

# Implementasi Modul ESP32 Sebagai Pengendali Kipas Angin Berbasis Jaringan WiFi

Nadia Alfi Ni'amah<sup>\*1</sup>, Husna Lathifunisa Arif<sup>2</sup>, Aryawangi Rahmawanto<sup>3</sup>, Edwyn Wahyu Prasetya<sup>4</sup>, Riza Akhsani Setyo Prayoga<sup>5</sup>, Cendra Devayana Putra<sup>6</sup>

<sup>1, 2, 3, 4, 5</sup>Prodi Pendidikan Teknologi Informasi, Universitas Negeri Surabaya  
Surabaya, Indonesia

<sup>6</sup>Prodi Sistem Informasi, Universitas Negeri Surabaya  
Surabaya, Indonesia

nadia.22111@mhs.unesa.ac.id<sup>1</sup>, husna.22084@mhs.unesa.ac.id<sup>2</sup>, aryawangi.22086@mhs.unesa.ac.id<sup>3</sup>, edwyn.22110@mhs.unesa.ac.id<sup>4</sup>, rizaprayoga@unesa.ac.id<sup>5</sup>, putracendra@gmail.com<sup>6</sup>

**Abstract.** *Conventional fans remain a widely used appliance, particularly in tropical regions such as Indonesia. On the other hand, conventional fans still rely on manual control; fans are often left running continuously without regard for the user's needs, resulting in inefficient energy use and comfort that is not always guaranteed. This research aims to design and implement SmartFan, an Internet of Things (IoT)-based fan control system using ESP32, MQTT, and a Flutter mobile application. The system was developed to provide more precise, responsive, and user-friendly control compared to conventional methods or third-party applications such as MQTT Dash. The Flutter application enables users to adjust the fan speed gradually (0–100%), control power, and monitor device connection status in real-time. On the hardware side, the ESP32 is equipped with a WiFi Manager, allowing the device to be configured on various networks without requiring firmware reprogramming, while also generating stable PWM signals to control motor speed through the L298N driver. Separating the power sources between the ESP32 and the motor using a power bank and a 12V adapter significantly improves system stability by preventing brownouts caused by motor current surges. The test results show that the MQTT communication runs reliably, control response is fast, and the integration between hardware and software functions as intended. Overall, SmartFan was successfully developed as an effective, flexible, and user-friendly IoT system. This research also opens opportunities for further development, including the addition of environmental sensors, condition-based automation, as well as the integration of MQTT Secure and smart home platforms.*

**Keywords:** *Smart Home, ESP32, WiFi Manager.*

**Abstrak.** Kipas angin konvensional masih menjadi perangkat yang banyak digunakan masyarakat, terutama di wilayah tropis seperti Indonesia. Di sisi lain kipas angin konvensional masih bergantung pada kontrol manual, kipas sering dibiarkan menyala terus-menerus tanpa memperhatikan kebutuhan pengguna, sehingga penggunaan energi menjadi kurang efisien dan kenyamanan tidak selalu terjamin. Penelitian ini bertujuan untuk merancang dan mengimplementasikan SmartFan, yaitu sistem pengendali kipas angin berbasis *Internet of Things* (IoT) menggunakan ESP32, MQTT, dan Flutter mobile application. Sistem ini dikembangkan untuk memberikan kendali kipas yang lebih presisi, responsif, dan mudah digunakan dibandingkan metode konvensional maupun aplikasi pihak ketiga seperti MQTT Dash. Aplikasi Flutter memungkinkan pengguna mengatur kecepatan kipas secara bertahap (0–100%), melakukan kontrol daya, serta memantau status koneksi perangkat secara *real-time*. Pada sisi perangkat keras, ESP32 dilengkapi dengan WiFi Manager sehingga dapat dikonfigurasi pada berbagai jaringan

tanpa proses pemrograman ulang, serta menghasilkan sinyal PWM yang stabil untuk mengatur putaran motor melalui driver L298N. Pemisahan sumber daya antara ESP32 dan motor melalui powerbank dan adaptor 12V terbukti meningkatkan stabilitas sistem dengan mencegah brownout yang biasanya terjadi akibat lonjakan arus motor. Hasil pengujian menunjukkan bahwa komunikasi MQTT berjalan stabil, respons kendali cepat, dan integrasi antara perangkat keras serta aplikasi berfungsi sesuai tujuan. Secara keseluruhan, SmartFan berhasil dikembangkan sebagai sistem IoT yang efektif, fleksibel, dan mudah digunakan. Penelitian ini juga membuka peluang pengembangan lanjutan melalui penambahan sensor lingkungan, otomatisasi berbasis kondisi dengan kendali jarak jauh berbasis mobile, serta integrasi keamanan MQTT Secure dan platform *smart home*.

**Kata Kunci:** Smart Home, ESP32, WiFi Manager.

## I. PENDAHULUAN

Dalam praktik sehari-hari, kipas angin konvensional masih menjadi perangkat yang banyak digunakan masyarakat, terutama di wilayah tropis seperti Indonesia. Popularitasnya dipengaruhi oleh harga yang terjangkau serta kemudahan pengoperasian. Namun, karena masih bergantung pada kontrol manual, kipas sering dibiarkan menyala terus-menerus tanpa memperhatikan kebutuhan pengguna, sehingga penggunaan energi menjadi kurang efisien dan kenyamanan tidak selalu terjamin [1]. Kondisi tersebut menunjukkan perlunya sistem pengendalian kipas yang lebih cerdas, fleksibel, dan dapat diakses jarak jauh. Hal inilah yang mendorong pengembangan solusi berbasis IoT, salah satunya melalui implementasi modul ESP32 sebagai pengendali kipas angin otomatis berbasis jaringan WiFi.

Seiring dengan perkembangan teknologi, konsep *Internet of Things* (IoT) telah membuka peluang baru dalam pengelolaan perangkat rumah tangga [2]. IoT memungkinkan integrasi antara perangkat fisik, sensor, dan jaringan internet sehingga perangkat seperti kipas angin dapat dipantau serta dikendalikan secara *real-time* [3]. Melalui aplikasi pada smartphone, pengguna dapat mengatur kecepatan kipas angin secara otomatis [4].

Perkembangan ini juga mendorong penggunaan perangkat cerdas yang mampu berkomunikasi melalui jaringan internet untuk meningkatkan efisiensi dan kenyamanan pengguna [5]. Salah satu mikrokontroler yang banyak digunakan dalam implementasi IoT adalah ESP32 karena memiliki konektivitas WiFi terintegrasi serta kemampuan pemrosesan yang stabil [6]. Melalui integrasi jaringan nirkabel WiFi, perangkat dapat menerima perintah, mengirim data, serta melakukan pembaruan status secara *real-time* tanpa koneksi fisik, sehingga kontrol jarak jauh dapat dilakukan dengan mudah menggunakan smartphone [7]. Keunggulan ini menjadikan ESP32 efektif digunakan untuk mengendalikan perangkat rumah tangga, termasuk kipas angin yang selama ini masih bergantung pada kontrol manual.

Penelitian-penelitian sebelumnya telah menunjukkan bahwa penerapan IoT pada sistem kendali kipas angin mampu meningkatkan efisiensi dan kenyamanan penggunaan [8]. Dalam mengembangkan sistem kipas angin pintar berbasis NodeMCU yang dapat mengatur kecepatan kipas secara otomatis berdasarkan parameter suhu dan kelembapan lingkungan. Sistem tersebut bekerja secara *real-time* dan terbukti meningkatkan efektivitas pengendalian [9].

Sementara itu otomatisasi kipas berbasis IoT mampu menurunkan konsumsi energi hingga 40%, terutama pada penggunaan jangka panjang. Meski demikian, sebagian besar proyek sebelumnya masih bergantung pada aplikasi pihak ketiga seperti Blynk atau MQTT Dash, yang memiliki keterbatasan dari segi tampilan antarmuka, fleksibilitas pengembangan fitur, serta kontrol penuh terhadap alur sistem [10]. Dengan keterbatasan Blynk atau MQTT DASH bisa menggunakan flutter dimana memiliki keunggulan dalam Cross Platform, UI yang menarik serta banyak plugin dan library yang bisa digunakan. Selain itu, sistem otomatisasi berbasis ESP32 mampu memberikan pengendalian perangkat yang lebih praktis dan efisien, terutama pada pengaturan kipas dan pencahayaan rumah yang membutuhkan respons cepat dan stabil [11]. Penerapan IoT pada sistem kendali kipas angin terbukti meningkatkan kenyamanan dan efisiensi energi [12].

Penelitian ini bertujuan untuk merancang dan membangun sebuah sistem *Internet of Things* (IoT) berbasis modul ESP32 yang mampu mengendalikan kipas angin secara otomatis melalui aplikasi SmartFan. Sistem ini dikembangkan agar dapat memberikan kemudahan bagi pengguna dalam mengatur kipas angin secara jarak jauh dengan memanfaatkan konektivitas WiFi [13]. Selain itu, penelitian ini bertujuan untuk mengembangkan antarmuka aplikasi SmartFan yang intuitif, mudah digunakan, serta mampu menampilkan perubahan kecepatan dan status kipas angin secara *real-time* sehingga pengguna dapat memantau kondisi perangkat secara langsung [14].

Signifikansi penelitian ini terletak pada kemampuannya menghadirkan solusi pengendalian kipas angin yang lebih modern, praktis, dan terintegrasi melalui pemanfaatan modul ESP32 dan jaringan WiFi. Dengan adanya sistem SmartFan yang dapat dikendalikan secara otomatis melalui aplikasi

mobile, pengguna dapat mengoperasikan kipas angin dari jarak jauh dengan lebih fleksibel dan efisien. Implementasi ini berpotensi meningkatkan kenyamanan pengguna sekaligus membantu mengoptimalkan penggunaan energi melalui pengendalian perangkat yang terpusat dan mudah diakses.

Melalui penelitian ini, diharapkan sistem pengendalian kipas angin berbasis ESP32 yang terintegrasi dengan aplikasi SmartFan dapat memberikan solusi yang lebih praktis, responsif, dan mudah digunakan oleh masyarakat. Sistem ini diharapkan mampu meningkatkan efisiensi dalam pengoperasian perangkat rumah tangga, khususnya melalui kemampuan kontrol jarak jauh dan pemantauan *real-time* yang tidak dimiliki oleh kipas angin konvensional [15]. Selain itu, diharapkan aplikasi SmartFan yang dikembangkan dapat menjadi contoh penerapan IoT yang sederhana namun efektif, sehingga dapat diimplementasikan secara luas dan dikembangkan lebih lanjut sesuai kebutuhan pengguna.

## II. METODE PENELITIAN

Dalam penggunaan sehari-hari, kipas angin konvensional sering kali dinyalakan secara terus-menerus tanpa mempertimbangkan kebutuhan aktual pengguna. Sistem kontrol manual membuat pengguna harus menekan tombol atau memutar saklar secara langsung untuk mengatur kecepatan atau mematikan kipas. Hal ini tidak hanya mengurangi kenyamanan, tetapi juga berpotensi menyebabkan pemborosan energi ketika kipas tetap menyala meskipun sedang tidak digunakan.

Selain itu, tidak adanya fitur kendali jarak jauh membuat pengguna tidak dapat mengakses atau memantau kondisi perangkat ketika berada jauh dari kipas. Kondisi tersebut menunjukkan perlunya sistem kendali kipas angin yang lebih fleksibel, efisien, dan dapat dioperasikan secara nirkabel menggunakan jaringan WiFi.

Tujuan penelitian meliputi mengembangkan sistem kendali kipas angin berbasis IoT menggunakan modul ESP32 yang dapat dioperasikan secara otomatis maupun manual melalui aplikasi SmartFan, Memungkinkan pengguna mengendalikan kecepatan dan status kipas dari jarak jauh dengan memanfaatkan konektivitas WiFi dan protokol komunikasi MQTT, Meningkatkan efisiensi penggunaan energi dengan menyediakan mekanisme pengendalian yang lebih terarah dan terpusat melalui sistem digital, Memberikan pengalaman pengguna yang lebih nyaman dan modern melalui antarmuka aplikasi yang responsif, *real-time*, dan mudah digunakan tanpa intervensi manual langsung pada perangkat. Adapun daftar alat penelitian tersaji pada Tabel 1, daftar bahan penelitian tersaji pada Tabel 2 dan daftar software penelitian pada Tabel 3.

Tabel 1. Daftar Alat Penelitian

No	Nama Alat	Jumlah	Keterangan
1	Laptop	1	Untuk pemrograman ESP32 dan

			pengembangan aplikasi.
2	Solder & Timah	1	Untuk menyolder dan menghubungkan komponen
3	Lem tembak	1	Untuk merekatkan dan menguatkan posisi komponen
4	Kabel Type C	1	Untuk menghubungkan ESP32 ke laptop

Tabel 2. Daftar Bahan Penelitian

No	Nama Bahan	Jumlah	Keterangan
1	Kipas Angin	1	Sebagai aktuator utama sistem
2	ESP32	1	Sebagai otak pemrosesan utama yang memiliki modul Wi-Fi dan Bluetooth terintegrasi
3	Module Driver (L298N atau MOSFET)	1	Untuk mengontrol kecepatan dan arah putaran motor DC
4	Oled i2c (0.96 inch)	1	Untuk menampilkan status koneksi, IP Address, dan kecepatan kipas.
5	Kabel jumper	15	Untuk menghubungkan jalur data antar komponen.
6	Adaptor 12v	1	Sumber daya eksternal untuk menyuplai tegangan ke Driver Motor L298N dan Kipas.
7	Powerbank 10.000 mAh	1	Sebagai sumber daya portabel khusus untuk ESP32 (5V) agar sistem kontrol tetap hidup terpisah dari beban motor.

Tabel 3. Daftar Software Penelitian

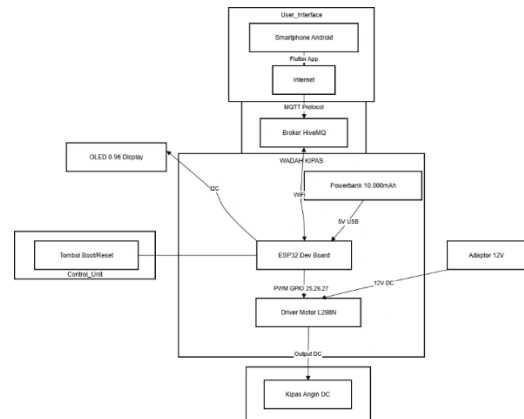
No	Nama Software	Jumlah	Keterangan
1	Arduino IDE	1	Untuk pemrograman ESP32
2	Flutter & Dart SDK	1	Untuk pengembangan aplikasi mobile SmartFan.
3	MQTT Broker	1	Sebagai perantara komunikasi

			ntara perangkat IoT
4	VS Code	1	Sebagai code editor

Sistem bekerja dengan topologi bintang di mana Broker MQTT berfungsi sebagai penengah.

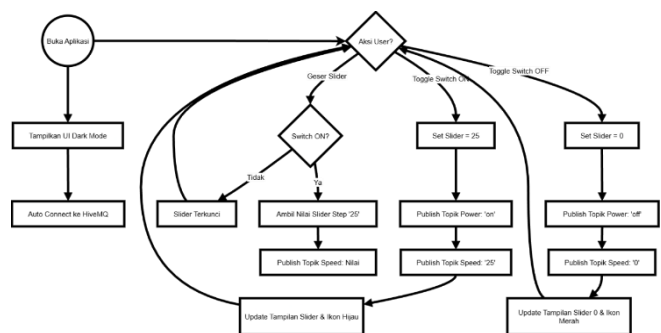
ESP32 sebagai perangkat IoT yang menerima sumber daya dari powerbank, menjalankan WiFi Manager untuk konfigurasi jaringan, mengirim/menerima data melalui MQTT, mengontrol kecepatan kipas melalui L298N, Menampilkan status sistem melalui OLED. MQTT Broker sebagai pusat komunikasi (publish-subscribe) antara ESP32 dan *Flutter mobile application*. Aplikasi SmartFan (Flutter) sebagai antarmuka pengguna yang menampilkan *slider* kecepatan kipas (0-100, step 25), mengirim perintah kecepatan ke broker, menerima *feedback* dari ESP32 secara *real-time*. OLED menampilkan status visual secara *real-time*.

Implementasi dilakukan ke dalam dua bagian utama yaitu implementasi perangkat keras meliputi ESP32 diprogram dengan Arduino IDE, OLED 0.96 digunakan untuk menampilkan status, L298N sebagai driver motor, Powerbank 10.000 mAh untuk ESP32, Adaptor 12V untuk kipas.



Gambar 1. Arsitektur Perangkat Keras

Implementasi Perangkat Lunak meliputi Flutter digunakan untuk membuat aplikasi SmartFan, Library MQTT client digunakan untuk komunikasi, Desain UI dibuat sesuai mockup, Firmware ESP32 memanfaatkan WiFi Manager untuk konfigurasi SSID.



Gambar 2. Arsitektur Perangkat Lunak

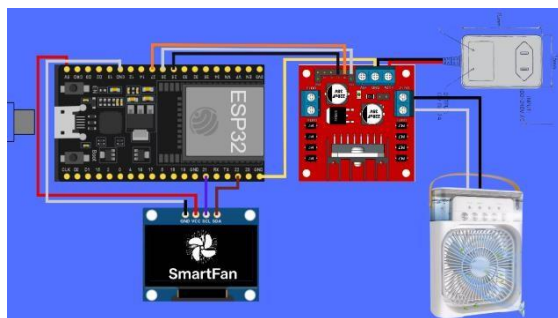
Pengujian Sistem terdiri dari Referensi Logika meliputi Logika Switch ON dimana saat tombol ON ditekan, *slider* otomatis ke 25 dan mengirim perintah ganda (Power ON & Speed 25). Logika Slider meliputi Slider mengirim data ke topik kecepatan hanya jika Switch dalam kondisi ON. Jika user menggeser *slider* saat switch masih OFF. Maka aplikasi memblokir geseran/disable *slider*.

Flowchart *Flutter mobile application* menggambarkan proses saat aplikasi SmartFan dijalankan seperti Aplikasi menampilkan halaman utama berupa *slider* dan tombol ON/OFF, Aplikasi melakukan koneksi awal ke MQTT Broker, Jika koneksi berhasil, aplikasi siap mengirimkan dan menerima data, Pengguna menggeser *slider* atau menekan tombol ON/OFF, Aplikasi mengirim nilai kecepatan ke topik MQTT, Aplikasi menerima *feedback* dari ESP32 untuk memastikan kecepatan sudah diterapkan.

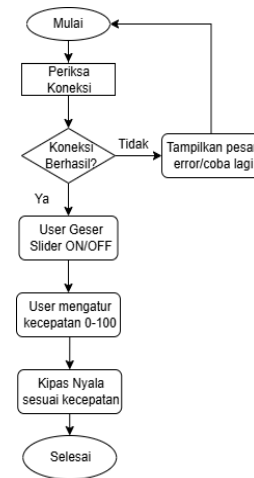
Skema rangkaian pada penelitian ini merupakan prototipe fungsional dari sistem pengendalian kipas angin otomatis berbasis *Internet of Things* (IoT) menggunakan mikrokontroler ESP32. Modul ESP32 berfungsi sebagai pusat kendali yang menerima perintah dari aplikasi SmartFan melalui jaringan WiFi, kemudian meneruskan sinyal kendali ke driver motor L298N. Driver L298N digunakan sebagai penguat arus untuk mengatur kecepatan motor kipas DC sesuai nilai yang dikirimkan melalui protokol MQTT.

Pada rangkaian ini, motor kipas terhubung ke output driver L298N, sementara input kendalinya berasal dari pin PWM pada ESP32 yang digunakan untuk mengatur variasi kecepatan putaran motor. Selain itu, rangkaian juga dilengkapi dengan catu daya eksternal untuk memastikan motor memperoleh suplai tegangan yang stabil tanpa membebani modul ESP32. Komunikasi antara aplikasi SmartFan dan ESP32 dilakukan secara nirkabel melalui WiFi, sehingga seluruh pengaturan kecepatan kipas dan status perangkat dapat dikendalikan secara jarak jauh tanpa intervensi manual.

Dengan konfigurasi rangkaian tersebut, sistem SmartFan mampu mengatur kecepatan kipas secara *real-time*, responsif, dan sesuai perintah pengguna. Implementasi ini tidak hanya memberikan fleksibilitas pengendalian, tetapi juga mendukung konsep *smart home* dengan memanfaatkan teknologi IoT untuk mengotomatisasi perangkat rumah tangga secara efisien.



Gambar 3. Skema Rangkaian SmartFan



Gambar 4. Flowchart Sistem

Rencana pengujian sistem disajikan pada Tabel 4.

Tabel 4. Rencana Pengujian

No	Nama Pengujian
1	Pengujian Koneksi WiFi Manager
2	Pengujian Latensi Komunikasi MQTT
3	Pengujian Kendali Kecepatan Kipas
4	Pengujian Tampilan OLED

### III. HASIL DAN PEMBAHASAN

#### A. Deskripsi Data

SmartFan merupakan sistem kendali kipas angin berbasis *Internet of Things* (IoT) yang dirancang untuk memungkinkan pengguna mengatur kecepatan kipas secara nirkabel melalui aplikasi mobile. Sistem ini mengintegrasikan perangkat keras, *firmware*, protokol komunikasi MQTT, dan aplikasi Android berbasis Flutter sehingga menghasilkan alur kerja yang cepat, responsif, dan mudah digunakan.

Secara umum, SmartFan memiliki empat komponen utama meliputi Unit Mikrokontroler (ESP32) yang berfungsi sebagai pusat kendali, menangani koneksi WiFi, komunikasi MQTT, pembacaan data dari aplikasi, serta pengaturan PWM untuk mengontrol kecepatan kipas. Driver Motor L298N, digunakan untuk mengatur suplai arus dan tegangan ke motor DC sehingga kipas dapat berputar sesuai kecepatan yang diperintahkan aplikasi. Motor Kipas DC sebagai aktuator utama untuk menghasilkan hembusan angin dengan kecepatan bertingkat (0%, 25%, 50%, 75%, 100%). Display OLED 0.96", digunakan untuk menampilkan status WiFi, proses koneksi MQTT, serta nilai kecepatan kipas secara *real-time*.

Sementara itu, aplikasi mobile SmartFan dibangun menggunakan Flutter dengan desain antarmuka modern dan minimalis. Aplikasi ini menyediakan Slider kontrol kecepatan, Tombol ON/OFF, Indikator status MQTT, Indikator kecepatan *real-time*.

Sistem SmartFan juga memanfaatkan fitur WiFi Manager pada ESP32. Apabila ESP32 tidak menemukan jaringan WiFi yang pernah tersimpan, perangkat akan memasuki mode konfigurasi (Setup Mode) dan membuat Access Point untuk menghubungkan pengguna ke halaman pengaturan WiFi melalui IP 192.168.4.1.

Setelah WiFi terhubung, sistem akan menyambungkan diri ke broker MQTT untuk memastikan komunikasi dua arah antara aplikasi dan perangkat berjalan secara stabil. Semua proses ini ditampilkan secara langsung melalui OLED, sehingga pengguna mendapat umpan balik visual tanpa harus membuka aplikasi.

Dengan integrasi penuh antara software dan hardware, SmartFan menjadi sistem IoT yang andal, mudah dipahami, serta memiliki respons tinggi terhadap kontrol yang diberikan melalui aplikasi.

**B. Hasil**

**Pengujian Koneksi WiFi Manajer**

Tabel 5. Pengujian Koneksi WiFi Manager

Skenario Jaringan	Jumlah Percobaan	Berhasil	Gagal	Rata-rata Waktu	Ket
Hotspot/Home WiFi (WPA2-PSK)	10	10	0	4.2 detik	Koneksi sangat stabil dan cepat.
Tidak Ada WiFi (Mode AP)	10	10	0	2.1 detik	ESP32 langsung membuat Hotspot mandiri.
Enterprise/High Sec (WPA3 / 802.1x)	10	0	10	-	Gagal Terhubung. ESP32 Terus merestart koneksi (looping)



Gambar 3. Tampilan OLED saat WiFi Manager aktif

WiFi Manager bekerja sempurna (100% berhasil) pada jaringan standar rumahan (WPA/WPA2). Namun, sistem mengalami kegagalan total saat dicoba pada jaringan dengan keamanan tingkat tinggi (WPA3 atau WPA2-Enterprise seperti WiFi kampus/kantor) karena library standar ESP32 belum

mendukung enkripsi tersebut secara native tanpa konfigurasi sertifikat khusus.

**Pengujian Latensi Komunikasi MQTT**

Tabel 6 Pengujian Latensi Komunikasi MQTT

Percobaan ke	Latensi	Status	Kondisi Jaringan
1	45 ms	Cepat	WiFi Stabil
2	38 ms	Cepat	WiFi Stabil
3	112 ms	Sedang	Ada sedikit gangguan traffic
4	56 ms	Cepat	WiFi Stabil
5	210 ms	Lambat	Spike (Lonjakan) jaringan sesaat
6	42 ms	Cepat	WiFi Stabil
7	65 ms	Cepat	WiFi Stabil
8	155 ms	Sedang	Jitter pada broker publik
9	48 ms	Cepat	WiFi Stabil
10	50 ms	Cepat	WiFi Stabil
Rata - Rata	<b>82.1 ms</b>	<b>Stabil</b>	<b>Dapat diterima untuk IoT</b>

Komunikasi MQTT sangat responsif dan stabil. Tidak ditemukan delay signifikan, bahkan ketika jaringan WiFi mengalami fluktuasi ringan.

Pengujian ini menguji respons motor DC saat slider aplikasi digeser pada step: 0% → 25% → 50% → 75% → 100%

Tabel 7. Pengujian Kendali Kecepatan Kipas

Kecepatan Aplikasi	PWM ESP32	Respons Motor	Delay
0%	0	Motor berhenti	< 200 ms
25%	64	Motor berputar pelan	< 200 ms
50%	128	Motor stabil	< 200 ms
75%	192	Motor cepat	< 200 ms
100%	255	Motor maksimum	< 200 ms

Motor DC memberikan respons sangat cepat dan linear terhadap perubahan slider. Tidak ada jitter/patah-patah pada kecepatan rendah. OLED diuji untuk melihat Kecepatan update teks, Keakuratan status, Keterbacaan tampilan.

Tabel 8 Pengujian Tampilan OLED

Parameter	Hasil
Waktu menampilkan splash screen	1.5 detik
Update status WiFi	Real-time
Update status MQTT	Real-time
Update kecepatan kipas	< 50 ms
Keterbacaan	Sangat jelas

OLED sangat responsif dan memberikan *feedback* langsung bagi pengguna, meningkatkan user experience secara signifikan. Pengujian integrasi dilakukan untuk memastikan seluruh komponen bekerja secara bersamaan.

Pengamatan Utama meliputi ESP32 tidak mengalami reset meskipun motor berputar kontinu, Aplikasi tetap stabil meskipun *slider* digeser cepat, WiFi Manager tidak crash selama 3 kali siklus konfigurasi, MQTT tidak mengalami timeout saat pengujian 30 menit, Dual power supply efektif mencegah gangguan arus motor ke ESP32.

Kesesuaian dengan Tujuan Pengembangan dimana setelah seluruh hasil dianalisis, dapat disimpulkan bahwa SmartFan meliputi Berhasil menyediakan kontrol kipas berbasis IoT secara *real-time*, Memiliki respons cepat antara aplikasi dan motor, Menggunakan WiFi Manager yang memudahkan pengguna, Mampu beroperasi stabil dalam durasi panjang, Memiliki antarmuka yang lebih baik dibanding aplikasi generik seperti MQTT Dash. Dengan demikian, SmartFan telah memenuhi semua tujuan pengembangan dan dapat digunakan sebagai prototype perangkat *smart home* yang siap untuk dikembangkan lebih lanjut.



Gambar 4. Rangkaian lengkap sistem SmartFan

### C. Pembahasan

Penggunaan Protokol MQTT dimana MQTT berperan sebagai penghubung antara aplikasi dan ESP32 sehingga perintah seperti mengubah kecepatan atau menyalakan kipas dapat dikirim dengan cepat dan menggunakan bandwidth yang sangat kecil. Sifatnya yang *real-time* membuat sistem mampu merespons perintah pengguna hampir tanpa jeda. Selain itu, struktur publish-subscribe memudahkan pengembangan fitur baru, seperti penambahan sensor suhu, integrasi ke *smart home*, atau penerapan dashboard monitoring. MQTT juga memungkinkan komunikasi tetap berjalan meskipun jaringan WiFi tidak stabil, sehingga meningkatkan keandalan sistem secara keseluruhan. Dengan demikian, protokol MQTT menjadi fondasi penting yang mendukung kinerja dan potensi pengembangan SmartFan di masa mendatang.

Pemrograman sistem dilakukan menggunakan Arduino IDE dengan bahasa C++, di mana seluruh proses pengendalian difokuskan pada *firmware* ESP32. Fitur WiFi Manager digunakan untuk memudahkan pengguna mengatur koneksi WiFi tanpa perlu memodifikasi program. Protokol MQTT

menjadi media utama komunikasi antara aplikasi SmartFan dan ESP32, sehingga perintah yang dikirim melalui smartphone dapat diterima secara *real-time*. Untuk mengendalikan kecepatan kipas, nilai *slider* dari aplikasi (0–100%) dikonversi menjadi nilai PWM (0–255), yang kemudian digunakan untuk mengatur intensitas putaran motor melalui driver L298N. Hasil pengujian menunjukkan bahwa motor merespons perubahan PWM dengan sangat cepat dan stabil.

Pengaruh ESP32 Terhadap Kinerja Sistem dimana Penggunaan ESP32 memberikan dampak besar terhadap performa sistem secara keseluruhan. ESP32 memiliki modul WiFi bawaan yang lebih stabil dan cepat, sehingga proses koneksi ke jaringan dapat dilakukan dalam waktu 3–6 detik, dan koneksi MQTT dapat tercapai dalam sekitar satu detik. Prosesor dual-core pada ESP32 memungkinkan pemisahan tugas antara pengendalian motor, tampilan OLED, dan komunikasi jaringan, sehingga sistem berjalan tanpa lag atau crash. Selama pengujian panjang, ESP32 terbukti stabil tanpa mengalami reset meskipun motor bekerja secara kontinu. Pemisahan sumber daya antara ESP32 dan motor juga membantu mencegah penurunan tegangan yang dapat menyebabkan sistem mati mendadak.

Penggunaan Aplikasi Berbasis Flutter. Pada penelitian SmartFan, aplikasi berbasis Flutter digunakan sebagai antarmuka utama yang menghubungkan pengguna dengan perangkat ESP32. Flutter dipilih karena mampu menghasilkan aplikasi dengan tampilan modern, responsif, dan berjalan pada berbagai perangkat Android tanpa perlu penyesuaian kode yang rumit. Melalui aplikasi ini, pengguna dapat mengontrol fitur-fitur utama seperti menghidupkan atau mematikan kipas, mengatur kecepatan putaran melalui *slider*, serta memantau status koneksi MQTT dan WiFi. Proses pengiriman perintah berlangsung secara *real-time* berkat integrasi Flutter dengan protokol MQTT, sehingga perubahan pada aplikasi langsung diterima oleh ESP32 tanpa jeda yang berarti.

Peningkatan dan Pengembangan Sistem. Sistem SmartFan memiliki potensi pengembangan yang luas. Salah satu peningkatan yang dapat dilakukan adalah menambahkan sensor suhu seperti DHT22 atau DS18B20 sehingga kipas dapat menyesuaikan kecepatan secara otomatis berdasarkan kondisi lingkungan. Sistem juga dapat ditambah dengan fitur mode malam, pembaruan *firmware* secara over-the-air (OTA), integrasi *smart home* seperti Google Home dan Alexa, serta dashboard monitoring berbasis web. Selain itu, penggunaan driver motor yang lebih efisien seperti MOSFET dapat meningkatkan performa motor dan mengurangi panas pada rangkaian.

Aplikasi Sistem di Kehidupan Sehari-Hari. SmartFan dapat digunakan dalam berbagai situasi, terutama dalam lingkungan rumah tangga, ruang kerja, laboratorium, atau tempat yang membutuhkan sirkulasi udara terkontrol. Pengguna dapat mengendalikan kipas melalui smartphone tanpa harus mendekati perangkat, sehingga sangat bermanfaat untuk lansia, anak-anak, atau penyandang disabilitas. Dengan

pengaturan kecepatan yang fleksibel, sistem membantu mengoptimalkan penggunaan energi, karena kipas tidak selalu beroperasi pada kecepatan maksimum. SmartFan juga berpotensi menjadi bagian dari sistem *smart home* yang lebih besar, terutama jika dikombinasikan dengan sensor lingkungan.

#### IV. KESIMPULAN

Berdasarkan proses perancangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa sistem SmartFan berhasil dikembangkan dengan baik sebagai perangkat kendali kipas berbasis IoT yang stabil, responsif, dan mudah digunakan. Penggantian aplikasi MQTT Dash dengan aplikasi mandiri berbasis Flutter terbukti memberikan peningkatan signifikan, baik dari segi presisi pengaturan kecepatan kipas maupun kenyamanan antarmuka pengguna. Integrasi WiFi Manager pada ESP32 turut mendukung fleksibilitas perangkat karena memungkinkan konfigurasi jaringan dilakukan tanpa pemrograman ulang, sehingga sistem menjadi lebih praktis dan siap digunakan pada berbagai lingkungan jaringan.

Pemisahan sumber daya antara ESP32 dan motor melalui powerbank serta adaptor 12V juga berkontribusi besar terhadap stabilitas sistem dengan mencegah terjadinya brownout. Selain itu, keseluruhan integrasi perangkat keras dan perangkat lunak—mulai dari pengaturan PWM, komunikasi MQTT *real-time*, hingga umpan balik visual melalui OLED—menunjukkan bahwa rancangan sistem telah bekerja sesuai tujuan. Aplikasi Flutter SmartFan juga berhasil memberikan kendali yang cepat dan akurat, serta menghadirkan pengalaman penggunaan yang intuitif bagi pengguna.

Secara keseluruhan, penelitian ini berhasil membangun sebuah sistem IoT yang fungsional dan layak digunakan, sekaligus memiliki potensi pengembangan lebih lanjut melalui penambahan sensor lingkungan, fitur otomatisasi, integrasi cloud, peningkatan keamanan komunikasi, serta perluasan aplikasi ke berbagai platform.

#### DAFTAR PUSTAKA

- [1] Y. Sofyan Arief and Y. Akbar, "Pengembangan Sistem Kipas Angin Pintar Berbasis IoT untuk Pemantauan Suhu dan Kelembapan secara Real-Time," *Jurnal Indonesia: Manajemen Informatika dan Komunikasi (JIMIK)*, vol. 6, no. 3, pp. 1931–1938, 2025, [Online]. Available: <https://journal.stmiki.ac.id>
- [2] D. Sembiring, "Evaluation Of Internet Of Things (Iot) For Monitoring And Control Of Home Electronic Appliances Deri Sembiring," *International Journal of Management, Economic and Accounting*, vol. 1, no. 3, pp. 289–293, 2023, doi: 10.61306/ijmea.
- [3] N. Ismail, M. R. Effendi, N. R. Evandi, and H. Fakrurroja, "Sistem Smart Home Berbasis IoT Dengan Integrasi Pengendalian Suara dan Aplikasi Smartphone," *Transmisi: Jurnal Ilmiah Teknik Elektro*, vol. 27, no. 3, pp. 176–184, Jul. 2025, doi: 10.14710/transmisi.27.3.176-184.
- [4] A. A. Jalil, M. J. Abdullah, M. Simatupang, and H. T. Mokui, "Systematic Literature Review: Penerapan *Internet of Things* (IoT) Untuk Optimalisasi Smart Home," *Jurnal Teknologi Informasi dan Komunikasi*, no. 2, 2025.
- [5] R. Rengaraj, M. Devesh Raj, G. R. Venkatakrishnan, and R. Sai Mugil, "Real-Time Home Environment Monitoring And Control Using Iot Technologies," *Int. J. Environ. Sci.*, vol. 11, no. 5, pp. 2229–7359, 2025, [Online]. Available: <https://www.theaspd.com/ijes.php>
- [6] P. A. Putri and H. Saputra, "Smart Home System Based on *Internet of Things* (IoT) for Monitoring and Controlling Room Temperature," *Journal on Internet Engineering Technologies and Applications*, vol. 1, no. 1, pp. 45–52, 2024.
- [7] V. Visayas, C. Cakra, and Y. Supit, "SISTEM KONTROL ALAT ELEKTRONIK DALAM RUMAH BERBASIS INTERNET OF THINGS (IOT)," *Jurnal Sistem Informasi dan Teknik Komputer*, vol. 9, no. 2, pp. 249–261, 2024.
- [8] R. Irfani, A. A. Rahmanto, and M. E. Gazazanata, "Rancang Bangun Sistem Exhaust Fan Otomatis Berbasis Sensor DHT11 dan Mikrokontroler ESP32 untuk Peningkatan Kualitas Udara di Smoking Area," *Jurnal Engine: Energi, Manufaktur, dan Material*, vol. 9, no. 1, pp. 102–112, Apr. 2025, doi: 10.30588/jeemm.v9i1.2161.
- [9] W. P. A. Pamungkas, N. Kholis, N. Nurhayati, and F. Baskoro, "Sistem Control Dan Keamanan Smart Home Berbasis Google Firebase," *Jurnal Teknik Elektro (JTE)*, vol. 11, pp. 40–46, 2022.
- [10] R. Moh. R. Djula *et al.*, "Smart WiFi Plug Untuk Penghematan Energi Kipas Angin Di Asarna PTDI-STTD," *Jurnal Manajemen Informatika Jayakarta*, vol. 5, no. 4, pp. 351–355, Dec. 2025, doi: 10.52362/jmijayakarta.v5i4.2013.
- [11] R. K. Habib, R. P. Bramadita, and A. N. Farida, "Sistem Pengendalian Suhu Otomatis pada Ruangan Berbasis IoT dengan Kipas PWM dan Sensor DHT11," *Seminar Nasional Teknologi Informasi dan Bisnis (SENATIB)*, pp. 771–778, 2025.
- [12] J. S. Ramadhan, M. Ilham, and S. Rochman, "Prototipe Kontrol Exhaust Fan Dengan Sensor MQ-2 Dan DHT22 Berbasis Internet Of Things Menggunakan ESP32," *Jurnal Teknik Elektro: Electronic Control, Telecommunication, Computer Information and Power System*, vol. 10, pp. 211–220, 2025.
- [13] I. Arifin and D. C. Deni, "Literature Review: Utilization of Technology and Computerization in Smart Homes," *SMART: Jurnal Teknologi Informasi dan Komputer*, vol. 1, pp. 34–38, 2022, doi: 10.58222/smart.v3i1.522.
- [14] Y. A. Rozzi, D. A. Dion, and P. W. Fadhel, "Rancang Bangun Sistem Smart Home Berbasis IoT untuk Kendali Lampu dan Pemantauan Suhu," *Jurnal Surya Energy*, pp. 24–30, Sep. 2025, doi: 10.32502/jse.v10i1.939.
- [15] P. Rahayu, S. Sularno, and I. U. Sari, "Perancangan Sistem Smart Home Berbasis IoT Menggunakan ESP32 dan Aplikasi Blynk untuk Otomatisasi Perangkat Rumah Tangga," *JISKA: Jurnal Sistem Informasi Dan Informatika*, vol. 3, no. 2, pp. 95–100, 2025, [Online]. Available: <http://jurnal.unidha.ac.id/index.php/jiska>