

PERANCANGAN SISTEM INKUBATOR TELUR **MULTIPURPOSE BERBASIS IOT DENGAN KENDALI FUZZY DAN KOMUNIKASI EDGE-MOBILE MELALUI JARINGAN LOKAL**

Ilham Arif Farabi¹, Dzikri Maulana², Moh. Dicky Purnama³, Susilawati⁴

Universitas Singaperbangsa Karawang, Karawang, Indonesia ^{1,2,3,4}

Email: ilhamfarabi33@gmail.com¹, dzikrimaulana1781945@gmail.com²,

mohdickyurnama@gmail.com³, susilawati.sobur@staff.unsika.ac.id⁴

Informasi	Abstract
Volume : 2 Nomor : 11 Bulan : November Tahun : 2025 E-ISSN : 3062-9624	<p><i>The reliance of conventional Internet of Things (IoT) systems on cloud connectivity often presents a significant barrier to the digitalization of livestock farming in areas with limited internet access. This study aims to design a multipurpose egg incubator system that operates independently using edge-mobile communication via a Local Area Network (LAN) and implements Fuzzy Logic algorithms for microclimate control. The system is constructed using an ESP32 as a client responsible for acquiring DHT22 sensor data and controlling actuators, and a Flutter-based smartphone acting as a local server and edge computing interface. Performance testing results demonstrate high system reliability with an average communication latency of 17.04 ms. The implementation of Fuzzy control proved effective in maintaining temperature stability with an operational average of 30.5°C. However, this study identified two operational limitations: first, the average humidity parameter was recorded at 80.7% (above the ideal range of 40-70%) due to the limited power capacity of the exhaust fan; second, continuous real-time data communication requires the smartphone application to remain in an active state (foreground). This research concludes that the LAN-based edge-mobile architecture is effective as a robust IoT alternative solution for remote areas, provided that mechanical actuator specifications and application state management are adjusted for optimal performance.</i></p>

Keyword: Egg Incubator, Internet of Things (IoT), Fuzzy Logic, Local Area Network (LAN), Edge Computing.

Abstrak

Ketergantungan sistem Internet of Things (IoT) konvensional pada koneksi cloud seringkali menjadi kendala utama dalam digitalisasi peternakan di wilayah dengan akses internet terbatas. Penelitian ini bertujuan untuk merancang sistem inkubator telur multipurpose yang beroperasi secara mandiri menggunakan komunikasi edge-mobile melalui Jaringan Lokal (LAN) dan menerapkan algoritma Fuzzy Logic untuk pengendalian mikroklimat. Sistem dibangun menggunakan ESP32 sebagai client yang bertugas mengakuisisi data sensor DHT22 dan mengendalikan aktuator, serta smartphone berbasis Flutter yang berfungsi sebagai local server dan antarmuka edge computing. Hasil pengujian kinerja menunjukkan reliabilitas sistem yang tinggi dengan latensi komunikasi rata-rata 17.04 ms. Penerapan kendali Fuzzy terbukti efektif menjaga kestabilan suhu dengan rata-rata operasional 30.5°C. Namun, penelitian ini mengidentifikasi dua batasan operasional: pertama, parameter kelembapan rata-rata tercatat sebesar 80.7% (di atas rentang ideal 40-70%) akibat keterbatasan kapasitas daya exhaust fan; kedua, kontinuitas komunikasi data real-time mensyaratkan aplikasi smartphone untuk tetap berada dalam kondisi aktif (foreground). Penelitian ini menyimpulkan bahwa arsitektur edge-mobile berbasis LAN efektif sebagai solusi alternatif IoT yang

tangguh untuk daerah terpencil, dengan catatan perlunya penyesuaian spesifikasi mekanik aktuator dan manajemen state aplikasi untuk performa optimal.

Kata Kunci: Inkubator Telur, Internet of Things (IoT), Fuzzy Logic, Local Area Network (LAN), Edge Computing.

A. PENDAHULUAN

Peternakan merupakan salah satu pilar penting dalam penyediaan pangan, terutama sebagai sumber protein hewani yang dibutuhkan oleh masyarakat. Pada subsektor unggas, proses penetasan telur menjadi tahapan kunci yang menentukan kualitas dan jumlah anak yang dihasilkan. Agar penetasan berjalan dengan baik, kondisi lingkungan di dalam inkubator perlu dijaga dalam rentang yang sesuai dengan kebutuhan perkembangan embrio.

Dua parameter utama yang sangat berpengaruh terhadap keberhasilan penetasan adalah suhu dan kelembaban (Zulhajji & Yantahin, 2022). Jika keduanya tidak dikelola dengan baik, dampaknya dapat sangat luas: daya tetas menurun (Mitchaothai dkk., 2024) dan perkembangan embrio menjadi tidak seimbang (Yalcin dkk., 2022). Perubahan suhu dan kelembaban yang terlalu besar bahkan dapat menyebabkan proses penetasan gagal secara keseluruhan. Penelitian (Agyekum dkk., 2022) menemukan bahwa profil suhu berbeda menghasilkan perbedaan pada waktu *pipping/hatching* sekaligus memengaruhi kualitas anak ayam. Artinya pengaturan suhu memengaruhi lama menetas dan outcome, oleh karena itu praktik manajemen inkubasi menjadi sangat penentu bagi produktivitas peternakan.

Dalam konteks ini, sistem monitoring berbasis *Internet of Things* (IoT) sering diusulkan sebagai solusi salah satunya adalah penelitian dari (Yoal dkk., 2023) yang menggunakan metode *fuzzy sugeno* sebagai basis utama pengambilan keputusan. Namun, implementasi IoT konvensional yang bergantung pada koneksi internet *cloud* menemui kendala nyata di lapangan. Banyak lokasi peternakan, terutama di daerah terpencil, memiliki koneksi internet yang tidak stabil atau bahkan tidak ada sama sekali. Kondisi ini menciptakan masalah baru: latensi tinggi (Abu Bakar, 2020) atau kegagalan pengiriman notifikasi yang justru gagal memenuhi tujuan awal sistem peringatan dini, hal ini menunjukkan bahwa sistem IOT berbasis *cloud* masih memiliki banyak keterbatasan (Samizadeh Nikoui dkk., 2021). Studi tentang komputasi tepi (*edge computing*) dalam ekosistem IoT menunjukkan bahwa arsitektur yang hanya mengandalkan *cloud* memang rentan mengalami keterlambatan layanan (*response time*) dan kurang cocok untuk aplikasi yang menuntut respons waktu nyata (Ali dkk., 2022).

Di sinilah urgensi penelitian ini muncul. Ketika koneksi internet gagal, peternak kehilangan "mata dan telinga"-nya terhadap kondisi inkubator, dan kembali pada metode manual yang rawan *error*. Oleh karena itu, dibutuhkan paradigma baru yang tidak bergantung pada koneksi internet eksternal. Solusi yang diusulkan adalah memanfaatkan jaringan *Local Area Network* (LAN) sebagai tulang punggung komunikasi, dengan *smartphone* yang berfungsi sebagai perangkat *edge*.

Konsep ini memindahkan pusat kendali dan notifikasi ke dalam jaringan lokal. *Smartphone* berinteraksi langsung dengan mikrokontroler pada inkubator, memungkinkan pemantauan data suhu dan kelembaban secara *real-time* tanpa *latency cloud*, serta memberikan peringatan dini secara instan meskipun tanpa akses internet.

Berangkat dari permasalahan tersebut, penelitian ini memiliki beberapa tujuan utama, yaitu:

1. Merancang dan mengimplementasikan sistem pemantauan inkubator berbasis IoT yang memanfaatkan jaringan lokal untuk komunikasi real-time antara sensor (DHT22), mikrokontroler (ESP32), dan *smartphone* sebagai perangkat *edge*.
2. Mengembangkan mekanisme peringatan dini yang instan pada *smartphone* melalui jaringan LAN ketika suhu atau kelembaban menyimpang dari batas aman, tanpa bergantung pada koneksi internet *cloud*.
3. Mengevaluasi keandalan dan responsivitas sistem dalam menjaga stabilitas lingkungan inkubasi serta mencegah penurunan daya tetas dan kematian embrio melalui intervensi yang lebih cepat.
4. Menyediakan antarmuka pemantauan dan kontrol berbasis *smartphone* yang memungkinkan peternak mengakses data serta melakukan tindakan korektif secara langsung dari perangkat mereka di dalam jaringan lokal.

Dari tujuan tersebut, beberapa manfaat yang diharapkan adalah:

1. Bagi Peternak/Usaha:
 - a. Meningkatkan keandalan operasional inkubator dengan sistem yang tetap berfungsi penuh meskipun tanpa koneksi internet.
 - b. Meningkatkan efisiensi waktu dan tenaga karena pemantauan tidak lagi dilakukan secara manual dan notifikasi abnormalitas diterima secara instan dan andal langsung di *smartphone*.
 - c. Mengurangi potensi kerugian ekonomi akibat kegagalan penetasan melalui deteksi dini dan respons yang lebih cepat terhadap penyimpangan kondisi inkubator.

2. Bagi Perkembangan Teknologi:

- a. Memberikan bukti konsep (*proof of concept*) arsitektur IoT berbasis LAN dan perangkat *edge* yang dapat diadopsi untuk aplikasi pemantauan kritis lainnya di lokasi dengan konektivitas internet terbatas.
- b. Menyediakan basis data historis suhu dan kelembaban secara lokal yang dapat digunakan untuk analisis lebih lanjut dan peningkatan akurasi kinerja inkubator di masa depan.

B. METODE PENELITIAN

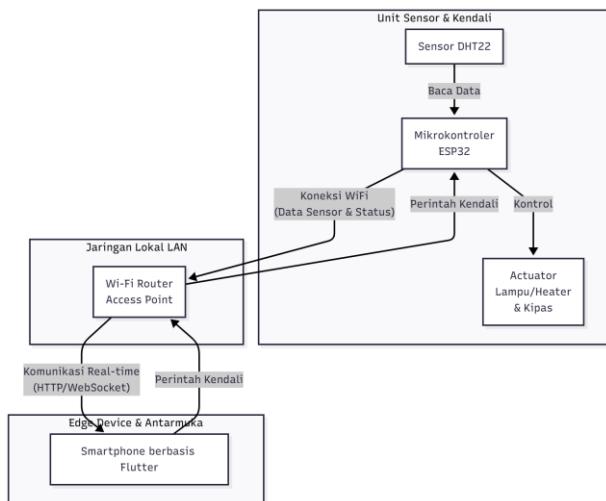
Penelitian ini menggunakan pendekatan *research and development* (R&D) dengan fokus pada perancangan dan pembangunan prototipe sistem monitoring inkubator telur berbasis IoT. Produk yang dikembangkan berupa sistem yang mampu:

- Membaca suhu dan kelembaban secara kontinu,
- Mengirimkan data ke *server*,
- Menyajikan informasi dalam bentuk tampilan grafis,
- Melakukan konfigurasi ideal sistem inkubator,
- Serta tindakan penanganan otomatis jika terjadi hal diluar batas normal.

Metode ini dipilih karena sesuai dengan tujuan untuk merancang, membangun, dan mengevaluasi fungsi serta keandalan sistem yang diusulkan.

1. Blok Diagram dan Alur Sistem

Secara keseluruhan, arsitektur sistem dapat dilihat pada Gambar 1, yang menunjukkan bagaimana setiap komponen saling terhubung membentuk satu kesatuan sistem pemantauan dan pengendalian berbasis IoT. Diagram tersebut memperlihatkan tiga blok utama, yaitu Unit Sensor & Kendali, Jaringan Lokal LAN, dan *Edge Device & Antarmuka*, yang bekerja secara terintegrasi untuk memastikan aliran data dan perintah berlangsung secara *real-time*.



Gambar 1. Blok diagram utama sistem

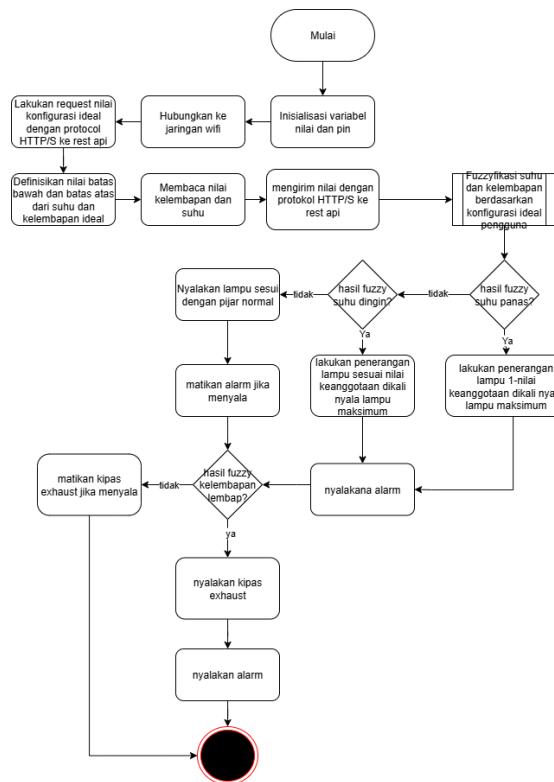
Pada sisi Unit Sensor & Kendali, sensor DHT22 berfungsi sebagai perangkat akuisisi data yang membaca kondisi lingkungan berupa suhu dan kelembapan. Data yang diperoleh kemudian dikirimkan ke mikrokontroler ESP32 sebagai pusat pemrosesan. ESP32 menjalankan dua tugas utama: pertama, mengolah dan mengirimkan data sensor melalui koneksi Wi-Fi; kedua, mengeksekusi logika kontrol untuk mengatur aktuator seperti lampu dan kipas berdasarkan nilai sensor atau perintah dari pengguna.

ESP32 terhubung dengan *Wi-Fi Router/Access Point* yang berada dalam Jaringan Lokal LAN, yang berfungsi sebagai jembatan komunikasi antar perangkat. Melalui *access point* ini, data sensor dan status sistem dikirimkan secara real-time ke perangkat pengguna, sementara perintah kendali dari pengguna diteruskan kembali ke ESP32. Jalur komunikasi ini dapat menggunakan HTTP atau *WebSocket*, tergantung kebutuhan responsivitas dan kontinuitas data.

Pada bagian *Edge Device & Antarmuka*, *smartphone* menjalankan aplikasi berbasis Flutter yang berfungsi sebagai UI utama bagi pengguna. Aplikasi ini menerima data sensor secara berkala, sekaligus memungkinkan pengguna mengirimkan perintah kontrol. Seluruh interaksi antara aplikasi dan jaringan lokal dilakukan secara dua arah, pengguna dapat memantau kondisi lingkungan sekaligus memberikan kendali langsung terhadap perangkat yang terhubung.

Untuk melengkapi pemahaman mengenai mekanisme kerja setiap bagian, khususnya urutan langkah pemrosesan data dan pengambilan keputusan pada mikrokontroler, pada Gambar 2 ditampilkan *flowchart* yang menggambarkan alur proses sistem secara lebih rinci. *Flowchart* tersebut menggambarkan tahapan mulai dari inisialisasi awal, proses koneksi ke jaringan Wi-Fi, pengambilan nilai konfigurasi ideal melalui protokol HTTP/S, hingga proses

pembacaan sensor suhu dan kelembapan. Selanjutnya ditunjukkan pula bagaimana mikrokontroler melakukan perhitungan *fuzzy* berdasarkan batas nilai suhu dan kelembapan ideal, kemudian menentukan aksi pengendalian yang sesuai, seperti menyalaakan lampu, heater, kipas *exhaust*, maupun alarm.



Gambar 2. *Flowchart* Sistem

2. Tahapan Penelitian

Penelitian ini akan dilaksanakan melalui beberapa tahapan utama:

a. Tahap Perancangan

Pada tahap ini, sistem dirancang berdasarkan kebutuhan dan blok diagram yang telah dibuat:

1) Perancangan Perangkat Keras

Menentukan dan merancang rangkaian untuk mikrokontroler yang terintegrasi dengan sensor DHT22 dan *actuator* seperti relay dan ac-dhimmer untuk mengontrol lampu pemanas dan kipas. Rancangan memastikan keselamatan operasional dengan menggunakan komponen yang sesuai dengan daya dan suhu kerja.

2) Perancangan Perangkat Lunak

Pada tahap ini dilakukan serangkaian proses perancangan perangkat lunak yang meliputi pembuatan *firmware* mikrokontroler, konfigurasi sistem komunikasi, serta

pengembangan antarmuka pengguna.

Firmware pada ESP32 dikembangkan menggunakan Arduino IDE (C++) dengan fungsi utama melakukan akuisisi data dari sensor DHT22, mengeksekusi logika kendali *fuzzy* untuk mengoperasikan aktuator, serta mengelola konektivitas Wi-Fi untuk komunikasi jaringan lokal.

Sistem komunikasi diimplementasikan menggunakan arsitektur *client-server* dalam lingkungan LAN, dimana ESP32 berfungsi sebagai *client* yang mengirimkan data sensor ke aplikasi mobile yang bertindak sebagai *server* lokal. Pertukaran data dilakukan melalui protokol HTTP REST API yang dipilih karena efisiensinya dalam mentransmisikan data real-time (Dwiyanto dkk., 2025) , serta kemudahan implemetasinya dalam konteks penelitian ini.

Antarmuka pengguna dikembangkan dengan *framework* Flutter yang memungkinkan pembuatan aplikasi mobile responsif yang dapat beroperasi tanpa koneksi internet eksternal. Aplikasi ini menampilkan visualisasi data sensor secara *real-time* dalam format numerik dan grafik, menyediakan notifikasi status sistem, serta memungkinkan konfigurasi parameter kendali secara manual melalui antarmuka yang intuitif.

b. Tahap Implementasi dan Pengujian

Tahap ini berfokus pada pelaksanaan rancangan yang telah dibuat, mulai dari proses implementasi perangkat keras dan perangkat lunak hingga pengujian fungsional serta pengujian kinerja sistem secara menyeluruh.

1) Tahap Implementasi

Implementasi dimulai dengan perakitan perangkat keras sesuai rancangan yang telah disusun pada tahap sebelumnya. Setelah perangkat dirakit, *firmware* kemudian diunggah ke mikrokontroler ESP32 untuk mengaktifkan fungsi pembacaan sensor, pengendalian aktuator, serta sistem konsumsi API (*Application Programming Interface*).

Pada tahap yang sama, antarmuka aplikasi yang telah dirancang turut diintegrasikan, sehingga seluruh sistem monitoring, kontrol, dan notifikasi dapat berjalan secara mandiri tanpa bergantung pada layanan eksternal.

2) Tahap Pengujian Fungsional

Pengujian fungsional dilakukan untuk memastikan seluruh komponen bekerja sesuai rancangan. Ruang lingkup pengujian mencakup verifikasi konektivitas LAN, yaitu memastikan *smartphone* dapat terhubung dengan ESP32 dalam jaringan lokal tanpa gangguan, serta pengujian respons aktuator untuk menilai apakah pemanas dan kipas beroperasi (menyalakan).

dan mati) sesuai logika pengendalian yang dipicu oleh perubahan suhu dan kelembapan.

3) Tahap Pengujian Kinerja

Pada tahap ini, pengujian difokuskan pada evaluasi efektivitas dan efisiensi sistem dalam kondisi operasional nyata. Pengujian kinerja yang dilakukan meliputi dua aspek utama. Pertama, uji stabilitas API *server* menggunakan k6 untuk memastikan kemampuan sistem dalam menangani beban permintaan secara berkelanjutan tanpa penurunan performa. Kedua, pengujian kestabilan lingkungan inkubator melalui pemantauan suhu dan kelembapan, untuk memastikan bahwa kedua parameter tersebut dapat dipertahankan dalam rentang ideal selama periode pengujian.

C. HASIL DAN PEMBAHASAN

1. Implementasi Sistem

Sistem monitoring inkubator berbasis LAN dengan antarmuka *smartphone* telah berhasil diimplementasikan. Berbeda dengan arsitektur umum yang menjadikan mikrokontroler sebagai *server*, pada sistem ini *smartphone* berperan sebagai *server* lokal, sementara ESP32 berfungsi sebagai *client* yang mengirimkan data sensor dan menerima perintah kendali dari *server*. Pendekatan ini memberikan fleksibilitas lebih besar dalam pemrosesan data, penyajian antarmuka, dan pengelolaan logika *fuzzy*. Sistem terdiri dari tiga komponen utama, yaitu perangkat keras inkubator, antarmuka *server* pada *smartphone*, dan mekanisme kendali berbasis *fuzzy logic* yang diolah langsung pada *smartphone*.

a. Implementasi Perangkat Keras

Subsistem perangkat keras dalam penelitian ini diimplementasikan menggunakan modul mikrokontroler ESP32 sebagai unit pengendali utama (*client*). ESP32 dipilih karena kekuatan pemrosesan cukup tinggi, efisiensi daya yang baik, dan konektivitas nirkabel terintegrasi, menjadikannya pilihan ideal untuk IoT modern dibanding MCU generasi lebih tua (Yan, 2025). ESP32 bertanggung jawab untuk mengakuisisi data dari sensor dan mengoperasikan aktuator berdasarkan logika kendali yang telah ditetapkan. Sensor DHT22 digunakan untuk membaca parameter lingkungan berupa suhu dan kelembaban udara secara berkala, sensor ini dipilih karena harganya yang ekonomis dan galat yang masih dalam batas toleransi (Puspasari dkk., 2020). Sebagai antarmuka kendali, sebuah modul relay diimplementasikan untuk mengoperasikan kipas exhaust, sementara sebuah dimmer AC digunakan untuk mengatur daya pada lampu pemanas guna mengontrol intensitas pemanasan.

Pada arsitektur jaringan, modul ESP32 dikonfigurasi dalam mode Stasiun (Station Mode/STA). Konfigurasi ini memungkinkan ESP32 untuk melakukan hal berikut:

- Melakukan koneksi secara mandiri ke jaringan Wi-Fi yang dibuat oleh *hotspot smartphone* atau jaringan lokal sejenis.
- Mengirimkan data hasil pembacaan sensor ke *server* lokal (aplikasi pada *smartphone*) secara periodik menggunakan protokol HTTP REST API, memastikan komunikasi data yang efisien dan terstandarisasi.

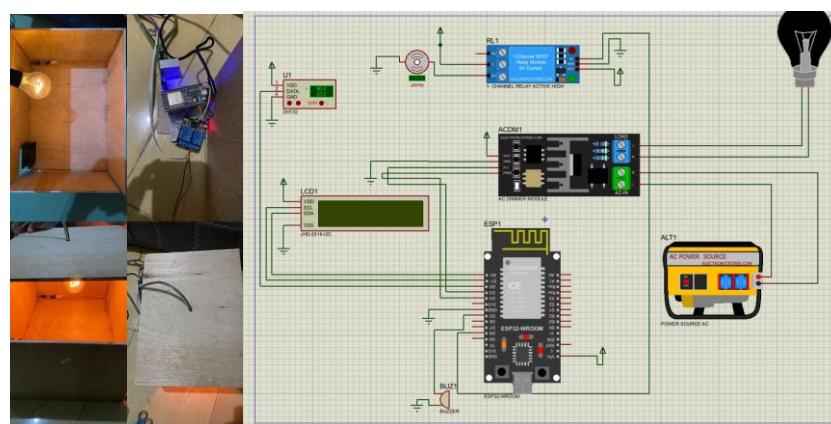
Subsistem aktuator terdiri dari:

- Sebuah lampu pijar sebagai sumber panas.
- Modul AC Dimmer (ACDM1) yang mengatur daya lampu pijar secara phase-control untuk menyesuaikan suhu secara gradual.
- Modul Relay (RLI) yang mengendalikan kipas *exhaust* untuk sirkulasi udara dan pengaturan kelembaban.
- Buzzer sebagai peringatan auditori jika sistem mendeteksi kondisi di luar batas normal.

Seluruh modul didukung oleh catu daya yang stabil, terdiri dari:

- Catu daya 5V DC untuk ESP32, sensor, dan buzzer.
- Catu daya 12V DC untuk kipas *exhaust*.
- Catu daya AC yang terhubung dengan dimmer dan lampu pijar.

Rangkaian lengkap dari sistem perangkat keras yang diimplementasikan dapat dilihat pada Gambar 3.



Gambar 3. Rangkaian Perangkat Keras Sistem

b. Perancangan Perangkat Lunak

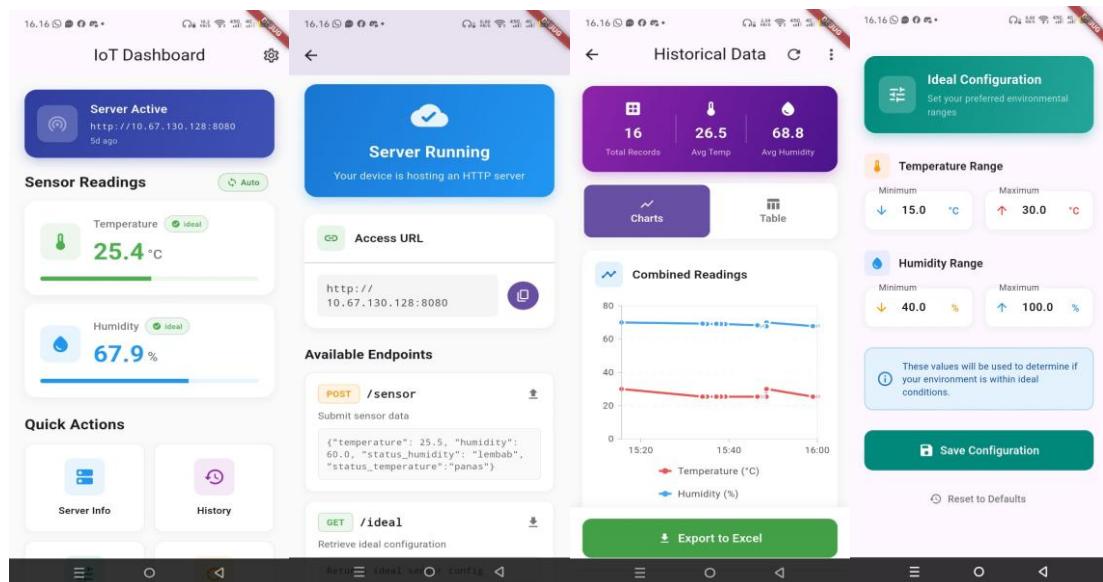
1) Antarmuka *Smartphone*

Dalam penelitian ini, peran antarmuka pengguna dan pusat pemrosesan data dilakukan oleh sebuah aplikasi Android berbasis Flutter. Aplikasi ini tidak hanya berfungsi sebagai antarmuka visual, tetapi juga sebagai *local server* yang meng-hosting *endpoint* untuk komunikasi dua arah dengan modul ESP32.

Fitur-fitur utama yang diimplementasikan meliputi:

- Menampilkan data suhu dan kelembaban terkini beserta grafik perubahannya dalam sebuah dashboard yang intuitif.
- Aplikasi menyimpan riwayat data sensor dan memungkinkan ekspor data ke format Excel untuk memudahkan analisis.
- Memberikan peringatan langsung kepada pengguna apabila terjadi anomali parameter yang melebihi batas yang ditentukan.
- Memfasilitasi pengguna dalam mengatur nilai setpoint suhu dan kelembaban ideal yang akan digunakan sebagai input dalam sistem logika *fuzzy*.

Implementasi visual dari antarmuka aplikasi ditunjukkan pada Gambar 4.



Gambar 4. Tampilan Antarmuka pada Smartphone

2) Implementasi Logika Kendali *Fuzzy* pada ESP32

ESP32 membaca nilai suhu dan kelembapan dari sensor, kemudian menghitung derajat keanggotaan menggunakan fungsi trapezoidal membership. Nilai batas atas dan bawah ditentukan oleh pengguna melalui aplikasi *mobile*, kemudian batas atas dan bawah diberi tambahan margin sebesar 20% untuk membentuk zona transisi di luar rentang ideal, sehingga perubahan kategori tidak terjadi secara tiba-tiba. Fungsi trapezoidal dipilih karena menghasilkan transisi yang halus, mudah dihitung, dan efisien untuk diproses oleh

mikrokontroler, dalam penelitian (Susilo & Lalay, 2025) juga disebutkan bahwa fungsi keanggotaan inilah yang paling ideal berdasarkan penggunaan energi.

Proses fuzzy suhu mengikuti alur keputusan seperti pada flowchart pada Gambar 2:

- 1) maka lampu akan menyala dengan intensitas yang berbanding lurus dengan nilai keanggotaan fuzzy:

$$\text{Intensitas lampu} = \mu \text{ dingin} \times I \text{ maks}$$

Artinya, semakin dingin suhu, nilai μ dingin semakin tinggi, dan lampu menyala lebih terang untuk memberikan kompensasi panas.

- 2) Jika suhu berada pada kategori "panas", intensitas lampu dihitung secara terbalik terhadap nilai keanggotaan:

$$\text{Intensitas lampu} = (1 - \mu \text{ panas}) \times I \text{ maks}$$

Dengan demikian, semakin tinggi suhu (semakin panas), nilai μ panas semakin besar dan lampu secara otomatis diturunkan intensitasnya hingga mendekati mati.

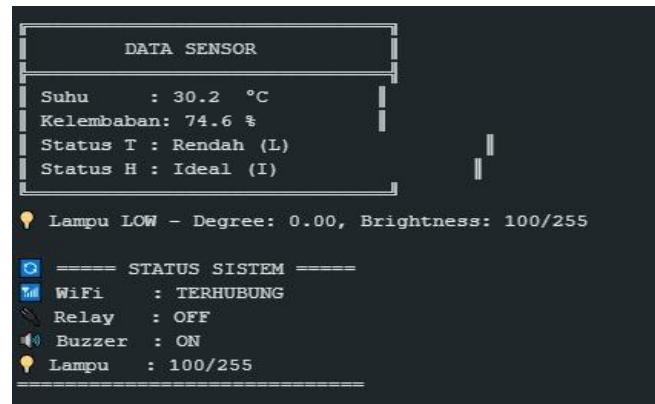
- 3) Jika tidak memenuhi kedua kondisi, suhu dianggap berada dalam rentang normal sehingga lampu tidak mengalami perubahan intensitas cahaya.

Fuzzy dipilih karena dapat menangani data *noisy* secara lebih sesuai (Athiyah dkk., 2021). Metode ini tidak hanya membedakan antara "ideal" dan "tidak ideal", tetapi juga menilai tingkat keanggotaan suatu nilai dalam kategori tertentu, sehingga pengendalian menjadi lebih adaptif dan halus.

2. Hasil Pengujian Fungsional

Pengujian fungsional dilakukan untuk memastikan bahwa sistem mampu menjalankan seluruh fungsi utamanya, mulai dari koneksi LAN hingga respons aktuator terhadap hasil pengolahan data. Hasil pengujian menunjukkan bahwa sistem bekerja secara konsisten dan sesuai dengan rancangan.

Pada pengujian koneksi LAN, *smartphone* dapat terhubung ke jaringan lokal yang dibentuk oleh ESP32 tanpa hambatan. Proses pairing hanya memerlukan beberapa detik, dan koneksi tetap stabil selama pengujian berlangsung. Aplikasi mobile dapat menerima data sensor, serta mengirimkan perintah kontrol tanpa kehilangan paket data atau jeda berarti seperti yang ditunjukkan pada Gambar 5. Perlu dicatat bahwa kontinuitas pertukaran data ini hanya dapat dipertahankan selama aplikasi pada perangkat *smartphone* berada dalam kondisi aktif karena sistem operasi Android memblokir background services yang berjalan terus-menerus tanpa foreground notification, sebagai bagian dari upaya penghematan energi dan perlindungan privasi (Sutter, 2020).



```
DATA SENSOR
Suhu : 30.2 °C
Kelembaban: 74.6 %
Status T : Rendah (L)
Status H : Ideal (I)

Lampu LOW - Degree: 0.00, Brightness: 100/255

===== STATUS SISTEM =====
WiFi : TERHUBUNG
Relay : OFF
Buzzer : ON
Lampu : 100/255
```

Gambar 5. Uji Fungsional Sistem dan Konektivitas

Pada pengujian respons aktuator, pemanas dan *exhaust fan* mampu merespons perubahan suhu dan kelembapan sesuai logika pengendalian *fuzzy* yang diimplementasikan pada ESP32. Ketika suhu turun memasuki wilayah dingin, intensitas nyala lampu pemanas meningkat secara bertahap berdasarkan nilai keanggotaan *fuzzy*. Sebaliknya, ketika suhu meningkat, intensitas lampu berkurang secara proporsional.

Untuk *exhaust fan*, hasil logika *fuzzy* kelembapan bekerja sesuai ekspektasi. Saat nilai *fuzzy* menunjukkan kondisi “lembap”, *exhaust* beralih ke mode HIGH, dan kembali ke LOW ketika tingkat kelembapan menurun.

3. Hasil Pengujian Kinerja

Pengujian kinerja diarahkan untuk mengevaluasi efektivitas dan efisiensi sistem dalam kondisi operasional nyata. Dua aspek utama yang diuji adalah stabilitas *API server* menggunakan K6 dan kestabilan parameter lingkungan inkubator (suhu dan kelembapan) selama periode pengujian.

a. Hasil Pengujian Stabilitas *API Server*

Pengujian dilakukan dengan mensimulasikan permintaan berulang ke *server* lokal aplikasi *mobile* melalui protokol HTTP REST API. Skenario k6 dirancang untuk mensimulasikan pola komunikasi riil antara ESP32 dan aplikasi *mobile*, yaitu polling data sensor secara periodik ditambah request kontrol aktuator pada kondisi tertentu.

```

execution: local
script: k6.js
output: -
scenarios: (100.00%) 1 scenario, 1 max VUs, 5m0s max duration (incl. graceful stop):
  * default: 1 looping VUs for 5m0s (gracefulStop: 30s)

THRESHOLDS
http_req_duration
  ✓ 'p(99)<1000' p(99)=70.03ms
  {endpoint:get_ideal}
    ✓ 'p(95)<300' p(95)=13.89ms
  {endpoint:post_sensor}
    ✓ 'p(95)<500' p(95)=45.53ms
http_req_failed
  ✓ 'rate<0.01' rate=0.00

TOTAL RESULTS
checks_total.....: 172 0.568512/s
✓ POST /sensor status is 200 or 201
✓ POST /sensor accepted data

HTTP
http_req_duration.....: avg=17.04ms min=5.9ms med=14.28ms max=75.13ms p(90)=25.35ms p(95)=30.46ms
{ endpoint:get_ideal }.....: avg=9.49ms min=5.9ms med=8.5ms max=31.08ms p(90)=12.16ms p(95)=13.89ms
{ endpoint:post_sensor }.....: avg=24.6ms min=12.51ms med=21.71ms max=75.13ms p(90)=28.57ms p(95)=45.53ms
{ expected_response:true }.....: avg=17.04ms min=5.9ms med=14.28ms max=75.13ms p(90)=25.35ms p(95)=30.46ms
http_req_failed.....: 0.00% 0 out of 86
http_reqs.....: 86 0.284256/s

EXECUTION
iteration_duration.....: avg=7.03s min=7.02s med=7.03s max=7.08s p(90)=7.05s p(95)=7.05s
iterations.....: 43 0.142128/s
vus.....: 1 min=1 max=1
vus_max.....: 1 min=1 max=1

NETWORK
data_received.....: 25 kB 82 B/s
data_sent.....: 13 kB 43 B/s

```

Gambar 6. Hasil Pengujian Stabilitas API Server menggunakan K6

Hasil pengujian stabilitas (*soak testing*) menggunakan k6 selama 5 menit dengan 1 *Virtual User* (VU) menunjukkan bahwa *server* lokal mampu menangani aliran permintaan secara konsisten. Selama durasi pengujian, sistem berhasil memproses total 86 HTTP *requests* tanpa terjadi error atau kegagalan (*rate* 0.00%), yang menunjukkan stabilitas komunikasi pada endpoint yang diuji.

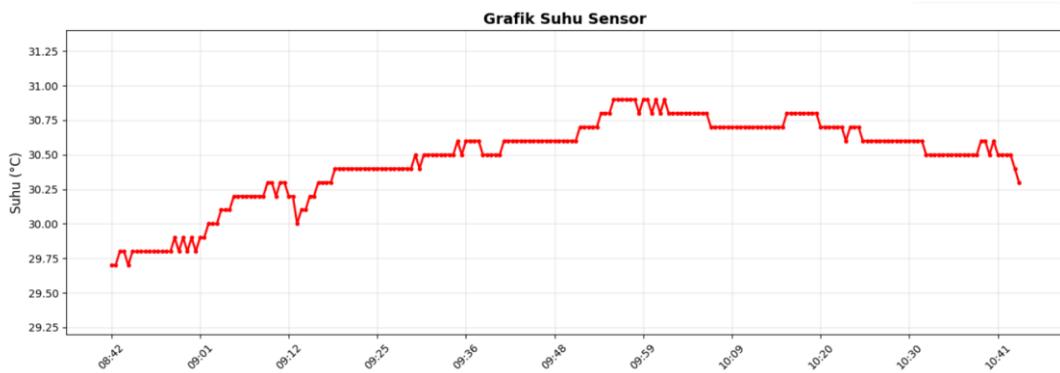
Tingkat respons tercatat sangat cepat dan stabil di bawah ambang batas yang ditentukan. Rata-rata durasi permintaan adalah 17.04 ms, dengan nilai p(95) sebesar 30.46 ms. Secara spesifik, *endpoint* *get_ideal* memiliki respons rata-rata 9.49 ms, sedangkan *endpoint* *post_sensor* merespons dalam 24.6 ms.

Temuan ini memvalidasi bahwa arsitektur *client-server* dalam LAN memberikan latensi rendah. Seluruh checks validasi berhasil lulus, membuktikan bahwa sistem mampu mempertahankan integritas data dan komunikasi real-time yang andal untuk skenario inkubasi yang berjalan terus-menerus.

b. Hasil Pengujian Kestabilan Suhu dan Kelembapan

Selama beberapa jam pengujian, data suhu dan kelembapan direkam secara kontinu untuk menilai apakah lingkungan inkubasi dapat dipertahankan pada rentang ideal.

1) Kestabilan Suhu



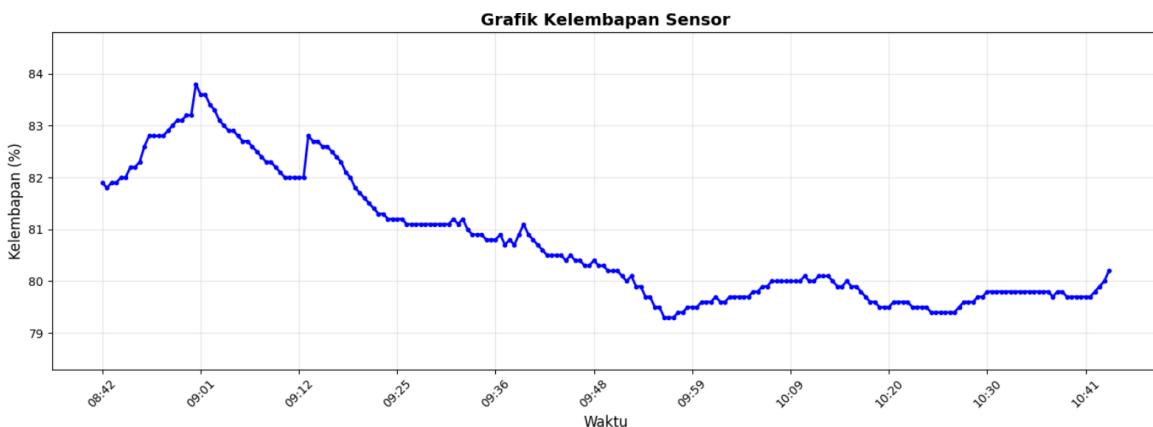
Gambar 7. Grafik Suhu Sistem Selama Kurun Waktu Tertentu

Berdasarkan pengujian yang dilakukan dari pukul 08:42 hingga 10:43 pada Gambar 7, sistem menunjukkan performa yang stabil dalam menjaga kondisi lingkungan inkubasi. Hasil monitoring suhu dapat dilihat pada grafik respon sistem pada Gambar 7. Data menunjukkan suhu terendah tercatat pada 29.7°C (saat *start-up*) dan suhu maksimum mencapai 30.9°C, dengan rata-rata suhu operasional sebesar 30.5°C.

Nilai rata-rata ini berada dalam rentang ideal yang ditetapkan, yaitu antara 30°C hingga 35°C. Grafik menunjukkan bahwa setelah mencapai titik setel (setpoint), fluktuasi suhu yang terjadi sangat minim dan terkontrol. Fenomena ini mengindikasikan bahwa penerapan algoritma *Fuzzy Logic* dengan fungsi keanggotaan trapesium bekerja efektif dalam mengatur intensitas pemanas.

Kontrol konvensional berbasis ON/OFF cenderung menghasilkan *overshoot* atau *undershoot* yang signifikan, hal ini juga sering menyebabkan fluktuasi suhu yang “tidak terkendali (Yusuf & Saputra, 2020), sebaliknya sistem ini menunjukkan transisi yang halus. Saat suhu mendekati ambang batas bawah ideal, sistem merespons dengan meningkatkan intensitas pemanas secara bertahap, dan sebaliknya, menurunkan atau mematikan pemanas saat mendekati batas atas untuk mencegah *overheating*. Absennya osilasi ekstrem pada grafik menegaskan bahwa kontroler mampu mencapai *steady state* dengan presisi yang baik.

2) Kestabilan Kelembapan



Gambar 8. Grafik Kelembapan Sistem Selama Kurun Waktu Tertentu

Pengujian parameter kelembapan dilakukan bersamaan dengan monitoring suhu dalam durasi waktu yang sama. Hasil pengukuran sensor menunjukkan data sebagai berikut:

- Kelembapan Minimum: 79.3%
- Kelembapan Maksimum: 83.8%
- Rata-rata: 80.7%

Berdasarkan grafik hasil pengujian pada Gambar 8, parameter kelembapan tercatat memiliki nilai rata-rata sebesar 80.7%, dengan nilai minimum 79.3% dan maksimum 83.8%. Nilai ini terpantau masih berada di atas rentang ideal yang ditargetkan (40% - 70%).

Secara fisik, sistem telah dilengkapi dengan aktuator *exhaust fan* yang berfungsi membuang udara lembap keluar dari ruang inkubasi. Grafik menunjukkan adanya tren penurunan kelembapan mulai pukul 09:01 seiring dengan naiknya suhu, yang menandakan adanya upaya sistem untuk menurunkan kelembapan relatif. Namun, laju penurunan tersebut berjalan lambat dan belum mampu mencapai setpoint bawah.

Hal ini mengindikasikan bahwa kapasitas debit udara dari *exhaust fan* yang digunakan belum memadai untuk volume ruang inkubator tersebut. Daya hisap *fan* yang rendah menyebabkan sirkulasi udara tidak cukup cepat untuk membuang akumulasi uap air secara efektif, meskipun logika kontrol telah memberikan sinyal aktif. Oleh karena itu, kestabilan kelembapan pada target ideal memerlukan peningkatan spesifikasi *exhaust fan* dengan RPM atau daya yang lebih tinggi.

D. KESIMPULAN

Penelitian ini telah berhasil mengembangkan dan mengimplementasikan sistem monitoring serta pengendalian inkubator telur berbasis IoT yang sepenuhnya beroperasi pada jaringan lokal (LAN), menjawab tantangan ketidakstabilan koneksi internet di wilayah pedesaan. Melalui arsitektur *edge-mobile*, *smartphone* terbukti mampu mengantikan peran

cloud server dalam mengolah data dan memberikan respons *real-time* dengan latensi yang sangat rendah. Penerapan algoritma *Fuzzy Logic* memberikan dampak signifikan pada kestabilan suhu ruang inkubasi, di mana sistem mampu mempertahankan suhu rata-rata pada 30.5°C dengan transisi pemanasan yang halus tanpa *overshoot* ekstrem. Meskipun demikian, dua keterbatasan utama teridentifikasi: Pertama, pengendalian kelembapan belum mencapai titik optimal dengan rata-rata 80.7% karena keterbatasan spesifikasi daya hisap pada aktuator *exhaust fan*, yang mengindikasikan bahwa logika kendali yang presisi harus didukung oleh kapasitas perangkat keras yang memadai. Kedua, kontinuitas komunikasi data dan pengendalian memerlukan aplikasi mobile pada perangkat *smartphone* untuk tetap berada dalam mode aktif. Secara keseluruhan, sistem ini menawarkan solusi teknologi tepat guna yang meningkatkan efisiensi dan keamanan proses penetasan telur tanpa ketergantungan pada penyedia layanan internet, serta memberikan landasan bagi pengembangan sistem IoT mandiri di sektor peternakan masa depan.

E. DAFTAR PUSTAKA

- Abu Bakar, M. T. (2020). Latency Issues in Internet of Things: A Review of Literature and Solution. *International Journal of Advanced Trends in Computer Science and Engineering*, 9(1.3), 83–91. <https://doi.org/10.30534/ijatcse/2020/1291.32020>
- Agyekum, G., Okai, M. A., Tona, J. K., Donkoh, A., & Hamidu, J. A. (2022). Impact of incubation temperature profile on chick quality, bone, and immune system during the late period of incubation of Cobb 500 broiler strain. *Poultry Science*, 101(9), 101999. <https://doi.org/10.1016/j.psj.2022.101999>
- Ali, O., Ishak, M. K., Bhatti, M. K. L., Khan, I., & Kim, K.-I. (2022). A Comprehensive Review of Internet of Things: Technology Stack, Middlewares, and Fog/Edge Computing Interface. *Sensors*, 22(3), 995. <https://doi.org/10.3390/s22030995>
- Athiyah, U., Handayani, A. P., Aldean, M. Y., Putra, N. P., & Ramadhani, R. (2021). Sistem Inferensi Fuzzy: Pengertian, Penerapan, dan Manfaatnya. *Journal of Dinda : Data Science, Information Technology, and Data Analytics*, 1(2), 73–76. <https://doi.org/10.20895/dinda.v1i2.201>
- Dwiyanto, R. A., Mutiara, G. A., & Sari, M. I. (2025). Performance analysis of REST API in a real-time IoT-based vehicle monitoring system. *International Journal of Reconfigurable and Embedded Systems (IJRES)*, 14(3), 766. <https://doi.org/10.11591/ijres.v14.i3.pp766-784>

- Mitchaothai, J., Lertpatarakomol, R., Trairatapiwan, T., & Lukkananukool, A. (2024). Influence of Incubation Temperature and Relative Humidity on the Egg Hatchability Pattern of Two-Spotted (*Gryllus bimaculatus*) and House (*Acheta domesticus*) Crickets. *Animals*, 14(15), 2176. <https://doi.org/10.3390/ani14152176>
- Puspasari, F., Satya, T. P., Oktiawati, U. Y., Fahrurrozi, I., & Prisyanti, H. (2020). Analisis Akurasi Sistem Sensor DHT22 berbasis Arduino terhadap Thermohygrometer Standar. *JURNAL FISIKA DAN APLIKASINYA*, 16(1), 40–45.
- Samizadeh Nikoui, T., Rahmani, A. M., Balador, A., & Haj Seyyed Javadi, H. (2021). Internet of Things architecture challenges: A systematic review. *International Journal of Communication Systems*, 34(4). <https://doi.org/10.1002/dac.4678>
- Susilo, S., & Lalay, A. A. (2025). ANALYSIS OF ENERGY EFFICIENCY PERFORMANCE USING THE MEMBERSHIP FUNCTION (MF) METHOD IN A FUZZY LOGIC CONTROL SYSTEM FOR RESIDENTIAL SPLIT AIR CONDITIONERS (AC). *Multidisciplinary Indonesian Center Journal (MICJO)*, 2(4), 4682–4695. <https://doi.org/10.62567/micjo.v2i4.1381>
- Sutter, T. (2020). Simple Spyware: Androids Invisible Foreground Services and How to (Ab)use Them.
- Yalcin, S., Özkan, S., & Shah, T. (2022). Incubation Temperature and Lighting: Effect on Embryonic Development, Post-Hatch Growth, and Adaptive Response. *Frontiers in Physiology*, 13. <https://doi.org/10.3389/fphys.2022.899977>
- Yan, P. (2025). Architectural Evolution and Performance Optimization in Embedded Systems: A Comparative Analysis of ESP8266, ESP32-S3, and ESP32-C6 Platforms. *International Journal of Advance in Applied Science Research*, 4(9), 51–56. www.h-tsp.com
- Yoal, H., Dirgantara, W., & Subairi, S. (2023). Monitoring Suhu dan Kelembaban pada Penetas Telur Otomatis Menggunakan Metode Fuzzy Sugeno Berbasis IoT. *Blend Sains Jurnal Teknik*, 2(2), 176–183. <https://doi.org/10.56211/blendsains.v2i2.356>
- Yusuf, C., & Saputra, D. I. (2020). Optimasi Kendali Suhu pada Sistem Nirkabel Penetasan Telur Berbasis PI dan PI Anti Windup. *Jurnal Otomasi Kontrol dan Instrumentasi*, 12(2), 79–101. <https://doi.org/10.5614/joki.2020.12.2.3>
- Zulhajji, Z., & Yantahin, M. (2022). Comparative Analysis of Temperature External and Internal and Relative Humidity of Electric Egg Incubator. *Indonesian Journal of Fundamental Sciences*, 8(1), 84. <https://doi.org/10.26858/ijfs.v8i1.33572>