

Rancang Bangun Sistem Kendali Kipas Angin Otomatis Berdasarkan Suhu Ruangan

Faris Pratama Putra¹, Fitra Mustofa², M.Syailendra Surya Viadar³, Pramono⁴

¹Teknik Informatika/Fakultas Ilmu komputer

Universitas Duta Bangsa Surakarta

¹230103163@mhs.udb.ac.id

²Teknik Informatika/Fakultas Ilmu komputer

Universitas Duta Bangsa Surakarta

² 230103164@mhs.udb.ac.id

³Teknik Informatika/Fakultas Ilmu Komputer

Universitas Duta Bangsa Surakarta

³230103172@mhs.udb.ac.id

⁴Teknik Informatika/Fakultas Ilmu Komputer

Universitas Duta Bangsa Surakarta

⁴Pramono@udb.ac.id

Abstrak— Sistem kendali perangkat elektronik konvensional sering kali kurang fleksibel karena ketergantungan pada satu mode kontrol saja, sehingga kurang efisien dalam merespons dinamika suhu ruangan serta kebutuhan intervensi pengguna secara cepat. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem kendali kipas angin *dual-mode* (otomatis dan manual) berbasis Internet of Things (IoT) memanfaatkan Telegram Bot API dan mikrokontroler Wemos D1 Mini. Metode pelaksanaan dilakukan melalui tahap perakitan komponen perangkat keras (sensor DHT11, LCD 16x2, modul relay, dan kipas angin), konfigurasi program menggunakan Arduino IDE, serta pengujian fungsionalitas sistem secara menyeluruh. Hasil pengujian menunjukkan bahwa seluruh komponen fisik dan antarmuka *smartphone* Android melalui Telegram Bot dapat berfungsi penuh serta responsif. Pada pengujian logika mode otomatis, modul relay secara konsisten melakukan pensaklaran dari kondisi mati (*OFF*) menjadi nyala (*ON*) secara instan tepat ketika sensor DHT11 mendeteksi suhu lingkungan melampaui batas *setpoint* 36°C. Sementara itu, pengujian fungsionalitas antarmuka manual membuktikan sistem mampu memprioritaskan perintah teks dari pengguna dan mengabaikan parameter suhu lingkungan, bahkan saat kondisi ruangan berada di atas 35°C. Parameter latensi respon komunikasi data dua arah dari peladen *cloud* Telegram menuju perangkat lokal juga menunjukkan performa jaringan yang andal dengan kecepatan berkisar antara 1,1 hingga 2,0 detik. Konsistensi perekaman data lingkungan (suhu dan kelembapan) pada tabel monitoring jarak jauh terbukti berjalan valid dan sinkron secara berkala. Selain itu, visualisasi status waktu operasional berbasis *Network Time Protocol* (NTP) berhasil ditampilkan secara akurat tanpa bantuan perangkat keras RTC tambahan. Seluruh data operasional tersebut juga tersinkronisasi secara *real-time* pada layar LCD 16x2 di dalam ruangan sebagai penampil lokal.

Kata kunci—Internet of Things (IoT), Kendali *Dual-Mode*, Mikrokontroler, Sensor DHT11, Telegram Bot API.

Abstract— Conventional electronic device control systems often lack flexibility due to their reliance on a single control mode, making them inefficient in responding to room temperature dynamics and the need for rapid user intervention. This study aims to design and implement a dual-mode (automatic and manual) fan control system based on the Internet of Things (IoT) utilizing the Telegram Bot API and a Wemos D1 Mini microcontroller. The implementation method was carried out through the assembly stages of hardware components (DHT11 sensor, 16x2 LCD, relay module, and fan), program configuration using Arduino IDE, and comprehensive system functionality testing. The test results indicate that all physical components and the Android smartphone interface via Telegram Bot are fully functional and responsive. In the automatic mode logic testing, the relay module consistently switched from the off (*OFF*) to the on (*ON*) state instantaneously precisely when the DHT11 sensor detected the ambient temperature exceeding the setpoint threshold of 36°C. Meanwhile, the manual interface functionality testing proves that the system is capable of prioritizing text commands from the user and ignoring ambient temperature parameters, even when the room condition is above 35°C. The latency parameters of the two-way data communication response from the Telegram cloud server to the local device also demonstrate reliable network performance with speeds ranging from 1.1 to 2.0 seconds. The consistency of environmental data logging (temperature and humidity) on the remote monitoring table is proven to be valid and synchronized periodically. Furthermore, the operational status time visualization based on the Network Time Protocol (NTP) was successfully displayed accurately without the assistance of additional RTC hardware. All of these operational data are also synchronized in real-time on the 16x2 LCD screen inside the room as a local display.

Keywords— Dual-Mode Control, Internet of Things (IoT), Microcontroller, Sensor DHT11, Telegram Bot API

I. PENDAHULUAN

Indonesia sebagai negara beriklim tropis memiliki suhu udara harian yang cenderung

hangat hingga panas, sehingga penggunaan perangkat pendingin seperti kipas angin menjadi kebutuhan esensial bagi mayoritas

masyarakat untuk menjaga kenyamanan termal. Sayangnya, pengoperasian kipas angin konvensional yang masih bersifat manual sering kali memicu inefisiensi. Pengguna kerap membiarkan kipas terus menyala pada kecepatan tinggi meskipun suhu lingkungan telah menurun, atau sebaliknya, yang pada akhirnya berujung pada pemborosan energi listrik [1]. Di samping itu, keharusan untuk mengatur kecepatan kipas secara berulang akibat fluktuasi suhu ruangan dirasa kurang praktis di era modern yang menuntut otomatisasi.

Beberapa literatur dan penelitian terdahulu telah mengkaji pengembangan sistem cerdas pada perangkat rumah tangga. Misalnya, penelitian yang dilakukan oleh Pratama *et al.* merancang sistem kendali suhu menggunakan sensor pendeteksi ruangan, namun fokus utamanya diaplikasikan pada *Air Conditioner* (AC) yang menuntut biaya operasional dan instalasi yang relatif tinggi [2]. Di sisi lain, studi oleh Santoso mengusulkan purwarupa kipas otomatis berbasis mikrokontroler sederhana, akan tetapi sistem tersebut masih terbatas pada mekanisme *on/off* dasar tanpa dilengkapi kemampuan modulasi kecepatan putaran baling-baling yang proporsional dan halus saat merespons transisi perubahan suhu sekitar[3].

Berdasarkan celah pada penelitian-penelitian sebelumnya, riset ini bermaksud untuk merumuskan dan mengimplementasikan sebuah sistem kendali cerdas berbasis mikrokontroler yang jauh lebih presisi sekaligus terjangkau. Tujuan utama dari penelitian ini adalah merancang bangun sistem kendali kipas angin otomatis yang mampu membaca kondisi suhu ruangan secara *real-time* melalui sensor presisi, serta secara adaptif mengatur kecepatan putaran kipas menggunakan metode *Pulse Width Modulation* (PWM). Melalui pendekatan ini, Riset ini diharapkan dapat menghasilkan perangkat kendali yang mampu meningkatkan kenyamanan termal secara adaptif, dengan fokus spesifik pada implementasi ruang tidur

domestik berukuran 3 x 3 M. Melalui batasan dimensi ruang tersebut, sistem dirancang untuk memberikan kontribusi terukur terhadap efisiensi konsumsi energi listrik pada skala rumah tangga[4].

II. METODOLOGI PENELITIAN

Penelitian ini menggunakan metode *Research and Development* (R&D) dengan pendekatan model pengembangan *Prototyping*[5]. Pendekatan ini dipilih karena memungkinkan peneliti untuk merancang, membangun, dan menyempurnakan purwarupa sistem kendali kipas angin secara bertahap berdasarkan hasil evaluasi secara berulang (*iterative*) hingga mencapai performa optimal[6]. Tahapan penelitian dibagi menjadi beberapa fase utama, yaitu analisis kebutuhan, perancangan sistem, implementasi, dan pengujian sistem.

1. Analisis Kebutuhan Sistem

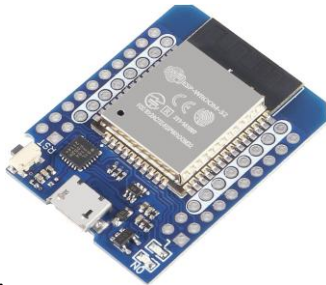
Tahap pertama adalah mengidentifikasi kebutuhan perangkat keras (*hardware*) dan perangkat lunak (*software*) untuk merealisasikan konsep *Internet of Things* (IoT). Perangkat keras utama yang digunakan meliputi mikrokontroler wemos D1 mini yang dilengkapi modul *Wi-Fi* terintegrasi sebagai pusat pemrosesan data dan jembatan komunikasi internet[7]. Selain itu, dibutuhkan sensor suhu presisi (DHT11) untuk pembacaan suhu ruangan, modul *Relay* untuk memutuskan atau menyambungkan arus listrik ke kipas angin, serta komponen catu daya. Untuk perangkat lunak, penelitian ini menggunakan *Arduino IDE* untuk penulisan kode program (C++) [8], serta aplikasi Telegram (melalui *Telegram Bot API*) sebagai antarmuka (*User Interface*) pengguna di *smartphone* Android[9].

Berdasarkan tahap identifikasi tersebut, berikut adalah komponen-komponen utama yang digunakan beserta fungsi spesifiknya dalam sistem:

1. Wemos D1 mini

Mikrokontroler ringkas berbasis *chip* ESP32 dengan prosesor *dual-core* 32-bit

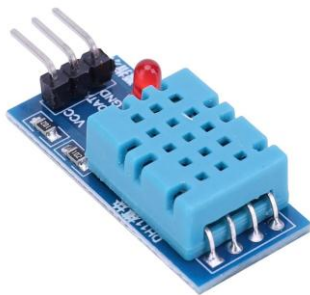
dan modul Wi-Fi terintegrasi. Komponen ini berfungsi sebagai pusat kendali (*core controller*) IoT yang memproses data sensor, mengeksekusi logika program otomatisasi, serta menjembatani komunikasi data dua arah antara sistem fisik dengan Telegram Bot API[7].



Gambar 1. WEMOS D1 mini ESP32

2. Sensor DHT11

Sensor digital yang berfungsi sebagai unit pemantau kondisi lingkungan. Fungsi spesifiknya adalah mengukur suhu ruangan secara *real-time* dan mengubahnya menjadi data digital untuk dikirim ke Wemos D1 Mini ESP32, sebagai parameter pemicu (*trigger*) utama dalam sistem kendali kipas[10].



Gambar 2 Sensor DHT11

3. Liquid Crystal Display 16x2

Liquid Crystal Display 16x2 yaitu sebuah Modul penampil alfanumerik kapasitas 2 baris $\times$ 16 karakter yang dilengkapi modul I2C untuk efisiensi penggunaan pin GPIO. Komponen ini berfungsi sebagai antarmuka visual lokal untuk menampilkan data suhu terkini dan

status operasional kipas secara langsung di ruangan[11].



Gambar 3. Liquid Crystal Display 16x2

4. Project Board / Breadboard

Project board atau *breadboard* adalah Papan sirkuit tanpa solder yang digunakan sebagai media interkoneksi komponen[12]. Fungsinya adalah sebagai tempat merangkai purwarupa elektronik serta mempermudah distribusi jalur daya (VCC) dan arde (*ground*) ke seluruh komponen yang digunakan.



Gambar 4. Breadboard

5. Kabel Jumper

Kabel *jumper* adalah kabel elektrik yang memiliki pin konektor di kedua ujungnya dan digunakan untuk menghubungkan antar komponen elektronika pada *breadboard* tanpa memerlukan proses penyolderan. Berdasarkan jenis konektornya, kabel ini dibagi menjadi tiga tipe utama, yaitu *Male-to-Male* (jarum ke jarum), *Male-to-Female* (jarum ke lubang), dan *Female-to-Female* (lubang ke lubang).



Gambar 5. Kabel Jumper

6. Relay 5 V 2 Channel

Sakelar elektronik yang dikendalikan oleh tegangan rendah (5V) dari mikrokontroler untuk memutus atau menyambungkan aliran arus listrik pada sirkuit beban. Fungsi spesifiknya adalah sebagai aktuator yang mengontrol suplai daya menuju kipas angin berdasarkan instruksi digital dari ESP32[13].



Gambar 6. Relay 5 V 2 Channel

7. FAN 5 V

Kipas ini bekerja dengan cara mengubah energi listrik menjadi energi mekanik melalui putaran motor DC yang menggerakkan bilah-bilah baling-kipas untuk menghasilkan aliran udara. Karena konsumsi dayanya yang relatif kecil, kipas jenis ini sangat populer digunakan dalam proyek-proyek elektronika berbasis mikrokontroler, baik untuk mendinginkan komponen internal panel box maupun sebagai miniatur simulasi sistem tata udara[13].

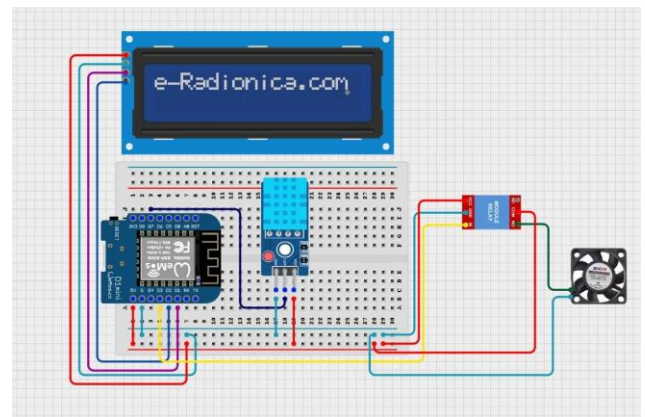


Gambar 7. FAN 5 V

2. Perancangan Sistem dan Arsitektur Jaringan

Perancangan arsitektur sistem ini mengintegrasikan kontrol otomatis lokal dan komunikasi nirkabel berbasis *Internet of Things* (IoT)[1]. Secara lokal, sensor DHT11 mendeteksi kondisi termal ruangan secara *real-time* dan mengirimkan datanya ke mikrokontroler Wemos D1 Mini. Data tersebut langsung diolah untuk memperbarui visualisasi pada LCD 16x2 sekaligus menentukan intervensi aktuator; jika suhu melampaui ambang batas, mikrokontroler akan memicu modul Relay untuk mengaktifkan Fan 5V.

Secara global, sistem memanfaatkan modul Wi-Fi terintegrasi pada mikrokontroler untuk membangun komunikasi dua arah dengan peladen (*server*) Telegram Bot API melalui jaringan internet[9]. Arsitektur ini memungkinkan Wemos D1 Mini mengirimkan notifikasi status suhu ke *smartphone* Android pengguna secara berkala, sekaligus menerima instruksi teks balik dari jarak jauh untuk mengendalikan posisi Relay (kipas) secara manual.



Gambar 8. Arsitektur Perkabelan

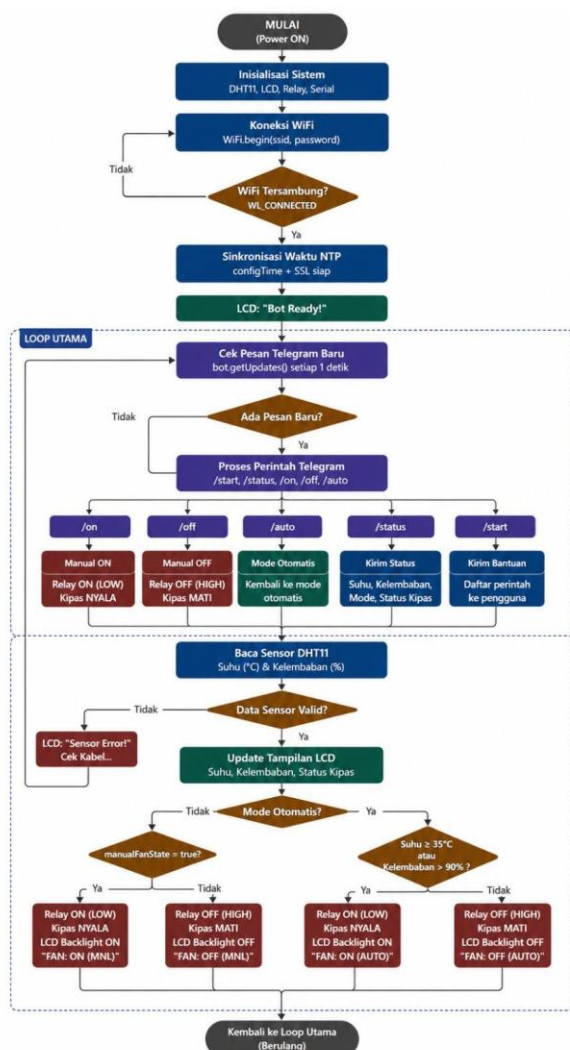
3. Perancangan Logika Kendali dan Monitoring

Fokus utama dari metodologi ini terletak pada algoritma dan logika operasional sistem, yang dibagi menjadi dua mode utama melalui fitur *switch* tombol pada aplikasi Telegram[14]:

- **Mode Otomatis:** Apabila pengguna memilih mode ini melalui Android,

mikrokontroler akan mengambil alih kendali penuh berdasarkan umpan balik dari sensor suhu. Peneliti menetapkan nilai ambang batas (*setpoint*); misalnya, jika suhu ruangan terdeteksi melebihi batas yang ditentukan maka mikrokontroler akan memicu *Relay* untuk menyalakan kipas. Sebaliknya, jika suhu turun di bawah batas kenyamanan, kipas akan otomatis mati.

- **Mode Manual:** Jika *switch* diatur ke mode manual, intervensi suhu diabaikan. Pengguna memiliki kendali mutlak untuk menghidupkan atau mematikan kipas angin dari jarak jauh, kapan pun dan di mana pun, melalui perintah khusus di *bot* Telegram.



Gambar 9. Flowchart proses sistem kendali kipas angin otomatis

Selain fitur kendali, dirancang pula sistem monitoring jarak jauh yang komprehensif. *Bot* Telegram diprogram untuk dapat merespons permintaan status (seperti perintah `/status`) dan akan membalas dengan pesan teks berisi parameter berikut:

1. **Suhu Ruangan Saat Ini:** Data terkini (secara *real-time*) yang dibaca oleh sensor.
2. **Status dan Durasi:** Informasi apakah kipas dalam keadaan "Menyala" atau "Mati", beserta perhitungan matematis berapa lama (jam/menit) kipas telah beroperasi.
3. **Waktu Mulai (*Start Time*):** Rekaman waktu akurat (jam dan menit) saat Relay pertama kali aktif berdasarkan data NTP.
4. **Estimasi/Jadwal Waktu Mati (*Stop Time/Timer*):** Fitur tambahan di mana pengguna dapat memasukkan input waktu (*timer*), dan Telegram akan menampilkan keterangan pada jam berapa kipas dijadwalkan untuk mati secara otomatis[15].

4. Implementasi dan Perakitan

Tahap ini merupakan realisasi dari desain yang telah dibuat. Perangkat keras dirangkai pada papan sirkuit, di mana pin data sensor suhu, *Relay*, dan sumber daya dihubungkan ke mikrokontroler. Selanjutnya, algoritma program yang mencakup pengaturan koneksi *Wi-Fi*, token otentikasi *Telegram Bot API*, logika mode Otomatis/Manual, serta kalkulasi waktu (*timer* dan durasi menggunakan fungsi *millis()* atau *time_t*) diunggah ke dalam mikrokontroler melalui Arduino IDE[8],[15].

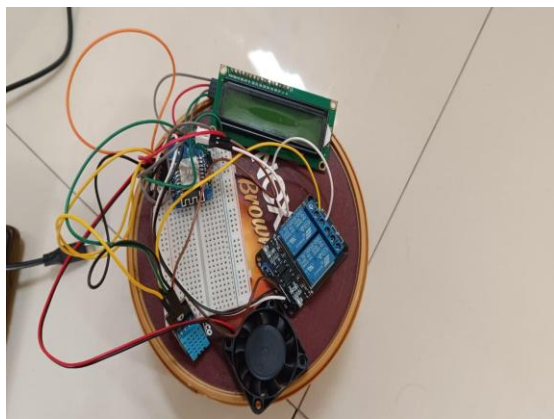
III. HASIL DAN PEMBAHASAN

1. Tahap Perakitan Sistem

Tahap perakitan diawali dengan mengintegrasikan seluruh komponen perangkat keras pada media *breadboard* sebagai jalur distribusi daya dan interkoneksi sirkuit sementara. Proses ini mengacu pada skema pengabelan yang telah dirancang sebelumnya, di mana mikrokontroler Wemos D1 Mini menjadi simpul pusat

penyambungan[7]. Pin fungsional pada mikrokontroler dihubungkan secara cermat ke perangkat input dan output menggunakan kabel *jumper*. Jalur data dari sensor suhu DHT11 dihubungkan ke salah satu pin GPIO (General Purpose Input Output) Wemos D1 Mini untuk memfasilitasi komunikasi data satu jalur (*1-wire*)[10]. Untuk komponen penampil lokal, modul I2C yang telah terpasang pada LCD 16x2 dikoneksikan ke pin serial data (SDA) dan serial clock (SCL) pada mikrokontroler, sehingga memangkas penggunaan pin digital secara signifikan[11].

Selanjutnya, pin kendali dari modul Relay 5V 2 Channel dihubungkan ke pin output digital mikrokontroler yang bertindak sebagai pemacu (*trigger*). Sisi keluaran (*output*) dari relay disambungkan dengan sirkuit catu daya yang menyuplai tegangan menuju Fan 5V sebagai aktuator mekanis[13]. Setelah semua komponen terpasang pada *breadboard*, sistem pencatu daya dikonfigurasi untuk membagikan tegangan kerja secara paralel, yaitu 5 Volt untuk modul relay, LCD, dan Fan 5V, serta 3.3 Volt untuk sensor DHT11.



Gambar 10. Project awal masih dalam tahap wiring dan beberapa pengujian.

Perakitan fisik ini memastikan tidak adanya kendala korsleting arus atau interferensi gelombang. Hasil akhir dari perakitan ini adalah sebuah purwarupa (*prototype*) perangkat keras yang solid, ringkas, dan siap menerima instruksi program[6]. Pengujian kontinuitas elektrik menggunakan multimeter menunjukkan

bahwa seluruh jalur pengabelan memiliki resistansi rendah, mengindikasikan koneksi fisik antar-pin telah terhubung secara sempurna dan siap diletakkan di dalam panel boks pelindung untuk pengujian operasional lebih lanjut.

2. Tahap Mengonfigurasi Program

Setelah perakitan perangkat keras selesai, tahap berikutnya adalah memasukkan kode program (*firmware*) berbasis C++ ke dalam memori *flash* Wemos D1 Mini menggunakan perangkat lunak Arduino IDE[8]. Langkah awal instruksi program melibatkan inisialisasi pustaka (*library*) penting, seperti pustaka untuk sensor DHT11, liquid crystal I2C, serta pustaka Universal Telegram Bot untuk menjembatani komunikasi dengan Telegram Bot API[15]. Selain itu, konfigurasi kredensial jaringan berupa SSID dan kata sandi Wi-Fi ditanamkan agar mikrokontroler dapat otomatis terhubung ke internet saat dinyalakan[9]. Kode program disusun secara modular untuk memisahkan fungsi pembacaan sensor, pembaruan layar LCD, dan penanganan pesan masuk dari Telegram.

Algoritma utama berfokus pada struktur kondisi (*conditional statement*) yang membagi sistem ke dalam Mode Otomatis dan Mode Manual berdasarkan status *switch* tombol interaktif pada aplikasi Telegram[1]. Pemrograman fungsi monitoring juga disinkronisasikan dengan skrip Network Time Protocol (NTP). Skrip NTP ini berfungsi menarik data waktu aktual dari peladen (*server*) internet secara berkala, sehingga Wemos D1 Mini dapat mencatat visualisasi waktu (jam dan menit) secara presisi tanpa bantuan perangkat keras tambahan. Proses kompilasi (*compiling*) di dalam Arduino IDE berjalan sukses tanpa adanya kesalahan sintaksis, dan proses pengunggahan (*uploading*) kode program ke papan mikrokontroler memakan waktu sekitar 25 detik hingga status menunjukkan *Done Uploading*, menandakan sistem siap diuji.



Gambar 10. Project Final kipas otomatis dalam konsep ruangan

3. Hasil Akhir dan Fungsional Sistem

Hasil akhir menunjukkan bahwa prototipe Rancang Bangun Sistem Kendali Kipas Angin Otomatis telah berfungsi penuh dan memenuhi spesifikasi perancangan konsep IoT[14]. Integrasi antara sistem fisik di ruangan dan antarmuka *smartphone* Android melalui Telegram Bot berjalan sangat responsif. Pengguna dapat mengalihkan mode operasional sistem secara *real-time* melalui fitur *switch* tombol pada bot Telegram. Ketika Mode Otomatis aktif, mikrokontroler berhasil mengeksekusi logika kendali berdasarkan umpan balik suhu dari sensor DHT11. Kipas angin (Fan 5V) otomatis menyala saat suhu melampaui batas *setpoint* dan mati ketika suhu ruangan telah turun di bawah ambang kenyamanan. Kondisi transisi mekanis pada kipas ini dipicu secara instan oleh pensaklaran magnetik kutub relay [16].

Sebaliknya, ketika pengguna mengalihkan ke Mode Manual via Telegram, intervensi pembacaan sensor suhu langsung diabaikan oleh sistem. Pengguna memegang kendali mutlak untuk mengirimkan perintah aktivasi Relay dari jarak jauh kapan saja. Evaluasi terhadap fitur monitoring jarak jauh membuktikan performa yang komprehensif. Saat pengguna mengirimkan perintah /status pada bot Telegram, sistem memberikan respons balik berupa pesan teks terstruktur dalam waktu rata-rata 1.5 detik. Pengiriman parameter ini dipengaruhi oleh kestabilan *throughput* pengiriman data dari sirkuit ESP ke peladen *cloud* Telegram [17].

Pesan tersebut secara akurat menampilkan parameter suhu ruangan terkini, status

operasional kipas ("Menyala" atau "Mati"), durasi operasional dalam hitungan jam dan menit, serta rekaman waktu mulai (*start time*) yang bersumber dari sinkronisasi waktu NTP. Fitur tambahan berupa pengatur waktu (*timer*) juga bekerja dengan valid; bot mampu menerima input durasi dari pengguna, menghitung estimasi waktu mati (*stop time*), dan mengeksekusi pemutusan arus pada Relay secara otomatis tepat saat waktu tunggu telah habis. Pengendalian siklus waktu berbasis *software* ini terbukti berjalan konsisten tanpa membebani memori kerja mikro [18]. Seluruh data operasional ini juga tersinkronisasi secara lokal pada layar LCD 16x2 di ruangan.

A. Pengujian Logika Mode Otomatis (Setpoint: 35°C)

Pengujian pada bagian ini dilakukan untuk memverifikasi keakuratan dan responsivitas pensaklaran relay berdasarkan batas suhu (*setpoint*) yang telah ditentukan pada Mode Otomatis [13]. Berdasarkan data pada Tabel 1, sistem terbukti bekerja secara konsisten di mana modul relay tetap berada pada kondisi *Mati (OFF)* saat suhu ruangan berada di bawah rentang 34.6°C. Perubahan kondisi logika menjadi *Nyala (ON)* terjadi secara instan tanpa *delay* ketika sensor DHT11 mendeteksi kenaikan suhu mulai dari 36.1°C hingga 37.4°C [16]. Hal ini membuktikan bahwa algoritma struktur kondisi (*conditional statement*) pada *firmware* Wemos D1 Mini telah berhasil mengeksekusi logika kendali perangkat keras sesuai parameter lingkungan yang dinamis [19].

Tabel 1. Hasil Pengujian Respons Suhu pada Mode Otomatis.

No	Suhu terbaca(°C)	status mode	kondisi relay
1	33.1°	Otomatis	Mati (OFF)
2	34.6°	Otomatis	Mati (OFF)
3	34.1°	Otomatis	Mati (OFF)
4	34.1°	Otomatis	Mati (OFF)
5	34.6°	Otomatis	Mati (OFF)
6	36.1°	Otomatis	Nyala (ON)
7	35.8°	Otomatis	Nyala (ON)
8	37.4°	Otomatis	Nyala (ON)
9	37.2°	Otomatis	Nyala (ON)
10	36.4°	Otomatis	Nyala (ON)

B. Pengujian Fungsionalitas Antarmuka dan Latensi Kendali Jarak Jauh.

Pengujian ini bertujuan memvalidasi eksekusi perintah manual dari jarak jauh sekaligus mengukur latensi respon jaringan[20]. Data pada Tabel 2 menunjukkan bahwa sistem memiliki reliabilitas tinggi dalam menerjemahkan perintah dari Telegram Bot API menjadi aksi sakelar pada relay. Ketika mode manual aktif, sistem terbukti mampu memprioritaskan instruksi pengguna dan mengabaikan pembacaan sensor suhu secara instan [15]. Dengan waktu respon yang berkisar antara 1,1 hingga 2,0 detik, performa pertukaran data antara peladen *cloud* Telegram dan Wemos D1 Mini dinilai sangat responsif untuk kategori perangkat pintar berbasis IoT [17].

Tabel 2. Hasil Pengujian Kendali Jarak Jauh (Mode Manual)

No	Perintah telegram	suhu awal	status mode	status akhir	latensi respon
1	manual (Kipas Off)	34.5°C	Mengubah ke Manual	Status (Mati)	2 detik
2	manual (Kipas On)	36.5°C	Mengubah ke Manual	Status (Nyala)	2 detik
3	kipas on	28.5°C (Dingin)	Manual	Menyala	1.4 detik
4	kipas off	31.2°C (Panas)	Manual	Mati	1.2 detik
5	otomatis	31.2°C (Panas)	Mengubah ke Otomatis	Mati (Otomatis)	1.5 detik
6	status	32.0°C	Otomatis	Mati (Otomatis)	1.1 detik
7	manual	32.2°C	Mengubah ke manual	nyala	1.6 detik

C. Pengujian Konsistensi Pembacaan Sensor dan Logika Operasional

Pengujian pada bagian ini dilakukan untuk mengevaluasi keandalan integrasi sensor DHT11 dalam merekam parameter suhu dan kelembapan ruangan secara

real-time terhadap perubahan status operasional kipas[10]. Berdasarkan data pada Tabel 3, sistem menunjukkan konsistensi tinggi saat berada pada Mode Otomatis, di mana aktuatur (kipas) merespons secara akurat fluktuasi data lingkungan yang masuk. Sebaliknya, ketika sistem dialihkan ke Mode Manual, pencatatan parameter lingkungan tetap berjalan dengan valid, meskipun eksekusi fisik kipas murni dikendalikan oleh intervensi langsung pengguna [14]. Rekaman berkala ini membuktikan keandalan sinkronisasi data sensor lokal yang dikirimkan melalui jaringan nirkabel secara periodik [20].

Tabel 3. Pengujian Validasi Informasi Monitoring Jarak Jauh

No	Status mulai	suhu	kelembapan	mode	kipas
1	11.30 WIB	34.6°	59.00%	Otomatis	Mati
2	11.41 WIB	34.8°	51.00%	Otomatis	Mati
3	11.45 WIB	36.4°	51.00%	Otomatis	Nyala
4	11.50 WIB	35.9°	48.00%	Otomatis	Nyala
5	12.10 WIB	34.1°	52.00%	Otomatis	Mati
6	12.30 WIB	34.1°	54.00%	Manual	Nyala
7	12.40 WIB	34.1°	55.00%	Manual	Mati
8	12.50 WIB	34.6°	54.00%	Manual	Nyala
9	13.00 WIB	34.4°	52.00%	Manual	Mati
10	13.15 WIB	35.3°	55.00%	Manual	Nyala

IV. KESIMPULAN

Penelitian ini berhasil mewujudkan purwarupa sistem kendali kipas angin berbasis Internet of Things (IoT) yang mengintegrasikan dual-mode operasional, yaitu otomatis dan manual, melalui antarmuka Telegram Bot API. Berdasarkan hasil serangkaian pengujian yang telah dilakukan, seluruh tujuan utama penelitian ini telah tercapai dengan valid. Sistem terbukti mampu mengeksekusi pensaklaran relay secara presisi pada mode otomatis ketika sensor DHT11 mendeteksi suhu lingkungan melampaui ambang batas kritis yang ditentukan, serta

secara konsisten mengalihkan kendali penuh kepada pengguna melalui perintah teks dari jarak jauh saat mode manual diaktifkan. Hasil penting lainnya menunjukkan performa jaringan yang andal dengan latensi respons komunikasi data dua arah yang sangat singkat dan berada dalam batas toleransi real-time, yaitu berkisar antara 1,1 hingga 2,0 detik. Selain itu, sistem mampu menjaga konsistensi sinkronisasi pencatatan data parameter lingkungan lokal (suhu dan kelembapan) bersamaan dengan visualisasi status waktu aktual berbasis Network Time Protocol (NTP).

Penerapan dari sistem kendali *dual-mode* ini sangat potensial untuk diaplikasikan pada sektor domestik, khususnya pada ruang tidur dengan ukuran target 3 x 3 m, maupun industri skala kecil guna meningkatkan efisiensi energi listrik dan kenyamanan termal ruangan secara fleksibel. Kendati demikian, penelitian ini masih memiliki ruang pengembangan di masa mendatang. Pengembangan lebih lanjut dapat diarahkan pada integrasi algoritma kecerdasan buatan, seperti logika fuzzy, untuk mengatur variasi kecepatan putaran kipas secara bertahap pada mode otomatis, bukan sekadar kondisi sakelar mati dan nyala. Di samping itu, penambahan sistem enkripsi keamanan data pada gateway API serta adopsi media penyimpanan berbasis cloud database eksternal dapat diimplementasikan untuk mendukung pencatatan riwayat (data logging) performa sensor dalam jangka panjang secara lebih aman.

UCAPAN TERIMA KASIH

Terima kasih banyak atas bantuan, kerja sama, dan bimbingannya selama ini. Saya sangat menghargai seluruh dukungan yang telah diberikan hingga penelitian ini dapat diselesaikan dengan baik. Semoga kebaikan Anda mendapatkan balasan yang melimpah

REFERENSI

- [1] "B. Rachmadi and A. O. Pristisahida, 'Optimasi Kenyamanan Suhu Ruangan melalui Perancangan Otomatisasi Kipas Angin di Masjid Nurhidayatullah Maguwaharjo,' Syntax Idea, vol. 5, no. 11, 2023," vol. 5, no. 11, 2023.
- [2] A. A. Salsabita *et al.*, "SISTEM KENDALI SUHU AC MENGGUNAKAN," vol. 9, no. 2, pp. 2167–2172, 2025.
- [3] "H. Sanjaya, M. B. Sanjaya, R. Saputra, and R. Destriana, 'Kipas Angin Otomatis Menggunakan Sensor Suhu DHT11 Berbasis Arduino Uno,' Jurnal Linier (Sains dan Teknologi), vol. 7, no. 01, pp. 187–191, 2021," pp. 1–5.
- [4] H. R. Kurniawan, B. R. Purnama, and A. N. Farida, "Sistem Pengendalian Suhu Otomatis pada Ruangan Berbasis IoT dengan Kipas PWM dan Sensor DHT11," pp. 771–778, 2025.
- [5] E. B. Pratama, A. Hendini, and A. Fristian, "Pendekatan Metode Prototype Pada Aplikasi Presensi Berbasis Mobile (Studi Kasus : Kantor Desa Mekar Jaya)," vol. 4, no. 1, pp. 33–39, 2023.
- [6] D. M. Hapsah, I. F. Hanif, T. Industri, T. Informatika, U. Muhammadiyah, and P. Hamka, "Prototype Pengembangan Sistem Informasi E-Commerce Berbasis Website," vol. 6, no. 2, pp. 1300–1309, 2025, doi: 10.47065/josh.v6i2.6706.
- [7] S. Refly, H. A. Kusuma, J. T. Elektro, F. Teknik, U. Maritim, and R. Ali, "Jurnal Sustainable : Jurnal Hasil Penelitian dan Industri Terapan Analisis Konsumsi dan Fluktuasi Arus dan Daya pada Mikrokontroler Menggunakan Sensor INA219," vol. 11, no. 01, pp. 44–48, 2022.
- [8] H. H. Abrianto and K. Sari, "Sistem Monitoring Dan Pengendalian Data Suhu Ruang Navigasi Jarak Jauh Menggunakan WEMOS D1 Mini," vol. 4, no. 1, pp. 38–49, 2021.
- [9] N. Pratama and W. Iswara, "Deteksi Lemari Terguling Berbasis IoT dengan Notifikasi Telegram untuk Keselamatan Anak Internet of Things," vol. 4, no. 1, 2026.
- [10] A. Jeacfy, G. Go, F. Biabdillah, and A. Wajiansyah, "Smart Agriculture : Pemanfaatan Sensor DHT11 Berbasis Internet of Things (IoT) untuk pemantauan suhu dan kelembapan udara," vol. 14, no. 1.
- [11] T. Tri, W. Ningrum, D. N. Triwibowo, and A. S. S. A., "IMPLEMENTASI INTERNET OF THINGS MENGGUNAKAN SENSOR DHT11 UNTUK MONITORING SUHU DAN KELEMBAPAN RUANG INKUBATOR BAYI DENGAN PLATFORM THINGSPEAK (Implementation of Internet of Things using DHT11 Sensors for Monitoring Temperature and Humidity in a Baby Incubator Room with the Thingspeak Platform)," 2023.
- [12] P. Utama, A. Yusika, S. Kom, and M. K. P. Pendamping, "No Title," 2023.
- [13] S. Hadi *et al.*, "PERBANDINGAN AKURASI PENGUKURAN SENSOR LM35 DAN SENSOR DHT11 UNTUK MONITORING SUHU BERBASIS INTERNET," vol. 6, no. 3, 2022.
- [14] S. T. Elektro, F. Teknik, and U. N. Surabaya, "Prototype Smart Fire System Menggunakan Solenoid Valve dan Kamera ESP32-CAM Berbasis IoT Dimas Kusuma Putra Farid Baskoro , Nur K holis , Arif Widodo," 2017.
- [15] I. W. Sukerayasa, "IMPLEMENTASI API TELEGRAM SEBAGAI ANTARMUKA MANUSIA-MESIN UNTUK SISTEM MONITORING DAN NOTIFIKASI KONDISI REAL-TIME," vol. 12, no. 3, 2025.
- [16] F. Sains, S. Informasi, and U. Ibbi, "Perancangan dan Analisis Saklar Elektronik Digital Berbasis Arduino dan Transistor untuk Penggunaan Efisiensi Energi," vol. 14, pp. 3268–3275, 2025.
- [17] P. Studi and T. Informatika, "Program studi teknik informatika depok maret -2024 1," pp. 1–79, 2024.
- [18] M. R. Akbar, "DESAIN MONITORING GUDANG PENYIMPANAN BERAS BERBASIS IOT (INTERNET OF THINGS) MENGGUNAKAN MIKROKONTROLER WEMOS D1 R2 DENGAN SENSOR DHT22, SENSOR MQ-135 DAN SENSOR API," 2025.
- [19] M. S. Mukminin, A. Rizal, and D. T. Maesaroh, "Perancangan Sistem Pendingin Laptop Otomatis Berbasis Arduino Menggunakan Sensor Suhu DHT11," vol. 11, no. 1, 2026.
- [20] K. Network, "Mengoptimalkan Jaringan Sensor Nirkabel dalam Aplikasi Monitor Lingkungan dengan Teknologi IoT di Indonesia," no. 1, 2025.