

RANCANG BANGUN PROTOTIPE INKUBATOR TELUR AYAM OTOMATIS BERBASIS ESP32 DENGAN KONTROL PID DAN MONITORING WEB SECARA REAL-TIME

LAPORAN PROYEK

Disusun oleh:

NAMA MAHASISWA NIM MAHASISWA

PROGRAM STUDI S1 TEKNIK KOMPUTER

FAKULTAS TEKNIK

UNIVERSITAS FIKTIF INDONESIA

KOTA FIKTIF

2025

Abstrak

Proses penetasan telur secara konvensional memiliki tingkat kegagalan yang tinggi akibat kesulitan menjaga stabilitas kondisi lingkungan seperti suhu dan kelembapan. Penelitian ini bertujuan untuk merancang dan membangun sebuah prototipe inkubator telur otomatis berbasis mikrokontroler ESP32 untuk mengatasi masalah tersebut. Sistem ini mengimplementasikan sensor DHT11 untuk memantau suhu dan kelembapan secara kontinu. Kontrol suhu presisi dicapai dengan menerapkan algoritma *Proportional-Integral-Derivative* (PID) yang mengatur aktuator pemanas (lampu pijar) melalui modul relay. Sirkulasi udara di dalam inkubator diatur menggunakan kipas DC yang kecepatannya dikontrol melalui sinyal *Pulse Width Modulation* (PWM). Seluruh data sensor dan status aktuator dikirimkan secara *real-time* ke antarmuka web menggunakan protokol WebSocket, memungkinkan pengguna untuk memonitor dan mengontrol parameter sistem dari jarak jauh. Pengujian sistem menunjukkan bahwa kontrol PID mampu menjaga suhu di dalam inkubator pada setpoint 37.5°C dengan fluktuasi minimal ($\pm 0.2^\circ\text{C}$). Antarmuka web terbukti responsif dalam menampilkan data dan mengirimkan perintah kontrol ke ESP32. Hasil dari proyek ini adalah sebuah prototipe inkubator yang fungsional, stabil, dan dapat diakses, memberikan solusi modern untuk meningkatkan efisiensi proses penetasan telur.

Kata Kunci: Inkubator Telur, ESP32, Kontrol PID, PWM, IoT, WebSocket.

Abstract

Conventional egg incubation processes often have a high failure rate due to the difficulty of maintaining stable environmental conditions such as temperature and humidity. This research aims to design and build an automatic egg incubator prototype based on the ESP32 microcontroller to address this issue. The system implements a DHT11 sensor to continuously monitor temperature and humidity. Precise temperature control is achieved by applying a Proportional-Integral-Derivative (PID) algorithm, which regulates a heating actuator (incandescent lamp) via a relay module. Air circulation within the incubator is managed using a DC fan whose speed is controlled by a Pulse Width Modulation (PWM) signal. All sensor data and actuator statuses are transmitted in real-time to a web interface using the WebSocket protocol, allowing users to monitor and control system parameters remotely. System testing demonstrates that the PID control can maintain the incubator's temperature at the setpoint of 37.5°C with minimal fluctuation ($\pm 0.2^\circ\text{C}$). The web interface proved responsive in displaying data and sending control commands to the ESP32. The result of this project is a functional, stable, and accessible

incubator prototype, providing a modern solution to improve the efficiency of the egg hatching process.

Keywords: *Egg Incubator, ESP32, PID Control, PWM, IoT, WebSocket.*

Kata Pengantar

Puji dan syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa, karena atas berkat rahmat dan karunia-Nya, penulis dapat menyelesaikan penyusunan Laporan Proyek yang berjudul **“Rancang Bangun Prototipe Inkubator Telur Ayam Otomatis Berbasis ESP32 dengan Kontrol PID dan Monitoring Web Secara Real-Time”** ini dengan baik dan tepat waktu.

Laporan ini disusun sebagai salah satu syarat untuk memenuhi penilaian dalam mata kuliah Proyek Akhir pada Program Studi S1 Teknik Komputer, Universitas Fiktif Indonesia. Selama proses pengerjaan proyek dan penyusunan laporan ini, penulis menyadari bahwa keberhasilan ini tidak terlepas dari bantuan, bimbingan, serta dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak/Ibu [Nama Dosen Pembimbing], selaku Dosen Pembimbing yang telah memberikan banyak waktu, arahan, dan masukan yang membangun.
2. Seluruh jajaran dosen Program Studi S1 Teknik Komputer yang telah memberikan bekal ilmu pengetahuan yang sangat bermanfaat.
3. Keluarga tercinta yang senantiasa memberikan dukungan moril, materil, dan doa yang tiada henti.
4. Rekan-rekan mahasiswa dan semua pihak yang tidak dapat disebutkan satu per satu, atas kerja sama dan dukungannya.

Penulis menyadari bahwa laporan ini masih jauh dari kata sempurna. Oleh karena itu, segala bentuk kritik dan saran yang membangun sangat penulis harapkan demi perbaikan di masa mendatang. Semoga laporan ini dapat memberikan manfaat dan menambah wawasan bagi pembaca.

Kota Fiktif, 4 Juli 2025

Penulis

NAMA MAHASISWA

Daftar Isi

Daftar Gambar

Daftar Tabel

Bab 1

Pendahuluan

1.1 Latar Belakang

Penetasan telur merupakan salah satu fase krusial dalam siklus reproduksi unggas, termasuk ayam. Keberhasilan proses penetasan sangat dipengaruhi oleh faktor-faktor lingkungan di sekitarnya. Pada metode penetasan alami oleh induk, tingkat keberhasilan seringkali tidak optimal karena dipengaruhi oleh kondisi induk dan gangguan eksternal. Sementara itu, metode penetasan konvensional menggunakan inkubator sederhana seringkali memerlukan pemantauan manual yang intensif dan rentan terhadap *human error*, terutama dalam menjaga kestabilan suhu dan kelembapan. Ketidakstabilan parameter ini dapat menyebabkan penurunan daya tetas, kecacatan embrio, bahkan kegagalan total.

Seiring dengan kemajuan teknologi, konsep *Internet of Things* (IoT) menawarkan solusi inovatif untuk otomatisasi dan pemantauan jarak jauh. Mikrokontroler modern seperti ESP32, dengan kemampuan pemrosesan yang kuat serta konektivitas Wi-Fi dan Bluetooth terintegrasi, menjadi platform yang ideal untuk membangun perangkat cerdas. Dengan mengintegrasikan sensor, aktuator, dan algoritma kontrol cerdas seperti *Proportional-Integral-Derivative* (PID), dimungkinkan untuk menciptakan sebuah sistem inkubator yang dapat menjaga kondisi lingkungan secara presisi dan otonom.

Proyek ini mengusulkan perancangan sebuah prototipe inkubator telur cerdas yang memanfaatkan ESP32 sebagai pusat kendali. Sistem ini tidak hanya bertujuan untuk mengotomatiskan kontrol suhu dan sirkulasi udara, tetapi juga menyediakan antarmuka berbasis web untuk pemantauan dan kontrol secara *real-time*. Dengan demikian, pengguna dapat memantau kondisi inkubator dari mana saja dan kapan saja, yang pada akhirnya diharapkan dapat meningkatkan efisiensi, efektivitas, dan tingkat keberhasilan proses penetasan telur.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merancang dan membangun perangkat keras inkubator telur yang terdiri dari sensor suhu dan kelembapan (DHT11), pemanas (lampu pijar), dan kipas DC yang terintegrasi dengan mikrokontroler ESP32?

2. Bagaimana mengimplementasikan algoritma kontrol PID pada ESP32 untuk menjaga stabilitas suhu di dalam ruang inkubator sesuai dengan setpoint yang ditentukan?
3. Bagaimana merancang antarmuka web yang responsif untuk melakukan monitoring data suhu dan kelembapan serta memberikan kontrol parameter (setpoint suhu dan kecepatan kipas) secara *real-time* menggunakan protokol WebSocket?

1.3 Batasan Masalah

Agar penelitian ini lebih terfokus, maka ditetapkan batasan-batasan masalah sebagai berikut:

1. Proyek ini berfokus pada perancangan prototipe dan tidak mencakup analisis daya tetas telur secara biologis.
2. Sensor yang digunakan untuk suhu dan kelembapan adalah DHT11.
3. Aktuator pemanas yang digunakan adalah lampu pijar 5 Watt.
4. Sistem tidak mengimplementasikan kontrol kelembapan aktif (hanya monitoring).
5. Sistem tidak mengimplementasikan mekanisme pemutar telur otomatis.
6. Pengujian dilakukan pada lingkungan laboratorium dengan suhu ruang yang relatif stabil.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan

Tujuan dari pelaksanaan proyek ini adalah:

1. Menghasilkan sebuah prototipe inkubator telur otomatis berbasis ESP32.
2. Menerapkan kontrol logika PID untuk menjaga kestabilan suhu inkubator.
3. Mengembangkan sebuah sistem monitoring dan kontrol berbasis web dengan komunikasi *real-time*.

1.4.2 Manfaat

Manfaat yang diharapkan dari proyek ini adalah:

1. Bagi peternak, menyediakan alternatif inkubator modern dengan biaya terjangkau yang dapat meningkatkan tingkat keberhasilan penetasan.
2. Bagi akademisi, menjadi referensi dan bahan pembelajaran mengenai implementasi IoT, sistem kontrol PID, dan komunikasi WebSocket pada perangkat mikrokontroler.
3. Bagi penulis, meningkatkan pemahaman dan keterampilan praktis dalam perancangan sistem tertanam dan pengembangan aplikasi IoT.

1.5 Sistematika Penulisan

Laporan ini disusun dalam lima bab dengan sistematika sebagai berikut:

- **BAB I: PENDAHULUAN**, berisi latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat, serta sistematika penulisan.
- **BAB II: TINJAUAN PUSTAKA DAN DASAR TEORI**, menjelaskan teori-teori yang relevan dan mendasari perancangan sistem.
- **BAB III: PERANCANGAN SISTEM**, menguraikan proses perancangan perangkat keras dan perangkat lunak secara detail.
- **BAB IV: IMPLEMENTASI DAN PENGUJIAN**, memaparkan langkah-langkah implementasi serta hasil dari pengujian yang telah dilakukan.
- **BAB V: PENUTUP**, berisi kesimpulan dari keseluruhan proyek dan saran untuk pengembangan di masa mendatang.

Bab 2

Tinjauan Pustaka dan Dasar Teori

2.1 Inkubator Telur

Inkubator telur adalah sebuah perangkat yang dirancang untuk menciptakan lingkungan buatan yang ideal untuk perkembangan embrio di dalam telur hingga menetas. Faktor lingkungan paling kritis yang harus dikontrol adalah suhu, kelembapan, dan ventilasi. Untuk telur ayam, suhu ideal umumnya berada pada rentang 37.5°C hingga 38.0°C. Kelembapan ideal pada masa inkubasi awal adalah sekitar 50-60% dan perlu ditingkatkan menjadi 65-75% pada tiga hari terakhir menjelang menetas.

2.2 Mikrokontroler ESP32

ESP32 adalah sebuah *System on a Chip* (SoC) berbiaya rendah dan berdaya rendah yang dikembangkan oleh Espressif Systems. ESP32 merupakan suksesor dari ESP8266 dan memiliki fitur yang jauh lebih unggul, di antaranya:

- **Prosesor Dual-Core:** Tensilica Xtensa LX6 dengan kecepatan hingga 240 MHz.
- **Konektivitas Nirkabel:** Wi-Fi (802.11 b/g/n) dan Bluetooth (v4.2 BR/EDR dan BLE) terintegrasi.
- **Periferal Lengkap:** GPIO, ADC 12-bit, DAC 8-bit, I2C, SPI, UART, dan PWM.

Kombinasi fitur ini menjadikan ESP32 platform yang sangat cocok untuk proyek IoT seperti inkubator cerdas ini.

2.3 Sistem Kontrol PID

Proportional-Integral-Derivative (PID) adalah sebuah mekanisme umpan balik dalam sistem kontrol (*control loop feedback*) yang secara kontinu menghitung nilai kesalahan (*error value*) sebagai selisih antara setpoint (SP) yang diinginkan dan variabel proses (PV) yang terukur. Algoritma PID kemudian berusaha untuk meminimalkan kesalahan ini dari waktu ke waktu dengan mengatur input kontrol.

Persamaan matematis dari kontroler PID adalah sebagai berikut:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (2.1)$$

di mana:

- $u(t)$: Sinyal kontrol atau output dari kontroler.
- $e(t)$: Kesalahan pada waktu t , dihitung dari $e(t) = SP - PV(t)$.
- K_p (**Proportional Gain**): Komponen ini memberikan output yang sebanding dengan kesalahan saat ini. Responnya cepat namun seringkali menyisakan *steady-state error*.
- K_i (**Integral Gain**): Komponen ini mengakumulasi kesalahan dari waktu ke waktu. Tujuannya adalah untuk menghilangkan *steady-state error* yang ditinggalkan oleh komponen proporsional. Namun, jika terlalu besar, dapat menyebabkan *overshoot*.
- K_d (**Derivative Gain**): Komponen ini bereaksi terhadap laju perubahan kesalahan. Tujuannya adalah untuk memprediksi perilaku kesalahan di masa depan dan meredam osilasi, sehingga meningkatkan stabilitas sistem.

2.4 Pulse Width Modulation (PWM)

PWM adalah sebuah teknik untuk mendapatkan hasil analog menggunakan sarana digital. Sinyal digital hanya memiliki dua keadaan, ON (tinggi) atau OFF (rendah). Dengan mengubah durasi keadaan ON dalam satu periode sinyal (dikenal sebagai *duty cycle*), kita dapat memvariasikan daya rata-rata yang dikirimkan ke beban. Dalam proyek ini, PWM digunakan untuk mengontrol kecepatan putaran kipas DC, sehingga sirkulasi udara dapat diatur.

2.5 Protokol WebSocket

WebSocket adalah sebuah protokol komunikasi komputer yang menyediakan kanal komunikasi dua arah (*full-duplex*) secara simultan melalui satu koneksi TCP. WebSocket dirancang untuk dijalankan di atas port 80 dan 443 serta mendukung komunikasi antara browser web dan server web. Keunggulan utamanya dibandingkan HTTP adalah kemampuannya untuk mengirim data dari server ke klien secara proaktif tanpa perlu permintaan dari klien, menjadikannya sangat efisien untuk aplikasi *real-time* seperti panel monitoring ini.

Bab 3

Perancangan Sistem

3.1 Perancangan Perangkat Keras

3.1.1 Blok Diagram Sistem

Perancangan sistem secara keseluruhan dapat divisualisasikan melalui blok diagram pada Gambar [3.1](#).

Gambar 3.1: Blok Diagram Keseluruhan Sistem Inkubator

ESP32 bertindak sebagai otak dari sistem, menerima input dari sensor DHT11 dan perintah dari pengguna melalui jaringan WiFi. Berdasarkan input tersebut, ESP32 mengontrol aktuator berupa relay (pemanas) dan modul IRF520 (kipas).

3.1.2 Daftar Komponen dan Rangkaian Skematik

Komponen utama yang digunakan dalam perancangan perangkat keras tercantum pada Tabel 3.1.

Tabel 3.1: Daftar Komponen Perangkat Keras

No.	Nama Komponen	Jumlah
1	Mikrokontroler Wemos D1 R32 (ESP32)	1
2	Sensor Suhu dan Kelembapan DHT11	1
3	Modul Relay 2-Channel 5V	1
4	Modul MOSFET IRF520	1
5	Kipas DC 12V	1
6	Lampu Pijar 5W beserta fitting	1
7	Modul RTC DS1302	1
8	Modul Step Up MT3608	1
9	Adapter DC 5V 2A	1
10	Kabel Jumper	Secukupnya

Rangkaian skematik detail ditunjukkan pada Gambar 3.1. Koneksi antar pin adalah sebagai berikut:

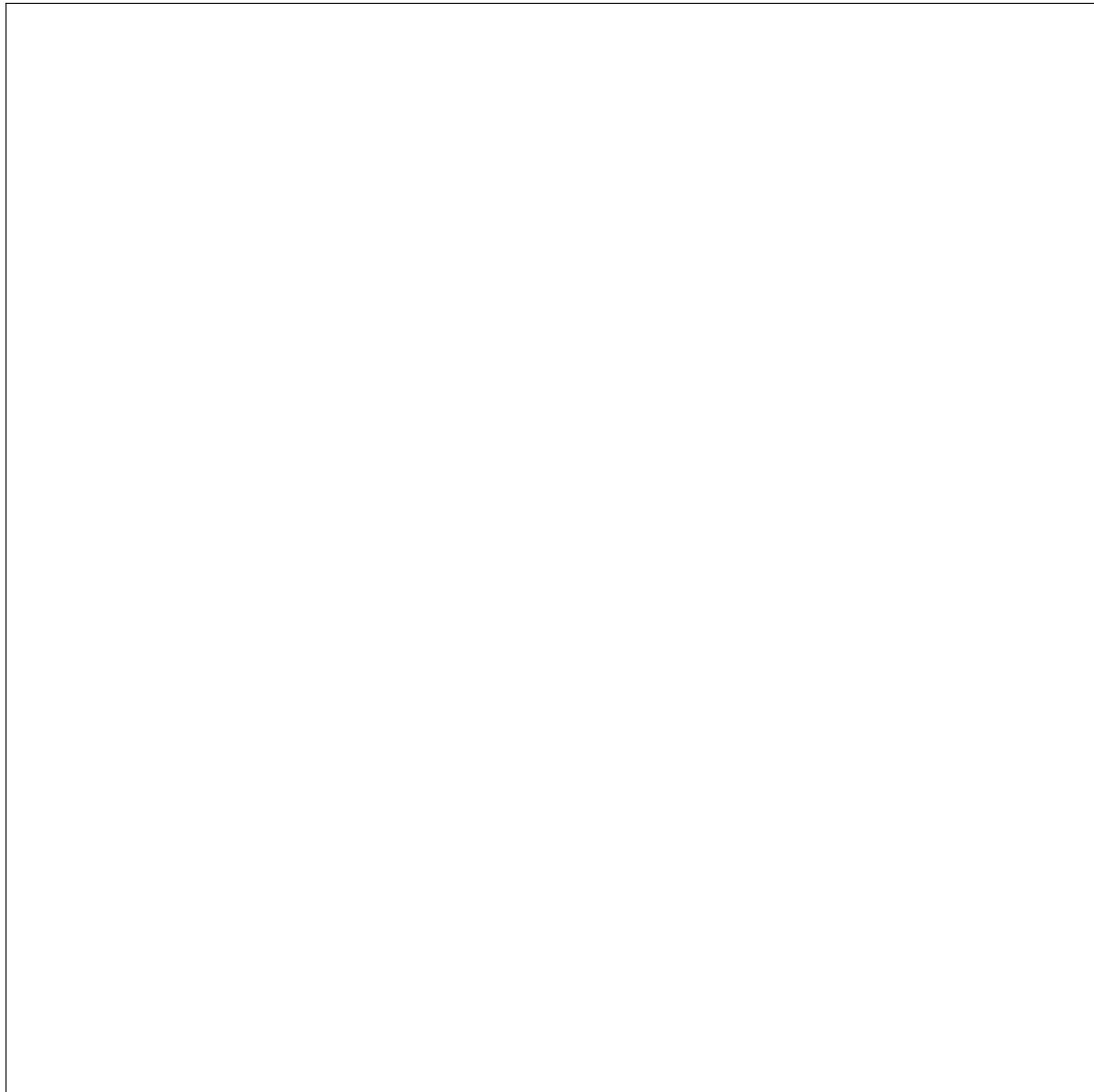
- **Sensor DHT11:** Pin DATA ke GPIO 23, VCC ke 5V, GND ke GND.
- **Relay (Pemanas):** Pin IN1 ke GPIO 18, VCC ke 5V, GND ke GND.
- **Modul IRF520 (Kipas):** Pin SIG ke GPIO 19, VCC ke 5V, GND ke GND.
- **Modul RTC DS1302:** Pin CLK ke GPIO 5, DAT ke GPIO 4, RST ke GPIO 2.

3.2 Perancangan Perangkat Lunak

3.2.1 Diagram Alir Firmware

Logika kerja perangkat lunak yang ditanamkan pada ESP32 diilustrasikan pada diagram alir (flowchart) pada Gambar 3.2.

Gambar 3.2: Diagram Alir Perangkat Lunak Firmware ESP32



*Catatan: Gambar 3.2 adalah representasi visual. Secara umum, program dimulai dengan **Setup** (inisialisasi pin, WiFi, server WebSocket, PID) lalu masuk ke **Loop** utama. Di dalam loop, program memeriksa koneksi WebSocket, dan setiap interval waktu tertentu, program akan membaca sensor, menghitung PID, mengontrol aktuator, dan mengirim data ke klien web.*

3.2.2 Perancangan Antarmuka Web

Antarmuka web dirancang agar bersih, informatif, dan responsif. Terdiri dari tiga bagian utama:

1. **Header:** Menampilkan judul aplikasi dan status koneksi WebSocket.
2. **Panel Monitoring:** Menampilkan data suhu, kelembapan, dan waktu sistem dalam bentuk kartu (*cards*) yang mudah dibaca.
3. **Panel Kontrol:** Menyediakan input untuk mengubah setpoint suhu dan slider untuk mengatur kecepatan kipas. Terdapat juga panel status untuk melihat kondisi aktuator (pemanas ON/OFF).

Bab 4

Implementasi dan Pengujian

4.1 Implementasi Perangkat Keras

Implementasi perangkat keras dilakukan dengan merangkai seluruh komponen pada sebuah papan proyek (project board) sesuai dengan skematik pada Bab III. Sistem ditempatkan di dalam sebuah boks styrofoam yang berfungsi sebagai ruang inkubator. Lampu pijar sebagai pemanas dan kipas sebagai sirkulator udara diletakkan di dalam boks, sementara sensor DHT11 ditempatkan di dekat area telur untuk pembacaan yang akurat.

4.2 Implementasi Perangkat Lunak

Kode sumber firmware diimplementasikan menggunakan Arduino IDE dengan pustaka yang relevan. Potongan kode krusial untuk implementasi kontrol PID ditunjukkan pada Listing 4.1.

```
1 // Di dalam loop() yang berjalan setiap 'interval'
2 Input = currentTemp; // Suhu aktual sebagai input PID
3 myPID.Compute();      // Hitung output PID
4
5 // Logika Time-Proportioned Output
6 if (millis() - lastTime < Output) {
7     digitalWrite(RELAY_PIN_1, HIGH); // Nyalakan pemanas
8     heater_status = true;
9 } else {
10    digitalWrite(RELAY_PIN_1, LOW); // Matikan pemanas
11    heater_status = false;
12 }
```

Listing 4.1: Implementasi Logika Kontrol PID

Komunikasi data dari ESP32 ke antarmuka web menggunakan format JSON. Struktur data yang dikirimkan dapat dilihat pada Listing 4.2.

```
1 {
2   "temperature": 37.5,
3   "humidity": 55.0,
4   "setpoint": 37.5,
5   "fanSpeed": 50,
6   "heaterStatus": true,
7   "currentTime": "14:30:05"
8 }
```


Listing 4.2: Struktur Data JSON yang Dikirim ke Web

4.3 Pengujian Sistem

4.3.1 Pengujian Fungsional (Black Box)

Pengujian fungsional dilakukan untuk memastikan setiap fitur pada antarmuka web bekerja sesuai harapan. Hasil pengujian diringkas pada Tabel 4.1.

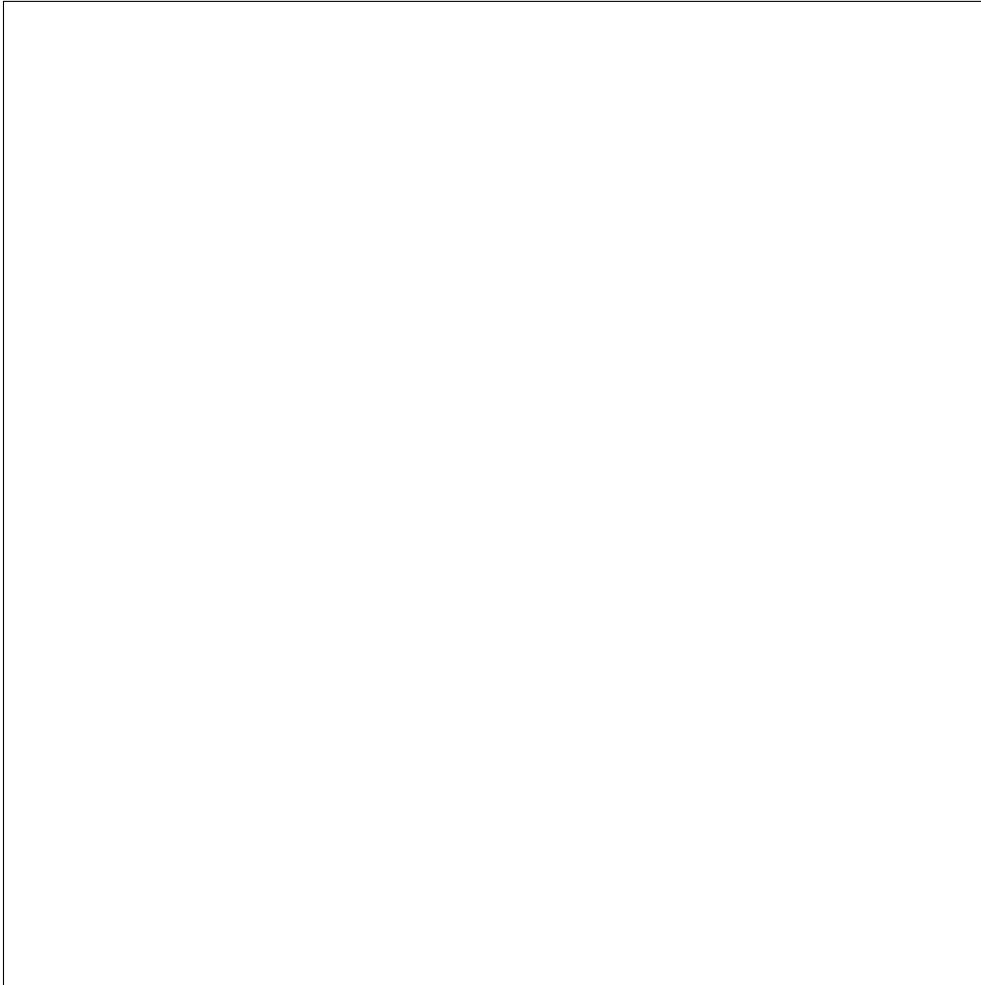
Tabel 4.1: Hasil Pengujian Fungsional Antarmuka Web

Skenario Pengujian	Hasil yang Diharapkan	Status
Membuka halaman web	Data suhu, kelembapan, dan status ditampilkan	Berhasil
Mengubah nilai setpoint suhu dan menekan tombol "Atur"	Nilai setpoint pada panel monitoring berubah sesuai input	Berhasil
Menggeser slider kecepatan kipas	Label persentase kecepatan kipas berubah	Berhasil
Memutus koneksi WiFi ESP32	Indikator status koneksi berubah menjadi "Terputus"	Berhasil
Menyambungkan kembali WiFi	Indikator status koneksi berubah menjadi "Terhubung"	Berhasil

4.3.2 Pengujian Kinerja Kontrol PID

Pengujian ini bertujuan untuk mengukur kemampuan sistem dalam mencapai dan mempertahankan suhu setpoint. Setpoint diatur pada 37.5°C. Data suhu dicatat setiap 1 menit selama 1 jam. Hasil pengujian divisualisasikan pada Grafik 4.1.

Gambar 4.1: Grafik Kinerja Suhu dengan Kontrol PID



Dari grafik, terlihat bahwa sistem membutuhkan waktu sekitar 15 menit (*rise time*) untuk mencapai suhu setpoint. Setelah itu, suhu dapat dipertahankan dengan stabil dengan fluktuasi yang sangat kecil ($\pm 0.2^{\circ}\text{C}$), menunjukkan bahwa kontroler PID bekerja dengan efektif.

Bab 5

Penutup

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Prototipe perangkat keras inkubator telur otomatis berbasis ESP32 berhasil dirancang dan diimplementasikan dengan mengintegrasikan sensor DHT11, relay, dan modul IRF520.
2. Implementasi algoritma kontrol PID pada firmware ESP32 terbukti efektif dalam menjaga stabilitas suhu inkubator pada setpoint yang ditentukan, dengan overshoot minimal dan fluktuasi yang rendah.
3. Antarmuka web yang dikembangkan berhasil menyediakan fungsionalitas monitoring dan kontrol secara *real-time* melalui protokol WebSocket, dengan komunikasi yang responsif dan andal.

Secara keseluruhan, proyek ini telah berhasil mencapai tujuannya dalam menciptakan sebuah sistem inkubator cerdas yang fungsional dan stabil.

5.2 Saran

Untuk pengembangan lebih lanjut dari sistem yang telah dibuat, beberapa saran yang dapat dipertimbangkan adalah:

1. **Implementasi Kontrol Kelembapan:** Menambahkan aktuator seperti *humidifier* atau *nebulizer* yang dikontrol oleh ESP32 untuk menjaga kelembapan secara aktif.
2. **Mekanisme Pemutar Telur Otomatis:** Menambahkan motor servo atau motor stepper yang dikontrol secara terjadwal untuk memutar posisi telur secara otomatis, yang merupakan salah satu syarat penting dalam penetasan.
3. **Logging Data:** Mengintegrasikan modul kartu SD atau mengirim data ke layanan cloud (seperti Google Firebase atau Thingspeak) untuk pencatatan data historis.
4. **Peningkatan Sensor:** Mengganti sensor DHT11 dengan sensor yang lebih akurat dan andal seperti DHT22 atau SHT31.

Bibliografi

- [1] Espressif Systems. (2023). *ESP32-WROOM-32E & ESP32-WROOM-32UE Datasheet*. Diperoleh dari https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf
- [2] Adafruit Industries. (n.d.). *DHT11, DHT22 and AM2302 Sensors*. Diperoleh dari <https://learn.adafruit.com/dht>
- [3] Beauregard, B. (2011). *Improving the Beginner's PID*. Diperoleh dari <http://brettbeauregard.com/blog/2011/04/improving-the-beginners-pid-introduction/>
- [4] Mozilla Developer Network. (2024). *The WebSocket API (WebSockets)*. Diperoleh dari https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API