SNTE-FORTEI
Seminar Nasional Teknik Elektro
Forum Pendidikan Tinggi Teknik Elektro Indonesia
e-ISSN : 3031-5321

Temperature-Responsive DC Fan Control System Using ESP32 and DHT22 for Improved Energy Efficiency

Muhammad Rafi Rizky Putra[1], Fauzia Nur Azmi[2], Muhammad Ilham Darmanto[3], Gathan Warino[4]
[1234]*Department of Electrical Engineering, Fakultas Sains dan Teknologi, Universitas Al-Azhar Indonesia, Jakarta Selatan, Indonesia*
[1]muhammadrafirizky@gmail.com, [2]fauzianurazmi04@gmail.com, [3]ilham.darmanto@gmail.com, [4]gathanwarino20@gmail.com

Article Info

Keyword:

ESP32
DHT22
PWM
Automatic Fan
Temperature Sensor
Microcontroller

ABSTRACT

Abstrak— This paper presents the design and development of an automatic fan control system based on the ESP32 microcontroller and a DHT22 temperature sensor. The system is designed to automatically turn the fan on or off and adjust its speed using PWM signals, depending on the ambient temperature. The fan remains off when the temperature is below 21°C and operates at increasing speeds for temperatures above 21°C, with PWM values ranging from 90 to 255. The primary goal is to offer an energy-efficient solution that can effectively respond to changing environmental conditions. Compared to manual systems, this automatic system offers faster response times, reduces energy consumption, and eliminates the need for human intervention. Testing demonstrated a clear and consistent relationship between temperature and fan speed, confirming that the control logic functioning as intended. This system can be effectively implemented in small-scale cooling systems requiring responsive and efficient temperature regulation.

Corresponding Author:
Putri Wulandari
Digital Business, Faculty of Business and Social Sciences,
Universitas Binawan, Dewi Sartika
East Jakarta, Indonesia 13630
Email: putri.wulandari@binawan.ac.id

I. INTRODUCTION

Maintaining a constant room temperature is essential for human comfort and the optional performance of electronic devices [25][24]. Fans are a widely used method to cool enclosed spaces, especially as the ambient temperature increase over time. However, manually operating a fan is often ineffective because it cannot adapt to real-time temperature conditions. Previous studies on automatic fan control systems have mostly relied on IoT connectivity, provided only basic on/off switching, or lacked layered PWM regulation. This study addresses those limitations by offering a fully offline, autonomous, and multi-level PWM-based control system.

With the development of affordable microcontroller and temperature sensor technologies, sensor based automated control systems have become increasingly accessible. One such sensor is DHT22, which offers improved accuracy and a wider measurement range than its predecessor. It can read ambient temperature and humidity in real time. Microcontrollers like the ESP32 provide programming flexibility and support the

PWM signals needed to regulate DC motor speed in stages. Additionally, using a motor driver enables more precise, safe, and responsive motor control than using a single MOSFET.

This research aims to create an automatic fan system that can turn on and adjust its rotation speed based on ambient temperature. The system uses a PWM signal generated by the ESP32 microcontroller to control the speed of the DC motor via the motor driver, with the speed adjusted based on the temperature detected by the DHT22 sensor. This approach not only improves comfort but also allows energy savings by activating the fan only when needed.

Key contributions to this research include the creation of a temperature control system using DHT22 and ESP32 sensors. These components are capable of accurately measuring and processing ambient temperature data in real time. The system applies Pulse Width Modulation (PWM) signals to implement a temperature control logic through a motor driver. This enables precise and gradual adjustment of fan speed according to temperature variations. Furthermore, this study presents an experimental analysis that demonstrates the correlation between temperature and fan speed, validating the effectiveness of the proposed control logic in providing responsive and energy-efficient cooling performance.

II. *RELATED WORD*

Automatic fan systems based on temperature sensors and microcontrollers have been the focus of various studies. Using an LM35 temperature sensor and an Arduino microcontroller, Budiyanto et al. [1] created a fan speed control system that utilized relays and Internet of Things (IoT) monitoring. However, this system did not use PWM signals to directly control motor speed in stages, and its operation depended on an internet connection. Septian and Syari [2] used NodeMCU to develop an automatic fan system based on the Internet of Things. Their online application enabled fan control. Ease of remote monitoring was not the primary goal of this system; it focused on local motor speed control. The IJARSCT study [3] employed a similar method. This study combined a DHT11 sensor with an ESP8266 to activate the fan based on room temperature. The system focuses on cloud integration and does not implement speed regulation using PWM.

Research conducted by Akin-Ponnle [4] introduced an energy-efficient temperature control system with fan control for tropical areas. However, this system still relies on IoT as a major component in its structure. On the other hand, Devmore et al. [8] created a temperature control project utilizing ESP32 with an emphasis on energy efficiency, but its primary use is more general for room regulation and does not specifically highlight DC motor actuation in layers.

Additional references, such as the work by Kanchanapalli et al. [12], investigated an ESP32-based temperature control system aimed at regulating fan speed through sensor feedback for general environmental monitoring. Although this study effectively demonstrated real-time temperature acquisition, it did not focus on layered DC motor speed adjustment using PWM logic. Similarly, Debele and Qian [13] developed an Arduino-based automatic room temperature regulation system using a DHT11 sensor, where the fan was controlled through simple on/off switching without proportional speed control.

In addition, research by Jain et al. [14][21] and Aggarwal and Verma [23] explored Arduino and LM35/DHT11-based fan control, achieving functional temperature-triggered operation but lacking offline PWM integration. Studies such as Sharma and Gupta [24] and Kim and Park [25] implemented PWM techniques and proportional control algorithms for energy efficiency, yet their systems targeted broader HVAC or AC fans rather than small-scale DC fans.

Other works, including Pahlevi [10], Rusimamto et al. [7], Ilupeju et al. [9], Setyawan and Setyawan [16], Pratomo et al. [17], Irfani et al. [18], and Syafriza et al. [19], contribute valuable insights into temperature monitoring, IoT integration, and ventilation improvement; however, they remain dependent on internet connectivity or do not specifically apply layered PWM control for DC motor fans. Even studies by Nasution and Sinaga [20] and Kalyani and Rajalakshmi [22], which utilized high-accuracy sensors such as DHT22, focused on IoT-based remote monitoring rather than standalone offline systems. To better understand the strengths and limitations of these previous studies, Table 1 provides a comparative summary:

Table 1. Comparative Summary of Related Research

Study	Microcontroller & Sensor	Strengths	Limitations
Septian & Syari [2]	NodeMCU	Remote fan control	Focus not on PWM/local system
IJARSCT [3]	ESP8266 + DHT11	Cloud integration	No speed control via PWM
Akin-Ponnle [4]	Not specified	Energy-efficient focus	Relies on IoT platform
Devmore et al. [8]	ESP32	ESP32 + energy efficiency	Not specific to DC fan control
Setia et al. [11]	ESP32 + DHT22	IoT-based fan system	Dependent on internet

From the comparative analysis in Table 1, it is evident that even in the most recent studies, the majority of automatic fan control systems remain focused on remote operation through IoT platforms or on providing only basic temperature monitoring without incorporating nuanced speed regulation. Although temperature and humidity sensors such as the DHT11 or DHT22 have been paired with microcontrollers like Arduino, ESP8266, or ESP32, their implementations generally lack direct PWM-based control to adjust DC motor speed in multiple stages. This limitation affects both the responsiveness and the energy efficiency of the system, especially in applications that require immediate adaptation to local temperature changes.

The five most recent studies—Septian & Syari [2], IJARSCT [3], Akin-Ponnle [4], Devmore et al. [8], and Setia et al. [11]—demonstrate advancements in IoT-based fan monitoring, energy-efficient control strategies, and integration of modern microcontrollers. However, these works also reveal common shortcomings: heavy dependence on internet connectivity, limited use of PWM for smooth speed control, and an emphasis on either monitoring or on/off actuation rather than layered fan speed regulation. While Setia et al. [11] introduced the DHT22 sensor in an exhaust fan application for energy optimization, their design still relied on cloud-based control. Similarly, Akin-Ponnle [4] and Devmore et al. [8] focused on energy efficiency but did not emphasize proportional speed adjustments for DC motors in small-scale cooling environments.

When these findings are considered alongside other relevant works—such as Kanchanapalli et al. [12] and Debele & Qian [13], which examined ESP32- and Arduino-based temperature control without layered PWM integration—the research gap becomes more apparent. Studies by Jain et al. [14][21], Aggarwal & Verma [23], Sharma & Gupta [24], and Kim & Park [25] contributed to PWM or proportional control methodologies, but their systems were designed primarily for AC or HVAC fans rather than low-voltage DC fans. Additional works, including Pahlevi [10], Rusimamto et al. [7], Ilupeju et al. [9], Setyawan & Setyawan [16], Pratomo et al. [17], Irfani et al. [18], Syafriza et al. [19], Nasution & Sinaga [20], and Kalyani & Rajalakshmi [22], offered valuable insights into IoT integration, temperature-humidity monitoring, and ventilation improvement. Nevertheless, they either depended on internet connectivity or did not address autonomous layered speed regulation via PWM in a fully offline context.

By addressing these gaps, the present study introduces a temperature-responsive fan system using the ESP32 microcontroller and DHT22 sensor that operates fully offline and autonomously. The system generates PWM signals corresponding to predefined temperature thresholds and transmits them through a motor driver to achieve multi-level fan speed control. This design not only improves energy efficiency and system reliability but also delivers a responsive and practical solution for small-scale cooling applications, such as personal spaces, enclosed work areas, and temperature-sensitive electronic equipment, where autonomous and proportional control is essential.

III. *METHODOLOGY*

This research aims to design an automatic fan control system that uses a temperature sensor, which is controlled by an ESP32 microcontroller. This system utilizes a DHT22 temperature sensor to measure the ambient temperature and a motor driver as a speed regulator on a DC motor via a PWM signal. The ESP32 microcontroller receives temperature information from the DHT22, then produces a PWM signal to regulate the motor speed via the motor driver.

A. *System Design*

This system consists of five essential parts: a DHT22 sensor, an ESP32 microcontroller, a motor controller, a DC motor, and a power source. The DHT22 sensor measures ambient temperature and send the data to ESP32. Upon receiving the data, the microcontroller processes it and generates a PWM signal to control the motor speed via the motor controller.

The system design includes the following key components: the DHT22 sensor, ESP32 microcontroller, motor driver, DC motor, and power supply. DHT22 reads the ambient temperature and transmits the data to the ESP32, which then calculates the appropriate PWM value. This signal is passed to the motor driver, which regulates the power sent to the DC motor. Acting as a fan, the DC motor spins at varying speeds based on the PWM input, while the battery supplies power to all components.

Table 2. Main Components of the System

Component	**Function**
Sensor DHT22	Reads ambient temperature with high accuracy and transmits the information to the ESP32.
ESP32	Process temperature information and generate PWM signals
Motor Driver	Manages power distribution to DC motors based on PWM signals
Motor DC	Acts as a fan, speed controlled by PWM
Battery	Supplies power to the entire system

B. *Diagram Block*

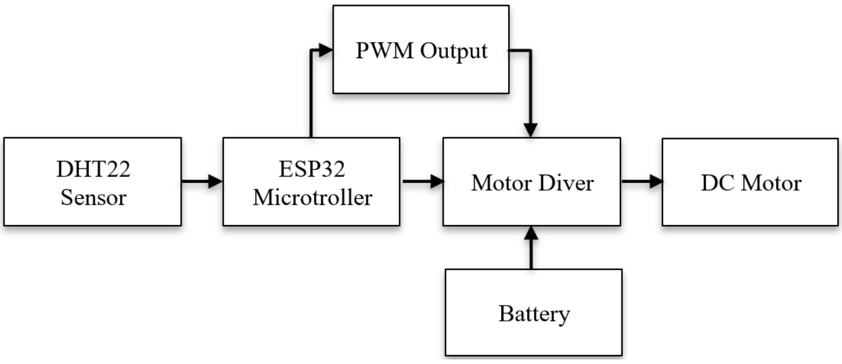

Figure 1. Block Diagram of the Automatic Fan System Based on DHT22 Temperature Sensor and ESP32

The block diagram illustrates the temperature-based fan control workflow system. The DHT22 sensor first reads the ambient temperature and sends the data to the ESP32 Microcontroller. Based on this data, the ESP32 generates a PWM signal and sends it to the motor driver. The motor driver then adjusts the voltage and current supplied to the DC motor to control fan speed. A power supply supports the entire system, ensuring stable and continuous operation.

C. *Logical Control*

The system routinely measures temperature using a DHT22 sensor. Based on the detected temperature, the ESP32 Microcontroller generates a PWM signal with a specific value to control the fan speed. The temperature control logic corresponding to each PWM value is defined as follows:

```
#include <DHT.h>
```

```
#define DHTTYPE DHT22
#define DHTPIN 12
#define OUTPIN1 25
#define OUTPIN2 26

DHT dht_22(DHTPIN, DHTTYPE);
void setup() {
  Serial.begin(9600);
  dht_22.begin();
  pinMode(OUTPIN1, OUTPUT);
  pinMode(OUTPIN2, OUTPUT);
}
void loop() {
  delay(1000);
  float temperature = dht_22.readTemperature();
  if (isnan(temperature)) {
        Serial.println("Failed to read from DHT sensor!");
        return;
  }
  if (temperature < 21) {
        ledcWrite(OUTPIN1, 0);
        ledcWrite(OUTPIN2, 0);
  } else if (temperature <= 25) {
        ledcWrite(OUTPIN1, 90);
        ledcWrite(OUTPIN2, 90);
  } else if (temperature <= 30) {
        ledcWrite(OUTPIN1, 145);
        ledcWrite(OUTPIN2, 145);
  } else if (temperature <= 35) {
        ledcWrite(OUTPIN1, 200);
        ledcWrite(OUTPIN2, 200);
  } else {
        ledcWrite(OUTPIN1, 255);
        ledcWrite(OUTPIN2, 255);
  }
}
```

Figure 2. Code snippet for PWM signal logic based on temperature thresholds

The code snippet in Figure 2 illustrates the implementation of temperature-based PWM control logic within the ESP32 microcontroller. The program begins by reading temperature data from the DHT22 sensor and then evaluates it against predefined thresholds, where each threshold corresponds to a specific PWM value sent to the motor driver to adjust the DC fan speed incrementally from low to maximum. This stepwise approach ensures smooth speed transitions and optimizes energy consumption by activating the fan only when necessary. The use of the ESP32's PWM functionality allows precise and consistent control over the motor's rotation, enabling responsive adaptation to real-time temperature variations.

D. Flowchart System

System Workflow of automatic fan can be explained with:

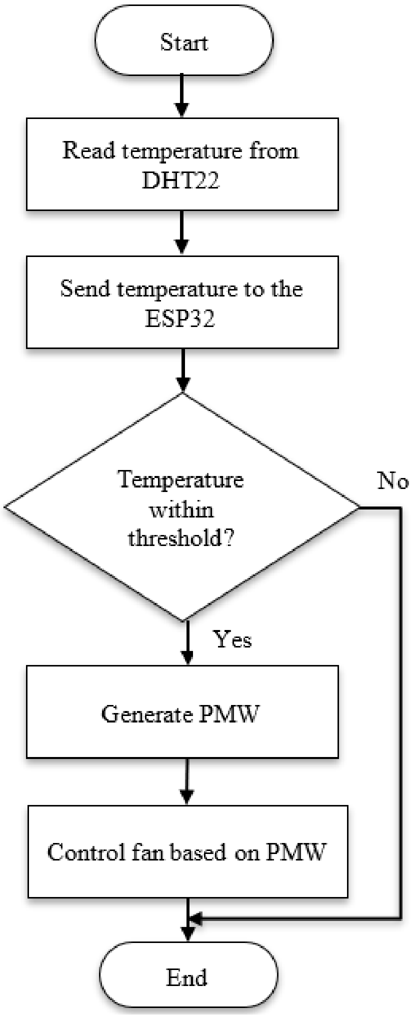

Figure 3. *Flowchart of the Automatic Fan System Workflow Based on DHT22 Sensor and ESP32*

The flowchart in Figure 3 illustrates the operational logic of the standalone automatic fan system, which uses an ESP32 microcontroller and the DHT22 temperature sensor. The process begins with sensor initialization, followed by continuous temperature monitoring at regular intervals. The DHT22 transmits ambient temperature data directly to the ESP32, which then evaluates the data based on predefined conditions. These conditions determine the appropriate PWM (Pulse Width Modulation) signal to control fan speed. For example, if the temperature is below 21°C, the PWM value is set to 0, turning the fan off. Higher temperature ranges trigger incremental PWM values 90, 145, 200, and finally 255 corresponding to increasing fan speeds from low to maximum.

The PWM signal generated by the ESP32 is sent directly to a motor driver, which regulates power to the DC motor accordingly. Since this system operates without internet connectivity, all processing and control take place locally, making it reliable and efficient for use in small-scale, enclosed environments. The cycle repeats continuously with a short delay, allowing the fan to respond dynamically to temperature changes in real time. This approach enables efficient cooling, reduces unnecessary energy use, and eliminates the need for manual operation or remote control systems.

IV. *RESULTS AND ANALYSIS*

The test was conducted by simulating temperature variations around the DHT22 sensor, then documenting the system's response in the form of the resulting PWM values and the state of the DC motor. Temperature variations were manually induced using a cold object such as an ice-filled cup to lower the surrounding temperature and a blower to raise it. These objects were positioned at incremental distances from the sensor, increasing in 3 cm intervals from 3 cm up to 18 cm, with each distance maintained for 10 seconds to ensure stable readings before moving to the next stage. This stepwise distance adjustment allowed gradual changes in ambient temperature while minimizing sudden fluctuations, enabling the system to be tested effectively at room temperatures ranging from 25°C to 35°C.

Tests on an automated fan system using a DHT22 temperature sensor and PWM controller were conducted at various temperatures to determine how the fan speed reacts to ambient temperature fluctuations. The optimal distance for sensor readings is approximately 15 cm, with a reading interval of approximately 5 seconds.

Table 3. Relationship between temperature, PWM signal, and fan status

Temperature (°C)	**PWM (0–255)**	**Fan Status**
20	0	OFF (not rotating)
24	90	Slow speed rotating
28	145	Medium speed rotating
32	200	High speed rotating
35	255	Maximum speed rotating

Table 3 shows the relationship between temperature, PWM signal, and fan status. A linear correlation can be observed between the ambient temperature and the PWM value generated by the ESP32 microcontroller. At low temperatures (20 °C), the fan remains off because the PWM value is zero. As the temperature rises to 25 °C, the system generates a PWM signal of 90, resulting in low-speed fan operation. Further increases in temperature cause the PWM to rise gradually, reaching a maximum speed at 35 °C with a PWM of 255. This confirms that the control logic implemented in the ESP32 successfully translates temperature variations into proportional fan speed regulation.

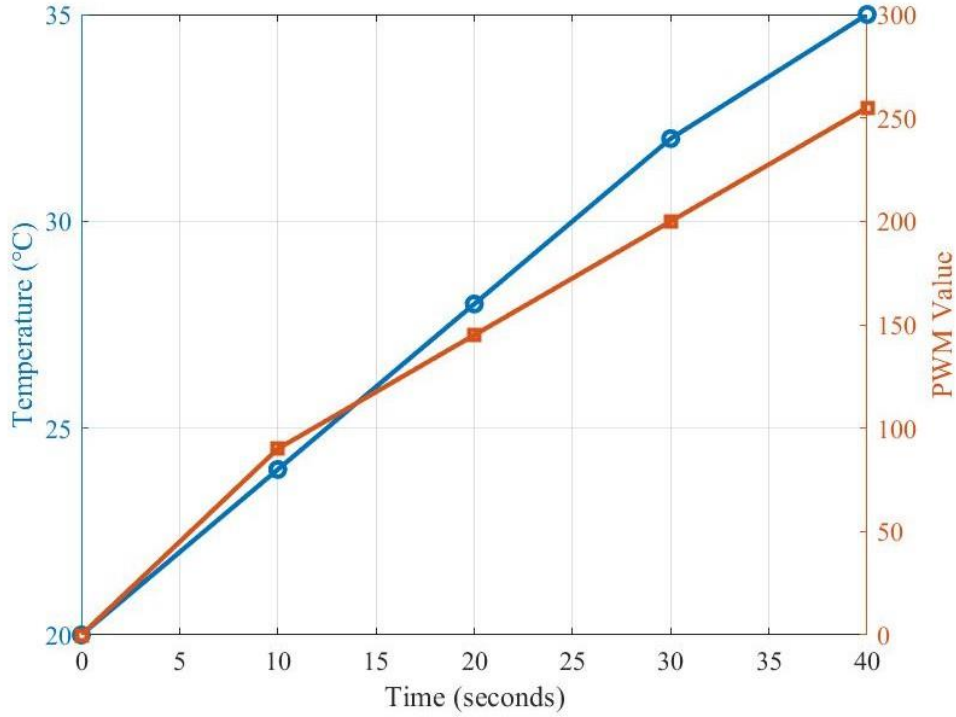

Figure 4. Temperature and PWM Response Over Time

In addition to Table 3, Figure 4 provides a visualization of the trend of temperature and PWM response over time. The graph shows that as the temperature increases, the PWM value rises in a stepwise manner, which corresponds to incremental changes in fan speed. This stepwise pattern demonstrates the stability of the system, ensuring smooth transitions between fan speeds and avoiding oscillations near threshold points. Furthermore, the visualization emphasizes the energy-efficient design of the system, since the fan only activates when required and increases its speed gradually rather than abruptly.

A. Analysis

The system's response to temperature variations indicates that the fan speed regulation operates proportionally to the detected temperature, demonstrating the effectiveness of PWM control in adjusting DC motor speed. Similar findings were also reported in previous studies, where PWM was applied to improve energy efficiency and system stability in fan control [24][25]. Compared to systems that relied only on on/off switching the proposed design demonstrates smoother speed transitions and reduced energy consumption [13][20]. Furthermore, unlike IoT-based systems that depend on internet connectivity this system is fully offline and therefore more reliable for standalone small-scale applications [2][11]. Furthermore, the ±5-second delay in sensor readings and the ±15 cm effective distance remain within acceptable limits for small room-based cooling applications or electronic device enclosures.

The PWM values observed during testing increased in a stepwise fashion according to predefined temperature thresholds. This structured approach ensured that the fan speed did not change erratically but rather followed a consistent logic based on sensor readings. For example, when the temperature was around 24°C, the PWM signal generated was 90, corresponding to a low fan speed. At 28°C, the PWM value increased to 145, demonstrating the system's capacity to scale motor speed in proportion to heat intensity. This behavior confirms the success of the logic mapping coded in the ESP32 microcontroller.

Additionally, the fan responded in real time to fluctuations in temperature, with only minimal lag attributed to the sensor's sampling delay and the system's loop interval. This responsiveness makes the system suitable for scenarios requiring immediate environmental adjustments, such as protecting temperature-sensitive electronic devices from overheating. The ±5-second delay remained stable across repeated trials, which is acceptable for passive cooling systems that do not require millisecond-level responsiveness.

Another important observation relates to the fan's behavior during borderline temperature readings. When the ambient temperature hovered near threshold points (such as between 25–26°C), the system maintained stability and avoided unnecessary switching. This shows that the PWM value assignment logic avoids "jitter" or unstable oscillations. Such stable performance at critical temperature margins highlights the system's robustness in maintaining desired conditions without frequent on-off cycles.

From an energy efficiency standpoint, the system only activates the fan when the environment surpasses certain temperature thresholds, thus reducing unnecessary power consumption. By remaining off at lower temperatures and running only as needed, the design effectively saves energy—especially important for battery-powered or embedded systems. The stepping PWM control also prevents sudden power surges that typically occur in on/off relay-based systems.

Finally, although the system demonstrated reliable performance under control conditions, some limitations were noted. For instance, the DHT11 sensor used in early testing has slower response time and lower accuracy than the DHT22, which is intended for final implementation. Environmental factors such as airflow direction and humidity may also influence readings slightly. Future work could include more precise thermal control experiments, long-duration tests, or deployment in real-use cases to further assess durability, calibration drift, and performance consistency over time.

V. *CONCLUSION*

Based on the design and testing results of an automatic fan system using an ESP32 microcontroller and a DHT22 temperature sensor, it can be concluded that the system successfully responds to changes in ambient temperature by automatically adjusting the DC motor speed via a PWM signal. The use of a motor controller proved effective in accurately regulating and stopping the DC motor. The system can automatically turn the fan on and off, as well as adjust the fan speed based on the detected temperature. This makes it suitable for use in small-scale environments that require automatic cooling, such as enclosed spaces or laboratory equipment.

Although the system performed according to its initial design, there are several potential improvements that could entrance its performance in the future. For example, adding a real-time temperature display module, such as an LCD or OLED, would make it easier for users to monitor environmental conditions directly and increase system interactivity. In terms of protection, implementation features such as fuses or current sensors is highly recommended to prevent damage caused by excessive current. Furthermore, the system could be expanded to be Internet of Things (IoT)-based application, enabling remote monitoring and control over a network. Additional testing in real application environments, such as inside electronic device enclosures or enclosed work areas, is essential to assess system performance under long-term operational conditions.

REFERENSI

[1] A. Budiyanto, G. B. Pramudita, and S. Adinandra, "Kontrol relay dan kecepatan kipas angin direct current (DC) dengan sensor temperature LM35 berbasis Internet of Things (IoT)," *Techné: Jurnal Ilmiah Elektroteknika*, vol. 19, no. 1, pp. 43–54, 2020. [Online]. Available: https://doi.org/10.31358/techne.v19i01.224

[2] R. Septian and M. A. Syari, "Design and build automatic fan control based Internet of Things," *ResearchGate*, 2024. [Online]. Available: https://www.researchgate.net/publication/385402726_Design_and_Build_Automatic_Fan_Control_Based_Internet_of_Things

[3] IJARSCT, "Temperature based fan speed control using NodeMCU (ESP8266)," *International Journal of Advanced Research in Science, Communication and Technology*, vol. 3, no. 12, pp. 45–49, 2023. [Online]. Available: https://ijarsct.co.in/Paper10642.pdf

[4] A. E. Akin-Ponnle, "An eco-friendly and low cost IoT-based room temperature control by fan speed regulation for tropical use," *ResearchGate*, 2024. [Online]. Available: https://www.researchgate.net/publication/383451082_An_Eco-Friendly_and_Low_Cost_IoT_based_Room_Temperature_Control_by_Fan_Speed_Regulation_for_Tropical_Use

[5] IJRPR, "IoT based fan speed control," *International Journal of Research Publication and Reviews*, vol. 6, no. 5, pp. 11226–11228, 2025. [Online]. Available: https://ijrpr.com/uploads/V6ISSUE5/IJRPR46203.pdf

[6] Circuit Digest, "ESP32 based smart temperature sensitive fan regulator," *Circuit Digest Projects*, Feb. 2022. [Online]. Available: https://circuitdigest.com/microcontroller-projects/esp32-based-smart-temperature-sensitive-fan-regulator

[7] P. W. Rusimamto, L. A. Endryansyah, and I. P. Indarto, "Implementation of Arduino Pro Mini and ESP32-CAM for temperature monitoring on automatic thermogun IoT-based," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 3, pp. 1615–1622, 2021. [Online]. Available: https://doi.org/10.11591/ijeecs.v21.i3.pp1615-1622

[8] E. Devmore, D. Shinde, and S. Patil, "ESP32 temperature control project with an emphasis on energy efficiency," *SSRN Electronic Journal*, 2024. [Online]. Available: https://doi.org/10.2139/ssrn.4841107

[9] A. M. Ilupeju, B. O. Adegoke, and A. Yakubu, "Temperature-dependent speed control device for an AC standing fan," *ResearchGate*, 2023. [Online]. Available: https://www.researchgate.net/publication/382891377_Temperature-dependent_Speed_Control_Device_for_an_Ac_Standing_Fan

[10] R. Pahlevi, "Room temperature monitoring and equalization system using sensors and fan," in *Proc. National Seminar on Information and Computer Technology*, STMIK Piksi Ganesha, pp. 169–174, 2024. [Online]. Available: https://journal.piksi.ac.id/index.php/proceedings/article/view/1859

[11] N. D. Setia, B. Suprianto, and R. P. Sari, "Perancangan sistem kendali exhaust fan berbasis IoT dengan sensor MQ-2 dan DHT22 untuk optimalisasi konsumsi energi," *Jurnal Teknik Elektro*, vol. 14, no. 1, pp. 1–7, 2025. [Online]. Available: https://ejournal.unesa.ac.id/index.php/JTE/article/view/69025

[12] B. Kanchanapalli, S. Banka, G. V. Bacchu, and P. R. Ganesh, "ESP32-based fan speed regulation system," in *Proc. 2024 Int. Conf. Sustainable Computing for Green and Secure Development (ICSGSD)*, pp. 1–5, 2024. [Online]. Available: https://doi.org/10.1109/ICSGSD60194.2024.10717499

[13] G. M. Debele and X. Qian, "Automatic room temperature control system using Arduino Uno R3 and DHT11 sensor," in *Proc. 2020 Int. Conf. Wavelet Active Media Technology and Information Processing (ICWAMTIP)*, pp. 320–324, 2020. [Online]. Available: https://doi.org/10.1109/ICWAMTIP51612.2020.9317307

[14] A. Jain, A. Sarkar, D. Ather, and D. Raj, “Temperature based automatic fan speed control system using Arduino,” in *Proc. Advancement in Electronics & Communication Engineering 2022, SSRN Electronic Journal*, 2022. [Online]. Available: https://doi.org/10.2139/ssrn.4159188

[15] R. Aulia, R. A. Fauzan, and I. Lubis, “Pengendalian temperature ruangan menggunakan fan dan DHT11 berbasis Arduino,” *CESS (Journal of Computer Engineering System and Science)*, vol. 6, no. 1, pp. 30–37, 2021. [Online]. Available: https://doi.org/10.24114/cess.v6i1.21113

[16] G. C. Setyawan and L. J. Setyawan, “Model sistem pengering tenaga surya berbasis IoT dengan ESP32 dan DHT-11,” *Progresif: Jurnal Ilmiah Komputer*, vol. 21, no. 1, 2025. [Online]. Available: https://doi.org/10.35889/progresif.v21i1.2398

[17] G. W. Pratomo, A. S. Puspaningrum, and I. Ismail, “Implementasi sistem otomatisasi sirkulasi udara menggunakan fan exhaust berbasis sensor DHT21 pada vertikal hidroponik,” *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 5, no. 2, pp. 656–663, 2025. [Online]. Available: https://doi.org/10.57152/malcom.v5i2.1660

[18] 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, 2025. [Online]. Available: https://doi.org/10.30588/jeemm.v9i1.2161

[19] M. Syafriza, D. Suherdi, and K. Sari, “Implementasi PWM pada sistem monitoring temperature dan kelembapan berbasis mikrokontroler,” *Jurnal Sistem Komputer*, vol. 3, no. 4, pp. 118–126, 2024. [Online]. Available: http://ojs.trigunadharma.ac.id/index.php/jskom/article/view/8916

[20] T. I. Nasution and S. H. Sinaga, “Design of monitoring system temperature and humidity using DHT22 sensor and NRF24L01 based on Arduino,” *Journal of Physics: Conference Series*, vol. 2421, no. 1, pp. 1–6, 2023. [Online]. Available: https://doi.org/10.1088/1742-6596/2421/1/012018

[21] A. Jain, A. Sarkar, D. Ather, and D. Raj, “Temperature based automatic fan speed control system using Arduino and DHT11 sensor,” *Proc. Advancement in Electronics & Communication Engineering 2022*, SSRN Electronic Journal, 2022. [Online]. Available: https://doi.org/10.2139/ssrn.4159188

[22] P. R. Kalyani and S. Rajalakshmi, “Development of IoT based intelligent temperature control system using ESP32 and DHT22,” *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, no. 2, pp. 113–116, 2019. [Online]. Available: https://www.ijeat.org/wp-content/uploads/papers/v9i2/B2167059219.pdf

[23] S. Aggarwal and M. Verma, “Temperature based automatic fan speed control system using Arduino and LM35,” *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 7, no. 5, pp. 2219–2224, 2018. [Online]. Available: https://doi.org/10.15662/IJAREEIE.2018.0705036

[24] A. Sharma and S. K. Gupta, “Smart fan control using microcontroller with PWM technique,” *International Journal of Scientific & Engineering Research (IJSER)*, vol. 10, no. 6, pp. 1381–1386, 2019. [Online]. Available: https://www.ijser.org/researchpaper/Smart-Fan-Control-Using-Microcontroller-with-PWM-Technique.pdf

[25] J. W. Kim and Y. S. Park, “Energy-efficient fan control system using proportional control algorithms and ambient sensors,” *IEEE Access*, vol. 9, pp. 55033–55041, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3071743