Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 5, **Number** 1, May 2025
https://doi.org/10.47709/brilliance.v5i1.6073

Design Of Automatic Laptop Cooling System Using Ds18b20 Temperature Sensor Based On Arduino Nano

Oktrison [1*], Erwinsyah Sipahutar[2], Rudi Arif Candra[3], Arie Budiansyah[4]
[1,2]Politeknik ATI Padang, Indonesia, [3]Politeknik Aceh Selatan, Indonesia, [4]Universitas Syiah Kuala, Indonesia
[1]oktrison88@gmail.com, [2]erwinsyah@poltekatipdg.ac.id, [3]rudiarifcandra@email.com, [4]arie.b@unsyiah.ac.id

open access

***Corresponding Author**

Article History:
Submitted: 11-06-2025
Accepted: 21-06-2025
Published: 24-06-2025

Keywords:
Tree felling ; Arduino Nano; Sensor DS18B20; Relay, Fan

MICROPY: Multidisciplinary Journal of Engineering, Science and Technology is licensed under a Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0).

ABSTRACT

Technological developments in the field of automation provide opportunities to improve efficiency and comfort in the operation of electronic devices. This research aims to design and implement an automatic cooling system on a laptop that uses an Arduino Nano-based DS18B20 temperature sensor. The system is designed to automatically regulate the laptop temperature by monitoring the temperature in real-time, and activating the cooling fan through a relay when the temperature reaches 33°C or more. This research method includes hardware design that involves the use of Arduino Nano as a microcontroller, a DS18B20 temperature sensor to detect temperature changes, and a relay to control the cooling fan. The software was developed using the Arduino programming language (C++) to process the data from the sensors and manage the work of the cooling system automatically. The test results show that the system can accurately detect the laptop temperature and respond in real-time by turning on the cooling fan when the temperature exceeds the 33°C limit. The system proved to be effective in preventing overheating, keeping the device temperature within safe limits, and optimizing power consumption by turning off the fan when the temperature returns to stable.

INTRODUCTION

The escalating computational demands of modern laptops have led to a significant challenge in thermal management, necessitating the development of efficient and adaptive cooling solutions (Jehhef, 2018). Temperature plays a crucial role in determining the quality and performance of electronic devices, and effective temperature control is paramount across diverse scientific research and productive sectors (Castro & Mestria, 2022). As electronic devices become increasingly compact and powerful, managing internally generated heat becomes a critical factor in ensuring their stable operation and longevity (Chen et al., 2023). Insufficient heat dissipation can lead to a cascade of problems, including decreased performance due to thermal throttling, reduced component lifespan, and even catastrophic hardware failures. Active cooling systems, particularly those employing forced air convection via fans, are frequently employed to mitigate heat accumulation within electronic devices, with their efficacy being directly proportional to the rate of airflow (El-Hasan et al., 2016). The need for efficient cooling mechanisms arises from the fundamental physics of computation, where electrical current flow inevitably generates heat, leading to temperature increases within the device (T.A et al., 2013). The efficiency of heat removal from chips is a crucial area of research, especially with the continuous development of CPU integration (Li & Wei, 2013). An effective cooling system must be able to respond dynamically to changing thermal loads, adjusting fan speeds to maintain optimal operating temperatures.

The overheating issue of electronic devices has spurred research into innovative thermal management solutions, including advanced heat sinks, vapor chambers, and liquid cooling systems. The integration of temperature sensors into cooling systems enables real-time monitoring of component temperatures, facilitating dynamic adjustments to fan speeds and other cooling parameters (Kanchanasatian, 2018). The goal of the present research is to create an automatic laptop cooling system that addresses these issues by using the DS18B20 temperature sensor and the Arduino Nano microcontroller.

Such systems, by carefully regulating internal temperatures, contribute to enhanced system reliability, prolonged component lifespan, and sustained peak performance. Moreover, the advancement of microelectronics technology has led to morphological transitions and diversification, with computers serving as subjects of heat transfer research (Nakayama, 2013). Traditional cooling methods, which frequently depend on passive heat sinks or fixed-speed fans, frequently fall short of providing the adaptability needed to effectively dissipate heat under varied workloads. In data centers, for example, cooling systems can consume a significant portion of the total energy, underscoring the need for more energy-efficient solutions (Azarifar et al., 2024).

The objective of this project is to design and implement a cost-effective, energy-efficient, and adaptable laptop cooling system that can maintain optimal operating temperatures under varying workloads, ultimately contributing to enhanced system reliability and performance. Furthermore, emerging cooling technologies, such as nanofluids, offer the potential for significantly enhanced heat transfer capabilities, paving the way for more compact and efficient cooling

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

E-ISSN : 2807-9035
Volume 5, **Number** 1, May 2025
https://doi.org/10.47709/brilliance.v5i1.6073

systems in the future (Das, 2006). Active cooling systems offer greater flexibility and ensure higher performance improvement, however they need operational energy and are more complex in design (Nižetić, 2020).

The integration of smart control algorithms, capable of predicting thermal loads and optimizing cooling performance accordingly, represents a promising avenue for future research in this area (Eini et al., 2021). Smart HVAC systems, leveraging temperature and humidity data, exemplify the trend toward intelligent climate control for optimized energy efficiency and comfort (Aliero et al., 2022). Such intelligent systems, by dynamically adjusting cooling parameters based on real-time conditions, can minimize energy consumption while maintaining optimal thermal performance.

The development of advanced cooling solutions is critical for maintaining the performance and reliability of electronic devices, driving innovation in thermal management technologies across diverse applications (Meijer, 2010). This project explores the application of microcontroller technology and temperature sensors to create a smart cooling solution that dynamically adjusts fan speed based on real-time temperature readings, ensuring optimal thermal performance and energy efficiency.

LITERATURE REVIEW

The design of an automatic laptop cooling system utilizing the DS18B20 temperature sensor based on Arduino Nano integrates various technological advancements to enhance thermal management in computing devices. This literature review synthesizes findings from multiple studies that highlight the effectiveness and application of the DS18B20 sensor in cooling systems. Overview of DS18B20 Temperature Sensor The DS18B20 is a digital temperature sensor known for its high precision and reliability, providing 12-bit digital output (Roihan, 2024). It is widely used in various applications, including automotive radiators and server rooms, demonstrating its versatility in temperature monitoring (Sharmila & Bhuvaneswari, n.d.). Applications in Cooling Systems in automotive applications, the DS18B20 is employed to monitor radiator temperatures, ensuring optimal cooling by activating fans when necessary (Roihan, 2024). The sensor has also been integrated into IoT-enabled cooling systems for server rooms, where it adjusts cooling based on real-time temperature data, thus preventing hardware damage (Sharmila & Bhuvaneswari, n.d.). Arduino Integration the Arduino Nano serves as a compact microcontroller platform that can effectively process data from the DS18B20, allowing for real-time temperature monitoring and control (Li, 2017). Projects have demonstrated the successful implementation of Arduino-based systems that utilize the DS18B20 for automated temperature regulation, enhancing energy efficiency and user comfort (Ngene & Gunda, 2018). While the DS18B20 sensor shows great promise in cooling applications, challenges such as sensor placement and calibration in varying environments may affect performance. Further research could explore these limitations to optimize system design.

METHOD

Instruments and materials needed in designing an automatic laptop cooling system using a DS18B20 temperature sensor based on Arduino Nano. Consisting of hardware and software, including: The hardware used in the design of an automatic laptop cooling system using a DS18B20 temperature sensor based on Arduino Nano includes.

Table 1. Hardware Requirements

No	**Name**	**Function**
1	Laptop THOSIBA windows 10	As a place to run applications in Tool Design
2	Temperature Sensor DS18B20	Measuring the temperature of a laptop
3	Mikrokontroler (Arduino Nano)	To process data from the DS18B20 sensor and control the cooling fan
4	*Fan*	Used to cool the laptop and this fan will be connected to a microcontroller
5	Relay	Activating and deactivating the cooling fan based on microcontroller commands
6	Breadboard	Assembling and connecting electronic components.
7	Jumper cable	Used to connect components
8	Power Supply	Used to supply power to all components in the system

Software required in the Design of an Automatic Laptop Cooling System Using a DS18B20 Temperature Sensor Based on Arduino Nano.

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

E-ISSN : 2807-9035
Volume 5, **Number** 1, May 2025
https://doi.org/10.47709/brilliance.v5i1.6073

Table 2. Software Requirements

No	Name	Function
1	IDE Arduino	Used to write program code, compile, and upload programs to the Arduino board
2	Library Arduino	Used to facilitate Arduino programming
3	Elektronics Simulator	Used to design and simulate electronic circuits before implementation.
4	Compiler	Programming and loading program code onto the Arduino board

Stages of Research

1. **Preparation Stage**
   In this stage, prepare the tools and materials needed to complete this final project, such as a PC, Arduino Nano, DS18B20 Temperature Sensor, Fan, 1-Channel Relay, Breadboard, Connecting Cables, and Power Supply.
2. **Design Stage**
   In this stage, the tools must be connected to each other to form a single unit.
3. **Programming Stage**
   In this stage, the program syntax is written into the Arduino IDE software.
4. **Testing Stage**
   After the previous stages have been completed correctly, in this stage, the laptop must be connected to the assembled device, and the program must be uploaded.

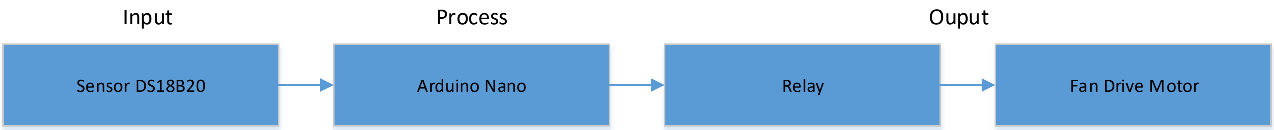


Figure 1. Schematic Design

1. Input Section
   The input section is where the DS18B20 sensor reads the temperature in a specific area of the laptop and then sends the input to the Arduino as the circuit controller.
2. Processing Section
   The Arduino, which controls the entire circuit, will process the input from the sensor according to the program that has been created and will then be forwarded by the Arduino to the relay to activate the fan motor.
3. Output Section
   The relay connected to the Arduino as the circuit's output and also connected to the fan motor will be in the on position to activate the fan motor, automatically turning on the cooling system when the laptop's temperature reaches a certain threshold.

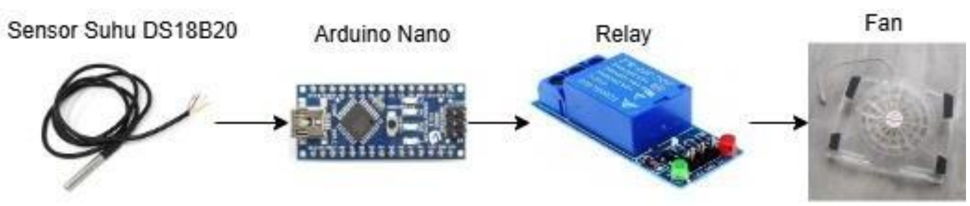


Figure 2. Equipment Connection Chart

1. The DS18B20 sensor functions as an input
2. The Arduino Nano functions as a circuit controller
3. The relay functions as a switch
4. The fan functions as a cooling system device

Starting with System Initialization, the Arduino Nano activates the DS18B20 temperature sensor and the relay connected to the cooling fan. Next, the DS18B20 sensor reads the laptop's temperature. Once the temperature is read, the Arduino Nano checks the laptop's temperature. If the temperature is below 33°C, the system keeps monitoring the temperature and doesn't activate the cooling fan. If the temperature is > 33°C, the system activates the cooling fan via the relay to prevent overheating. Once the fan is on, the system continues to monitor the laptop's temperature by monitoring the temperature on the laptop. If the temperature drops back below 33°C, the fan is turned off to conserve

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

E-ISSN : 2807-9035
Volume 5, **Number** 1, May 2025
https://doi.org/10.47709/brilliance.v5i1.6073

power. This process repeats continuously, with the system maintaining the laptop's temperature at a stable level without manual intervention from the user. As can be seen in Figure 3, the system design flowchart. The schematic design of the device shown in the figure below is used to determine which pins will be used and to connect all components into a single unit that can be followed when assembling an automatic laptop cooling device using a DS18B20 sensor.

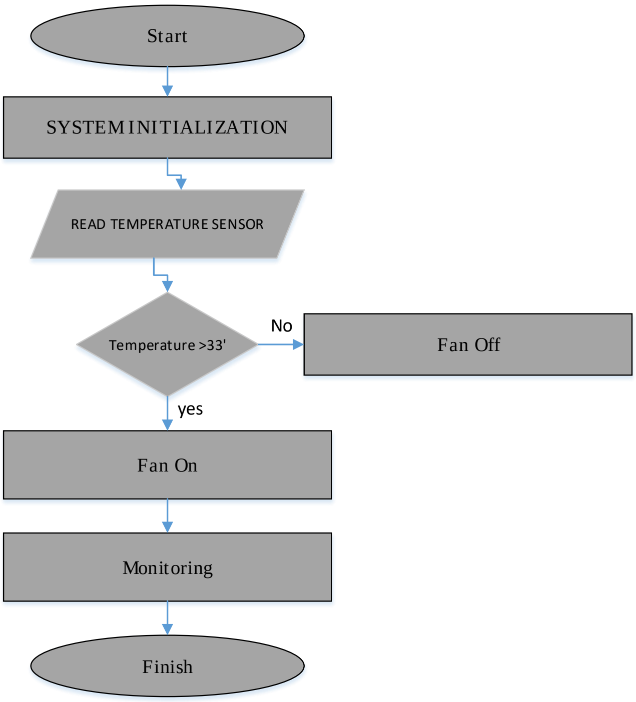


Figure 3. Flowchart system

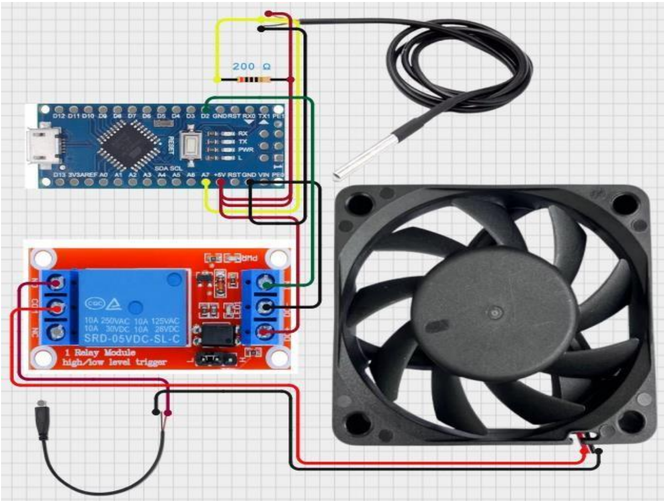


Figure 4. Schematic Design of the Device

All components are connected to each other, where these components are connected using pins whose functions correspond to the input/output of the automatic laptop cooling device using the DS18B20 sensor. The sensor pins connected to the Arduino are GND, VCC, and A0. The GND and VCC pins are connected to the GND and 5V pins on the Arduino Nano, which serves as the power source from the Arduino to the sensor, using a resistor as a current limiter for the sensor. The A0 pin is connected to the A5 pin on the Arduino as the sensor signal pin. The GND and VCC pins on the relay are connected to the GND and VCC pins on the Arduino, and the in1 pin is connected to D3 on the Arduino. The positive and negative pins on the fan are connected to the VCC pin on the Arduino, while the positive pin is connected to the MOSFET as a current balancer from the adapter and is also connected to the relay pin.

The sketch of the device shown in the image below is to provide an illustration of where the device will be placed and connected to the fan as a laptop cooler.

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 5, Number 1, May 2025
https://doi.org/10.47709/brilliance.v5i1.6073

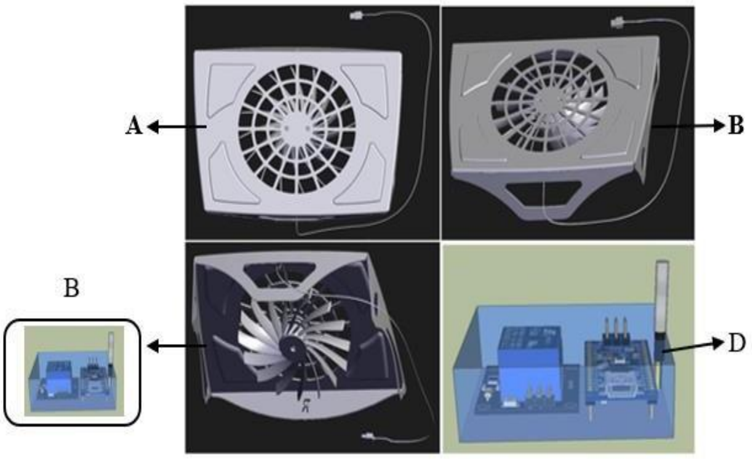


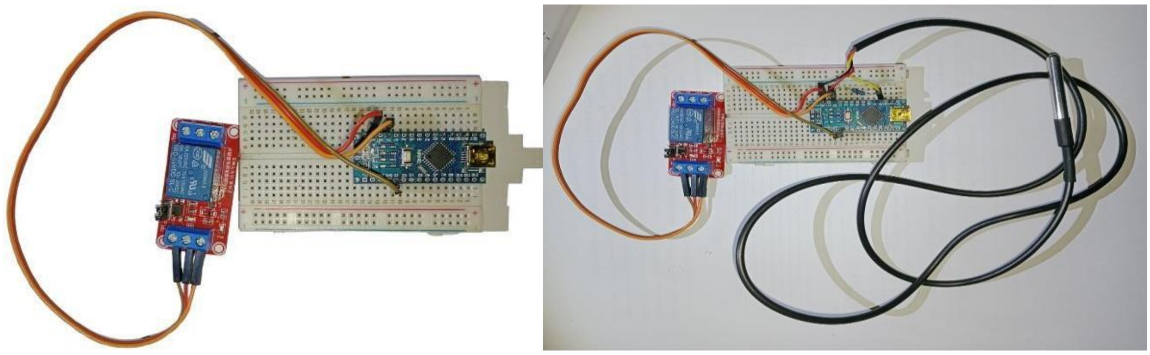
Figure 5. Sketch of the Tool

A. Front view of the laptop cooling fan using 1 cooling fan.
B. View from below and placement of the assembled device connected to the cables in the laptop fan.
C. Side view.
D. The shape of the device after assembly and placement inside the acrylic box so that when installed under the fan, the device is safe because it is protected inside the box.

RESULT

The devices that have been assembled together will display the desired results, where each module will be connected to the Arduino using pins that function as current and pins that function as signals, both analog and digital signals, which are used to supply current from the Arduino to other modules and receive input/output according to the specified functions. The 5V pin on the Arduino is connected to the VCC on the relay to provide power to the relay, while the GND pin on the relay is connected to the GND on the Arduino, ensuring that both have the same ground path. Pin A1 is connected to the data pin on the sensor, which serves as the data input from the sensor to the Arduino. Next, pin D8 on the Arduino is connected to pin IN 1 on the relay and will activate or deactivate the relay based on certain conditions. Pin A1 on the Arduino is connected to pin AO on the temperature sensor to receive an analog signal indicating the laptop's temperature. Pin GND on the temperature sensor is connected to GND on the Arduino to ensure that both have the same ground path. Meanwhile, pin VCC on the Arduino is connected to VCC on the temperature sensor to provide power to the sensor. Based on the temperature input received from the temperature sensor, the Arduino will process the data and send commands to the relay to turn the cooling fan on or off. If the laptop's temperature is detected to be high, the relay will activate the fan to cool the laptop, and if the temperature drops, the relay will turn off the fan.

The entire set of tools works as follows: when the temperature sensor detects a high temperature in the laptop, it sends a signal to the Arduino Nano as input. The Arduino then processes the data from the temperature sensor and determines whether the cooling fan needs to be turned on or off. If the laptop's temperature is detected to be high, the Arduino will activate the relay, which functions to turn on the cooling fan, and if the laptop's temperature has dropped or stabilized, the relay will turn off the fan to save power.



(a) (b)

Figure 6. Complete Set of Tools

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

E-ISSN : 2807-9035
Volume 5, **Number** 1, May 2025
https://doi.org/10.47709/brilliance.v5i1.6073

The overall appearance of the automatic laptop cooling device using the DS18B20 sensor based on Arduino Nano can be seen in the image below.

First appearance before the device is run

The appearance of the device before it is run can be seen in the following image: The image shows the appearance of the automatic laptop cooling device before it is turned on and connected to the laptop.

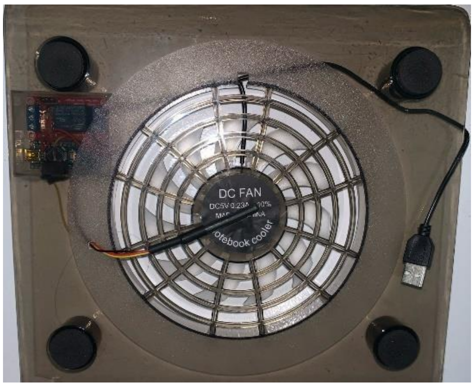


Figure 7. First view before the tool is run

Initial display after the device is connected to the laptop

The second display shows the device after it is connected to the laptop with the temperature still below 33^0 C and the fan not yet activated, as shown in the following image.

DISCUSSION

This automatic laptop cooling system can still be developed to improve its efficiency and ease of use. One possible development is the addition of a user-adjustable temperature threshold setting feature, such as using a potentiometer or control via a mobile app. With this feature, users can set the temperature threshold at which the fan should turn on or off. Additionally, integration with Wi-Fi or Bluetooth technology could be a good step, as users can control the fan remotely, enabling more flexible operation without needing to be near the laptop.

Here's a comparison of laptop temperature monitoring test results with and without the tool. This comparison was made to see how effective the tool is in keeping the laptop cool, both in an air-conditioned room and without AC. The following table shows the results of several tests conducted on the device to ensure that the performance of all components is as expected and in line with the functions of those components. The device produced significant results, as it was able to operate on 5 volts of DC power from a laptop USB port, powering both the Arduino component and the fan. This table shows the test results of the device after 3 hours of laptop use with several applications running, including Microsoft Word, Mendeley Desktop, Google Chrome, and Idle Arduino.

Table 1. Measuring Laptop Temperature Using Tools

Time (Minutes)	**Temperature (^{o}C), Ruang non AC.**	**Fan Active (Yes/ No)**	**Temperature (^{o}C), AC Room (25^{o}C),**	**Fan Aktif (Yes/ No)**
0	37.13 ^{o}C	Yes	33.06 ^{o}C	**Yes**
30	36.31 ^{o}C	Yes	32.94 ^{o}C	No
60	35.88 ^{o}C	Yes	32.94 ^{o}C	No
90	35.51 ^{o}C	Yes	32.94 ^{o}C	No
120	35.13 ^{o}C	Yes	33.06 ^{o}C	Yes
150	35.09 ^{o}C	Yes	32.94 ^{o}C	No
180	34.69 ^{o}C	Yes	32.94 ^{o}C	No

In the first experiment with the device in a room without air conditioning, the temperature of the laptop when running an application was 37.13°C, dropping to 36.31°C in the first 30 minutes and thereafter. The device performed well in the test, maintaining the temperature and preventing overheating when using the laptop.

In the second experiment, the device was run in an air-conditioned room with a room temperature of 25°C. The initial temperature was 33.06°C and decreased to 32. 94°C during the testing process with the fan in the on/off state, as the temperature remained between 33.06°C and 32.94°C, and vice versa. As a result, the temperature remained the same for the next 30 minutes. The testing results showed that the laptop's temperature remained stable without overheating, even when used for an extended period.

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 5, **Number** 1, May 2025
https://doi.org/10.47709/brilliance.v5i1.6073

CONCLUSION

After designing and implementing an Arduino Nano-based automatic laptop cooling system, the conclusions are as follows:

1. The automatic laptop cooling system was successfully designed and implemented. The Arduino Nano is capable of reading the laptop's temperature and controlling the cooling fan in real-time. After undergoing the design and testing phases, the system functions effectively to maintain the laptop's temperature at a stable level. The temperature sensor detects the laptop's temperature conditions and sends a signal to the Arduino to turn the cooling fan on or off as needed. With this system, the laptop remains cool without requiring manual adjustments.
2. Test results show that when the laptop's temperature is detected to be high, the cooling fan will automatically turn on. Conversely, when the laptop's temperature has dropped or stabilized, the fan will turn off to save power. This system is highly useful for preventing the laptop from overheating and maintaining optimal device performance.

REFERENCES

Aliero, M. S., Asif, M., Ghani, I., Pasha, M. F., & Jeong, S. R. (2022). Systematic Review Analysis on Smart Building: Challenges and Opportunities. Sustainability, 14(5), 3009. https://doi.org/10.3390/su14053009

Azarifar, M., Arık, M., & Chang, J.-Y. (2024). Liquid cooling of data centers: A necessity facing challenges. Applied Thermal Engineering, 247, 123112. https://doi.org/10.1016/j.applthermaleng.2024.123112

Castro, A. C. M. M. C. e, & Mestria, M. (2022). Temperature Control System Using Mobile Application Interface. Deleted Journal, 5(1), 1. https://doi.org/10.26417/729pbt84

Chen, X., Gong, Z., Zhao, X., Zhou, W., & Yao, W. (2023). A machine learning surrogate modeling benchmark for temperature field reconstruction of heat source systems. Science China Information Sciences, 66(5). https://doi.org/10.1007/s11432-021-3645-4

Das, S. K. (2006). Nanofluids—The Cooling Medium of the Future. Heat Transfer Engineering, 27(10), 1. https://doi.org/10.1080/01457630600904585

Eini, R., Linkous, L., Zohrabi, N., & Abdelwahed, S. (2021). Smart building management system: Performance specifications and design requirements. Journal of Building Engineering, 39, 102222. https://doi.org/10.1016/j.jobe.2021.102222

El-Hasan, T. S., Alia, M. A. K., Saluos, W. A. A., & Al-Janaideh, A. (2016). Arduino and Labview Based Control for Efficient Drive of Cooling Fan System. Research Journal of Applied Sciences Engineering and Technology, 13(10), 771. https://doi.org/10.19026/rjaset.13.3351

Jehhef, K. A. (2018). Experimental and Numerical Study Effect of Using Nanofluids in Perforated Plate Fin Heat Sink for Electronics Cooling. Journal of Engineering, 24(8), 1. https://doi.org/10.31026/j.eng.2018.08.01

Kanchanasatian, K. (2018). Automatic Speed Control and Turning ON/OFF for Smart Fan by Temperature and Ultrasonic Sensor. IOP Conference Series Materials Science and Engineering, 325, 12022. https://doi.org/10.1088/1757-899x/325/1/012022

Li, M. D., & Wei, S. (2013). Heat Generation Characteristics and Heat Dissipation Experiment of CPU Chip. Applied Mechanics and Materials, 437, 1057. https://doi.org/10.4028/www.scientific.net/amm.437.1057

Meijer, G. I. (2010). Cooling Energy-Hungry Data Centers. Science, 328(5976), 318. https://doi.org/10.1126/science.1182769

M.M., S., Mehedy, M., Rahman, Md. M., Ali, Md. T., Rahman, M., & Bagdadee, A. H. (2022). Using microcontroller based solar power system for reliable power supply. Journal of Applied and Advanced Research, 18. https://doi.org/10.21839/jaar.2022.v7.7550

Nakayama, W. (2013). Heat in Computers: Applied Heat Transfer in Information Technology. Journal of Heat Transfer, 136(1). https://doi.org/10.1115/1.4025377

Nižetić, S. (2020). Thermal Management Of Silicon Photovoltaic Panels: A Review (In the press) [Review of THERMAL MANAGEMENT OF SILICON PHOTOVOLTAIC PANELS: A REVIEW (In the press)]. Resource-Efficient Technologies, 3. https://doi.org/10.18799/24056537/2020/3/276

Rao, D. M. N., Devi, P. L., Mahitha, K. K., & Kumar, K. P. (2017). Temperature based fan speed control and observing utilizing arduino. International Journal of Engineering & Technology, 7, 685. https://doi.org/10.14419/ijet.v7i1.1.10828

T.A, A., M.S.K, K., H.S.A, G., S.K, K., & P.O, N. (2013). Heat Dissipation in a Computer. Deleted Journal, 3(6), 43. https://www.iiste.org/Journals/index.php/JETP/article/download/6464/6560

Roihan, R. (2024). Analisa Kelayakan Sensor Suhu dengan DS18B20 di Radiator Mobil Berbasis Arduino. Journal of Informatics and Communication Technology (JICT), 6(1), 127–131. https://doi.org/10.52661/jict.v6i1.269

Ngene, C. U., & Gunda, M. B. (2018). An Arduino-Based Thermal Comfort System. ATBU Journal of Science, Technology and Education, 6(4), 284–292. https://www.atbuftejoste.com/index.php/joste/article/view/513/pdf_424

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

E-ISSN : 2807-9035
Volume 5, **Number** 1, May 2025
https://doi.org/10.47709/brilliance.v5i1.6073

Saha, R., Biswas, S., Sarmah, S., Karmakar, S., & Das, P. (2021). A Working Prototype Using DS18B20 Temperature Sensor and Arduino for Health Monitoring. 2(1), 33. https://doi.org/10.1007/S42979-020-00434-2

Khairi, M. G., Gurning, M. I., & Furqan, M. (2024). Perancangan Sistem Kontrol Pendingin Udara Otomatis Berbasis Suhu Ruangan Menggunakan Arduino. E-Jurnal Ilmu Komputer Dan Sistem Informasi, 3(1), 61–71. https://doi.org/10.70340/jirsi.v3i1.96

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.