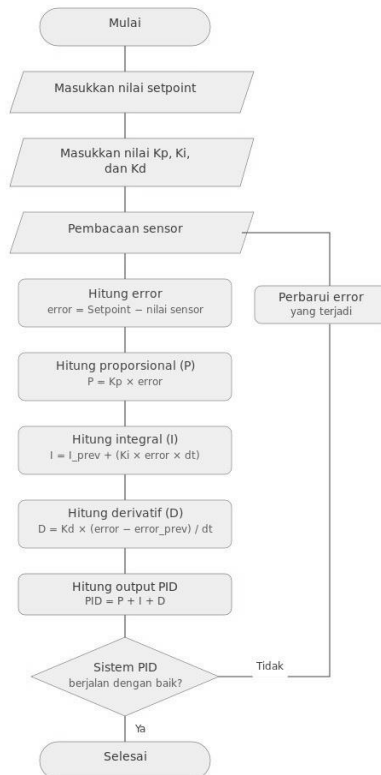
Sistem kontrol PID digunakan untuk mengontrol suhu elemen heater sesuai dengan *setpoint* yang telah ditentukan. Perancangan dilakukan untuk menentukan nilai K_p , K_i , K_d dari plant yang terkontrol. Dalam perancangan sistem kontrol PID, yang perlu dilakukan ialah mengatur nilai parameter P, I dan D agar sistem dapat menanggapi masukan sinyal dan dapat mengeluarkan *output* yang diinginkan. Rancangan desain alir kontrol PID dapat dilihat pada 3.10 .



Gambar 3. 10 Diagram alir PID

Dalam rancangan ini, sistem kontrol PID terbagi dalam beberapa tahapan. Yang pertama, ialah menetapkan nilai *setpoint* agar mendapatkan nilai luaran yang diinginkan setelah mendapatkan sinyal masukan tertentu. Sinyal yang diteruskan ke kontrol PID berupa suhu aktual yang didapat dari sensor DHT 22. kemudian, komponen PID akan menganalisa atau menjumlahkan untuk menghasilkan nilai dari proporsional, integral dan derivatif setelah mendapat masukan sinyal tertentu.

Proses tersebut akan bekerja secara *close loop* atau mengalami pengulangan terus menerus hingga suhu yang diinginkan telah stabil. *Flowchart* dari kontrol PID dapat dilihat pada gambar 3.11.

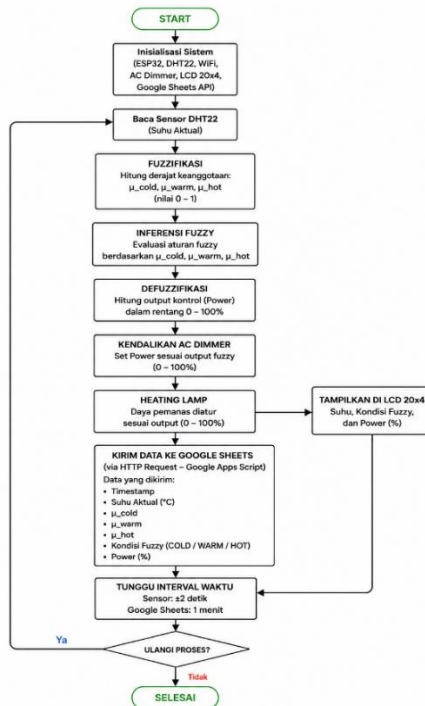


Gambar 3. 11 Flowchart PID

3.5.2 Perancangan Logika Fuzzy

Metode *Fuzzy Logic* pada penelitian ini digunakan untuk menentukan tingkat daya pemanasan berdasarkan kondisi suhu aktual di dalam kandang DOC. Sistem menggunakan suhu aktual sebagai masukan utama yang diperoleh dari sensor DHT22. Nilai suhu tersebut tidak langsung digunakan sebagai keputusan *ON* atau *OFF*, tetapi terlebih dahulu dipetakan ke dalam fungsi keanggotaan sehingga diperoleh derajat keanggotaan pada kondisi *COLD*, *WARM*, dan *HOT*.

Hasil fuzzifikasi berupa μ_{COLD} , μ_{WARM} , dan μ_{HOT} digunakan pada proses inferensi berdasarkan basis aturan Fuzzy. Setiap aturan menentukan kecenderungan keluaran berdasarkan kondisi suhu yang terdeteksi. Hasil dari aturan yang aktif kemudian diproses melalui defuzzifikasi untuk menghasilkan satu nilai numerik sebagai keluaran sistem.



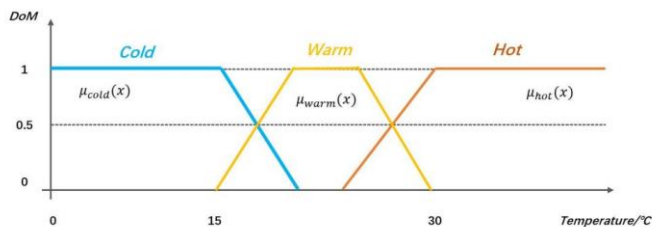
Gambar 3. 12 Diagram Alir Logika Fuzzy

Keluaran tersebut digunakan untuk mengatur daya *AC light dimmer* yang terhubung dengan *heating lamp*. Dengan demikian, sistem dapat memberikan daya pemanasan yang berbeda sesuai dengan kondisi suhu aktual. Penggunaan *AC light dimmer* memungkinkan keluaran daya diatur secara bertahap, sehingga pengendalian tidak terbatas pada dua kondisi seperti sistem *ON/OFF*.

Pada penelitian ini digunakan *setpoint* sebesar 35°C. Nilai tersebut digunakan sebagai kondisi target yang harus dipertahankan oleh sistem selama proses pengujian. Pemilihan *setpoint* yang sama pada pengujian PID dan Fuzzy Logic dilakukan agar kedua metode dapat dibandingkan menggunakan kondisi target yang setara.

3.5.2.1 Fungsi Keanggotaan Input

Fungsi keanggotaan input merupakan bagian awal dalam proses pengolahan *Fuzzy Logic* yang digunakan untuk mengubah nilai suhu aktual hasil pembacaan sensor DHT22 menjadi nilai derajat keanggotaan. Pada penelitian ini, suhu aktual digunakan sebagai satu-satunya variabel input karena kondisi suhu di dalam kandang *DOC* menjadi dasar utama dalam menentukan besarnya daya pemanasan yang diberikan oleh *heating lamp*. Nilai suhu yang dibaca oleh DHT22 selanjutnya dipetakan ke dalam tiga kondisi linguistik, yaitu *COLD*, *WARM*, dan *HOT*.



Gambar 3.13 Fungsi Keanggotaan Input

Ketiga kondisi tersebut digunakan untuk menggambarkan tingkat kesesuaian suhu aktual terhadap kondisi yang telah ditentukan pada sistem. Kondisi *COLD* menunjukkan keadaan ketika suhu aktual masih berada pada daerah yang membutuhkan pemanasan lebih besar. Kondisi *WARM* menunjukkan keadaan ketika suhu telah mendekati daerah suhu target sehingga kebutuhan pemanasan dapat dikurangi secara bertahap.

Sementara itu, kondisi *HOT* menunjukkan keadaan ketika suhu telah berada pada daerah suhu tinggi sehingga sistem perlu mengurangi keluaran daya pemanasan. Pembagian kondisi tersebut memungkinkan sistem memberikan respons yang lebih bertahap dibandingkan pengendalian yang hanya menggunakan keputusan *ON* dan *OFF*.

Berdasarkan Gambar 3.13, sumbu horizontal menunjukkan nilai suhu aktual dalam satuan °C, sedangkan sumbu vertikal menunjukkan derajat keanggotaan dengan rentang nilai 0 sampai 1. Kurva pada grafik menggambarkan hubungan antara suhu aktual dengan tiga kondisi linguistik, yaitu *COLD*, *WARM*, dan *HOT*. Semakin mendekati nilai 1, semakin tinggi tingkat kesesuaian suhu aktual terhadap kondisi linguistik tersebut. Sebaliknya, nilai yang mendekati 0 menunjukkan bahwa suhu tersebut memiliki tingkat kesesuaian yang rendah terhadap kondisi yang bersangkutan.

Pada kondisi *COLD*, derajat keanggotaan digunakan untuk menunjukkan seberapa kuat suhu aktual termasuk ke dalam kondisi suhu rendah. Apabila suhu berada pada daerah *COLD*, sistem memberikan kecenderungan keluaran daya yang lebih besar karena suhu masih membutuhkan peningkatan pemanasan. Ketika suhu mulai meningkat dan memasuki daerah peralihan menuju *WARM*, nilai keanggotaan *COLD* akan berkurang. Perubahan tersebut tidak terjadi secara langsung, tetapi mengikuti bentuk fungsi keanggotaan yang telah ditentukan sehingga sistem dapat memberikan respons secara bertahap.

Kondisi *WARM* digunakan untuk menggambarkan daerah suhu yang berada di sekitar kondisi target. Pada daerah ini, suhu dapat memiliki derajat keanggotaan *WARM* yang tinggi, tetapi pada daerah peralihan tertentu dapat pula memiliki derajat keanggotaan terhadap *COLD* atau *HOT*. Hal tersebut merupakan salah satu karakteristik utama *Fuzzy Logic*, karena suatu nilai input tidak harus secara mutlak hanya termasuk dalam satu kondisi. Sebagai contoh, ketika suhu berada pada daerah peralihan *COLD* dan *WARM*, kedua kondisi tersebut dapat memiliki nilai derajat keanggotaan secara bersamaan. Nilai tersebut kemudian digunakan dalam proses inferensi untuk menentukan keluaran pengendalian.

Kondisi *HOT* digunakan untuk menunjukkan bahwa suhu aktual telah berada pada daerah suhu tinggi. Ketika suhu semakin mendekati atau memasuki daerah *HOT*, derajat keanggotaan *HOT* akan meningkat, sedangkan derajat keanggotaan *COLD* dan *WARM* akan menyesuaikan berdasarkan posisi suhu terhadap fungsi keanggotaan masing-masing.

Kondisi tersebut menyebabkan sistem memberikan kecenderungan keluaran daya yang lebih rendah sehingga pemanasan tidak terus diberikan dengan tingkat daya yang tinggi ketika suhu telah mencapai kondisi yang lebih panas.

Hasil proses fungsi keanggotaan ini menghasilkan tiga nilai yang digunakan pada tahap berikutnya, yaitu μ_COLD , μ_WARM , dan μ_HOT . Ketiga nilai tersebut menjadi dasar dalam proses inferensi *Fuzzy Logic*. Dengan demikian, pembacaan suhu dari DHT22 tidak langsung diterjemahkan menjadi nilai daya *AC light dimmer*, tetapi terlebih dahulu melalui proses fuzzifikasi untuk mengetahui kondisi suhu berdasarkan derajat keanggotaannya. Selanjutnya, hasil tersebut diproses menggunakan basis aturan Fuzzy dan defuzzifikasi untuk memperoleh nilai keluaran yang digunakan dalam pengaturan daya *heating lamp*.

Fungsi keanggotaan yang digunakan pada penelitian ini juga harus memiliki parameter yang sama dengan implementasi program pada ESP32. Hal tersebut penting agar grafik fungsi keanggotaan yang ditampilkan pada Bab III benar-benar menggambarkan algoritma yang digunakan pada sistem aktual. Dengan kesesuaian tersebut, nilai μ_COLD , μ_WARM , dan μ_HOT yang diperoleh selama pengujian pada Bab IV dapat ditelusuri kembali berdasarkan fungsi keanggotaan yang telah dirancang.

3.5.2.2 Basis Aturan Fuzzy

Formulasi aturan Fuzzy digunakan untuk menerjemahkan hubungan antara kondisi suhu dan keluaran pengendalian ke dalam aturan *IF-THEN*. Nilai derajat keanggotaan yang diperoleh dari tahap fuzzifikasi digunakan untuk menentukan tingkat aktivasi dari setiap aturan. Dengan demikian, perubahan suhu dapat menghasilkan perubahan tingkat aktivasi aturan dan selanjutnya memengaruhi keluaran sistem.

Apabila suhu memiliki derajat keanggotaan yang tinggi pada kondisi *COLD*, aturan yang berhubungan dengan kebutuhan pemanasan tinggi akan memiliki kontribusi yang lebih besar terhadap keluaran. Sebaliknya, apabila suhu memiliki derajat keanggotaan yang lebih tinggi pada kondisi *HOT*, aturan yang menghasilkan daya pemanasan lebih rendah akan lebih dominan.

Beberapa aturan dapat aktif secara bersamaan apabila suhu berada pada daerah peralihan fungsi keanggotaan. Hasil dari aturan-aturan tersebut kemudian digabungkan pada proses inferensi. Setelah proses inferensi selesai, keluaran yang dihasilkan diproses melalui defuzzifikasi sehingga diperoleh satu nilai numerik yang dapat digunakan untuk mengatur daya *AC light dimmer*.

Tabel 3. 2 Aturan Fuzzy Berdasarkan Kondisi Suhu

No	Rentang Suhu	μ_COLD	μ_WARM	μ_HOT	Perbandingan	Tingkat Daya Lampu
1	≤ 30	1.00	0.0	0.00	<i>COLD</i>	Tinggi
2	31	0.80	0.20	0.00	<i>COLD</i>	Tinggi
3	32	0.60	0.40	0.00	<i>COLD</i>	Tinggi
4	33	0.40	0.60	0.00	<i>WARM</i>	Sedang
5	34	0.20	0.80	0.00	<i>WARM</i>	Sedang
6	35	0.00	1.00	0.00	<i>WARM</i>	Sedang Rendah
7	36	0.0	0.80	0.20	<i>WARM</i>	Sedang Rendah
8	37	0.00	0.60	0.40	<i>WARM</i>	Sedang Rendah
9	38	0.00	0.40	0.60	<i>HOT</i>	Rendah
10	39	0.00	0.20	0.80	<i>HOT</i>	Rendah
11	≥ 40	0.00	0.00	1.00	<i>HOT</i>	Sangat Rendah

Dari tabel 3.2 dapat dilihat bahwa pada zona transisi *WARM* (30°C - 35°C), meskipun kedua derajat keanggotaan *COLD* dan *HOT* dapat bernilai rendah atau bahkan 0, keputusan tetap dapat diambil berdasarkan perbandingan relatif antara keduanya. Hal ini memungkinkan sistem untuk tetap responsif terhadap perubahan suhu yang kecil sekalipun.

3.5.2.3 Metode Inferensi Fuzzy Berbasis Pembobotan Derajat Keanggotaan

Metode inferensi Fuzzy pada penelitian ini digunakan untuk menentukan keluaran daya pemanasan berdasarkan nilai derajat keanggotaan yang diperoleh dari suhu aktual.

Sistem menggunakan suhu sebagai variabel *input* yang dibaca oleh sensor DHT22, kemudian nilai tersebut diproses melalui fungsi keanggotaan *COLD*, *WARM*, dan *HOT*. Hasil proses tersebut berupa nilai μ_{COLD} , μ_{WARM} , dan μ_{HOT} yang menggambarkan tingkat keanggotaan suhu aktual terhadap masing-masing kondisi.

Berbeda dengan metode inferensi Mamdani yang menggunakan keluaran berupa himpunan Fuzzy dan proses implikasi serta agregasi, implementasi pada penelitian ini menentukan nilai keluaran secara langsung berdasarkan pembobotan derajat keanggotaan. Pembobotan dilakukan pada daerah suhu yang berada di antara kondisi *COLD* dan *WARM*. Pada daerah tersebut, nilai μ_{COLD} dan μ_{WARM} digunakan untuk menentukan besarnya daya yang diberikan kepada AC light dimmer. Persamaan yang digunakan untuk menentukan keluaran daya adalah sebagai berikut ;

$$P = \mu_{COLD} / (\mu_{COLD} + \mu_{WARM}) \times 100\%$$

Dengan P merupakan persentase keluaran daya, μ_{COLD} merupakan derajat keanggotaan terhadap kondisi *COLD*, dan μ_{WARM} merupakan derajat keanggotaan terhadap kondisi *WARM*. Hasil perhitungan tersebut kemudian digunakan sebagai nilai pengendalian daya AC light dimmer.

Pada suhu yang berada pada atau di bawah 30°C, sistem menetapkan kondisi *COLD* dan memberikan keluaran daya sebesar 100%. Kondisi tersebut menunjukkan bahwa suhu masih berada pada daerah rendah sehingga sistem memberikan daya pemanasan maksimum. Ketika suhu berada di antara 30°C dan 35°C, sistem berada pada daerah peralihan *COLD* dan *WARM*. Pada daerah tersebut, kedua derajat keanggotaan digunakan dalam pembobotan sehingga keluaran daya dapat berubah secara bertahap mengikuti perubahan suhu.

Sebagai contoh, pada suhu aktual 33,40°C diperoleh nilai μ_{COLD} sebesar 0,320 dan μ_{WARM} sebesar 0,680. Dengan memasukkan kedua nilai tersebut ke dalam persamaan, diperoleh:

$$P = 0,320 / (0,320 + 0,680) \times 100\% = 32\%$$

Dengan demikian, pada suhu 33,40°C sistem menghasilkan keluaran daya sebesar 32%. Nilai tersebut kemudian digunakan untuk mengatur AC light dimmer, sehingga daya yang diberikan kepada *heating lamp* dapat disesuaikan dengan kondisi suhu aktual.

Ketika suhu mencapai 35°C atau lebih, program menetapkan keluaran daya sebesar 0%. Kondisi tersebut menunjukkan bahwa suhu telah mencapai atau melewati *setpoint* sehingga kebutuhan pemanasan dihentikan. Meskipun nilai μ_{HOT} tetap dihitung oleh sistem, nilai tersebut tidak digunakan secara langsung dalam persamaan pembobotan daya pada implementasi program. Nilai μ_{HOT} tetap dapat dicatat sebagai bagian dari data pengujian untuk menunjukkan kondisi keanggotaan suhu pada saat sistem beroperasi.

Dengan demikian, mekanisme inferensi yang diterapkan pada penelitian ini menghasilkan keluaran daya yang berubah secara bertahap pada daerah 30°C sampai 35°C. Semakin tinggi suhu mendekati 35°C, nilai μ_{COLD} semakin kecil dan keluaran daya semakin rendah. Sebaliknya, semakin rendah suhu mendekati 30°C, nilai μ_{COLD} semakin besar sehingga keluaran daya semakin tinggi. Mekanisme tersebut memungkinkan *heating lamp* dikontrolkan menggunakan perubahan tingkat daya melalui AC light dimmer, bukan hanya menggunakan keputusan *ON* atau *OFF*.

Secara keseluruhan, proses pengendalian terdiri atas pembacaan suhu aktual, fuzzifikasi, pembentukan nilai μ_{COLD} , μ_{WARM} , dan μ_{HOT} , pembobotan derajat keanggotaan, penentuan keluaran daya, serta pengaturan AC *light dimmer*. Oleh karena itu, metode inferensi pada sistem ini dapat dikategorikan sebagai inferensi Fuzzy berbasis pembobotan derajat keanggotaan (*weighted average*), karena nilai keluaran diperoleh secara langsung dari perbandingan derajat keanggotaan yang aktif dan kemudian dinyatakan dalam bentuk persentase daya.

3.6 Alat Dan Bahan

Dalam pelaksanaan penelitian ini, digunakan sejumlah alat dan bahan yang mendukung perancangan dan implementasi sistem kontrol suhu berbasis ESP32 pada kandang ayam *Day Old Chick* (DOC). Alat dan bahan tersebut meliputi komponen elektronika, perangkat keras pendukung, serta material fisik yang diperlukan dalam proses perancangan prototipe sistem. Keseluruhan alat dan bahan yang digunakan bersumber dari dana pribadi peneliti. Adapun rincian alat dan bahan yang digunakan dalam penelitian ini disajikan pada Tabel 3.3.

Tabel 3. 3 Alat dan bahan

No	Alat/bahan	Harga Satuan (Rp.)	Jumlah	Total (Rp.)	Keterangan
1	ESP32 Dev-kit V1	Rp. 82.000,00	1 pcs	Rp. 82.000,00	Dana Pribadi
2	DHT22 AM2302 Sensor Suhu & Kelembaban	Rp. 22.900,00	1 pcs	Rp. 22.900,00	Dana Pribadi
3	LCD 2004 <i>Blue Backlight</i> + I2C <i>Serial Interface Module</i>	Rp. 59.900,00	1 pcs	Rp. 59.900,00	Dana Pribadi
4	I/O <i>Expansion</i> ESP32 Doit V1 30P	Rp. 27.900,00	1 pcs	Rp. 27.900,00	Dana Pribadi
5	Resistor 4.7k Ohm	Rp. 2.490,00	1 pcs	Rp. 2.490,00	Dana Pribadi
6	Kapasitor keramik 104 (100 nF)	Rp. 250,00	5 pcs	Rp. 1.250,00	Dana Pribadi
7	Kabel <i>jumper Female to Female</i>	Rp. 500,00	7 pcs	Rp. 3.500,00	Dana Pribadi
8	Kabel <i>jumper male to Female</i>	Rp. 500,00	3 pcs	Rp. 1.500,00	Dana Pribadi
9	Lampu UVA <i>Intense Basking Spot</i>	Rp. 55.000,00	1 pcs	Rp. 55.000,00	Dana Pribadi
10	Termometer ruangan kayu merk GEA	Rp. 19.950,00	1 pcs	Rp. 19.950,00	Dana Pribadi

No	Alat/bahan	Harga Satuan (Rp.)	Jumlah	Total (Rp.)	Keterangan
11	Ayam DOC Kampung super	Rp. 13.000,00	8 ekor	Rp. 104.000,00	Dana Pribadi
12	<i>Black Box</i> X5	Rp 15.000,00	1 pcs	Rp 15.000,00	Dana Pribadi
13	<i>Black box</i> X3	Rp. 9.000,00	1 pcs	Rp. 9.000,00	Dana Pribadi
14	Fitting lampu socket E27	Rp. 2.400,00	1 pcs	Rp. 2.400,00	Dana Pribadi
15	Kandang ayam DOC	Rp. 250.000,00	1 pcs	Rp. 250.000,00	Dana Pribadi
16	Triplek uk. 3mm	Rp. 78.700,00	1 keping	Rp. 78.700,00	Dana Pribadi
17	Lasdop	Rp. 500,00	4 pcs	Rp. 2.000,00	Dana Pribadi
18	kayu reng jati belanda	Rp. 15.000,00	1 Batang	Rp. 15.000,00	Dana Pribadi
19	Kabel instalasi	Rp. 10.000,00	1 Meter	Rp. 10.000,00	Dana Pribadi
20	<i>Glue gun</i>	Rp. 35.000,00	1 pcs	Rp. 35.000,00	Dana Pribadi
21	Lem tembak	Rp. 2.000,00	2 pcs	Rp. 2.000,00	Dana Pribadi

No	Alat/bahan	Harga Satuan (Rp.)	Jumlah	Total (Rp.)	Keterangan
22	Kabel power AC	Rp. 7.000,00	1 pcs	Rp. 7.000,00	Dana Pribadi
23	T dus 3 cabang	Rp. 5.000,00	1 pcs	Rp. 5.000,00	Dana Pribadi
24	Terminal <i>Block</i> krustin	Rp. 17.200,00	1 Batang	Rp. 17.200,00	Dana Pribadi
25	Modul Adaptor Samsung Bekas	Rp. 10.000,00	1 pcs	Rp. 10.000,00	Dana Pribadi
26	AC <i>Light</i> Dimmer Robotdyn Module 1 Channel 220V 110V logic 5V 3.3V	Rp.168.850,00	1 pcs	Rp.168.850,00	Dana pribadi
Total harga Keseluruhan				Rp. 1.007.540,00	

3.7 Jadwal Pelaksanaan

Pelaksanaan penelitian ini direalisasikan selama 2 sesi yang terjadwal untuk seminar proposal dan untuk Tugas akhir. Pada masing – masing sesi dijadwalkan berlangsung selama 4 bulan.

Pada sesi ke-1 berfokus pada pengerjaan proposal yang dilaksanakan di bulan agustus hingga November pada tahun 2023. Jadwal pelaksanaan dimulai dari studi literature dan identifikasi masalah yang dilakukan pada minggu ke-1 - 3 di bulan agustus. Kemudian penyusunan Bab 1 Pendahuluan (proposal) dikerjakan pada minggu ke-4 Agustus – minggu ke-2 di bulan September. Untuk penyusunan Bab 2 Tinjauan Pustaka (proposal) dimulai pada minggu ke-3 September - minggu ke-1 Oktober. Perancangan mekanikal dan elektrikal sistem dilaksanakan pada minggu ke-2 – 4 di bulan oktober. Untuk Penyusunan Bab 3 dimulai pada minggu ke-1 - 3 di bulan November. kemudian dilanjutkan dengan revisi proposal serta persiapan seminar proposal yang dijadwalkan pada minggu ke-4 di bulan November.

Pada sesi ke-2 berfokus pada implementasi dari proposal yang telah dibuat. Pada bulan maret 2026 dari minggu ke-1 - 4 dilakukan proses instalasi alat sesuai dengan konsep yang telah dibangun di proposal. Untuk pengujian alat dilaksanakan pada bulan April di minggu ke-1 – 3. Untuk proses pengambilan dan analisa data dikerjakan di minggu ke-4 bulan April – minggu ke-3 bulan Mei. Adapun untuk penyusunan bab 4 dikerjakan di minggu ke-4 bulan Mei – minggu ke-2 bulan Juni. Untuk penyusunan bab 5 dan daftar pustaka dikerjakan di minggu ke-3 bulan Juni. Minggu ke-4 di bulan juni merupakan periode penyelesaian atau revisi pada buku TA sebelum melaksanakan sidang tugas akhir. Adapun ringkasan penelitiannya dapat dilihat pada tabel 3.4.

Tabel 3. 4 Jadwal penelitian

Tahun 2023																	
Aktivitas	Indicator	Agustus				September				Oktober				November			
Minggu		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Studi Literature & identifikasi masalah	Penentuan																
Penyusunan Bab 1	Penulisan bab 1 pendahuluan (proposal)																
Penyusunan Bab 2	Penulisan bab 2 tinjauan pustaka (proposal)																
Perancangan Mekanikal	Perancangan Mekanikal yang akan diteliti																
Perancangan hardware	Perancangan Hardware (Elektrical alat)																
Penyusunan Bab 3	Penulisan bab 3 Metodologi penelitian (proposal)																

Tahun 2023																	
Aktivitas	Indicator	Agustus				September				Oktober				November			
Minggu		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Persiapan Seminar Proposal	Revisi Proposal																

Tahun 2026																	
Aktivitas	Indicator	Maret				April				Mei				Juni			
Minggu		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Instalasi alat	Memastikan alat sudah Sesuai dengan konsep yang diinginkan																
Pengujian alat	Uji coba dan <i>Monitoring</i> kerja alat																
Pengambilan data	Analisa data																
Penyusunan Bab 4	Penulisan Bab 4 berupa hasil dan pembahasan (TA)																
Penyusunan Bab 5	Penulisan kesimpulan dan saran dari penelitian yang dikerjakan																
Penyusunan daftar pustaka	Penulisan daftar pustaka																
Persiapan Sidang Tugas Akhir	Revisi Buku TA																

Bab 4. Hasil Dan Pembahasan

4.1 Hasil Pengujian

4.1.1 Pengujian Sensor

Pada pengujian sensor yaitu DHT22, dilakukan perbandingan pembacaan data suhu antara sensor DHT22 dan termometer *wooden* GEA. Pengujian ini bertujuan untuk menguji keandalan dan keakuratan sensor DHT22 dalam pengukuran suhu, serta melakukan perbandingan dengan hasil yang diperoleh dari termometer *wooden* GEA sebagai alat ukur pembanding. Metode pengujian dilakukan dengan cara mengkalibrasi nilai *output* suhu dari DHT 22 dan data pada termometer pembanding, kemudian selisih yang didapat (-1.60) dimasukkan ke program codingan. Pengujian sensor suhu DHT22 dapat dilihat dalam Tabel 4.1.

Tabel 4. 1 Pengujian sensor

No	Suhu (°C)		Selisih (°C)	Persentase (%)
	DHT-22	Termometer <i>wooden</i> GEA		
1	32,0	31,9	0,1	0,31 %
2	32,5	32,3	0,3	0,93 %
3	33,2	33,5	0,3	0,90 %
4	34,0	34,5	0,5	1,45 %
5	35,0	35,4	0,4	1,13 %
6	36,2	36,5	0,3	0,82 %
7	37,0	37,2	0,2	0,54 %
8	39,7	39,9	0,2	0,50 %
9	42,0	42,1	0,1	0,24 %
10	43,8	43,0	0,8	1,86 %

Hasil pengujian menunjukkan bahwa suhu terendah yang diperoleh oleh sensor DHT22 adalah 32°C, sementara suhu tertinggi mencapai 43,8°C.

Perbedaan suhu yang terukur antara DHT22 dan termometer digital berada dalam rentang 0,1°C hingga 0,8°C, menunjukkan fluktuasi yang masih wajar tetapi tetap stabil dalam hasil pengukuran. Dan untuk Persentase *error* tertinggi adalah 1,86% (No. 8) dan terendah 0,24% (No. 7), yang menunjukkan sensor DHT-22 cukup akurat dibandingkan termometer *wooden* GEA. Berdasarkan pengujian ini dapat diambil kesimpulan bahwa sensor DHT22 menunjukkan kinerja yang sangat baik, dengan perbedaan yang cukup kecil dalam pengukuran suhu.

4.1.2 Pengujian Parameter PID

Pengujian dilakukan untuk menentukan parameter kontroler PID secara otomatis menggunakan metode osilasi tertutup melalui rumus Ziegler-Nichols dengan dua parameter eksperimental, yaitu *Ultimate Gain* (K_u) dan *Ultimate Period* (T_u), yang diperoleh langsung dari respon dinamis sistem, yang kemudian dikalkulasi ke bentuk parameter K_p , K_i dan K_d sesuai perintah pada codingan.

Pengujian sistem *autotune* dilakukan dengan mengaktifkan mode hanya proporsional ($K_i = 0$, $K_d = 0$), kemudian nilai K_p dinaikkan secara bertahap hingga *output* sistem mengalami osilasi yang stabil, tidak teredam maupun tidak meledak. Nilai K_p pada kondisi tersebut dicatat sebagai K_u , sedangkan periode satu siklus osilasi penuh diukur sebagai T_u . Pengujian dilakukan pada tiga titik *setpoint* berbeda, yaitu 32 °C, 33.5 °C, dan 35 °C, guna mengevaluasi konsistensi dan kinerja metode pada rentang operasi yang bervariasi.

4.1.2.1 Pengujian Autotune Ziegler-Nichols

Pengujian *autotune* pada sistem kontrol suhu *brooder* dilakukan menggunakan metode *relay feedback* Ziegler-Nichols. Metode ini bekerja dengan cara mengalihkan *output* aktuator (AC dimmer) secara bergantian antara dua level intensitas, yaitu intensitas tinggi sebesar 60% dan intensitas rendah sebesar 15%, setiap kali suhu terukur melewati nilai *setpoint* acuan. Pergantian intensitas ini sengaja tidak dibuat penuh (*on/off*), melainkan tetap berupa dua level intensitas cahaya/panas yang berbeda, sesuai dengan karakteristik aktuator AC dimmer yang bersifat kontinu.

Selama proses ini berlangsung, sistem mencatat waktu setiap kali terjadi pergantian intensitas serta nilai puncak (maksimum) dan lembah (minimum) suhu yang tercapai.

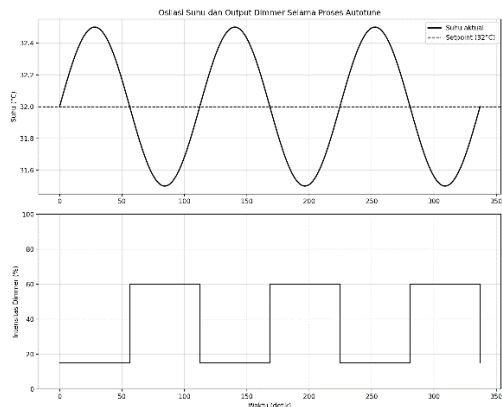
Data tersebut kemudian digunakan untuk menghitung dua parameter utama, yaitu periode osilasi *ultimate* (Pu) dan gain kritis (Ku), yang menjadi dasar perhitungan parameter kontrol PID (Kp, Ki, Kd) menggunakan rumus Ziegler-Nichols klasik. Hasil *autotune* yang diperoleh dari pengujian sistem ditunjukkan pada Tabel 4.2 berikut.

Tabel 4. 2 Hasil perhitungan *autotune* Ziegler-Nichols

Parameter	Nilai
Gain kritis (Ku)	57,296
Periode <i>ultimate</i> (Pu)	112,328 detik
Kp	34,377
Ki	0,612
Kd	482,694

4.1.2.2 Grafik Osilasi Suhu dan *Output Dimmer*

Berdasarkan nilai Ku dan Pu yang diperoleh, dapat direkonstruksi bentuk osilasi suhu dan *output dimmer* selama proses *autotune* berlangsung, sebagaimana ditunjukkan pada Gambar 4.1.



Gambar 4. 1 Grafik osilasi suhu dan output dimmer selama proses *autotune*

Grafik pada Gambar 4.1 memperlihatkan dua besaran yang saling berkaitan. Pada grafik bagian atas, suhu aktual bergerak naik-turun secara periodik di sekitar garis putus-putus yang merepresentasikan nilai *setpoint*. Pada grafik bagian bawah, *output* dimmer berbentuk gelombang kotak yang berpindah antara intensitas rendah (15%) dan intensitas tinggi (60%) setiap kali suhu melewati *setpoint*, dengan waktu pergantian tiap setengah periode, yaitu sekitar 56,16 detik.

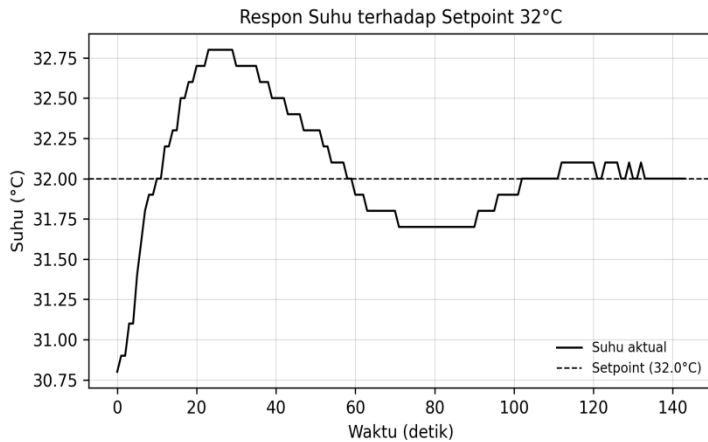
Amplitudo osilasi suhu (b) dihitung menggunakan hubungan *describing function relay feedback*:

$$Ku = 4a / (\pi \times b)$$

Dengan a adalah setengah selisih antara intensitas tinggi dan intensitas rendah ($a = (60\% - 15\%) / 2 = 22,5\%$). Dari nilai $Ku = 57,296$ yang diperoleh, didapatkan amplitudo osilasi suhu sebesar $b \approx 0,5^\circ\text{C}$. Dengan demikian, selama proses *autotune* suhu *brooder* berosilasi pada rentang $31,5^\circ\text{C}$ hingga $32,5^\circ\text{C}$ di sekitar nilai *setpoint* 32°C , dengan satu siklus osilasi penuh membutuhkan waktu $P_u = 112,328$ detik. Periode osilasi $P_u = 112,328$ detik yang relatif panjang mengindikasikan bahwa sistem *brooder* memiliki reoverspons termal yang cenderung lambat, yaitu waktu yang dibutuhkan suhu untuk merespons perubahan intensitas pemanas cukup besar. Hal ini konsisten dengan karakteristik pemanasan pada ruang *brooder* yang memiliki kapasitas panas (*thermal mass*) tertentu, sehingga perubahan suhu tidak terjadi secara instan setelah intensitas dimmer diubah.

1) Respons Suhu pada *Setpoint* 32°C

Pengujian pada *setpoint* 32°C dimulai dari suhu awal $30,8^\circ\text{C}$. Grafik respon suhu terhadap waktu ditunjukkan pada Gambar 4.2.



Gambar 4. 2 Grafik Respons Suhu terhadap *Setpoint* 32°C

Gambar 4.2 menunjukkan bahwa suhu awal sistem sebesar 30,8°C naik secara progresif menuju *setpoint* 32°C. Sistem mengalami *Rise time* selama 5 detik dan mencapai suhu puncak sebesar 32,8°C pada detik ke-23, sehingga menghasilkan *Overshoot* sebesar 2,42% (0,775°C) terhadap *setpoint*. Setelah melewati puncak, suhu mengalami penurunan (*undershoot*) hingga menyentuh nilai terendah di sekitar 31,7°C pada rentang detik ke-70 hingga ke-90, sebelum berosilasi kembali dan mendekati garis *setpoint*. Respons kemudian masuk dan menetap dalam pita toleransi $\pm 2\%$ pada detik ke-36 (*settling time*), dengan *error* keadaan tunak yang sangat kecil, yaitu $-0,025^\circ\text{C}$ atau setara $-0,078\%$. Osilasi kecil yang tampak pada rentang detik ke-110 hingga ke-135 menunjukkan penyesuaian akhir pengendali sebelum suhu benar-benar stabil pada nilai *setpoint*. Tabel parameter respons transien pada *setpoint* 32°C disajikan pada tabel 4.3.

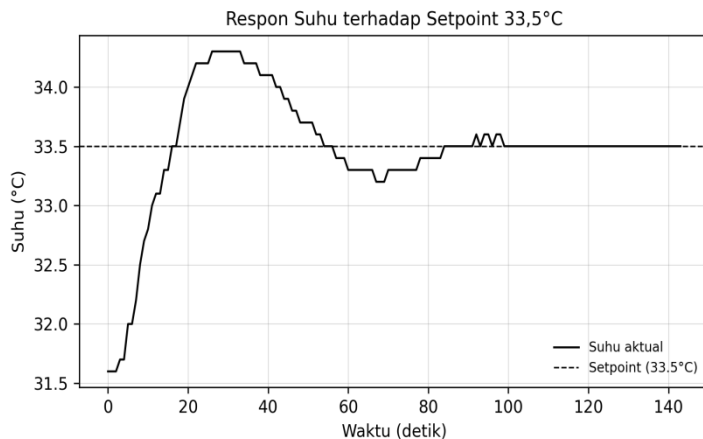
Tabel 4. 3 Parameter Respons Transien pada *Setpoint* 32°C

Parameter	Persamaan	Nilai
Suhu awal	Nilai awal sistem	30,8°C

Parameter	Persamaan	Nilai
Waktu naik (rise time)	$t(90\%) - t(10\%)$	5 detik
Suhu puncak (peak)	Nilai tertinggi output	32,8°C pada detik ke-23
<i>Overshoot</i>	$(C_{peak} - C_{ss}) / C_{ss} \times 100\%$	2,42% (0,775 °C)
<i>Settling time</i> (pita $\pm 2\%$)	t saat respons masuk dan tetap berada dalam pita toleransi $\pm 2\%$ SP	36 detik
<i>Error keadaan</i> tunak	$SP - C_{ss}$	-0,025 °C ($\approx$ -0,078%)

2) Respons Suhu pada Setpoint 33,5°C

Pengujian pada setpoint 33,5°C dimulai dari suhu awal 31,6°C. Grafik respon suhu terhadap waktu ditunjukkan pada Gambar 4.3.



Gambar 4. 3 Grafik Respons Suhu terhadap Setpoint 33,5°C

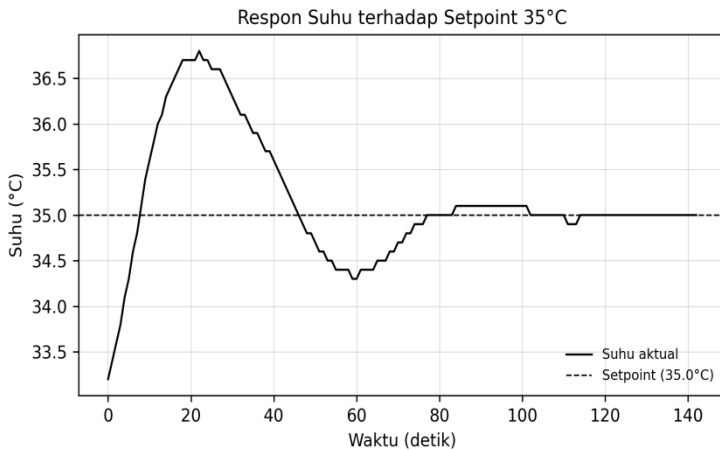
Pada Gambar 4.3 memperlihatkan kenaikan suhu yang cukup cepat dengan *Rise time* selama 11 detik, kemudian mencapai suhu puncak sebesar 34,3°C pada detik ke-26 yang menghasilkan *Overshoot* sebesar 2,39% (0,8°C). Setelah mencapai puncak, suhu menurun dan sempat berada di bawah *setpoint* pada kisaran 33,2°C–33,3°C selama rentang detik ke-55 hingga ke-77, sebelum kembali naik secara bertahap. Sistem tercatat masuk ke dalam pita toleransi $\pm 2\%$ pada detik ke-38 (*settling time*) dan akhirnya stabil tepat pada nilai *setpoint* dengan *error* keadaan tunak sebesar 0,0°C (0%). Beberapa riak kecil pada rentang detik ke-90 hingga ke-98 menunjukkan penyesuaian minor pengendali sebelum suhu benar-benar konstan hingga akhir pengujian. Tabel parameter respons transien pada *setpoint* 33.5°C disajikan pada tabel 4.4.

Tabel 4. 4 Parameter Respons Transien pada *Setpoint* 33.5°C

Parameter	Persamaan	Nilai
Suhu awal	Nilai awal sistem	31,6°C
Waktu naik (rise time)	$t(90\%) - t(10\%)$	11 detik
Suhu puncak (peak)	Nilai tertinggi output	34,3°C pada detik ke-26
<i>Overshoot</i>	$(C_{peak} - C_{ss}) / C_{ss} \times 100\%$	2,39% (0,8 °C)
<i>Settling time</i> (pita $\pm 2\%$)	t saat respons masuk dan tetap berada dalam pita toleransi $\pm 2\%$ SP	38 detik
<i>Error</i> keadaan tunak	$SP - C_{ss}$	0,0 °C (0%)

3) Respons Suhu pada *Setpoint* 35°C

Pengujian pada *setpoint* 35°C dimulai dari suhu awal 33,2°C. Grafik respon suhu terhadap waktu ditunjukkan pada Gambar 4.4.



Gambar 4. 4 Grafik Respons Suhu terhadap *Setpoint* 35°C

Gambar 4.4 menunjukkan respons suhu pada pengujian dengan *setpoint* 35°C, dengan suhu awal sistem sebesar 33,2°C. Sistem menunjukkan *Rise time* tercepat di antara ketiga pengujian, yaitu 7 detik, dan mencapai suhu puncak tertinggi sebesar 36,8°C pada detik ke-22. Kondisi ini menghasilkan *Overshoot* terbesar di antara ketiga *setpoint*, yaitu 5,14% (1,8°C). Setelah melewati puncak, suhu menurun cukup signifikan hingga mencapai titik terendah 34,3°C pada detik ke-60, kemudian naik secara bertahap menuju *setpoint*. Respons masuk ke dalam pita toleransi $\pm 2\%$ pada detik ke-38 (*settling time*) dan akhirnya menetap pada nilai *setpoint* dengan *error* keadaan tunak sebesar 0,0°C (0%). Osilasi kecil yang muncul pada rentang detik ke-83 hingga ke-113 mengindikasikan proses stabilisasi akhir pengendali sebelum suhu benar-benar konstan pada 35°C. Tabel parameter respons transien pada *setpoint* 35°C disajikan pada tabel 4.5.

Tabel 4. 5 Parameter Respons Transien pada *Setpoint* 35°C

Parameter	Persamaan	Nilai
Suhu awal	Nilai awal sistem	33,2°C

Parameter	Persamaan	Nilai
Waktu naik (rise time)	$t(90\%) - t(10\%)$	7 detik
Suhu puncak (peak)	Nilai tertinggi output	36,8°C pada detik ke-22
<i>Overshoot</i>	$(C_{peak} - C_{ss}) / C_{ss} \times 100\%$	5,14% (1,8 °C)
<i>Settling time</i> (pita $\pm 2\%$)	t saat respons masuk dan tetap berada dalam pita toleransi $\pm 2\%$ SP	38 detik
<i>Error</i> keadaan tunak	$SP - C_{ss}$	0,0 °C (0%)

4.1.2.3 Analisis Perbandingan Ketiga *Setpoint*

Untuk memudahkan perbandingan kinerja pengendali PID pada ketiga kondisi pengujian, parameter respons transien dari *setpoint* 32°C, 33,5°C, dan 35°C direkapitulasi pada Tabel 4.6 berikut.

Tabel 4. 6 Perbandingan Parameter Respons Transien pada Ketiga *Setpoint*

Parameter	<i>Setpoint</i> 32°C	<i>Setpoint</i> 33,5°C	<i>Setpoint</i> 35°C
Suhu awal	30,8°C	31,6°C	33,2°C
Waktu naik (rise time)	5 detik	11 detik	7 detik
Suhu puncak (peak)	32,8°C (detik ke-23)	34,3°C (detik ke-26)	36,8°C (detik ke-22)
<i>Overshoot</i>	2,42% (0,775°C)	2,39% (0,8°C)	5,14% (1,8°C)

Parameter	Setpoint 32°C	Setpoint 33,5°C	Setpoint 35°C
<i>Settling time</i> ($\pm 2\%$)	36 detik	38 detik	38 detik
Error keadaan tunak	-0,025°C ($\approx$ -0,078%)	0,0°C (0%)	0,0°C (0%)

Berdasarkan Tabel 4.6 dapat diamati bahwa pengendali PID mampu membawa suhu *brooder* mendekati nilai *setpoint* yang ditentukan pada ketiga kondisi pengujian, meskipun terdapat sedikit perbedaan karakteristik respons pada masing-masing *setpoint*. Dari sisi suhu awal, ketiga pengujian dimulai dari kondisi yang berbeda (30,8°C; 31,6°C; dan 33,2°C), sehingga selisih terhadap *setpoint* yang harus dikejar oleh sistem juga tidak sama, yaitu berturut-turut 1,2°C; 1,9°C; dan 1,8°C.

Dari segi kecepatan respons, *setpoint* 32°C memiliki *Rise time* tercepat, yaitu 5 detik, sedangkan *setpoint* 33,5°C memiliki *Rise time* terlama, yaitu 11 detik, dan *setpoint* 35°C berada di antara keduanya dengan 7 detik. Variasi ini menunjukkan bahwa kecepatan kenaikan suhu tidak selalu berbanding lurus dengan besar selisih suhu awal terhadap *setpoint*, melainkan juga dipengaruhi oleh kondisi termal *brooder* pada saat pengujian berlangsung.

Nilai *Overshoot* cenderung meningkat seiring dengan kenaikan nilai *setpoint*, yaitu 2,42% pada 32°C, 2,39% pada 33,5°C (relatif setara), dan meningkat cukup signifikan menjadi 5,14% pada 35°C. *Overshoot* terbesar pada *setpoint* 35°C mengindikasikan bahwa pada suhu target yang lebih tinggi, energi panas yang diberikan elemen pemanas pada fase transien awal masih berlebih sebelum aksi kontrol PID berhasil menekan laju kenaikan suhu, sehingga suhu aktual melampaui *setpoint* dengan selisih yang lebih besar (1,8°C) dibandingkan dua *setpoint* lainnya (0,775°C dan 0,8°C).

Adapun *Settling time* pada ketiga pengujian relatif konsisten, berada pada rentang 36–38 detik, yang menunjukkan bahwa pengendali PID memiliki kemampuan stabilisasi yang cukup seragam meskipun nilai *setpoint* dan besar *Overshoot* yang dihasilkan berbeda-beda.

Untuk *error* keadaan tunak, sistem menghasilkan *error* yang sangat kecil pada *setpoint* 32°C ($-0,025^{\circ}\text{C}$ atau $\approx -0,078\%$) dan bahkan mencapai 0% pada *setpoint* 33,5°C dan 35°C, yang berarti suhu aktual pada kondisi tunak benar-benar berhimpit dengan garis *setpoint*.

Secara keseluruhan, hasil perbandingan ini menunjukkan bahwa parameter PID yang digunakan sudah cukup efektif dalam mengendalikan suhu *brooder* pada berbagai nilai *setpoint*, dengan *Overshoot* yang masih berada pada batas wajar (di bawah 6%), *Settling time* yang relatif seragam, serta *error* keadaan tunak yang sangat kecil hingga nol.

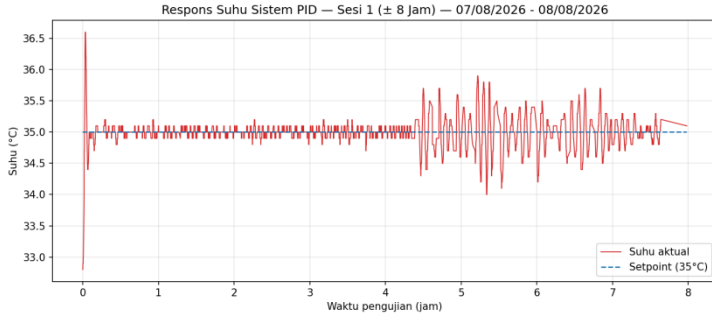
4.1.2.4 Pengujian Keandalan Sistem

Pengujian ini menyajikan hasil pengolahan dan analisis data keandalan sistem kontrol suhu berbasis PID pada mikrokontroler ESP32, yang digunakan untuk menjaga suhu pada nilai *setpoint* 35°C.

Pengujian keandalan dilakukan dalam tiga sesi berbeda dengan durasi yang bervariasi untuk melihat konsistensi performa sistem dalam jangka waktu singkat hingga panjang, termasuk pada rentang waktu yang melewati kondisi lingkungan berbeda (siang, malam, dini hari). Setiap baris data berisi stempel waktu, suhu aktual (Suhu) dan nilai acuan (*Setpoint*).

1) PENGUJIAN SESI 1 ($\pm$ 8 JAM)

Pengujian Sesi 1 dilaksanakan pada tanggal 07–08 Agustus 2026 selama kurang lebih 8 jam (19:35:46 – 03:34:38). Pada awal pengujian, suhu ruang berada pada 32,8°C, jauh di bawah *setpoint* 35°C, sehingga data ini menggambarkan respons transien penuh sistem dari kondisi awal hingga tercapainya kondisi tunak (*steady state*). Gambar 4.5 menunjukkan perbandingan antara suhu aktual terhadap *setpoint* sepanjang durasi pengujian.



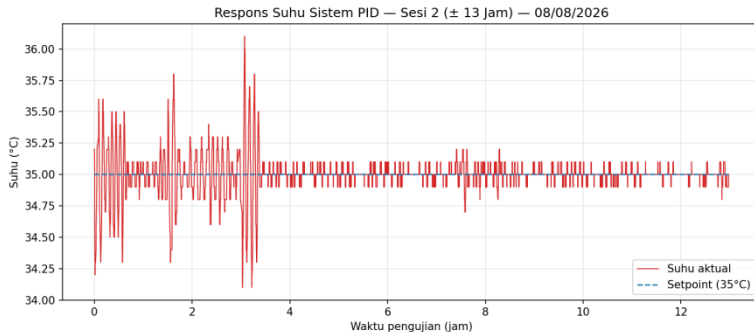
Gambar 4. 5 Grafik respons suhu terhadap waktu — Sesi 1 (± 8 jam)

Seperti yang terlihat pada gambar 4.5, Pada menit-menit awal (0–5 menit), suhu naik cepat dari 32,8°C menuju *setpoint* dengan intensitas dimmer yang besar, kemudian terjadi lonjakan (*Overshoot*) singkat hingga 36,6°C sebelum sistem berosilasi dan berangsur menetap di sekitar 35°C.

Pada rentang jam ke-4,4 hingga jam ke-7 terlihat osilasi dengan amplitudo yang jauh lebih besar (mencapai $\pm 0,8$ – $0,9^{\circ}\text{C}$ dari *setpoint*), kemungkinan disebabkan oleh gangguan (*disturbance*) lingkungan pada dini hari yaitu penurunan suhu ambient di sekitar sistem yang memaksa aktuator bekerja lebih agresif untuk mempertahankan *setpoint*. Menjelang akhir pengujian (jam ke-7,5 ke atas) osilasi kembali mengecil dan suhu kembali stabil mendekati 35°C, menunjukkan sistem PID mampu pulih dari gangguan tanpa kehilangan kestabilan.

2) PENGUJIAN SESI 2 (± 13 JAM)

Pengujian Sesi 2 dilaksanakan pada tanggal 08 Agustus 2026 selama kurang lebih 13 jam (04:01:38 – 17:00:15), merupakan sesi pengujian keandalan dengan durasi terpanjang di antara ketiga sesi. Suhu awal pengujian adalah 35,2°C, sudah sangat dekat dengan *setpoint*, sehingga sesi ini lebih merepresentasikan kemampuan sistem mempertahankan suhu dalam jangka panjang dibanding respons transien *start-up*.



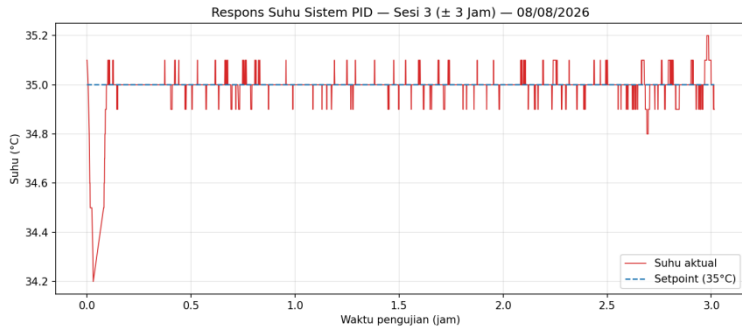
Gambar 4. 6 Grafik respons suhu terhadap waktu — Sesi 2 (± 13 jam)

Seperti yang terlihat pada gambar grafik respons suhu diatas, Pada 3,5 jam pertama pengujian terlihat osilasi suhu yang relatif lebih besar dibandingkan sisa durasi pengujian, dengan puncak fluktuasi mencapai $36,1^{\circ}\text{C}$ dan titik terendah $34,1^{\circ}\text{C}$, khususnya di sekitar jam ke-3 di mana lonjakan paling tajam terjadi. Pola ini kemungkinan berkaitan dengan perubahan kondisi lingkungan pada pagi hari (04:00–07:30) saat suhu ambient masih berfluktuasi akibat transisi dini hari menuju siang.

Setelah jam ke-3.5, sistem menunjukkan performa yang jauh lebih stabil dan konsisten hingga akhir pengujian pada jam ke-13, dengan osilasi suhu yang sangat kecil (umumnya dalam rentang $34,9^{\circ}\text{C}$ – $35,1^{\circ}\text{C}$). Secara statistik, 99,30% data Sesi 2 berada dalam pita toleransi $\pm 2\%$ dari *setpoint*, dengan simpangan baku keseluruhan hanya $0,113^{\circ}\text{C}$ — lebih baik dibandingkan Sesi 1 ($0,211^{\circ}\text{C}$). Hal ini menunjukkan bahwa dalam durasi pengujian yang lebih panjang, sistem tetap mampu menjaga keandalan pengendalian suhu tanpa mengalami penyimpangan jangka panjang.

3) PENGUJIAN SESI 3 (± 3 JAM)

Pengujian Sesi 3 dilaksanakan pada tanggal 08 Agustus 2026 selama kurang lebih 3 jam (18:00:02 – 21:00:54), merupakan sesi pengujian dengan durasi terpendek. Suhu awal pengujian adalah $35,1^{\circ}\text{C}$, hampir tepat pada *setpoint*, sehingga sesi ini murni menguji kestabilan sistem pada kondisi tunak (*steady-state maintenance*) tanpa fase transien besar.



Gambar 4. 7 Grafik respons suhu terhadap waktu — Sesi 3 ($\pm$ 3 jam)

Pada menit – menit awal pengujian suhu turun sesaat ke 34,2°C, sebelum kembali naik dan menetap. Suhu sepanjang Sesi 3 sangat konsisten berada pada rentang sempit 34,9°C–35,1°C, dengan hanya satu penyimpangan kecil ke 34,8°C di sekitar jam ke-2,7. Simpangan baku Sesi 3 adalah yang terkecil di antara ketiga sesi, yaitu 0,051°C, dengan 99,94% data berada dalam pita toleransi $\pm 2\%$ dari *setpoint*.

4.1.2.5 Kesimpulan Pengujian Keandalan Sistem PID

Berdasarkan hasil pengolahan dan analisis data ketiga sesi pengujian keandalan sistem kontrol suhu berbasis PID pada ESP32, dapat disimpulkan beberapa hal berikut:

- Gangguan lingkungan pada rentang dini hari/pagi hari (teramati pada Sesi 1 sekitar jam ke-4,4–7 dan Sesi 2 pada 3,5 jam pertama) sempat menyebabkan osilasi suhu yang lebih besar, namun sistem PID terbukti mampu pulih dan kembali stabil tanpa kehilangan kestabilan kontrol.
- Secara keseluruhan, sistem menunjukkan keandalan yang baik untuk pengendalian suhu jangka panjang, dengan lebih dari 97% data pada seluruh sesi berada dalam pita toleransi $\pm 2\%$ dari *setpoint*.

4.1.3 Pengujian Sistem Kontrol Logika Fuzzy

Pengujian sistem kontrol Logika Fuzzy dilakukan untuk mengetahui kemampuan sistem dalam mempertahankan suhu kandang DOC pada *setpoint* 35°C selama pengoperasian dalam jangka waktu panjang. Pengujian dilakukan dalam tiga sesi dengan durasi masing-masing sekitar delapan jam sehingga total waktu pengamatan mencapai 24 jam. Data suhu dibaca oleh sensor DHT22 dengan interval pencatatan satu menit dan diproses oleh ESP32 menggunakan algoritma Logika Fuzzy yang telah dirancang.

Setiap data hasil pengujian terdiri atas waktu pengambilan data, suhu aktual, nilai derajat keanggotaan μ_{COLD} , μ_{WARM} , dan μ_{HOT} , kondisi Fuzzy, serta persentase keluaran daya AC light dimmer. Dengan struktur data tersebut, evaluasi tidak hanya dilakukan terhadap kestabilan suhu, tetapi juga terhadap hubungan antara perubahan suhu, derajat keanggotaan, dan perubahan daya pemanas.

4.1.3.1 Konfigurasi Pengujian

Pengujian sistem kontrol logika Fuzzy dilakukan untuk mengetahui kemampuan sistem dalam mempertahankan suhu kandang DOC pada kondisi yang telah ditentukan dengan mengatur daya *heating lamp* melalui AC light dimmer. Pengujian dilakukan dengan mengamati perubahan suhu aktual yang dibaca oleh sensor DHT22 serta perubahan keluaran daya yang dihasilkan oleh algoritma Fuzzy. Berbeda dengan sistem kontrol yang hanya memberikan keputusan *ON* atau *OFF*, sistem yang digunakan pada penelitian ini menghasilkan keluaran daya dalam bentuk persentase sehingga besar daya pemanasan dapat disesuaikan terhadap perubahan suhu.

Pengujian juga dilakukan untuk melihat respons sistem ketika terjadi perubahan suhu selama periode pengamatan. Data yang diamati meliputi waktu pengambilan data, suhu aktual, nilai derajat keanggotaan μ_{COLD} , μ_{WARM} , dan μ_{HOT} , kondisi suhu, serta persentase daya yang diberikan kepada AC light dimmer. Data tersebut digunakan untuk mengetahui hubungan antara perubahan suhu dengan perubahan keluaran pengendalian.

Selain mengamati kestabilan suhu, pengujian digunakan sebagai dasar untuk mengevaluasi karakteristik pengendalian logika Fuzzy.

Hasil pengujian selanjutnya dapat dibandingkan dengan hasil pengujian kontrol PID pada parameter kinerja yang digunakan dalam penelitian, sehingga dapat diketahui perbedaan respons kedua metode dalam mempertahankan suhu kandang *DOC*.

4.1.3.2 Kondisi dan Konfigurasi Pengujian

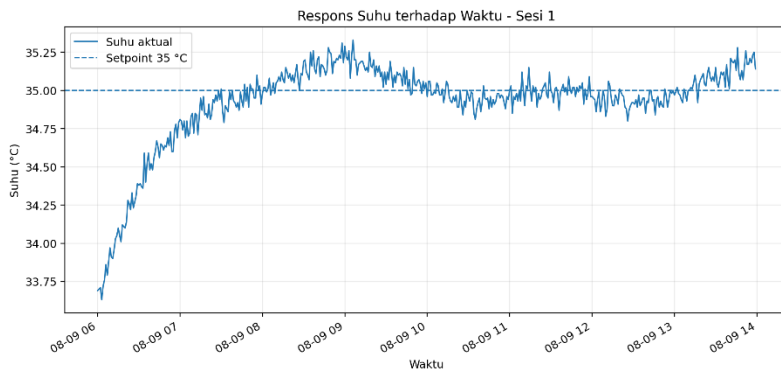
Pengujian dilakukan menggunakan sistem kontrol logika Fuzzy yang menggunakan komponen ESP32 sebagai pengendali utama, sensor DHT22 sebagai pembaca suhu, dan AC light dimmer sebagai pengatur daya *heating* lamp. Suhu aktual yang diperoleh dari DHT22 diproses oleh ESP32 melalui algoritma Fuzzy untuk menghasilkan nilai keluaran daya. Nilai tersebut kemudian digunakan sebagai dasar pengaturan daya listrik yang diberikan kepada *heating* lamp.

Parameter utama yang digunakan dalam pengujian adalah suhu *setpoint* sebesar 35°C. Sistem menggunakan tiga kondisi linguistik suhu, yaitu *COLD*, *WARM*, dan *HOT*. Pada kondisi suhu rendah, sistem memberikan daya pemanasan yang lebih besar. Ketika suhu mendekati *setpoint*, keluaran daya diturunkan secara bertahap. Ketika suhu telah mencapai atau melewati 35°C, sistem menghentikan keluaran daya pemanasan sesuai dengan algoritma yang diterapkan pada program.

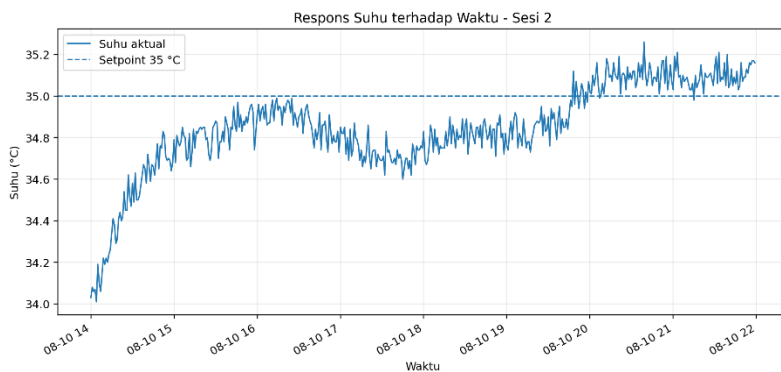
Pengujian dilakukan dalam tiga sesi pada tanggal 09, 10, dan 11 Agustus. Waktu pengujian dibuat bervariasi untuk mengetahui respons sistem pada kondisi lingkungan yang berbeda. Selama pengujian, data suhu dan keluaran sistem dicatat secara berkala sehingga perubahan suhu dapat diamati terhadap waktu.

4.1.3.3 Hasil Pengujian Sistem Kontrol Logika Fuzzy

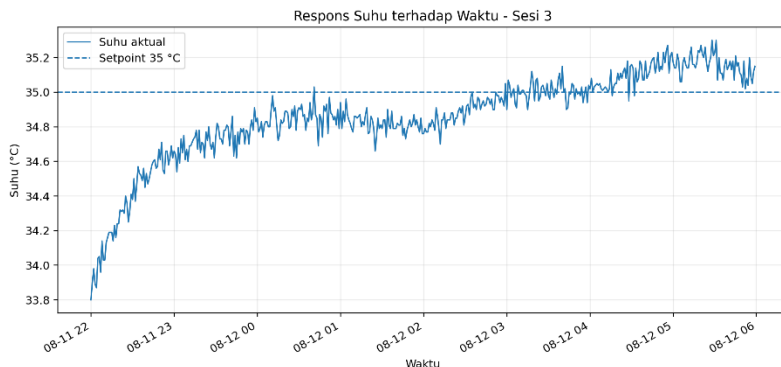
Pengujian sistem kontrol logika Fuzzy dilakukan selama tiga sesi dengan total waktu pengamatan selama 24 jam. Setiap sesi berlangsung selama 8 jam dengan interval pencatatan data satu menit, sehingga diperoleh 480 titik data pada masing-masing sesi dan 1.440 titik data secara keseluruhan. Data yang diperoleh meliputi waktu pengukuran, suhu aktual, derajat keanggotaan μ_{COLD} , μ_{WARM} , dan μ_{HOT} , kondisi Fuzzy, serta persentase keluaran daya AC light dimmer. Seluruh parameter tersebut digunakan untuk melihat hubungan antara perubahan suhu aktual dengan respons pengendalian yang diberikan sistem. Grafik respons suhu terhadap waktu di setiap sesi dapat dilihat pada gambar 4.8, 4.9 dan 4.10.



Gambar 4. 8 Grafik Respons Suhu terhadap Waktu Sesi 1



Gambar 4. 9 Grafik Respons Suhu terhadap Waktu Sesi 2



Gambar 4. 10 Grafik Respons Suhu terhadap Waktu Sesi 3

Tabel 4. 7 Hasil Statistik Suhu dan Keluaran Daya Setiap Sesi

Parameter	Sesi 1	Sesi 2	Sesi 3
Jumlah Data	480	480	480
Durasi	8 jam	8 jam	8 jam
Suhu Minimum (°C)	33.63	34.01	33.80
Suhu Maksimum (°C)	35.33	35.26	35.30
Suhu Rata-rata (°C)	34.924	34.847	34.862
Simpangan Baku (°C)	0.284	0.215	0.258
Intensitas dimmer Minimum (%)	0	0	0
Intensitas dimmer Maksimum (%)	27.4	19.8	24.0
Intensitas dimmer Rata-rata (%)	2.42	3.55	3.46
<i>WARM</i> (%)	53.125	73.750	67.917
<i>HOT</i> (%)	46.875	26.250	32.083

Berdasarkan tabel 4.7, Pada Sesi 1 suhu aktual berada pada rentang 33,63–35,33°C dengan nilai rata-rata 34,924°C. Kondisi *COLD* tidak muncul selama pengamatan karena suhu terendah masih berada di atas 30°C. Sistem bekerja pada kondisi *WARM* selama 255 titik data atau 53,125% dari keseluruhan waktu pengujian, sedangkan kondisi *HOT* muncul pada 225 titik data atau 46,875%. Keluaran daya dimmer berada pada rentang 0–27,4%, dengan nilai rata-rata 2,42%.

Hal tersebut menunjukkan bahwa sebagian besar proses pengendalian berlangsung di sekitar daerah *setpoint*, sehingga kebutuhan pemanasan relatif kecil.

Pada Sesi 2, suhu aktual berada pada rentang 34,01–35,26°C dan memiliki nilai rata-rata 34,847°C. Rentang suhu yang lebih sempit dibandingkan Sesi 1 menunjukkan bahwa perubahan temperatur selama sesi ini relatif lebih kecil. Kondisi *WARM* muncul sebanyak 354 titik data atau 73,75%, sedangkan kondisi *HOT* sebanyak 126 titik data atau 26,25%. Keluaran daya berada pada rentang 0–19,8% dengan rata-rata 3,55%. Kondisi tersebut menunjukkan bahwa sistem lebih sering berada pada daerah *WARM* dan melakukan penyesuaian daya dengan tingkat keluaran yang relatif rendah

Pada Sesi 3, suhu tercatat antara 33,80–35,30°C dengan nilai rata-rata 34,862°C. Sebanyak 326 titik data atau 67,917% berada pada kondisi *WARM*, sedangkan 154 titik data atau 32,083% berada pada kondisi *HOT*. Keluaran daya berada pada rentang 0–24,0% dengan rata-rata 3,46%.

Dibandingkan Sesi 1, rentang suhu pada Sesi 3 lebih sempit, sedangkan dibandingkan Sesi 2, variasi suhu sedikit lebih besar.

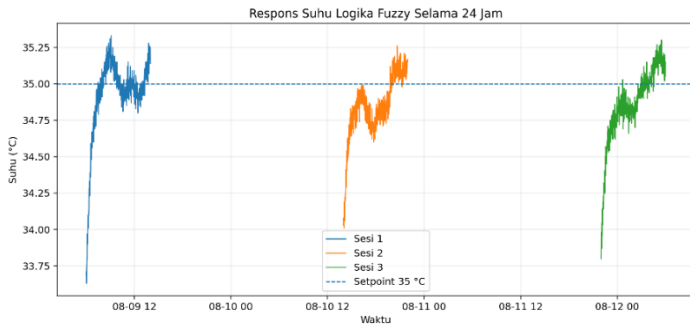
4.1.3.4 Analisis Respons Suhu Terhadap *Setpoint*

Berdasarkan hasil pengujian selama tiga sesi, suhu aktual berada relatif dekat dengan *setpoint* sebesar 35°C. Dari keseluruhan 1.440 data, suhu minimum yang tercatat adalah 33,63°C dan suhu maksimum sebesar 35,33°C. Dengan demikian, rentang perubahan suhu keseluruhan selama pengujian adalah 1,70°C. Nilai suhu rata-rata keseluruhan sebesar 34,878°C menunjukkan bahwa sistem mampu mempertahankan temperatur di sekitar nilai *setpoint* yang ditentukan. Berikut hasil statistik keseluruhan pengujian yang disajikan pada tabel 4.8.

Tabel 4. 8 Statistik Keseluruhan Pengujian

Parameter	Hasil
Total Data	1.440
Total Durasi	24 Jam
<i>Setpoint</i>	35°C
Suhu Minimum	33.63°C
Suhu Maksimum	35.33°C
Suhu Rata-rata	34.878°C
Simpangan Baku	0.256°C
Selisih Rata-rata Terhadap <i>Setpoint</i>	0.122°C

Perbedaan antara suhu rata-rata dan *setpoint* adalah sekitar 0,122°C. Nilai tersebut menunjukkan bahwa temperatur rata-rata selama pengujian berada sedikit di bawah target. Kondisi ini berkaitan dengan karakteristik algoritma pengendalian yang menurunkan keluaran daya ketika suhu semakin mendekati 35°C. Pada saat suhu mencapai atau melewati 35°C, program menetapkan keluaran intensitas dimmer sebesar 0%, sehingga sistem tidak terus memberikan pemanasan ketika temperatur telah mencapai target. Berikut gambar grafik respon suhu sistem kontrol logika fuzzy selama 24 Jam.



Gambar 4. 11 Respons Suhu Sistem Kontrol Logika Fuzzy Selama 24 Jam

Berdasarkan gambar 4.11, Garis horizontal 35°C menunjukkan *setpoint* sedangkan kurva suhu menunjukkan perubahan temperatur yang dibaca DHT22 selama tiga sesi pengujian. Kurva tidak menunjukkan perubahan yang sangat jauh dari *setpoint*, dengan sebagian besar data berada pada daerah sekitar 35°C.

Dari hasil tersebut dapat dilihat bahwa sistem tidak mempertahankan suhu dengan cara mempertahankan satu nilai daya tertentu. Ketika suhu berada lebih rendah dari 35°C, masih terdapat keluaran daya yang diberikan melalui AC light dimmer. Sebaliknya, ketika suhu mencapai atau melewati 35°C, keluaran daya menjadi 0%. Pola tersebut sesuai dengan algoritma pengendalian yang digunakan pada ESP32.

4.1.3.5 Distribusi Keluaran Daya AC Light Dimmer

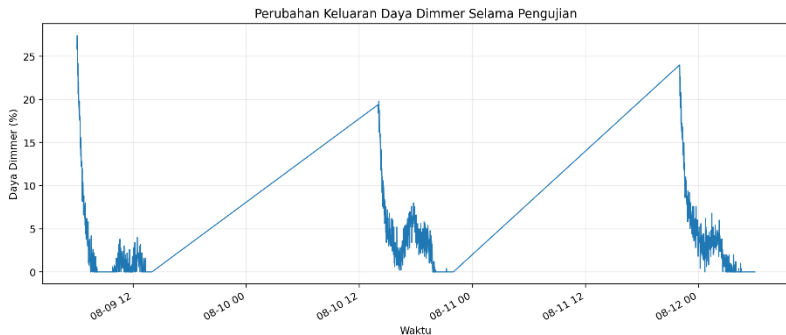
Pengamatan terhadap keluaran daya diperlukan untuk menunjukkan karakteristik pengendalian logika Fuzzy yang digunakan dalam penelitian. Keluaran sistem tidak hanya berupa keputusan ON atau OFF, tetapi berupa nilai persentase daya/intensitas dimmer yang berubah berdasarkan posisi suhu terhadap fungsi keanggotaan.

Pada daerah suhu 30–35°C, keluaran daya dihitung menggunakan pembobotan derajat keanggotaan *COLD* dan *WARM*. Karena data pengujian berada pada rentang 33,63–35,33°C, sebagian besar keluaran berada pada nilai daya rendah. Ketika suhu semakin mendekati 35°C, nilai μ_{COLD} semakin kecil sehingga keluaran daya juga semakin rendah. Distribusi keluaran daya AC light dimmer dapat dilihat pada tabel 4.9.

Tabel 4. 9 Distribusi Keluaran Daya AC Light Dimmer

Parameter	Sesi 1	Sesi 2	Sesi 3
Daya Minimum (%)	0	0	0
Daya Maksimum (%)	27.4	19.8	24.0
Daya Rata-rata (%)	2.42	3.55	3.46

Hubungan antara tabel 4.9 pada grafik seperti yang disajikan pada gambar 4.12, menunjukkan kecenderungan bahwa semakin tinggi suhu aktual menuju 35°C, semakin kecil keluaran daya. Hal ini merupakan konsekuensi langsung dari persamaan pembobotan yang digunakan dalam program. Pada suhu yang lebih rendah dalam daerah *WARM*, μ_{COLD} masih memiliki nilai yang lebih besar sehingga daya yang diberikan lebih tinggi. Ketika suhu mendekati 35°C, μ_{COLD} mendekati nol dan daya yang dihasilkan juga mendekati nol.



Gambar 4. 12 Hubungan Suhu Aktual terhadap Keluaran

Dengan demikian, perubahan keluaran dimmer merupakan bagian penting dari mekanisme pengendalian sistem. Sistem melakukan pengurangan daya secara bertahap sebelum mencapai *setpoint*, kemudian menghentikan pemanasan ketika suhu mencapai atau melewati 35°C.

4.1.3.6 Distribusi Derajat Keanggotaan Fuzzy

Tahap fuzzifikasi menghasilkan tiga parameter utama, yaitu μ_{COLD} , μ_{WARM} , dan μ_{HOT} . Nilai tersebut menunjukkan tingkat keanggotaan suhu aktual terhadap masing-masing himpunan Fuzzy.

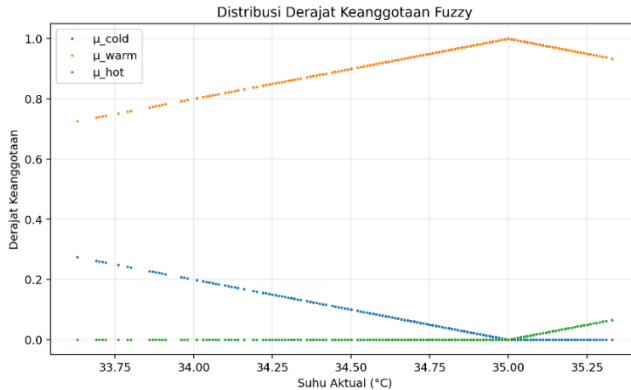
Pada data pengujian, nilai μ_COLD dan μ_WARM menjadi parameter yang paling berpengaruh terhadap keluaran daya karena keduanya digunakan secara langsung dalam persamaan pembobotan pada daerah 30–35°C. Berikut tabel Distribusi Derajat Keanggotaan Fuzzy.

Tabel 4. 10 Distribusi Derajat Keanggotaan Fuzzy

Kondisi Suhu	μ_COLD	μ_WARM	μ_HOT
30°C	1.00	0.00	0.00
31°C	0.80	0.20	0.00
32°C	0.60	0.40	0.00
33°C	0.40	0.60	0.00
34°C	0.20	0.80	0.00
35°C	0.00	1.00	0.00
>35°C	0.00	menurun	meningkat

Berdasarkan tabel 4.10, pada hasil pengujian suhu tidak mencapai daerah *COLD* sehingga nilai μ_COLD maksimum yang muncul lebih kecil dari 1. Sebaliknya, pada saat suhu berada di atas 35°C, μ_HOT mulai memiliki nilai lebih besar dari nol. Meskipun demikian, sesuai implementasi program, nilai μ_HOT tidak digunakan dalam perhitungan keluaran daya dan hanya menjadi informasi mengenai kondisi keanggotaan suhu.

Perubahan μ_COLD dan μ_WARM pada daerah 30–35°C menunjukkan proses transisi kondisi suhu yang digunakan untuk menentukan besar keluaran daya. Sementara itu, μ_HOT mulai meningkat setelah suhu melewati 35°C. Grafik pada gambar 4.13, memperlihatkan hubungan antara suhu aktual dengan ketiga derajat keanggotaan.



Gambar 4. 13 Distribusi Derajat Keanggotaan *COLD*, *WARM*, dan *HOT* terhadap Suhu Aktual

4.1.3.7 Analisis Statistik Hasil Pengujian Pada Setiap Sesi

Analisis statistik digunakan untuk mengetahui tingkat penyebaran data suhu pada setiap sesi dan melihat konsistensi sistem selama 24 jam pengujian. Parameter yang digunakan meliputi suhu minimum, suhu maksimum, suhu rata-rata, simpangan baku, serta keluaran daya rata-rata.

Sesi 2 memiliki simpangan baku paling kecil, yaitu 0,215°C. Nilai tersebut menunjukkan bahwa penyebaran suhu pada Sesi 2 lebih sempit dibandingkan Sesi 1 dan Sesi 3. Sesi 1 memiliki simpangan baku terbesar sebesar 0,284°C, sedangkan Sesi 3 sebesar 0,258°C.

Suhu rata-rata ketiga sesi juga menunjukkan nilai yang relatif berdekatan, yaitu 34,924°C pada Sesi 1, 34,847°C pada Sesi 2, dan 34,862°C pada Sesi 3. Perbedaan tersebut menunjukkan bahwa karakteristik pengendalian tidak berubah secara signifikan antar periode pengujian. Tabel perbandingan statistik pada setiap sesi disajikan pada tabel 4.11.

Tabel 4. 11 Perbandingan Statistik Tiga Sesi

Parameter	Sesi 1	Sesi 2	Sesi 3
Suhu Minimum (°C)	33.63	34.01	33.80
Suhu Maksimum (°C)	35.33	35.26	35.30
Rata-rata (°C)	34.924	34.847	34.862
Simpangan Baku (°C)	0.284	0.215	0.258
Daya Rata-rata (%)	2.42	3.55	3.46

Berdasarkan tabel diatas, sistem menunjukkan karakteristik yang relatif konsisten pada ketiga sesi. Selisih suhu rata-rata tertinggi dan terendah antar sesi hanya sebesar 0,077°C. Kondisi ini menunjukkan bahwa perubahan periode pengujian tidak menghasilkan perbedaan besar terhadap suhu rata-rata yang dipertahankan sistem.

4.1.3.8 Analisis *Rise-Time*

Pada penelitian ini, *rise-time* dihitung berdasarkan formulasi yaitu menggunakan acuan *setpoint* 35 °C sebagai nilai tujuan, bukan menggunakan nilai *steady-state* (T_{ss}) pada masing-masing sesi. Pendekatan ini digunakan agar perbandingan antara kontrol PID dan Logika Fuzzy dilakukan pada dasar yang konsisten.

Secara matematis, nilai suhu pada batas 10% dan 90% ditentukan dengan persamaan:

$$T_{10} = T_{awal} + 0,1 (SP - T_{awal})$$

$$T_{90} = T_{awal} + 0,9 (SP - T_{awal})$$

$$\Delta T = SP - T_{awal}$$

Dengan **T_{awal}** adalah suhu awal setiap sesi pengujian, SP adalah *setpoint* sebesar 35°C, **T₁₀** adalah suhu pada saat respons mencapai 10% perubahan, dan **T₉₀** adalah suhu pada saat respons mencapai 90% perubahan. *Rise-time* kemudian dihitung sebagai:

$$T_r = t_{90} - t_{10}$$

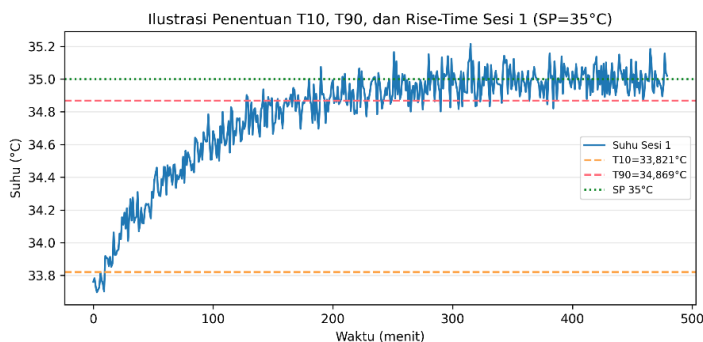
Dengan **t₁₀** adalah waktu ketika suhu aktual pertama kali mencapai atau melewati **t₉₀**, dan **t₉₀** adalah waktu ketika suhu aktual pertama kali mencapai atau melewati **T₉₀**.

Pencarian t_{10} dan t_{90} dilakukan dengan metode *forward search* pada data mentah berinterval 1 menit yang tersimpan di Google Sheets (480 data per sesi).

Tabel 4. 12 Hasil Perhitungan Rise-time Sistem Logika Fuzzy

Sesi	Tawal (°C)	$\Delta T =$ SP- Tawa l (°C)	T10 = Tawal+0, 1 ΔT (°C)	T90 = Tawal+0, 9 ΔT (°C)	Keterangan
Sesi 1	33,69	1,31	33,821	34,869	SP=35°C
Sesi 2	34,03	0,97	34,127	34,903	SP=35°C
Sesi 3	33,80	1,20	33,920	34,880	SP=35°C

Berdasarkan data statistik yang tersedia pada Tabel 4.12, suhu awal setiap sesi adalah sebagai berikut: Sesi 1 sebesar 33,69 °C, Sesi 2 sebesar 34,03 °C, dan Sesi 3 sebesar 33,80 °C. Dengan menggunakan $SP = 35$ °C, diperoleh grafik nilai batas 10% dan 90% sebagai berikut:



Gambar 4. 14 Penentuan T10, T90, dan Rise-Time pada Tiga Sesi Pengujian

Berikut rumus perhitungan *Rise-Time* untuk Sesi 1 ;

- Suhu awal $T_{awal} = 33,69$ °C.
- $\Delta T = 35 - 33,69 = 1,31$ °C.
- $T_{10} = 33,69 + 0,1 \times 1,31 = 33,69 + 0,131 = 33,821$ °C. $T_{90} = 33,69 + 0,9 \times 1,31 = 33,69 + 1,179 = 34,869$ °C.

Selanjutnya dari data mentah 1 menit, dicari waktu t_{10} ketika suhu pertama kali $\geq 33,821^\circ\text{C}$ dan waktu t_{90} ketika suhu pertama kali $\geq 34,869^\circ\text{C}$. Selisih $t_{90} - t_{10}$ merupakan T_r Sesi 1. Prosedur yang sama diterapkan untuk Sesi 2 dan Sesi 3.

4.1.3.9 Analisis *Settling-time*

Settling-time (T_s) menunjukkan waktu yang dibutuhkan sistem untuk mencapai dan mempertahankan kondisi stabil di sekitar *setpoint*. Dengan $SP = 35^\circ\text{C}$, maka toleransi $\pm 2\%$ dihitung sebagai:

$$\Delta T_s = 0,02 \times T_{sp} = 0,02 \times 35 = 0,7^\circ\text{C}$$

$$T_{lower} = T_{sp} - 0,7 = 35 - 0,7 = 34,3^\circ\text{C}$$

$$T_{upper} = T_{sp} + 0,7 = 35 + 0,7 = 35,7^\circ\text{C}$$

$$\text{Kriteria stabil: } 34,3^\circ\text{C} \leq T(t) \leq 35,7^\circ\text{C}$$

Settling-time ditentukan dengan metode *backward search*: mulai dari data terakhir (menit ke-480) mundur ke depan, cari waktu pertama ketika suhu keluar dari pita $34,3\text{--}35,7^\circ\text{C}$. Waktu setelah titik tersebut, di mana seluruh data selanjutnya berada di dalam pita, ditetapkan sebagai T_s . Jika setelah masuk pita suhu kembali keluar, maka pencarian dilanjutkan. Hasil perhitungan *settling-time* sistem logika fuzzy dapat dilihat pada tabel 4.13.

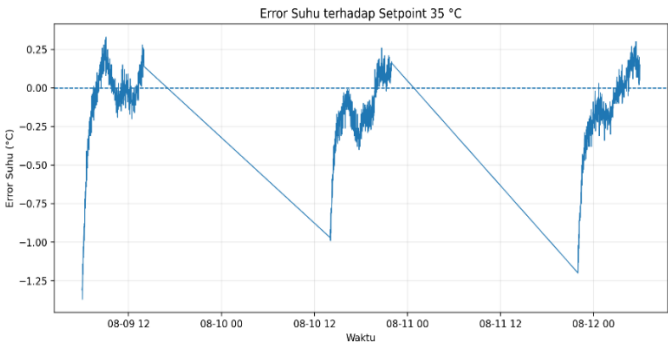
Tabel 4. 13 Hasil Perhitungan *Settling-time* Sistem Logika Fuzzy

Sesi	Pita $\pm 2\%$ SP (35°C)	Keterangan
Sesi 1	34,3–35,7	Pita baru lebih ketat di batas atas
Sesi 2	34,3–35,7	Konsisten SP
Sesi 3	34,3–35,7	Konsisten SP

Berdasarkan kriteria pita $\pm 2\%$ terhadap *setpoint* 35°C , batas bawah dan batas atas *settling-time* adalah $34,3^\circ\text{C}$ dan $35,7^\circ\text{C}$. Kriteria ini digunakan secara konsisten untuk ketiga sesi pengujian.

4.1.3.10 Analisis *Steady-state error* (SSE)

Steady-state error (SSE) merupakan selisih absolut antara *setpoint* dan suhu pada kondisi tunak. SSE dihitung dengan persamaan $SSE = |Tsp - Tss|$, dengan $Tsp = 35\text{ }^{\circ}\text{C}$ dan Tss adalah suhu rata-rata pada kondisi *steady-state* yang telah ditentukan setelah sistem mencapai *settling-time*. Berikut grafik *Error* suhu terhadap *setpoint* $35\text{ }^{\circ}\text{C}$ Selama 24 Jam.



Gambar 4. 15 Error Suhu terhadap Setpoint $35\text{ }^{\circ}\text{C}$ Selama 24 Jam

Secara keseluruhan, suhu rata-rata 24 jam sebesar $34,878\text{ }^{\circ}\text{C}$ menghasilkan SSE global sebesar $|35 - 34,878| = 0,122\text{ }^{\circ}\text{C}$. Nilai ini menunjukkan bahwa rata-rata suhu selama 24 jam hanya $0,122\text{ }^{\circ}\text{C}$ di bawah *setpoint*. Berikut hasil perhitungan *Steady-state error* (SSE) yang ditunjukkan pada Tabel 4. 14.

Tabel 4. 14 Hasil Perhitungan *Steady-state error* (SSE)

Sesi	Tss (10 data terakhir) °C	SSE (pendekatan lama) °C	Tavg Sesi °C	SSE Terhadap SP (Baru) °C
Sesi 1	35,184	0,184	34,924	0,076
Sesi 2	35,130	0,130	34,847	0,153
Sesi 3	35,096	0,096	34,862	0,138
Rata-rata 24 jam	35,137	0,137	34,878	0,122

Berdasarkan tabel 4.14. T_{ss} dihitung dari rata-rata sesi yang memang berada pada kondisi stabil, berikut perhitungannya ;

- Sesi 1 rata-rata 34,924 °C → $SSE = |35 - 34,924| = 0,076$ °C,
- Sesi 2 rata-rata 34,847 °C → $SSE = 0,153$ °C,
- Sesi 3 rata-rata 34,862 °C → $SSE = 0,138$ °C.

4.1.3.11 Analisis *Overshoot*

Overshoot didefinisikan sebagai kondisi ketika suhu maksimum melebihi *setpoint* selama proses menuju kondisi stabil. Formulasi yang digunakan adalah sebagai berikut ;

$$M_p = (T_{max} - T_{sp}) / T_{sp} \times 100\%, \text{ jika } T_{max} > T_{sp}, \\ \text{selain itu } M_p = 0\%$$

Dengan formulasi tersebut, *Overshoot* menunjukkan persentase penyimpangan maksimum di atas target yang ditetapkan. Pada sistem kandang DOC, *Overshoot* yang besar perlu dihindari karena dapat menyebabkan stres termal pada DOC akibat suhu yang terlalu tinggi melebihi kebutuhan *brooding*.

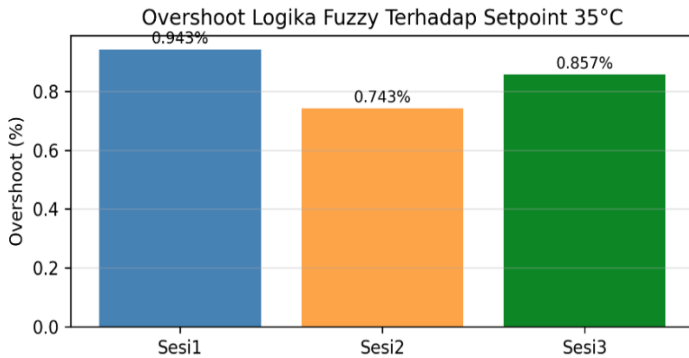
Berdasarkan data hasil pengujian pada Tabel 4.15, suhu maksimum setiap sesi adalah: Sesi 1 sebesar 35,33 °C, Sesi 2 sebesar 35,26 °C, dan Sesi 3 sebesar 35,30 °C. Dengan $T_{sp} = 35$ °C, maka perhitungan *Overshoot* adalah sebagai berikut:

- Sesi 1: $M_p = (35,33 - 35)/35 \times 100\% = 0,33/35 \times 100\% = 0,943\%$
- Sesi 2: $M_p = (35,26 - 35)/35 \times 100\% = 0,26/35 \times 100\% = 0,743\%$
- Sesi 3: $M_p = (35,30 - 35)/35 \times 100\% = 0,30/35 \times 100\% = 0,857\%$

Tabel 4. 15 Hasil Perhitungan *Overshoot* Sistem Logika Fuzzy

Sesi	Tmax (°C)	Tsp (°C)	<i>Overshoot</i> Mp (%) Terhadap SP
Sesi 1	35,33	35	0,943
Sesi 2	35,26	35	0,743
Sesi 3	35,30	35	0,857
Rata- rata	35,30	35	0,848

Pada tabel 4.15 menjelaskan nilai *Overshoot* rata-rata 0,848% menunjukkan bahwa penyimpangan maksimum di atas *setpoint* hanya sekitar 0,30 °C. Hal ini jauh lebih kecil dibandingkan *Overshoot* pada kontrol PID pada *setpoint* yang sama, yaitu sebesar 5,14% (1,8 °C) sebagaimana tercantum pada Tabel 4.5. Karakteristik *Overshoot* yang kecil pada Logika Fuzzy berkaitan dengan mekanisme pengurangan daya secara bertahap ketika suhu mendekati 35 °C, serta penghentian daya (0%) ketika suhu mencapai atau melewati *setpoint*. Pada gambar 4.16 menampilkan grafik perbandingan *Overshoot* disetiap sesi.



Gambar 4. 16 *Overshoot* Sistem Logika Fuzzy Tiap Sesi Terhadap Setpoint 35°C

4.1.4 Pengujian Kedua Metode Kontrol Pada Anak Ayam

Pengujian kedua metode kontrol menggunakan *setpoint* 35°C untuk kebutuhan thermal bagi ayam DOC. Hasil pengujian secara langsung terhadap perilaku ayam DOC di dalam kandang membuktikan efektivitas kedua sistem kontrol. Pada kondisi suhu belum stabil (sebelum *setpoint* tercapai di angka 35°C), ayam cenderung berkumpul di satu titik sebagai respons terhadap stres termal akibat suhu yang terlalu rendah. Setelah sistem kontrol baik PID maupun Logika Fuzzy berhasil menstabilkan suhu kandang mendekati *setpoint*, ayam tidak lagi berkumpul dan bergerak lebih bebas ke seluruh area kandang. Perubahan perilaku ini mengkonfirmasi bahwa suhu kandang telah berada pada kondisi nyaman dan optimal bagi DOC.

Kondisi kandang sebelum suhu stabil ditunjukkan pada gambar berikut, di mana ayam tampak berkumpul di satu titik sebagai indikasi ketidaknyamanan termal :



Gambar 4. 17 Respon Anak Ayam Ketika Suhu Belum Stabil

Pada Gambar 4.17 terlihat ayam-ayam DOC berkumpul mengelompok di satu titik di dalam kandang. Perilaku ini merupakan respons fisiologis terhadap suhu lingkungan yang masih berada di bawah titik *setpoint*, di mana ayam secara naluriah berusaha saling menghangatkan satu sama lain untuk mempertahankan suhu tubuh.

Setelah suhu kandang mencapai dan stabil pada *setpoint* yang telah ditentukan, perubahan perilaku ayam dapat diamati sebagaimana ditunjukkan pada gambar berikut:

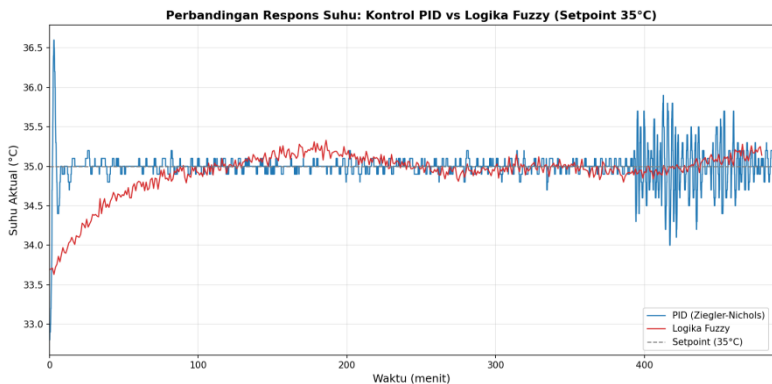


Gambar 4. 18 Respon Anak Ayam Ketika Suhu Telah Stabil

Pada Gambar 4.18 terlihat bahwa posisi ayam sudah tidak lagi berkumpul di satu titik. Ayam bergerak bebas ke seluruh area kandang, yang mengindikasikan bahwa suhu kandang telah mencapai kondisi nyaman dan stabil sesuai dengan suhu yang telah ditentukan.

4.1.5 Perbandingan Langsung PID (*Setpoint* 35 °C) vs Logika Fuzzy

Pengujian dilakukan pada sistem kontrol suhu berbasis ESP32 dengan dua metode kontrol yang berbeda, yaitu kontrol PID (hasil tuning Ziegler-Nichols) dan Logika Fuzzy, dengan *setpoint* suhu yang sama yaitu 35°C. Data suhu aktual direkam secara kontinu selama pengujian berlangsung (PID pengujian sesi 1 ±8,1 jam dan Fuzzy Sesi 1 ±8 jam) untuk mengevaluasi respons transien dan kestabilan masing-masing metode terhadap nilai acuan (*setpoint*). Berikut grafik perbandingan pada kedua metode kontrol.



Gambar 4. 19 Perbandingan respons suhu aktual terhadap waktu untuk kontrol PID dan Logika Fuzzy pada *setpoint* 35°C.

Pada rentang waktu $t \approx 394\text{--}460$ menit (menit ke-394 hingga ke-460 dari total durasi pengujian), data suhu pada sesi PID menunjukkan osilasi tajam dan berulang di luar pita toleransi $\pm 2\%$, dengan simpangan mencapai sekitar $\pm 1^\circ\text{C}$ dari *setpoint* sebelum akhirnya kembali stabil pada menit ke-460. Pola ini kemungkinan disebabkan oleh gangguan eksternal (misalnya perubahan suhu lingkungan, aliran udara, atau gangguan pembacaan sensor) dan bukan merupakan bagian dari respons transien awal terhadap *setpoint*. Di luar rentang tersebut, kontrol PID menunjukkan kestabilan yang sangat baik. Fenomena ini tidak teramati pada data Logika Fuzzy, yang tetap stabil sepanjang sesi pengujian.

Evaluasi kinerja dilakukan berdasarkan empat parameter standar dalam analisis sistem kontrol, yaitu *rise time*, *Overshoot*, *settling time*, dan *steady-state error*. Tabel parameter performansinya dapat dilihat pada tabel 4.16.

Tabel 4. 16 Perbandingan Parameter Kinerja Kontrol PID vs Logika Fuzzy

Parameter	Kontrol PID	Logika Fuzzy	Metode Lebih Baik
<i>Rise time</i> (10–90%)	1,08 menit (± 65 detik)	69,00 menit	PID
<i>Overshoot</i>	4,57% (puncak $36,60^\circ\text{C}$, $t \approx 2,9$ menit)	0,43% (puncak $35,15^\circ\text{C}$, $t \approx 137$ menit)	Fuzzy
<i>Settling time</i> (pita $\pm 2\%$)	3,92 menit (fase transien awal)	27,00 menit	PID
<i>Steady-state error</i> (rata-rata)	$\approx 0,001^\circ\text{C}$ (<i>error</i> rata-rata $0,046^\circ\text{C}$, std $0,078^\circ\text{C}$)	$0,007^\circ\text{C}$ (<i> error </i> rata-rata $0,076^\circ\text{C}$, std $0,097^\circ\text{C}$)	PID

Berdasarkan perbandingan parameter kinerja di kedua metode kontrol pada tabel 4.16, Berikut penjelasan parameter performansinya ;

1) Kecepatan respons (*Rise time & settling time*).

Kontrol PID jauh lebih unggul dalam kecepatan mencapai *setpoint* hanya membutuhkan sekitar 1 menit untuk naik dari 10% ke 90% rentang suhu dan sekitar 4 menit untuk menetap stabil, dibandingkan Logika Fuzzy yang membutuhkan puluhan menit. Hal ini konsisten dengan karakteristik kontrol PID hasil tuning Ziegler-Nichols yang memang dirancang untuk respons agresif dan cepat.

2) *Overshoot*.

Kecepatan respons PID ini membawa konsekuensi berupa *Overshoot* yang lebih besar (4,57%, suhu sempat melonjak ke 36,6°C) dibandingkan Logika Fuzzy yang praktis tidak menghasilkan *Overshoot* berarti (0,43%). Pada aplikasi yang sensitif terhadap suhu berlebih misalnya kandang unggas, di mana lonjakan suhu dapat membuat anak ayam stres atau dehidrasi — karakteristik Fuzzy yang landai dan minim *Overshoot* menjadi nilai tambah tersendiri meskipun responsnya lebih lambat.

3) *Steady-state error*.

Pada kondisi tunak, kedua metode menunjukkan galat yang sangat kecil dan sebanding (PID $\approx 0,046^{\circ}\text{C}$, Fuzzy $\approx 0,076^{\circ}\text{C}$ *error* absolut rata-rata), menandakan keduanya mampu mempertahankan suhu mendekati *setpoint* dengan baik setelah fase transien.

4) Ketahanan terhadap gangguan (*Noise*).

Temuan menarik adalah munculnya osilasi signifikan pada data PID di pertengahan sesi pengujian (menit ke-394–460), yang tidak terjadi pada Logika Fuzzy. Meski PID kembali stabil setelah gangguan tersebut, hal ini mengindikasikan bahwa kontrol PID linier dengan parameter tetap (*fixed-gain*) berpotensi lebih sensitif terhadap perubahan kondisi lingkungan dibandingkan Logika Fuzzy, yang basis aturannya secara inheren lebih adaptif terhadap variasi kondisi input.

4.2 Pembahasan

Hasil pengujian menunjukkan bahwa kedua metode kontrol mampu mempertahankan suhu kandang DOC di sekitar *setpoint* 35 °C, namun dengan karakteristik yang berbeda. Kontrol PID dengan parameter hasil *autotune* Ziegler-Nichols $K_u = 57,296$ dan $P_u = 112,328$ detik menghasilkan $K_p = 34,377$, $K_i = 0,612$, dan $K_d = 482,694$. Pada *setpoint* 35 °C, PID menghasilkan *rise-time* 7 detik, *settling-time* 38 detik, *Overshoot* 5,14% (36,8 °C), dan SSE 0,0 °C. Kecepatan respons yang tinggi tersebut relevan untuk fase *brooding* awal ketika suhu kandang jauh di bawah *setpoint*.

Kemudian untuk pengujian keandalan sistem kontrol suhu berbasis PID, sistem mampu membawa suhu dari kondisi awal 32,8°C menuju *setpoint* 35°C dengan *rise time* $\approx 47,5$ detik, *overshoot* $\approx 4,57\%$, *settling time* ≈ 174 detik, dan *steady-state error* rata-rata mendekati nol ($\approx 0,003^\circ\text{C}$). Keempat parameter tersebut menunjukkan bahwa sistem memiliki respons transien yang relatif cepat dengan lonjakan yang masih dalam batas wajar, serta akurasi jangka panjang yang baik berkat komponen integral pada kontroler. Pada pengujian terdapat gangguan lingkungan pada rentang dini hari/pagi hari (teramati pada Sesi 1 sekitar jam ke-4,4–7 dan Sesi 2 pada 3,5 jam pertama) sempat menyebabkan osilasi suhu yang lebih besar, namun sistem PID terbukti mampu pulih dan kembali stabil tanpa kehilangan kestabilan kendali. Secara keseluruhan, sistem menunjukkan keandalan yang baik untuk pengendalian suhu jangka panjang, dengan lebih dari 97% data pada seluruh sesi berada dalam pita toleransi $\pm 2\%$ dari *setpoint*.

Sebaliknya, kontrol Logika Fuzzy dengan tiga himpunan *COLD* ($\leq 30^\circ\text{C}$, $\mu=1$), *WARM* ($30\text{--}40^\circ\text{C}$ transisi, 35°C $\mu=1$), dan *HOT* ($\geq 40^\circ\text{C}$, $\mu=1$) menghasilkan keluaran daya berupa persentase dimmer yang dihitung dengan pembobotan μ_{COLD} dan μ_{WARM} pada daerah $30\text{--}35^\circ\text{C}$. Berdasarkan data 24 jam (1.440 titik), sistem Fuzzy memiliki rentang suhu $33,63\text{--}35,33^\circ\text{C}$, rata-rata $34,878^\circ\text{C}$, simpangan baku $0,256^\circ\text{C}$, dan *Overshoot* rata-rata $0,848\%$ terhadap SP. SSE global $0,122^\circ\text{C}$ menunjukkan ketepatan yang baik meskipun tidak sekecil PID.

Perbedaan mendasar terletak pada filosofi pengendalian yaitu PID berusaha menghilangkan *error* secara agresif melalui aksi proporsional, integral, dan derivatif, sehingga *rise-time* cepat namun berpotensi *Overshoot* lebih besar. Logika Fuzzy mengurangi daya secara bertahap berdasarkan derajat keanggotaan, sehingga *Overshoot* sangat kecil ($<1\%$) dan konsumsi daya dimmer hanya $0\text{--}27,4\%$ dengan rata-rata $2,42\text{--}3,55\%$, jauh lebih hemat energi.

Hal ini sesuai dengan kebutuhan peternak skala kecil-menengah yang menggunakan lampu pijar sebagai pemanas.

Dengan demikian, pemilihan metode dapat disesuaikan dengan prioritas yaitu jika kecepatan mencapai *setpoint* menjadi prioritas utama, PID lebih unggul dan jika stabilitas tanpa lonjakan suhu dan efisiensi energi menjadi prioritas, Logika Fuzzy lebih unggul. Kombinasi hybrid, yaitu PID untuk fase pemanasan awal dan Fuzzy untuk fase pemeliharaan, dapat menjadi arah pengembangan selanjutnya.

Bab 5. Penutup

5.1 Simpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan pada sistem kontrol suhu kandang DOC berbasis ESP32 menggunakan metode PID (Ziegler-Nichols *autotune*) dan Logika Fuzzy, dapat ditarik beberapa kesimpulan sebagai berikut; Sensor DHT22 yang digunakan pada sistem menunjukkan tingkat akurasi yang baik, dengan persentase *error* terhadap termometer *wooden* GEA sebagai alat ukur pembandingan berada pada rentang 0,24% hingga 1,86%. Hasil ini menunjukkan bahwa DHT22 layak digunakan sebagai sensor utama pembacaan suhu pada sistem kontrol *brooder*. Proses *autotune* Ziegler-Nichols berhasil menghasilkan parameter PID secara otomatis melalui metode relay feedback, dengan nilai gain kritis (K_u) sebesar 57,296 dan periode ultimate (P_u) sebesar 112,328 detik, yang kemudian dikonversi menjadi $K_p = 34,377$, $K_i = 0,612$, dan $K_d = 482,694$. Parameter ini terbukti mampu mengendalikan suhu *brooder* pada tiga titik *setpoint* (32°C, 33,5°C, dan 35°C) dengan rise time 5–11 detik, *settling time* yang relatif konsisten pada 36–38 detik, dan *error* keadaan tunak yang sangat kecil hingga 0%. *Overshoot* yang dihasilkan cenderung meningkat seiring kenaikan *setpoint*, dari 2,39% pada 32–33,5°C menjadi 5,14% pada 35°C. Pengujian keandalan kontrol PID selama tiga sesi (± 8 jam, ± 13 jam, dan ± 3 jam) pada *setpoint* 35°C menunjukkan lebih dari 97% data berada dalam pita toleransi $\pm 2\%$ terhadap *setpoint*. Sistem sempat mengalami osilasi suhu yang lebih besar akibat gangguan lingkungan pada dini/pagi hari, namun mampu pulih dan kembali stabil tanpa kehilangan kestabilan kontrol, menunjukkan ketahanan sistem yang cukup baik terhadap gangguan jangka pendek. Pada sistem kontrol Logika Fuzzy yang diuji selama 24 jam (tiga sesi @ 8 jam) pada *setpoint* 35°C mampu mempertahankan suhu pada rentang 33,63°C–35,33°C dengan rata-rata keseluruhan 34,878°C, menghasilkan *steady-state error* global sebesar 0,122°C dan *Overshoot* rata-rata hanya 0,848% — jauh lebih kecil dibandingkan *Overshoot* kontrol PID pada *setpoint* yang sama (5,14%). Karakteristik ini berkaitan dengan mekanisme pengurangan daya secara bertahap serta penghentian daya pemanasan (0%) begitu suhu mencapai atau melewati *setpoint*.

Perbandingan langsung kedua metode pada *setpoint* 35°C menunjukkan adanya *trade-off* yang jelas yaitu kontrol PID unggul pada kecepatan respons (*rise time* 1,08 menit dan *settling time* 3,92 menit, dibandingkan logika Fuzzy yang membutuhkan 69,00 menit dan 27,00 menit) serta *steady-state error* yang sedikit lebih kecil ($\approx 0,046^{\circ}\text{C}$ berbanding $\approx 0,076^{\circ}\text{C}$). Sebaliknya, Logika Fuzzy unggul dalam kehalusan respons (*Overshoot* 0,43% berbanding 4,57%) dan tidak menunjukkan gejala osilasi akibat gangguan lingkungan yang teramati pada sesi PID (menit ke-394–460). Pengujian langsung terhadap perilaku DOC di dalam kandang membuktikan efektivitas praktis kedua sistem kontrol yaitu anak ayam yang semula berkumpul di satu titik (indikasi stres termal akibat suhu rendah) bergerak lebih bebas ke seluruh area kandang setelah suhu berhasil distabilkan mendekati *setpoint*, mengkonfirmasi bahwa kedua metode kontrol mampu menciptakan kondisi termal yang nyaman bagi DOC. Secara keseluruhan, sistem kontrol suhu berbasis ESP32 dengan aktuator AC *light* dimmer berhasil diimplementasikan dan diverifikasi dengan baik melalui pengujian data historis maupun pengujian langsung pada objek nyata (DOC). Pemilihan metode kontrol yang lebih unggul bergantung pada prioritas aplikasi. Untuk PID lebih sesuai jika kecepatan respons menjadi prioritas utama, sedangkan Logika Fuzzy lebih sesuai untuk aplikasi brooding DOC yang mengutamakan kestabilan dan minimnya lonjakan suhu.

5.2 Saran

Untuk pengembangan lebih lanjut, sistem ini dapat dikembangkan dengan menambahkan lebih banyak parameter masukan selain suhu, seperti kelembaban udara, intensitas cahaya, serta kadar amonia di dalam kandang sehingga sistem mampu mengambil keputusan yang lebih komprehensif terhadap kondisi lingkungan. Selain itu, metode kontrol dapat dikembangkan menjadi Fuzzy-PID untuk meningkatkan performa sistem pada kondisi lingkungan yang lebih dinamis. Penggunaan *platform* IOT juga dapat diperluas dengan penambahan fitur notifikasi otomatis apabila suhu kandang berada di luar batas aman. Dari sisi mekanik, desain kandang dapat disempurnakan agar distribusi panas lebih merata sehingga efisiensi pemanasan dapat ditingkatkan. Pengujian juga disarankan dilakukan dalam periode yang lebih panjang, misalnya selama masa pemeliharaan DOC secara penuh, sehingga diperoleh data yang lebih beragam dan mampu menggambarkan performa sistem dalam berbagai kondisi cuaca maupun perubahan lingkungan.

Daftar Pustaka

- [1] A. H. Fattah, R. Faridah, A. H. N. Amalia, dan Khaeruddin, “Pengaruh Pengaturan Suhu dan Kelembaban di Kandang Closed House Terhadap Performa Broiler,” *Musamus Journal of Livestock Science*, vol. 6, no. 1, pp. 12–20, 2023.
- [2] S. Shafiudin dan N. Kholis, “Sistem *Monitoring* dan Pengontrolan Temperatur pada Inkubator Penetas Telur Berbasis PID,” *Jurnal Teknik Elektro*, vol. 7, no. 1, 2018.
- [3] D. O. Aborisade dan S. Oladipo, “Poultry House Temperature Control Using Fuzzy-PID *Controller*,” *International Journal of Engineering Trends and Technology (IJETT)*, vol. 11, no. 6, pp. 310–314, 2014.
- [4] B. G. Caesario, E. Setiawan, dan R. Pramananda, “Sistem Pengendalian Suhu pada Kandang Ayam Broiler menggunakan PID *Controller*,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIK)*, vol. 7, no. 3, pp. 1336–1344, 2023.
- [5] H. A. Wijaya dan S. Hartati, “Implementasi Logika Fuzzy Dalam Sistem Pendingin Otomatis Kandang Ayam Broiler Closed House,” *Indonesian Journal of Electronics and Instrumentation Systems (IJEIS)*, vol. 14, no. 2, 2024.
- [6] R. R. Hidayah, S. Nurcahyo, dan D. Dewatama, “Implementasi Pengaturan Suhu Menggunakan Mikrokontroler ESP32,” *Journal of Mechanical and Electrical Technology (METROTECH)*, vol. 3, no. 3, pp. 117–124, 2024.
- [7] Masyhuda, M. Ula, dan Kurniawati, “Sistem *Monitoring* dan Kontrol Suhu Kelembaban dan Gas Amonia dengan Penerapan Fuzzy Tsukamoto untuk Kandang Ayam Tertutup Skala Kecil,” dalam *Prosiding SENASTIKA Universitas Malikussaleh*, 2024.

- [8] A. P. Aji dan O. A. Dhewa, "Chicken Cage Incubator Cooling Control System Using Fuzzy *Logic*," *Media of Computer Science (MCS)*, vol. 1, no. 1, pp. 17–30, 2024.
- [9] A. F. Trinaldi, A. K. Ningsih, dan Melina, "Sistem Kontrol dan *Monitoring* Suhu Kelembaban Kandang pada Peternakan Ayam Broiler dengan Metode Logika Fuzzy Mamdani Berbasis Internet of Things," dalam *Prosiding Seminar Nasional Sains dan Teknologi (PSNST)*, vol. 12, no. 1, 2024.
- [10] N. A. Fahila, S. A. Wibowo, dan F. X. Ariwibisono, "Implementasi Fuzzy Mamdani Pada Sistem Automasi Dan *Monitoring* Ayam Broiler Berbasis *Internet of Things* (IoT)," *Jurnal Mahasiswa Teknik Informatika (JATI)*, vol. 8, no. 2, pp. 1827–1834, 2024.
- [11] D. N. Praing, R. Y. Kalaway, dan R. T. Abineno, "Perancangan IoT Pengendalian Suhu Kandang Ayam Broiler Untuk Peningkatan Produktivitas Menggunakan Metode Fuzzy Mamdani," *Sustainable Agricultural Technology Innovation (SATI)*, pp. 74–84, 2025.
- [12] M. H. Tamzil, B. Indarsih, I. N. S. Jaya, dan N. K. D. Haryani, "Stres Pengangkutan pada Ternak Unggas, Pengaruh dan Upaya Penanggulangan," *Livestock and Animal Research*, vol. 20, no. 1, pp. 48–58, 2022.
- [13] I. Lahlouh, A. Elakkary, dan N. Sefiani, "Design and Implementation of State-PID Feedback *Controller* for Poultry House System: Application for Winter Climate," *Advances in Science, Technology and Engineering Systems Journal (ASTESJ)*, vol. 5, no. 1, pp. 135–141, 2020.
- [14] T. Hadyanto dan M. F. Amrullah, "Sistem *Monitoring* Suhu dan Kelembaban pada Kandang Anak Ayam Broiler Berbasis Internet of Things," *Jurnal Sistem Informasi Politeknik Bisnis Kaltara*, 2021.
- [15] A. B. S. Pratama, Y. Z. Maulana, dan H. Pujiharsono, "Sistem Pengendali Suhu pada Kandang Ayam Broiler Berbasis Kontrol PID," *Journal of Telecommunication, Electronics, and Control Engineering (JTECE)*, vol. 6, no. 2, pp. 117–124, 2024.

- [16] P. A. Topan, A. Satriawansyah, I. Darmawan, dan T. Anddriani, "Rancang Bangun Pengendali Suhu dan Kelembaban Ruangan Budidaya Jamur Tiram Menggunakan ESP32 dan Sensor SHT22," *HEXAGON (Jurnal Teknik dan Sains)*, vol. 5, no. 2, 2024.
- [17] M. C. A. Prabowo, I. Sayekti, S. Astuti, S. T. Nursaputro, dan Supriyati, "Development of an IoT-Based Egg Incubator with PID Control System and Mobile Application," *International Journal on Informatics Visualization (JOIV)*, vol. 8, no. 1, pp. 465–472, 2024.
- [18] R. Adji Fadlani, "Implementasi Metode PID dalam Sistem Kontrol Temperatur Pada Ruang Produksi 'Clean Room' Berbasis IoT," *Mars: Jurnal Teknik Mesin, Industri, Elektro dan Ilmu Komputer*, vol. 3, no. 1, pp. 77–93, 2025.
- [19] Z. I. Awwal, A. Murtono, dan S. Adhisuwignjo, "Implementasi Kontrol PID untuk Pengaturan Suhu dan Kelembaban pada Lemari Pengering Pakaian," *Kohesi: Jurnal Multidisiplin Saintek*, vol. 2, no. 8, pp. 1–15, 2024.
- [20] A. D. Attaqiroh, A. R. Chaidir, dan Sumardi, "Sistem Pengendalian Suhu pada Inkubator Fermentasi Tempe dengan Metode Proportional Integral Derivative (PID) Secara Digital," *MITOR: Jurnal Teknik Elektro*, 2023.
- [21] R. Firdaus dan W. Zulfikar, "Pengontrol Suhu Ruangan menggunakan Metode PID," *TELEKONTRAN*, vol. 4, no. 2, pp. 71–77, 2016.
- [22] A. Hendriawan, "Inkubator Telur Ayam Menggunakan Lampu DC Dengan Kontrol PID," *TELEKONTRAN*, vol. 4, no. 2, pp. 78–85, 2016.
- [23] A. K. T. Laksmi, T. Andromeda, dan A. Triwiyatno, "Pengendalian Suhu Menggunakan Kontroler PID pada Prototipe Mesin Pengering Fluidisasi Gabah," *Transient*, vol. 7, no. 2, 2018.
- [24] R. Muhammadi, Stasiswaty, dan Subardin, "Implementasi Logika Fuzzy Dalam Rancang Bangun Sistem Otomatisasi Ruangan Pemeliharaan Ayam Berbasis Internet of Things," *ANIMATOR*, vol. 2, no. 2, pp. 18–26, 2024.

- [25] K. Senawong, S. Winitchai, dan T. Radpukdee, "Humidity and Temperature Control in an Evaporative Cooling System of a Poultry House," *KKU Engineering Journal*, vol. 39, no. 1, pp. 95–111, 2012.
- [26] Imam dan N. Abdillah, "Sistem Pemantau Suhu dan Kelembapan pada Kandang Anak Ayam Berbasis Internet of Things," *Jurnal Rekayasa Sistem Informasi dan Teknologi (JRSIT)*, vol. 2, no. 1, 2024.
- [27] M. R. Wirajaya dan S. Abdussamad, "Rancang Bangun Mesin Penetas Telur Otomatis Menggunakan Mikrokontroler Arduino Uno," *Jambura Journal of Electrical and Electronics Engineering (JEEEE)*, vol. 2, no. 1, 2020.
- [28] Pasnur, A. Hendra, M. S. Hadis, dan Ardiansa, "Sistem Kontrol Suhu Ideal Kandang Ayam Broiler Berbasis Teknologi *Internet of Things* (IoT)," dalam *Prosiding Seminar Nasional Komunikasi dan Informatika #3*, pp. 79–82, 2019.
- [29] T. Hadyanto dan M. F. Amrullah, "Sistem *Monitoring* Suhu dan Kelembapan pada Kandang Anak Ayam Broiler Berbasis Internet of Things," *Jurnal Sistem Informasi Politeknik Bisnis Kaltara*, 2022.
- [30] S. K. Risandriya, E. M. Lubis, M. Syahroni, I. Aryeni, dan A. Setyawan, "Pengendalian Suhu Terhadap Proses Penetasan Telur Ayam dengan Kontrol Logika Fuzzy Menggunakan IoT sebagai *Monitoring*," *Journal of Applied Electrical Engineering*, vol. 8, no. 1, pp. 51–57, 2024.
- [31] Ziegler, J. G., & Nichols, N. B. (1942). Optimum settings for automatic controllers. *Transactions of the ASME*, 64, 759–768.
- [32] Ogata, K. (2010). *Modern control engineering* (5th ed.). Prentice Hall.
- [33] Åström, K. J., & Hägglund, T. (1995). *PID controllers: Theory, design, and tuning*. Instrument Society of America.
- [34] University of Michigan. (n.d.). CTM PID tutorial: Effect of Kp, Ki, Kd on system response. University of Maryland. <https://terpconnect.umd.edu/~sandborn/courses/ENME462/Studio7/Part1/PID.htm>

[35] Wevolver. (n.d.). PID explained: Theory, *tuning* and implementation of PID *controllers*. <https://www.wevolver.com/article/pid-explained-theory-tuning-and-implementation-of-pid-controllers>

[36] Rao, C. V. N., & Rao, T. V. (2020). Ziegler–Nichols *tuning* method: Understanding the PID *controller*. *Resonance*, 25(10), 1385–1397. <https://www.ias.ac.in/public/Volumes/reso/025/10/1385-1397.pdf>

[37] ResearchGate. (2018). PID *tuning* using Ziegler-Nichols method for DC motor. <https://www.researchgate.net/publication/327816077>

[38] Wikipedia. (n.d.). Ziegler–Nichols method. In Wikipedia. https://en.wikipedia.org/wiki/Ziegler%E2%80%93Nichols_method

[39] Rifqi Mulyawan. “IOT: Pengertian, Apa itu Internet of Things? Definisi Menurut Ahli, Sejarah, Prinsip, Kelebihan dan Kekurangannya!”. Internet: <https://rifqimulyawan.com/blog/pengertian-iot-internet-of-things/>, [Okt. 22, 2023]

[40] Krysna Yudha Maulana. “Apa Itu ESP32, Salah Satu Modul Wi-Fi Populer”. Internet: <https://www.anakteknik.co.id/krysnayudhamaulana/articles/apa-itu-esp32-salah-satu-modul-wi-fi-poppuler>, Des. 30, 2022 [Okt. 22, 2023]

[41] <https://dte.telkomuniversity.ac.id/peran-sensor-dht22-dalam-Monitoring-lingkungan-berbasis-iot/>

[42] <https://www.kmtech.id/post/mengenal-perangkat-lunak-arduino-ide>

[43] <https://chickin.id/blog/berapa-suhu-ideal-untuk-ayam-doc/>

[44] <https://robotdyn.com/ac-light-dimmer-module-1-channel-3-3v-5v-logic-ac-50-60hz-220v-110v>

[45] Ross, T. J. (2010). Fuzzy Logic with Engineering Applications (3rd ed.). John Wiley & Sons.

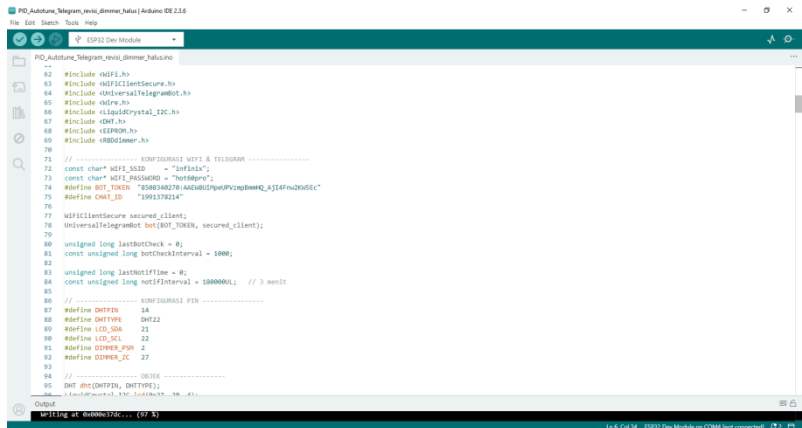
Lampiran

Lampiran 1

- 1) Pengujian kontrol PID : https://drive.google.com/file/d/1-ljDPgWKu7fr0NwQ_XqGbcI97bRnLc8_/view?usp=drive_link
- 2) Pengujian kontrol logika fuzzy:
https://drive.google.com/file/d/17kzbYNMRrT8FgDMSSxtDnBVnFOTtjMGc/view?usp=drive_link
- 3) Pengujian sistem kontrol suhu pada anak ayam :
https://drive.google.com/file/d/1GcoBXQt4AdvSj9khj5pBDM8iqIQQyzev/view?usp=drive_link

Lampiran 2

1) Code PID



```
PC2_Autotune_Telegram_versi_dimmer_naha | Arduino IDE 2.1.6
File Edit Search Tools Help
SPI2 Dev Module
PC2_Autotune_Telegram_versi_dimmer_naha.c
--
62 #include <SPI2.h>
63 #include <WiFiClientSecure.h>
64 #include <UniversalTelegramBot.h>
65 #include <Arduino.h>
66 #include <LiquidCrystal_I2C.h>
67 #include <GSM.h>
68 #include <EEPROM.h>
69 #include <DS16C29.h>
70
71 // ----- KONTROL Suhu B TELEGRAM -----
72 const char* WIFI_SSID = "Infina";
73 const char* WIFI_PASSWORD = "not00pr";
74 #define BOT_TOKEN "5983467914408010960:VjmgdwwQ_AJIfnuDMS0E"
75 #define CHAT_ID "1091378114"
76
77 WiFiClientSecure secured_client;
78 UniversalTelegramBot bot(BOT_TOKEN, secured_client);
79
80 unsigned long lastBotCheck = 0;
81 const unsigned long botCheckInterval = 1000;
82
83 unsigned long lastHotline = 0;
84 const unsigned long voiceInterval = 1800000; // 3 menit
85
86 // ----- KONTROL Suhu PID -----
87 #define DHTPIN 14
88 #define DHTTYPE DHT22
89 #define I2C_SDA 21
90 #define I2C_SCL 22
91 #define DHTWRK_PIN 2
92 #define DHTWRK_IC 27
93
94 // ----- OTHER -----
95 DHT dht(DHTPIN, DHTTYPE);
96 DS16C29 ds16c29(A4, A5);
97
98 Output
Writing at 0x00017d0... (97 B)
Ln 4, Col 34 SPI2 Dev Module on C:\M4 (not connected) 21 81
```

```
File Edit Search Tools Help
ESP8266 Dev Module
PIO_Autotune_Telegram_vcsil_gimmar_hakuo | Arduino IDE 2.3.6

93
94 // ----- DHTXX -----
95 DHT dht(DHTPIN, DHTTYPE);
96 LiquidCrystal_I2C lcd(I2C1, 20, 4);
97 dimmerLamp dimmer(SDIMER_PIN, SDIMER_SIZE);
98
99 // ----- PARAMETER PID -----
100 double setpoint = 32.0;
101 double kp = 0, Ki = 0, Kd = 0; // errorSum = 0, errorDelta = 0;
102 double errorSum = 0, errorLast = 0, errorSum = 0, errorDelta = 0;
103 double outputPID = 0; // INTENSITAS dimmer yang diminta PID (0-100%),
    langsung dikirim
104 unsigned long lastPIDTime = 0;
105 const unsigned long pidInterval = 1000;
106
107 const double outputMin = 0;
108 const double outputMax = 100;
109
110 // ----- OUTPUT DIMMER: PENUN -> MEREDUP + ANTI-FLICKER -----
111 double outputTarget = 0; // target intensitas dimmer (100% saat
    jam dari setpoint, PID saat mendekati)
112 double outputSmooth = 0; // intensitas setelah dibalaskan (low-pass
    filter), nilai yang dikirim ke dimmer
113 int outputTerakhir = -1; // nilai terakhir yang benar-benar dikirim
    ke dimmer.setPower()
114
115 const double bandWidethSetpoint = 3.0; // (C) mulai meredup jika suhu
    sudah sejauh ini menuju setpoint
116 const double filterAlphaDimmer = 0.12; // 0..1, makin kecil = transisi
    makin halus/lembut (kurangi flicker)
117 unsigned long lastDimmerSmoothTime = 0;
118 const unsigned long dimmerSmoothInterval = 50; // update halus tiap 50ms,
    independen dari siklus PID 1 detik
119
120
Output
Writing at 0x00073c... (97 B)
Lo & Cat 34 ESP8266 Dev Module on COM4 (not connected)

File Edit Search Tools Help
ESP8266 Dev Module
PIO_Autotune_Telegram_vcsil_gimmar_hakuo | Arduino IDE 2.3.6

120 float suhuTerakhir = 0;
121
122 // ----- PARAMETER AUTOTUNE -----
123 // dua level INTENSITAS terang/redup (bukan on/off penuh)
124 bool autotuneMode = false;
125 bool relayState = false;
126 double intensitasTinggi = 80.0;
127 double intensitasRendah = 15.0;
128 double tuneSetpoint = 32.0;
129
130 double peakHigh = 1000, peakLow = 1000;
131 unsigned long lastSetpointTime = 0;
132 unsigned long peakTimeStamp[30];
133 int peakCount = 0;
134 double Ki = 0, Pn = 0;
135 bool autotuneFinished = false;
136
137 // ----- KALIBRASI SUHU (SUDERHANA - OFFSET) -----
138 double suhuOffset = 0.0; // offset koreksi, default 0 = tanpa koreksi
139 bool suhuKalibrasiTersedia = false;
140
141 // ----- EEPROM -----
142 #define EEPROM_SIZE 64
143 #define ADDR_IP 0
144 #define ADDR_KI 8
145 #define ADDR_KD 16
146 #define ADDR_SUHU_FLAG 24
147 #define ADDR_SUHU_OFFSET 28
148
149 // ----- SETUP -----
150 // -----
151 void setup() {
152   Serial.begin(115200);
153 }
```

The image displays the Arduino IDE interface with two main windows. The top window, titled 'PIDAutotune_Telegram_v01c_dimmer_halusino.ino', shows the source code for a C++ program designed for an ESP32. The code includes a setup function for serial communication and pin configuration, a loop function for reading sensor data and controlling a dimmer, and an autotuning function for PID control. The bottom window, titled 'Output', shows the compilation and upload process, indicating that the code was successfully compiled and uploaded to the ESP32 Dev Module. The status bar at the bottom of the IDE shows 'Uploading to ESP32 Dev Module'.

```

151 // void setup() {
152   Serial.begin(115200);
153
154   dht.begin();
155
156   Wire.begin(I2C_SDA, I2C_SCL);
157   I2C.begin();
158   I2C.begin();
159   I2C.begin();
160
161   dimmer.begin(DIMMER_ADDR, DIMMER_ADDR);
162   dimmer.setPower(0);
163   outputTarget = 0;
164   outputSmooth = 0;
165   outputTargetInterval = 0;
166
167   EEPROM.begin(EEPROM_SIZE);
168   loadFromEEPROM();
169   loadInitialValuesFromEEPROM();
170
171   I2C.setCursor(0, 0);
172   I2C.print("Sistem PID Dimmer-");
173   I2C.setCursor(0, 1);
174   I2C.print("Menyambung WiFi...");
175
176   connectWiFi();
177   secureClient.startSecure();
178
179   bot.sendMessage(CHAT_ID, "Sistem brooder online. Ulatkatik /start untuk daftar perintah.", "");
180
181   I2C.clear();
182 }
183
184 // =====
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

PID_Autotune_Magnet_resit_dimmer_huhu | Arduino IDE 2.8
File Edit Sketch Tools Help

ESP8266 Dev Module

PID_Autotune_Magnet_resit_dimmer_huhu.ino
212
213 if (autotunehulu) {
214     resetAutotune(suhu);
215 } else {
216     readPID(suhu);
217 }
218
219 updateDimmerHulu();
220
221 tampilKiri.CD(suhu);
222 kiriSerial(suhu);
223 kiriInfoHuluKiri(suhu);
224
225 delay(50);
226 }
227
228 // =====
229 // KALIBRASI SUDU SEDIKITA (SATU TITIK OFFSET)
230 // =====
231
232 // Penghitung & langsung menerapkan offset: baca sensor SEKARANG,
233 // bandingkan dengan nilai referensi (termometer standar) yang diberikan.
234 void kalibrasiSuhu(double referensi) {
235     float bacaan = dtc.readTemperature();
236     if (isnan(bacaan)) {
237         bot.sendMessage(CHAT_ID, "Gagal membaca sensor DHT22, coba lagi.", "");
238         return;
239     }
240     suhuOffset = referensi - bacaan;
241     suhuKalibrasiTersebut = true;
242     saveKalibrasiSuhuToEEPROM();
243     String pesan = "Kalibrasi suhu diterapkan!";
244     pesan += "Bacaan sensor asli : " + String(bacaan, 2) + " C\n";
245
246     Output
247     Writing at 0x000070c... (97 B)
248     In 6, Out 34 ESP8266 Dev Module on COM4 (not connected)
249
250 PID_Autotune_Magnet_resit_dimmer_huhu | Arduino IDE 2.8
251 File Edit Sketch Tools Help
252
253 ESP8266 Dev Module
254
255 PID_Autotune_Magnet_resit_dimmer_huhu.ino
256
257 String pesan = "Kalibrasi suhu diterapkan!";
258 pesan += "Bacaan sensor asli : " + String(bacaan, 2) + " C\n";
259 pesan += "Referensi : " + String(referensi, 2) + " C\n";
260 pesan += "Offset : " + String(suhuOffset, 2) + " C\n";
261 pesan += "Nilai sekarang, suhu = bacaan + offset.";
262 bot.sendMessage(CHAT_ID, pesan, "");
263 Serial.println(pesan);
264 }
265
266 // Menghapus offset, kembali ke bacaan asli sensor.
267 void resetKalibrasiSuhu() {
268     suhuOffset = 0.0;
269     suhuKalibrasiTersebut = false;
270     saveKalibrasiSuhuToEEPROM();
271     bot.sendMessage(CHAT_ID, "Kalibrasi suhu direset. Menakal bacaan sensor asli
272     (tanpa koreksi).", "");
273 }
274
275 // =====
276 // WiFi
277 // =====
278 void connectWiFi() {
279     WiFi.mode(WIFI_STA);
280     WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
281     int percobaan = 0;
282     while (WiFi.status() != WL_CONNECTED && percobaan < 30) {
283         delay(100);
284         Serial.print(".");
285         percobaan++;
286     }
287     if (WiFi.status() == WL_CONNECTED) {
288         Serial.println("WiFi tersambung : " + WiFi.localIP().toString());
289     } else {
290         Serial.println("Gagal tersambung ke WiFi, akan dicoba lagi.");
291     }
292 }
293
294 Output
295 Writing at 0x000070c... (97 B)
296 In 6, Out 34 ESP8266 Dev Module on COM4 (not connected)

```

```
File Edit Sketch Tools Help
PID_Autotune_Tekang_revisi_dimmer_huhu | Arduino IDE 2.3.6
ESP8266 Dev Module
274 if (WiFi.status() == WL_CONNECTED) {
275   Serial.println("WiFiIP tersambung: " + WiFi.localIP().toString());
276 } else {
277   Serial.println("Tidak tersambung ke WiFi, akan dicoba lagi.");
278 }
279 }
280
281 // =====
282 // TELEGRAM BOT
283 // =====
284 void cekPesanTelegram() {
285   int jumlahPesanBaru = bot.getUpdates(bot.last_message_received + 1);
286   while (jumlahPesanBaru) {
287     for (int i = 0; i < jumlahPesanBaru; i++) {
288       String chat_id = bot.messages[i].chat_id;
289       if (chat_id != String(Chat_ID)) continue;
290
291       String teks = bot.messages[i].text;
292       prosesPerintahTelegram(teks, chat_id);
293     }
294     jumlahPesanBaru = bot.getUpdates(bot.last_message_received + 1);
295   }
296 }
297
298 void prosesPerintahTelegram(String teks, String chat_id) {
299   teks.trim();
300
301   if (teks == "/start") {
302     String pesan = "Halo Perintah Bot Berhasil!";
303     pesan += "/status - lihat suhu & status saat ini!";
304     pesan += "/set cangkang - ubah setpoint, contoh: /set 33.5!";
305     pesan += "/autotune - mulai autotune PID!";
306     pesan += "/kalibrasi suhu <referensi> - kalibrasi suhu, contoh: /kalibrasi_suhu 33.2!";
307     pesan += "/kalibrasi_suhu_reset - hapus koreksi suhu!";
308
309     pesan = "/kalibrasi_suhu <referensi> - kalibrasi suhu, contoh: /kalibrasi_suhu 33.2!";
310     pesan += "/kalibrasi_suhu_reset - hapus koreksi suhu!";
311     pesan += "/tuning - lihat nilai kp, ki, kd!";
312     bot.sendMessage(chat_id, pesan, "");
313   }
314   else if (teks == "/status") {
315     String pesan = "Suhu saat ini: " + String(suhuTerakhir, 1) + " C!";
316     pesan += "Setpoint: " + String(setpoint, 1) + " C!";
317     pesan += "Mode: " + String(autotuneMode, 1) + "AUTOTUNE: " + String(autotuneMode, 1) + "!";
318     pesan += "Intensitas dimmer: " + String((int)round(outputSmooth)) + " %!";
319     pesan += "Kalibrasi suhu: " + String(suhuKalibrasiTerakhir, 1) + "C!";
320     offset = String(suhuOffset, 2) + " C";
321     bot.sendMessage(chat_id, pesan, "");
322   }
323   else if (teks.startsWith("/set")) {
324     String nilaiStr = teks.substring(4);
325     nilaiStr.trim();
326     if (nilaiStr.length() > 0) {
327       double nilaiBaru = nilaiStr.toFloat();
328       if (nilaiBaru > 0 && nilaiBaru < 50) {
329         setpoint = nilaiBaru;
330         bot.sendMessage(chat_id, "Setpoint diubah menjadi " + String(setpoint, 1) + " C, """);
331       } else {
332         bot.sendMessage(chat_id, "Nilai tidak valid. Gunakan rentang 0-50 C.", "");
333       }
334     }
335   }
336   else if (teks == "/autotune") {
337     if (autotuneMode) {
338
339       Writing at 0x00007f0c... (97.5)
340       In 6, Out 34 ESP8266 Dev Module on COM4 (not connected)
341
342 File Edit Sketch Tools Help
343 PID_Autotune_Tekang_revisi_dimmer_huhu | Arduino IDE 2.3.6
344 ESP8266 Dev Module
345
346 300     pesan = "/kalibrasi_suhu <referensi> - kalibrasi suhu, contoh: /kalibrasi_suhu 33.2!";
347 307     pesan += "/kalibrasi_suhu_reset - hapus koreksi suhu!";
348 308     pesan += "/kalibrasi_suhu_status - lihat offset aktif!";
349 309     pesan += "/tuning - lihat nilai kp, ki, kd!";
350 310     bot.sendMessage(chat_id, pesan, "");
351 311   }
352 312   else if (teks == "/status") {
353 313     String pesan = "Suhu saat ini: " + String(suhuTerakhir, 1) + " C!";
354 314     pesan += "Setpoint: " + String(setpoint, 1) + " C!";
355 315     pesan += "Mode: " + String(autotuneMode, 1) + "AUTOTUNE: " + String(autotuneMode, 1) + "!";
356 316     pesan += "Intensitas dimmer: " + String((int)round(outputSmooth)) + " %!";
357 317     pesan += "Kalibrasi suhu: " + String(suhuKalibrasiTerakhir, 1) + "C!";
358 318     offset = String(suhuOffset, 2) + " C";
359 319     bot.sendMessage(chat_id, pesan, "");
360 320   }
361 321   else if (teks.startsWith("/set")) {
362 322     String nilaiStr = teks.substring(4);
363 323     nilaiStr.trim();
364 324     if (nilaiStr.length() > 0) {
365 325       double nilaiBaru = nilaiStr.toFloat();
366 326       if (nilaiBaru > 0 && nilaiBaru < 50) {
367 327         setpoint = nilaiBaru;
368 328         bot.sendMessage(chat_id, "Setpoint diubah menjadi " + String(setpoint, 1) + " C, """);
369 329       } else {
370 330         bot.sendMessage(chat_id, "Nilai tidak valid. Gunakan rentang 0-50 C.", "");
371 331       }
372 332     }
373 333   }
374 334   else if (teks == "/autotune") {
375 335     if (autotuneMode) {
376
377       Writing at 0x00007f0c... (97.5)
378       In 6, Out 34 ESP8266 Dev Module on COM4 (not connected)
```

```

PID_Autotune_Megam_versi_dimmer_huhu | Arduino IDE 2.3.6
File Edit Sketch Tools Help

PID_Autotune_Megam_versi_dimmer_huhu.ino
232 }
233 }
234 }
235 else if (teks == "/autotune") {
236   if (autotuneMode) {
237     startAutotune();
238     bot.sendMessage(chat_id, "Autotune dimulai. Kamu akan diberi tahu saat selesai.", "");
239   } else {
240     bot.sendMessage(chat_id, "Autotune sedang berjalan.", "");
241   }
242 }
243 else if (teks == "/kalibrasi_suhu_reset") {
244   resetKalibrasiSuhu();
245 }
246 else if (teks == "/kalibrasi_suhu_status") {
247   String pesan = "Status kalibrasi suhu:\n";
248   pesan += "Offset: " + String(subKalibrasiTersebut * "ya" : "tidak") + "\n";
249   pesan += "Offset saat ini: " + String(suhuOffset, 2) + " C";
250   bot.sendMessage(chat_id, pesan, "");
251 }
252 else if (teks.startsWith("/kalibrasi_suhu")) {
253   String nilaiStr = teks.substring(String("kalibrasi_suhu").length());
254   nilaiStr.trim();
255   if (nilaiStr.length() > 0) {
256     double referensi = nilaiStr.toFloat();
257     kalibrasiSuhu(referensi);
258   } else {
259     bot.sendMessage(chat_id, "Format salah. Contoh: /kalibrasi_suhu 33.2\n(33.2 = suhu sebenarnya dari termometer standar saat ini)", "");
260   }
261 }
262 else if (teks == "/tuning") {
263   String pesan = "Parameter PID saat ini:\n";
264   pesan += "Kp = " + String(Kp, 3) + "\n";
265   pesan += "Ki = " + String(Ki, 3) + "\n";
266   pesan += "Kd = " + String(Kd, 3);
267   bot.sendMessage(chat_id, pesan, "");
268 }
269 else {
270   bot.sendMessage(chat_id, "Perintah tidak dikenali. Ketik /start untuk melihat daftar perintah.", "");
271 }
272 }
273
274 void kirimNotifikasiSuhu(float suhu) {
275   if (millis() - lastNotifTime > notifInterval) {
276     lastNotifTime = millis();
277     String pesan = "Status Buzzer (update tiap 3 menit):\n";
278     pesan += "Suhu: " + String(suhu, 1) + " C\n";
279     pesan += "Setpoint: " + String(setpoint, 1) + " C\n";
280     pesan += "Mode: " + String(autotuneMode ? "AUTOTUNE" : "PID") + "\n";
281     pesan += "Intensitas dimmer: " + String((int)round(outputSmooth) * " N";
282     bot.sendMessage(chat_id, pesan, "");
283   }
284 }
285
286 // =====
287 // FUNGSI PID (output langsung berupa INTENSITAS dimmer, 0-100%)
288 // =====
289 void runPID(float suhu) {
290   unsigned long now = millis();
291   if (now - lastPIDTime < pidInterval) return;
292   double dt = (now - lastPIDTime) / 1000.0;
293   lastPIDTime = now;
294   errorNow = setpoint - suhu;
295 }
296
Output
Writing at 0x00007d6... (97.5)
In 6, Col 34 ESP8266 Dev Module on COM4 (not connected)

```

```

PID_Autotune_Megam_versi_dimmer_huhu | Arduino IDE 2.3.6
File Edit Sketch Tools Help

PID_Autotune_Megam_versi_dimmer_huhu.ino
363 String pesan = "Parameter PID saat ini:\n";
364 pesan += "Kp = " + String(Kp, 3) + "\n";
365 pesan += "Ki = " + String(Ki, 3) + "\n";
366 pesan += "Kd = " + String(Kd, 3);
367 bot.sendMessage(chat_id, pesan, "");
368 }
369 else {
370   bot.sendMessage(chat_id, "Perintah tidak dikenali. Ketik /start untuk melihat daftar perintah.", "");
371 }
372 }
373
374 void kirimNotifikasiSuhu(float suhu) {
375   if (millis() - lastNotifTime > notifInterval) {
376     lastNotifTime = millis();
377     String pesan = "Status Buzzer (update tiap 3 menit):\n";
378     pesan += "Suhu: " + String(suhu, 1) + " C\n";
379     pesan += "Setpoint: " + String(setpoint, 1) + " C\n";
380     pesan += "Mode: " + String(autotuneMode ? "AUTOTUNE" : "PID") + "\n";
381     pesan += "Intensitas dimmer: " + String((int)round(outputSmooth) * " N";
382     bot.sendMessage(chat_id, pesan, "");
383   }
384 }
385
386 // =====
387 // FUNGSI PID (output langsung berupa INTENSITAS dimmer, 0-100%)
388 // =====
389 void runPID(float suhu) {
390   unsigned long now = millis();
391   if (now - lastPIDTime < pidInterval) return;
392   double dt = (now - lastPIDTime) / 1000.0;
393   lastPIDTime = now;
394   errorNow = setpoint - suhu;
395 }
396
Output
Writing at 0x00007d6... (97.5)
In 6, Col 34 ESP8266 Dev Module on COM4 (not connected)

```

```

PIC_Autotune_Magnum_versi_dimmer_hulu | Arduino IDE 2.3.6
File Edit Sketch Tools Help

PIC_Autotune_Magnum_versi_dimmer_hulu
-----
384 lastPIDTime = now;
385
386 errorMax = setpoint - 100;
387 errorSum += errorMax * dt;
388 errorDelta = (errorMax - errorLast) / dt;
389
390 double maxIntegral = outputMax / (KI * 0.0001);
391 errorSum = constrain(errorSum, -maxIntegral, maxIntegral);
392
393 outputPID = (Kp * errorMax) + (KI * errorSum) + (Kd * errorDelta);
394 outputPID = constrain(outputPID, outputMin, outputMax);
395
396 errorLast = errorMax;
397
398 // ----- Menyala penutup selama masih jauh di bawah setpoint, lalu meredup -----
399 if (errorMax > bandDekatSetpoint) {
400 // suhu masih jauh di bawah setpoint -> dimmer menyala penuh (100%)
401 outputTarget = outputMax;
402 errorSum = 0; // anti-silang: jangan biarkan integral menumpuk saat masih
403 full power
404 } else {
405 // sudah masuk zona mendekati/di atas setpoint -> PID yang atur redupnya
406 outputTarget = outputPID;
407 }
408 // outputTarget di atas TMAX langsung dikirim ke dimmer; updateDimmerLast()
409 // yang akan meng-haluskan transisinya secara bertahap tiap siklus.
410 }
411
412 // =====
413 // MEMULAI SETPOINT DIMMER (MIDI-PIEXTER)
414 // Meng-ramp outputSmooth menuju outputTarget sedikit demi sedikit tiap
415 // 50ms (low-pass filter), dan hanya memanggil dimmer.setPower() saat
416 // nilai halatnya benar-benar berubah, supaya tidak ada ledakan daya
417 // mendadak yang menyebabkan kedip pada lampu.
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
29
```

```

PID_Autotune_Magnum_versi_dimmer_hulu | Arduino IDE 2.3.8
File Edit Sketch Tools Help

ESP32 Dev Module

PID_Autotune_Magnum_versi_dimmer_huluino
657 lastSwitchTime = millis();
658
659 dimmer.setPower((int)intensitasTinggi);
660
661
662 void runAutotune(float suhu) {
663   if (autotuneFinished) return;
664
665   if (relayState && suhu > tuneSetpoint) {
666     relayState = false;
667     dimmer.setPower((int)intensitasRendah);
668     catatMaknaluPeak();
669   } else if ((relayState && suhu <= tuneSetpoint) {
670     relayState = true;
671     dimmer.setPower((int)intensitasTinggi);
672     catatMaknaluPeak();
673   }
674
675   if (suhu > peakHigh) peakHigh = suhu;
676   if (suhu < peakLow) peakLow = suhu;
677
678   if (peakCount >= 5) {
679     hitungIgarWichoi();
680   }
681 }
682
683 void catatMaknaluPeak() {
684   unsigned long now = millis();
685   if (peakCount < 10) {
686     peakTimestamps[peakCount] = now;
687   }
688   peakCount++;
689   lastSwitchTime = now;
690 }
691
Output
Writing at 0x00007f0... (97.5%)
In 6, Col 34 ESP32 Dev Module on COM4 (not connected)

PID_Autotune_Magnum_versi_dimmer_hulu | Arduino IDE 2.3.8
File Edit Sketch Tools Help

ESP32 Dev Module

PID_Autotune_Magnum_versi_dimmer_huluino
692 void hitungIgarWichoi() {
693   unsigned long totalPeriod = 0;
694   int jumlahSiklus = 0;
695   for (int i = 2; i < peakCount && i < 10; i += 2) {
696     totalPeriod += (peakTimestamps[i] - peakTimestamps[i - 2]);
697     jumlahSiklus++;
698   }
699   if (jumlahSiklus == 0) return;
700
701   Pu = (totalPeriod / (double)jumlahSiklus) / 1000.0;
702
703   double a = (intensitasTinggi - intensitasRendah) / 2.0;
704   double b = (peakHigh - peakLow) / 2.0;
705   if (b <= 0) b = 0.01;
706
707   Ku = (4.0 * a) / (PI * b);
708
709   Kp = 0.5 * Ku;
710   Ki = 1.2 * Ku / Pu;
711   Kd = 0.075 * Ku * Pu;
712
713   saveTuningToEEPROM();
714
715   autotuneMode = false;
716   autotuneFinished = true;
717
718   // reset state dimmer & PID supaya mulai bersih (tampa limpasan daya) saat
719   // mengemil silih
720   outputPID = 0;
721   outputTarget = 0;
722   outputSmooth = 0;
723   outputFaktorInterakhir = -1;
724   errorSum = 0;
725   errorLast = 0;
726   lastSwitchTime = millis();
727 }
728
Output
Writing at 0x00007f0... (97.5%)
In 6, Col 34 ESP32 Dev Module on COM4 (not connected)

```

```

File Edit Sketch Tools Help
ESP8266 Dev Module
PIO_Autotune_Megam_vcs1_glimmer_haha | Arduino IDE 2.3.6

525 lastIPDTime = millis();
526 dimer.setPower(0);
527
528 Serial.println(">>> AUTOTUNE SELESAI <<<");
529 Serial.print("Kp = "); Serial.print(Kp);
530 Serial.print("Pu = "); Serial.print(Pu);
531 Serial.print("Kp = "); Serial.print(Kp);
532 Serial.print("Kd = "); Serial.print(Kd);
533 Serial.print("Kd = "); Serial.print(Kd);
534
535 String pesan = "Autotune selesai!\n";
536 pesan += "Kp = " + String(Kp, 3) + "\n";
537 pesan += "Pu = " + String(Pu, 3) + " detik\n";
538 pesan += "Kp = " + String(Kp, 3) + "\n";
539 pesan += "Kd = " + String(Kd, 3) + "\n";
540 pesan += "Kd = " + String(Kd, 3);
541 bot.sendMessage(Chat_ID, pesan, "");
542
543 lcd.clear();
544 lcd.setCursor(0, 0);
545 lcd.print("Autotune Selesai!");
546 lcd.setCursor(0, 1);
547 lcd.print("Kp:"); lcd.print(Kp, 3);
548 lcd.setCursor(0, 2);
549 lcd.print("Kp:"); lcd.print(Kd, 3);
550 lcd.setCursor(0, 3);
551 lcd.print("Kd:"); lcd.print(Kd, 3);
552 delay(1000);
553 lcd.clear();
554 }
555
556 // =====
557 // ESPROM
558 // =====
559 void saveFirmwareToEPROM() {
560   EEPROM.put(ADDR_EP, Kp);
561   EEPROM.put(ADDR_KI, Kd);
562   EEPROM.put(ADDR_KD, Kd);
563   EEPROM.commit();
564 }
565
566 void loadFirmwareFromEPROM() {
567   EEPROM.get(ADDR_EP, Kp);
568   EEPROM.get(ADDR_KI, Kd);
569   EEPROM.get(ADDR_KD, Kd);
570 }
571
572 if (isnan(Kp) || Kp < 0 || Kp > 1000) Kp = 2.0;
573 if (isnan(KI) || KI < 0 || KI > 1000) KI = 0.5;
574 if (isnan(KD) || KD < 0 || KD > 1000) KD = 1.0;
575
576 void saveKalibrasiSuhoToEPROM() {
577   byte flag = suhoKalibrasiTerselesa ? 1 : 0;
578   EEPROM.put(ADDR_SUHA_FLAG, flag);
579   EEPROM.put(ADDR_SUHA_OFFSET, suhoOffset);
580   EEPROM.commit();
581 }
582
583 void loadKalibrasiSuhoFromEPROM() {
584   byte flag = 0;
585   EEPROM.get(ADDR_SUHA_FLAG, flag);
586
587   if (flag == 1) {
588     EEPROM.get(ADDR_SUHA_OFFSET, suhoOffset);
589     suhoKalibrasiTerselesa = true;
590   } else {
591

```

Output
Writing at 0x000070c... (97.5)

File Edit Sketch Tools Help
ESP8266 Dev Module
PIO_Autotune_Megam_vcs1_glimmer_haha | Arduino IDE 2.3.6

```

590 // =====
591 // ESPROM
592 // =====
593 void saveFirmwareToEPROM() {
594   EEPROM.put(ADDR_EP, Kp);
595   EEPROM.put(ADDR_KI, Kd);
596   EEPROM.put(ADDR_KD, Kd);
597   EEPROM.commit();
598 }
599
600 void loadFirmwareFromEPROM() {
601   EEPROM.get(ADDR_EP, Kp);
602   EEPROM.get(ADDR_KI, Kd);
603   EEPROM.get(ADDR_KD, Kd);
604 }
605
606 if (isnan(Kp) || Kp < 0 || Kp > 1000) Kp = 2.0;
607 if (isnan(KI) || KI < 0 || KI > 1000) KI = 0.5;
608 if (isnan(KD) || KD < 0 || KD > 1000) KD = 1.0;
609
610 void saveKalibrasiSuhoToEPROM() {
611   byte flag = suhoKalibrasiTerselesa ? 1 : 0;
612   EEPROM.put(ADDR_SUHA_FLAG, flag);
613   EEPROM.put(ADDR_SUHA_OFFSET, suhoOffset);
614   EEPROM.commit();
615 }
616
617 void loadKalibrasiSuhoFromEPROM() {
618   byte flag = 0;
619   EEPROM.get(ADDR_SUHA_FLAG, flag);
620
621   if (flag == 1) {
622     EEPROM.get(ADDR_SUHA_OFFSET, suhoOffset);
623     suhoKalibrasiTerselesa = true;
624   } else {

```

Output
Writing at 0x000070c... (97.5)

File Edit Sketch Tools Help
ESP8266 Dev Module on COM4 (not connected)

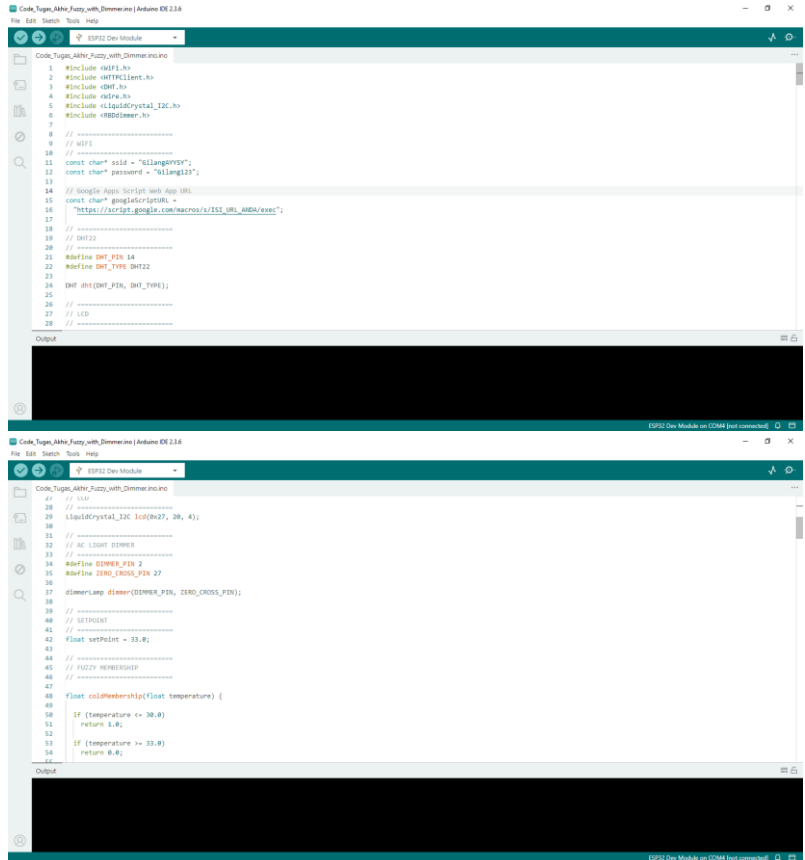
```
File Edit Sketch Tools Help
ESP8266 Dev Module

PID_Autotune_Magnet_yeni_jimmer_huhu | Arduino IDE 2.8

588 //EEPROM.get(AZODR_SIHU_OFFSET, suhuOffset);
589 suhuKalibrasiTersema = true;
590 } else {
591 suhuOffset = 0.0;
592 suhuKalibrasiTersema = false;
593 }
594
595
596 // =====
597 // TAMPILAN LCD & SERIAL
598 // =====
599 void tampilkanLCD(float suhu) {
600 lcd.setCursor(0, 0);
601 lcd.print("Suhu : ");
602 lcd.print(suhu, 1);
603 lcd.print(" C ");
604
605 lcd.setCursor(0, 1);
606 lcd.print("Set : ");
607 lcd.print(setpoint, 1);
608 lcd.print(" C ");
609
610 lcd.setCursor(0, 2);
611 if (autotuneMode) {
612 lcd.print("Mode : AUTOTUNE ");
613 } else {
614 lcd.print("Intensitas : ");
615 lcd.print((int)round(outputSmooth));
616 lcd.print("W ");
617 }
618
619 lcd.setCursor(0, 3);
620 lcd.print("Vga"); lcd.print(Vga, 1);
621 lcd.print(" K1"); lcd.print(K1, 1);
622 lcd.print(" Kd"); lcd.print(Kd, 1);
623
624 }
625
626 void tampilkanError() {
627 lcd.setCursor(0, 0);
628 lcd.print("Sensor DHT22 Error!");
629 }
630
631 void kirimSerial(float suhu) {
632 Serial.print("Suhu"); Serial.print(suhu);
633 Serial.print(" Set"); Serial.print(setpoint);
634 Serial.print(" Intensitas"); Serial.print((int)round(outputSmooth));
635 Serial.print(" Mode"); Serial.print(autotuneMode ? "AUTOTUNE" : "PID");
636 }
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069

```

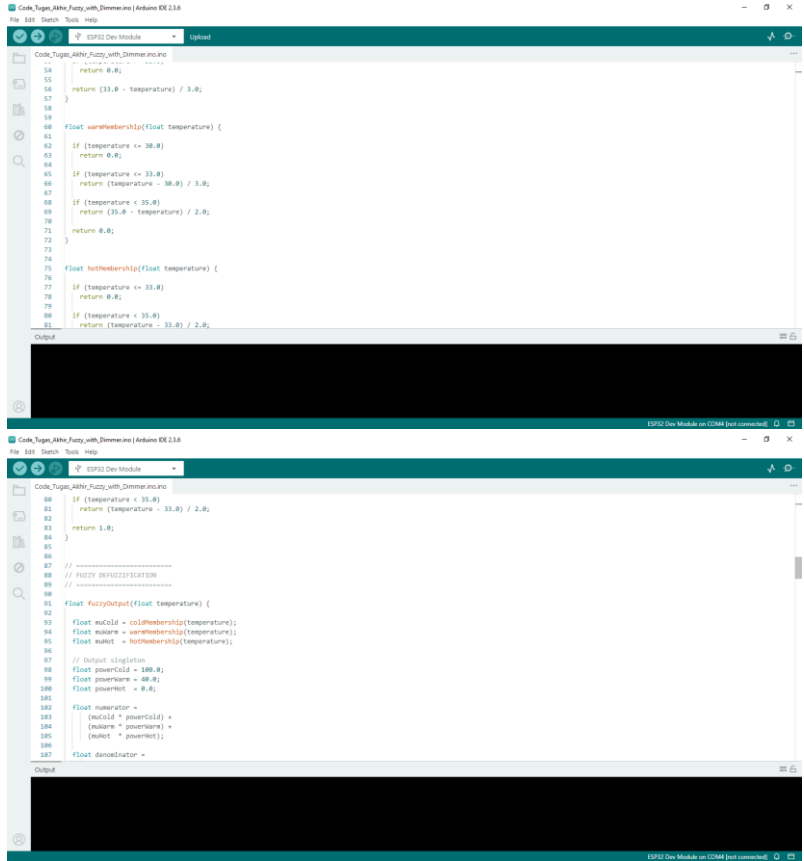
2) Code Logika Fuzzy

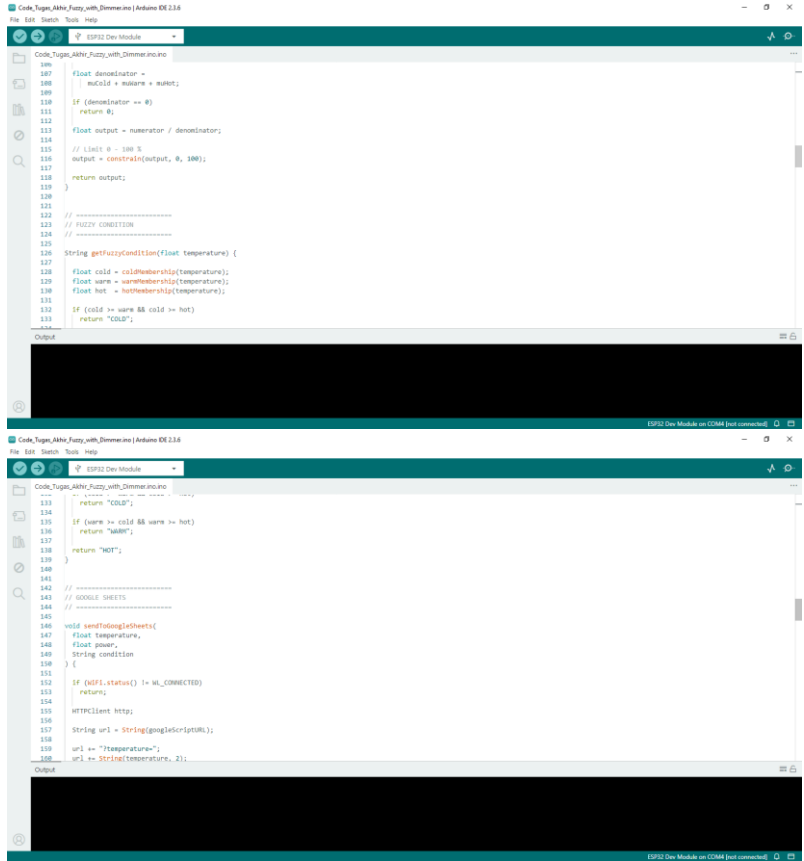


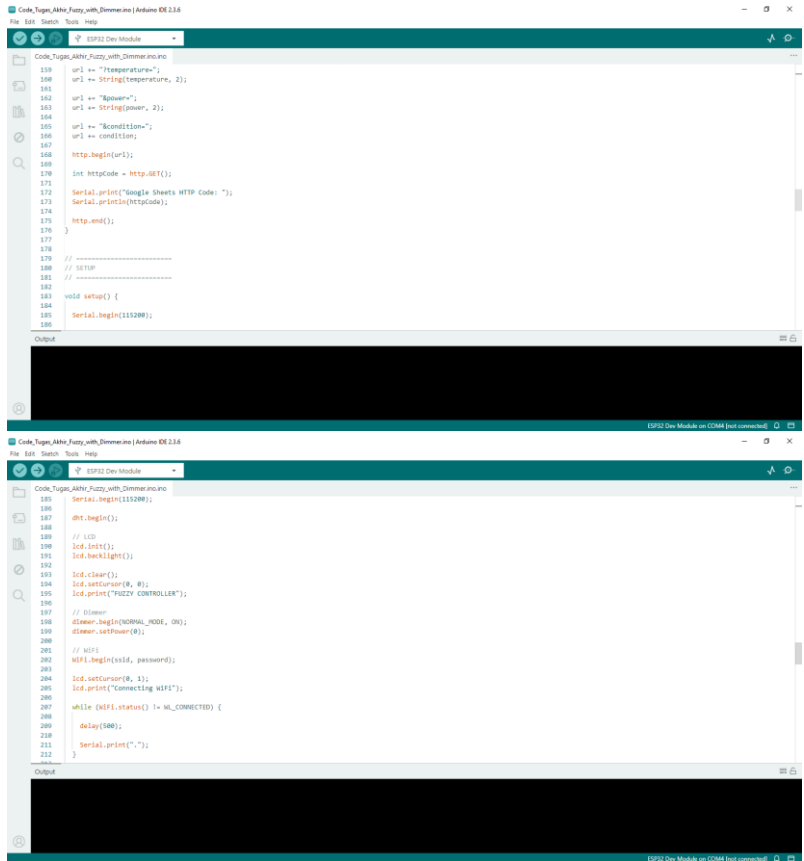
```
Code_Tugas_Akhu_Fuzzy_with_Dimmer.ino | Arduino IDE 2.0.6
File Edit Sketch Tools Help

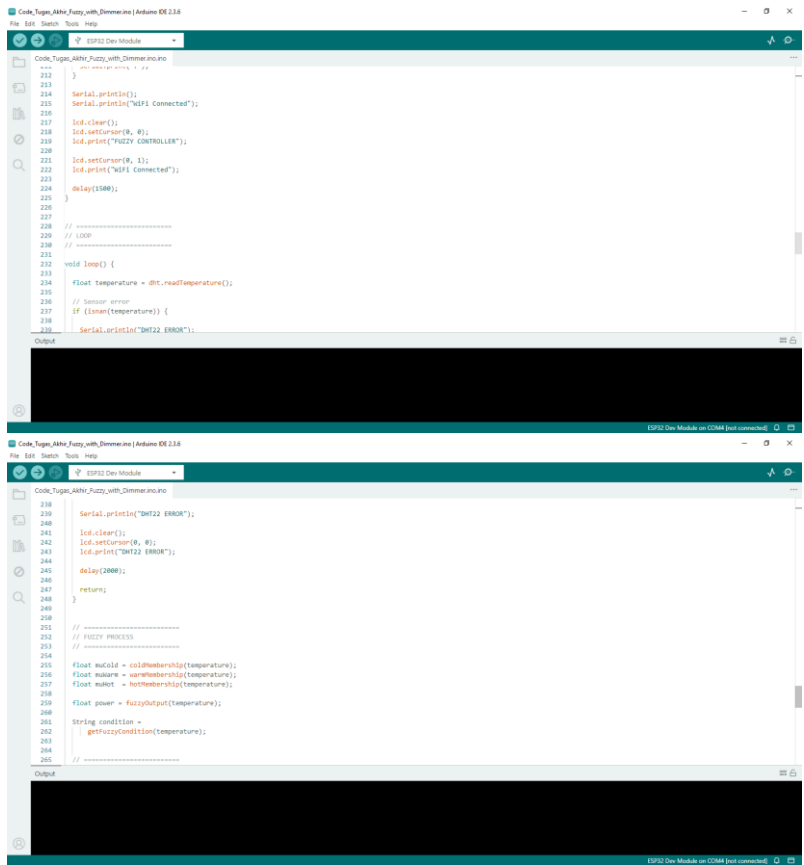
Code_Tugas_Akhu_Fuzzy_with_Dimmer.ino
1 #include <WiFi.h>
2 #include <HTTPClient.h>
3 #include <DHT.h>
4 #include <Wire.h>
5 #include <liquidCrystal_I2C.h>
6 #include <RGBMatrix.h>
7
8 // =====
9 // WiFi
10 // =====
11 const char* ssid = "GilaNganyot";
12 const char* password = "GilaNg123";
13
14 // Google Apps Script web App URL
15 const char* googleScriptURL =
16   "https://script.google.com/macros/s/F5I_URL_BMDA/exec";
17
18 // =====
19 // DHT22
20 // =====
21 #define DHT_PIN 14
22 #define DHT_TYPE DHT22
23
24 DHT dht(DHT_PIN, DHT_TYPE);
25
26 // =====
27 // LCD
28 // =====
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628

```









```

Code_Tugas_Akhir_Fuzzy_with_Dimension [Arduino IDE 2.0.6]
File Edit Sketch Tools Help

ESP8266 Dev Module

Code_Tugas_Akhir_Fuzzy_with_Dimension.ino
264
265 // =====
266 // CONTROL DIPSWER
267 // =====
268 digitalWrite(power);
269
270
271
272 // =====
273 // SERIAL MONITOR
274 // =====
275
276 Serial.println("=====");
277
278 Serial.print("Temperature : ");
279 Serial.println(temperature);
280 Serial.println(" C");
281
282 Serial.print("No COLD : ");
283 Serial.println(noCold, 3);
284
285 Serial.print("No Warm : ");
286 Serial.println(noWarm, 3);
287
288 Serial.print("No HOT : ");
289 Serial.println(noHot, 3);
290
291 Serial.print("Condition : ");
292
Output

```

```

Code_Tugas_Akhir_Fuzzy_with_Dimension [Arduino IDE 2.0.6]
File Edit Sketch Tools Help

ESP8266 Dev Module

Code_Tugas_Akhir_Fuzzy_with_Dimension.ino
293
294 Serial.print("Condition : ");
295 Serial.println(condition);
296
297 Serial.print("Lamp Power : ");
298 Serial.println(power);
299 Serial.println(" W");
300
301 // =====
302 // LCD
303 // =====
304
305 lcd.clear();
306
307 lcd.setCursor(0, 0);
308 lcd.print("P02V LOGIC");
309
310 lcd.setCursor(0, 1);
311 lcd.print("T : ");
312 lcd.print(temperature, 3);
313 lcd.print("C");
314
315 lcd.setCursor(0, 2);
316 lcd.print("COLD : ");
317 lcd.print(condition);
318
319 lcd.setCursor(0, 3);
Output

```

```

320 lcd.print(condition);
321
322 lcd.setCursor(0, 3);
323 lcd.print("P02V");
324
325 // =====
326 // SERIAL MONITOR
327 // =====
328
329 sendTelegramSheets(
330   temperature,
331   power,
332   condition
333 );
334
335 // Sampling Interval
336 delay(6000);
337
Output

```

Lampiran 3









