ComTech: Computer, Mathematics and Engineering Applications, 13(2), December 2022, 75−85 P-ISSN: 2087-1244
DOI: 10.21512/comtech.v13i2.7433 E-ISSN: 2476-907X

Modeling Automatic Room Temperature and Humidity Monitoring System with Fan Control on the Internet of Things

Fivtatianti Hendajani[1]*, Arif Mughni[2], Ire Puspa Wardhani[3], and Abdul Hakim[4]

[1-4]Department of Information Systems, STMIK Jakarta STI&K
Jln. BRI No 17, Jakarta Selatan 12140, Indonesia
[1]fivtatiantihendajani@gmail.com; [2]arifmughni98@gmail.com;
[3]irepuspa@gmail.com; [4]hkiem09@gmail.com

Received: 15th June 2021/ **Revised:** 29th November 2021/ **Accepted:** 30th November 2021

How to Cite: Hendajani, F., Mughni, A., & Wardhani, I. P., Hakim, A. (2022). Modeling Automatic Room Temperature and Humidity Monitoring System with Fan Control on the Internet of Things. *ComTech: Computer, Mathematics and Engineering Applications, 13*(2), 75−85. https://doi.org/10.21512/comtech.v13i2.7433

***Abstract* -** The Internet of Things (IoT) aims to expand the benefits of being connected to the Internet network continuously. It functions as a control system that has been widely applied in various fields because in certain case people are not allowed in certain rooms for security reasons. The research aimed to create a temperature and humidity monitoring system using as many fan controls as expected by utilizing IoT. The model used input in the form of temperature and humidity sensors. The output was a motor driver that drove a fan and used a microcontroller as the main processor. IoT-based systems consisted of hardware and software. Hardware included NodeMCU ESP8266 V3, DHT22 sensor, L298N motor driver module, fan, and computer. Meanwhile, the microcontroller software was made using Arduino IDE. From the test results, the system model works well. Fan control is set manually based on desired room temperature and humidity monitoring based on IoT. A mobile phone can also monitor temperature and humidity and control fans. The DHT22 sensor can read temperature and humidity every two seconds so that the resulting data is stable to display. Then, the L298N motor driver can adjust the fan speed with Pulse Width Modulation (PWM) using analog data ranging from 1 to 1.024.

Keywords: automatic room temperature, humidity monitoring system, fan control, Internet of Things (IoT)

I. INTRODUCTION

Technological advances have continued growing rapidly and created various tools to help complete human tasks. Internet of Things (IoT) is a term that has recently begun to emerge, but only some people understand the meaning of this term. In general, it can be interpreted as objects around people that can communicate with each other through the Internet (Adani & Salsabil, 2019).

Approximately there are two billion Internet users around the world. The Internet is the major platform for surfing the web, sending and receiving emails, social connections/networking, audio/video connectivity, online games, and others (Shahid & Aneja, 2017). The IoT environment requires infrastructure to store and process data for decision-making. The data collected globally are not enough to be stored centrally in one location, or it can be more than one and distributed location. So, cloud services have become an integral solution for IoT applications. The IoT architecture consists of five phases: data collection, data transmission, protocols for data delivery, storage and processing platforms, and decision-making. Research on the use of IoT is growing rapidly because every sector in society seems to replace humans in certain situations. IoT applications can automate the environment by minimizing human intervention (Paul & Jeyaraj, 2019). Moreover, connecting objects around people with a powerful Internet network, making it able to communicate between objects and users, is a concept from the IoT. Hence, this concept will make it easier for users if applied in life (Putra, Lhaksmana, & Adytia, 2018). For example, used IoT for vehicle maintenance monitoring systems (Attubel, Siswanto, & Mukhsim, 2019).

Many IoT-based temperature and humidity monitoring systems have been made with various specifications. For instance, it is recommended to use a temperature and humidity monitoring system for successful oyster mushroom cultivation. The

*Corresponding Author

system captures information in cultivation every three minutes. The data are obtained from the server. It checks the Cayenne application from the smartphone. Smartphones are used to monitor nodes, not only on Android but also on iOS-based phones. If there is no Internet connection, farmers can check the temperature and humidity conditions through the Liquid Crystal Display (LCD) on the device (Adhiwibowo, Daru, & Hirzan, 2020).

Another example is the design and implementation of a temperature and humidity monitoring system in research conducted by (Ab Rahman, Hashim, & Ahmad, 2020). It discusses how to get a better perspective on the temperature and humidity distribution of data center data at the Muadzam Shah Polytechnic. IoT platforms, such as AT&T M2X, have greatly reduced the hassle of setting up complex storage to record and analyze temperature and humidity.

Then, research related to monitoring temperature and humidity is also carried out in relation to controlling Air Conditioner (AC) using Raspberry Pi, a DHT22 sensor as a temperature sensor, and Infrared as an AC control sensor and companion apps. The communication protocols are required to implement IoT, namely Message Queuing Telemetry Transport (MQTT) protocol (Vinola, Rakhman, & Sarjana, 2020). The IoT-based server room temperature and humidity monitoring system is also conducted by x using Raspberry Pi and Arduino, assisted by a DHT22 sensor (Utomo, Aziz, & Harjito, 2019). Based on existing research, a temperature and humidity monitoring model is made with manual fan control to obtain the expected value.

Based on the explanation, the research problem is how to monitor temperature and humidity automatically via IoT. Once the temperature and humidity match the expected values, the fan can also be controlled via IoT. The research is expected to be applied to special rooms, such as server rooms, production rooms, and other rooms.

II. METHODS

The system is designed to make it easier for humans to monitor the temperature and humidity in the room and control fans based on IoT technology. Making a temperature and humidity monitoring system model in a room with IoT-based fan control includes hardware manufacturing. The research uses the NodeMCU ESP8266 V3 as the operating brain. It is supported by several other components, such as DHT22 as a temperature and humidity detector which will be processed by the microcontroller and controlling the fan to cool the temperature when the room is hot based on IoT. The block diagram of the system can be seen in Figure 1.

The DHT22 is a temperature and humidity sensor using a polymer capacitor detecting elements and outputting a digital signal with conversion and calculation by the sensor itself. This sensor works at 3,3–6 Volt Direct Current (VDC) and has a wide measurement range of 0 to 100% for humidity and -40°C to 80°C for temperature (Mihai, 2016). The choice of using the DHT22 sensor is based on research conducted by (Puspasari, Satya, Oktiawati, Fahrurrozi, & Prisyanti, 2020). The average error value in temperature and humidity measurements between the DHT22 sensor and a standard thermohygrometer produces a value of 2,99% for humidity and -2,31% for temperature. The difference in temperature and humidity value on the DHT22 sensor is according to the datasheet in the DHT22 sensor with a range between 2%–5% for the measured humidity and ±5°C for temperature.

The ESP8266 is the name of a microcontroller designed by Espressif Systems. The ESP8266 is a self-contained Wi-Fi networking solution offering a bridge from the existing microcontroller to Wi-Fi and can also run self-contained applications. This module has a built-in USB connector and a rich assortment of pinouts. With a micro-USB cable, it can connect the NodeMCU devkit to the laptop and flash it without

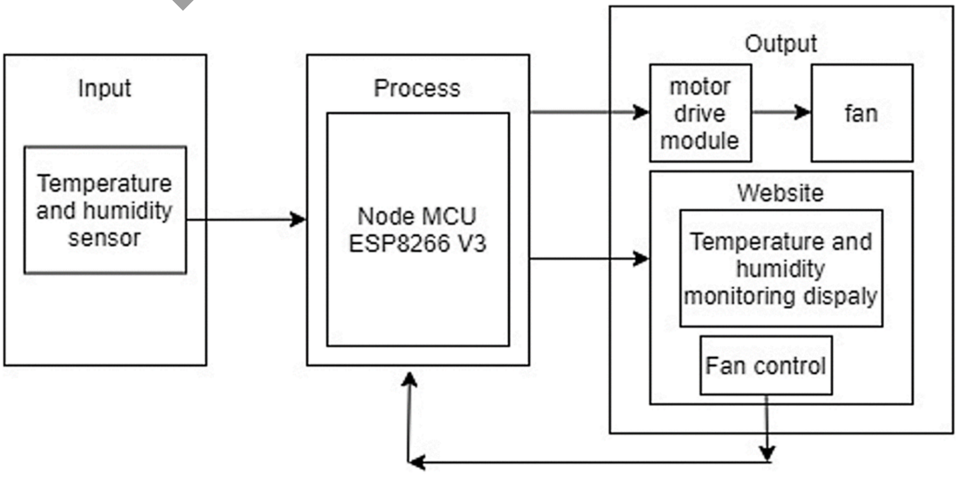


Figure 1 Block Diagram System

any trouble, just like Arduino. It is also immediately breadboard friendly. The shape of the NodeMCU ESP8266 board can be seen in Figure 2 (Selvarasi, Aswini, & Pandian, 2019).

According to (Al Dahoud & Fezari, 2018), NodeMCU V3 is an open-source firmware and development kit that plays a vital role in designing an IoT product using a few script lines. Multiple General Purpose Input/Output (GPIO) pins on the board allow the researchers to connect the board with other peripherals and generate Pulse Width Modulation (PWM), Inter Integrated Circuit (I2C), Serial peripheral Interface (SPI), and Universal Asynchronous Receiver Transmitter (UART) serial communications. The module interface is mainly divided into two parts, including firmware and hardware, where the former runs on the ESP8266 Wi-Fi SoC and later is based on the ESP-12 module.

L298N is used as a Direct Current (DC) motor driver. This Integrated Circuit (IC) uses the working principle of H-Bridge. Each H-Bridge is controlled using a Transistor-Transistor Logic (TTL) voltage level from the microcontroller output. The motor driver module pin schematic can be seen in Figure 3. L298N can control two DC motors. The voltage that can be used to control the dc motor can reach 35 VDC, and the current reaches 2 A for each channel (Budiyanta, Tanudjaja, & Mulyadi, 2019).

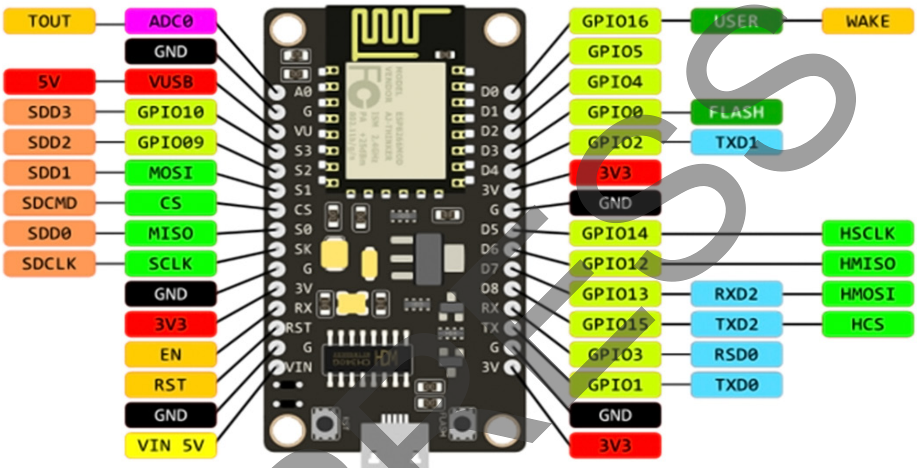


Figure 2 NodeMCU ESP8266 V3

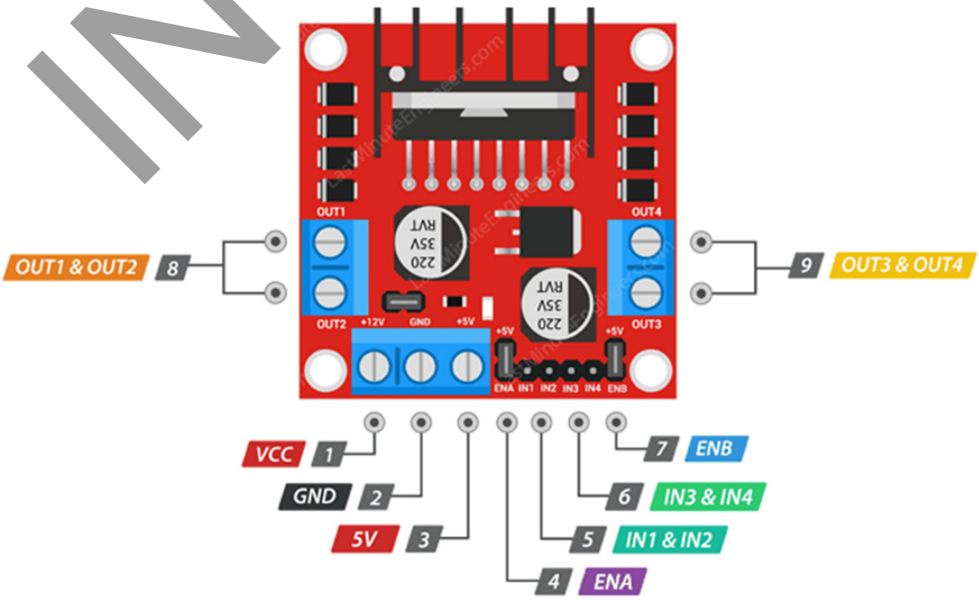


Figure 3 L298N Motor Driver Module Pin Schematic

The working principle of DHT22 is basically two-wire bidirectional communication. Bidirectional communication is run through a reciprocal relationship (sender and receiver) between the sensor and the Micro Controller Unit (MCU) to avoid communication errors. Synchronization is needed using Serial Clock Input (SCK). This sensor system has one data line on the second leg, which addresses and reads commands. Next, Pin D8 NodeMCU serves to receive data sent by DHT22. When the sensor is active HIGH (1), the data sent from the DHT22 is in the form of digital data so that the MCU can directly process it for display. When the sensor is not active LOW (0), the digital data will stop being sent to the MCU and will not be processed. The voltage connected to the 3,3 V pin is connected to the VDD pin, and the GND pin is connected to the GND pin located on the NodeMCU ESP8266 V3 (Putri & Rahayu, 2020).

III. RESULTS AND DISCUSSIONS

In the DHT22 sensor module circuit, the Voltage Drain Drain (VDD) pin on the module is connected to the 3,3 Volt pin as a power source. Then, the Ground (GND) pin on the module is connected to the GND pin as Ground, and the data pin on the sensor is connected to the D8 pin on the NodeMCU ESP8266 V3 as a data line for readings from the surrounding environment. When the sensor is active (HIGH), the data sent from the DHT22 are in the form of digital data, so there is no need for conversion such as Analog to Digital Converter (ADC). The data received are processed by the microcontroller to be displayed on the website page in the form of temperature and humidity. When the sensor is not active (LOW), the digital data will stop being sent to the MCU because it will not be processed and will not be displayed on the website page.

After the program runs, NodeMCU ESP8266 V3 connects to the specified Access Point (AP). If it fails to connect to the AP, it will continue to try to reconnect until it is successfully connected and gets an IP address. IP addresses are obtained by means of the Dynamic Host Configuration Protocol (DHCP). It is a service that allows devices to distribute IP addresses to hosts on a network automatically. The DHCP server on the AP will respond to requests sent by the DHCP client on the microcontroller. Then, this IP address can be used to open a website page that has been created to display the output of the DHT22 in the form of temperature and humidity and display fan control using the available buttons on the website page.

The fan is a Brushless Motor DC (BLDC) using an electronic circuit and serves to provide voltage output to the coil/winding. So, the winding raises a force (push) to rotate the propeller (impeller). It distinguishes a BLDC motor from a DC motor which uses mechanical brushes located on the commutator to set the time and provide a magnetic field to the windings.

In the fan circuit with the L298N module, there are two lines for output: lines A and B. This circuit only uses the output of line A because it only uses one fan. Then, the L298N module will be given a 9 V voltage as an additional power source. The IN_1 and IN_2 pins on the L298N module are connected to pins D6 and D7 as fan inputs so that they can be programmed by the microcontroller using HIGH (1) and LOW (0). Then, the EnableA pin on the L298N module is connected to pin D5 on the NodeMCU ESP8266 V3. The EnableA pin is used to adjust the PWM to control the fan speed using analog data that can be set from 1 to 1.024. The fan on this prototype tool uses 1.024 analog data so that the results of maximum speed will work. Then, the OUT1 pin for the positive pin and OUT2 for the negative pin is connected to the fan as an output.

The fan can be controlled via the available ON/OFF button on the website. When entering the beginning of the website, the fan does not turn on, and there is a fan status of OFF. The fan will turn on when the button is first pressed, and the fan status changes to ON. The fan speed rotates according to the analog data that have been set. Then, when the button is pressed again, the fan will stop spinning, and the fan status will return to OFF. On the working principle of the fan, in a magnetic field, the same poles will repel each other. Meanwhile, if the poles are different, they will attract each other. So, if we have two magnets and mark one side of the magnet with north and south, the north side will try to attract the south. Otherwise, the north side of the first magnet will repel the second north side, and so on if both sides of a magnet have the same poles. The fan in this circuit will work when the button is pressed through the website that has been created. The Voltage Common Collector (VCC) pin on the fan is connected to the OUT1 pin and the GND to the OUT2 pin on the L298N module.

Then, the module is given a 9 V power supply for additional voltage. If a 5 V power supply is used, the voltage will be less, and the fan cannot turn on. Next, the ENA (Enable A) pin on the L298N module is used to adjust the fan speed using PWM, which can be set in analog data ranging from 1 to 1.024. As in the program using a value of 1.024, the fan rotation speed will be maximum. This pin is connected to pin D5 on the NodeMCU. Meanwhile, pin IN_1 is connected to pin D6 on NodeMCU, and pin IN_2 is connected to pin D7 on NodeMCU. The complete series of the system made can be seen in Figure 4.

Next, a website is a collection of various web pages. They are summarized in a domain or subdomain located on the World Wide Web (WWW) on the Internet. Web pages are usually documents written in Hyper Text Markup Language (HTML) format, which can be accessed via Hyper Text Transfer Protocol (HTTP). Then, HTTP is a protocol that conveys various information from the website server to be displayed to users via a web browser. Website is often also called the web, which can be interpreted as a collection of pages that display various kinds of text information, data, still or moving images, animation data, sound, and video or a combination of all of them, both static and dynamic to form a series of interconnected buildings. Each of them is linked by a network of pages or hyperlinks (Jacksi & Abass, 2019).

The first text that should always be sent is the <! DOCTYPE>, which indicates sending HTML code. The <meta> viewport element makes the web page responsive in the web browser. Meanwhile, the title tag defines the page title. Next, it uses Cascading Style Sheets (CSS) to style the appearance of the web page. It chooses the Helvetica font, specifying the content displayed as inline-block and centered. Then, it sets the color, font, and margins. Setting the web page's title, such as the image below the text, can be changed to whatever is desired. Meanwhile, to dynamically display temperature and humidity values, the researchers put those values in paragraph tags. These values are displayed as integers. The researchers use the HTML entity to display the degree symbol, like in Figure 5.

The flow chart can be seen in Figure 6. It can be explained that this system first initializes all devices consisting of active Wi-Fi when looking for an AP with a HIGH status (1), a DHT22 Sensor with a HIGH status (1), and an initial fan that is LOW (0) on the Microcontroller. After the program runs, the NodeMCU ESP8266 V3 will connect to the specified AP. If it fails to connect to the AP, it will continue to try to reconnect until it is successfully connected and gets an Internet Protocol (IP) address. This IP address can open a website page that has been installed.

The temperature and humidity monitoring

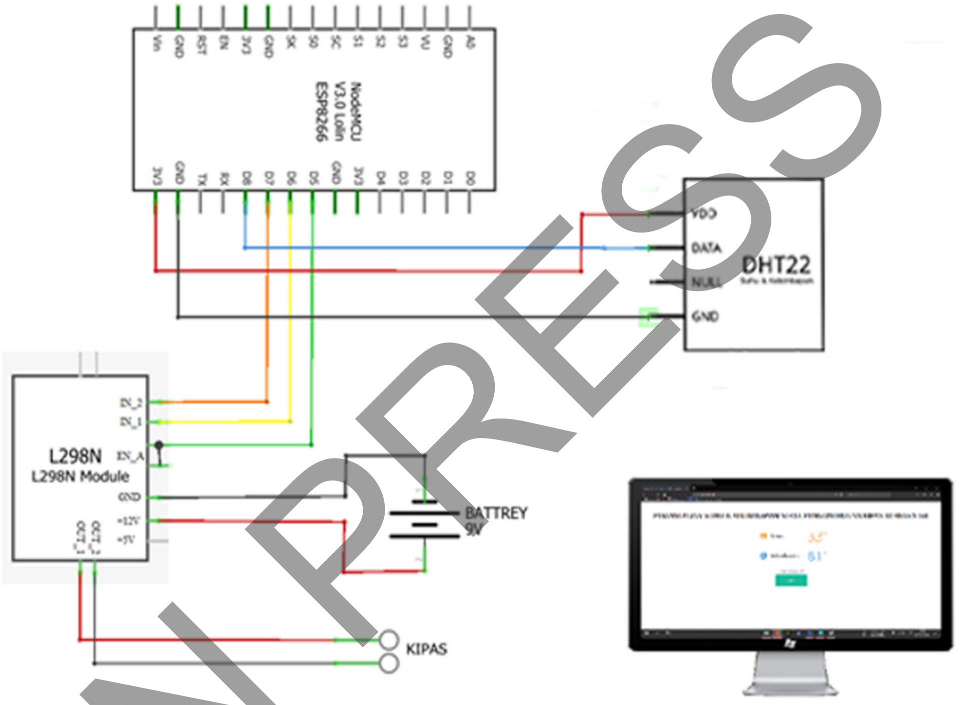
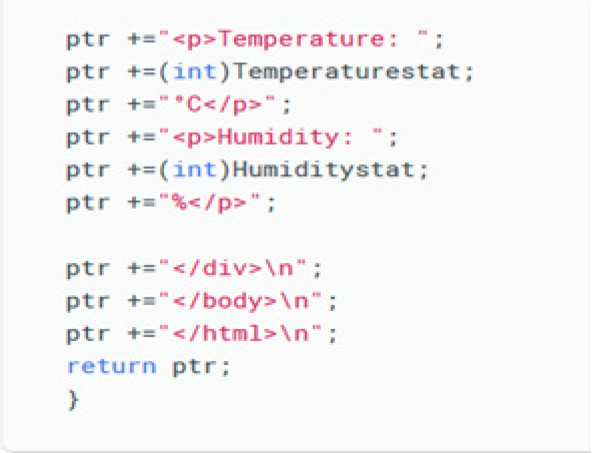


Figure 4 Automatic Room Temperature and Humidity Monitoring System with Fan Control on the Internet of Things



```
ptr +="<p>Temperature: ";
ptr +=(int)Temperaturestat;
ptr +="°C</p>";
ptr +="<p>Humidity: ";
ptr +=(int)Humiditystat;
ptr +="%</p>";

ptr +="</div>\n";
ptr +="</body>\n";
ptr +="</html>\n";
return ptr;
}
```

Figure 5 Displaying Temperature and Humidity on the Website

system model in this room takes data through the DHT22 sensor in the form of temperature and humidity that has been processed by the microcontroller and displays through the website. If the DHT22 sensor changes in temperature or humidity, the data obtained will change every two seconds and send it back to the web for display. Delay in every two seconds means reloading the website page (refreshing) automatically by programming on the Arduino Integrated Development Environment (IDE) software. The button on the website page must be pressed to control the fan. The fan will spin when the ON button is pressed and will stop spinning when pressing the OFF button.

The program that has been made is written to the NodeMCU ESP8266 V3 memory using special software, namely Arduino IDE. It is free-of-charge software, so users only need to download it. The Arduino IDE allows users to add and remove existing libraries.

In this programming design, the NodeMCU ESP12E Module board is used. It is equipped with a Universal Serial Bus (USB) to serial-chip so that users do not need additional hardware to program the Sera microcontroller. The required voltage is from USB, so the NodeMCU ESP8266 V3 board can work. Before starting to write the program code that will be downloaded to the microcontroller, the settings are made. It looks at the port to see whether the NodeMCU ESP8266 V3 has been connected or not. Next, the link "http://arduino.esp8266.com/stable/package_esp8266com_index.json" is entered to download the board in the additional board manager. Then, it looks for the Tools menu, selects Board, and clicks Board Manager to search for boards on the Arduino IDE. Last, it selects the installed board and NodeMCU 1.0 (ESP-12E Module) again by clicking on the Board Manager option.

After the setup process that the ESP8266 1.0 board (ESP-12E Module) has been connected to the laptop/PC, the next step is to write coding or program code which will be downloaded to the ESP8266 V3 NodeMCU memory. Before the program is downloaded, the user must perform the compilation process by pressing the Verify button or by pressing Ctrl + R. If the program code written is correct or there are no errors, the download process is carried out by pressing the Upload button or Ctrl + U as shown in Figure 7 when the downloading process is completed.

The first test starts with testing the DHT22 sensor using a serial monitor to determine the resulting output, namely temperature and humidity, by activating and programming the connectivity on the pins found on the DHT22 sensor with the NodeMCU ESP8266 V3 pin. As seen in Figure 8, the DHT22 sensor gets input and is successfully processed by the microcontroller. The results show the output in the form of temperature and humidity detected from the surrounding environment with changes every two seconds. The result is displayed on the serial monitor. This test is carried out twice during the day and night, with temperature and humidity data taken every ten minutes, as shown in Table 1.

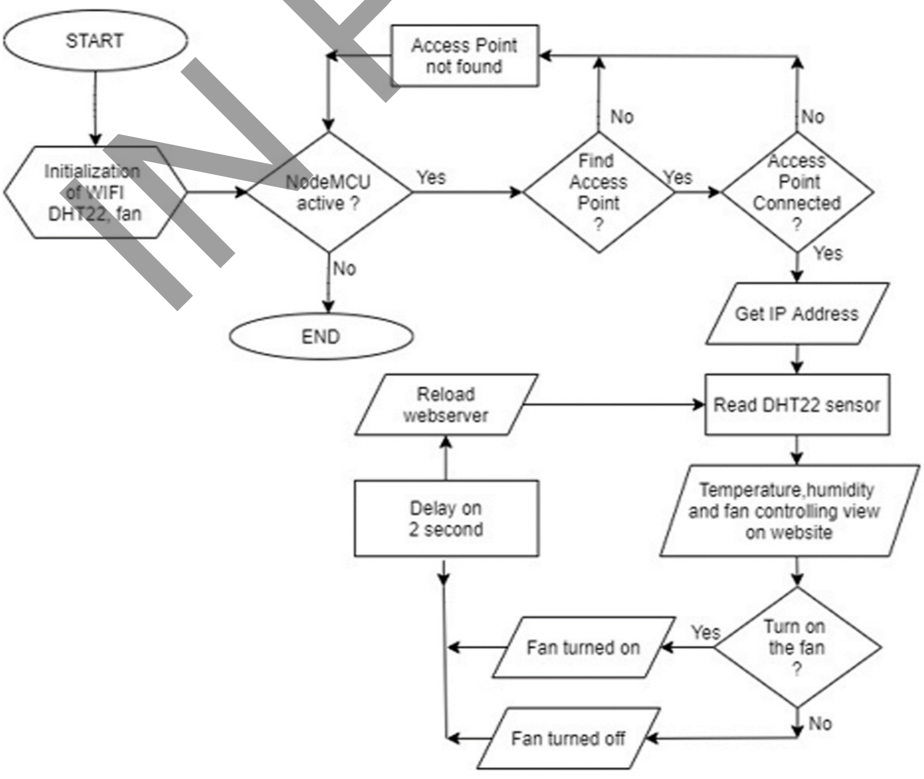
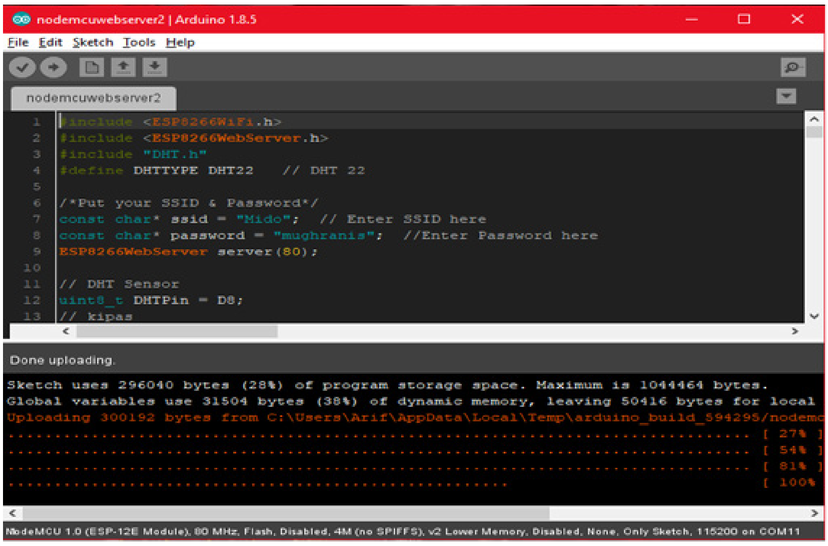


Figure 6 Flowchart System


nodemcuwebserver2 | Arduino 1.8.5
File Edit Sketch Tools Help
nodemcuwebserver2

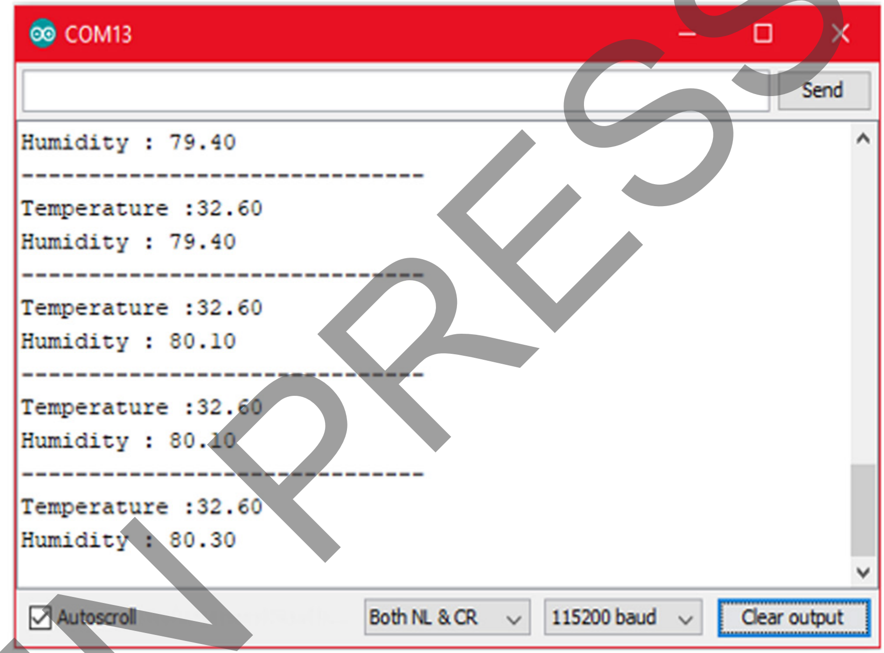
```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include "DHT.h"
#define DHTTYPE DHT22   // DHT 22

/*Put your SSID & Password*/
const char* ssid = "Mido";  // Enter SSID here
const char* password = "mughranis";  //Enter Password here
ESP8266WebServer server(80);

// DHT Sensor
uint8_t DHTPin = D8;
// kipas
```

```
Done uploading.
Sketch uses 296040 bytes (28%) of program storage space. Maximum is 1044464 bytes.
Global variables use 31504 bytes (38%) of dynamic memory, leaving 50416 bytes for local
Uploading 300192 bytes from C:\Users\Arif\AppData\Local\Temp\arduino_build_594295/nodemc
............................................................................ [ 27% ]
............................................................................ [ 54% ]
............................................................................ [ 81% ]
.....................................................                        [ 100%
NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled, 4M (no SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM11
```

Figure 7 Downloading Process is Completed and Successfully Done



```
Humidity : 79.40
--------------------------------
Temperature :32.60
Humidity : 79.40
--------------------------------
Temperature :32.60
Humidity : 80.10
--------------------------------
Temperature :32.60
Humidity : 80.10
--------------------------------
Temperature :32.60
Humidity : 80.30
```

☑ Autoscroll

Figure 8 Test Results Regarding Output from DHT22

Table 1 DHT22 Sensor Test during Day and Night

No	Temperature (Celcius)		Humidity (%)	
	Day	Night	Day	Night
1	32,5	30,5	76,9	73,5
2	32,8	30,3	74,6	75,7
3	33,1	29,7	73,8	74,0
4	32,9	30,1	69,7	77,4
5	33,4	29,6	72,3	80,2
6	33,6	28,4	69,8	74,1
7	33,2	27,9	67,4	81,4
8	33,7	28,0	73,7	75,6
9	34,2	28,2	75,0	79,5
10	32,8	27,5	70,3	82,4

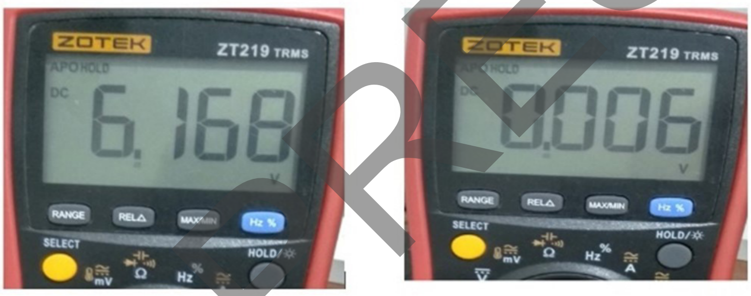
In the second test, a fan with an L298N motor driver is used to see the results obtained when the motor driver gets input to control the fan with a pin connected to the microcontroller. The measurement results can be observed using a digital multimeter when the fan is on, the pin connected to the fan will be logic 1 (HIGH), the fan is on (ON), as shown in Figure 9a and will be logic 0 (LOW) which means the fan is in a state of ON. off (OFF) in Figure 9b.

The third test is carried out to see the appearance of the website page using a device that is already connected to each other's AP. It can open the website page with a browser on a computer or smartphone using the IP address that has been obtained. The appearance of the website can be seen in Figure 10. It explains *sistem pemantauan suhu & pengontrolan kipas berbasis* IoT (temperature monitoring and fan controlling system based on IoT), *suhu* (temperature), *kelembapan* (humidity), and *kipas* (fan). Before doing the test, check the Wireless Fidelity (Wi-Fi) first, whether it is connected to the Access Point (AP).

After the program runs or resets, the users should make sure that the AP is on. After it is connected, and the IP address is recognized, users can open a web page with a browser on a computer or smartphone. Website pages can be accessed using a device that is already connected to the same AP or by connecting the Wi-Fi emitted by the microcontroller. The view of the website on the computer can be seen in Figure 11, and the display on the smartphone is shown in Figure 12.

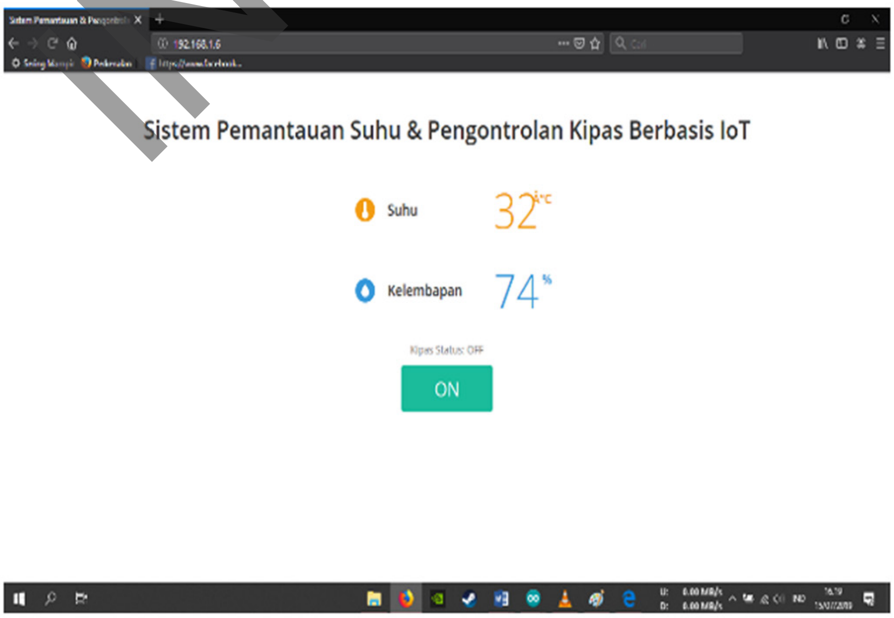
On the website page accessed, the results of the DHT22 sensor readings in the form of temperature and humidity can be seen. It also shows the display of buttons for controlling the fan in the form of buttons that can be used to activate or deactivate the fan through the website page. In Figure 13, the fan button is activated to the desired humidity value. Then, the fan is turned off again.

Testing all designs on the tool is carried out in an active state. Meanwhile, testing the tool uses a miniature that represents temperature and humidity monitoring and IoT-based fan control first to detect the state of DHT22. It can be observed when the DHT22 gets input from the surrounding environment.



(a) (b)

Figure 9 Fan Voltage Measurement, (a) On (b) Off



Sistem Pemantauan Suhu & Pengontrolan Kipas Berbasis IoT

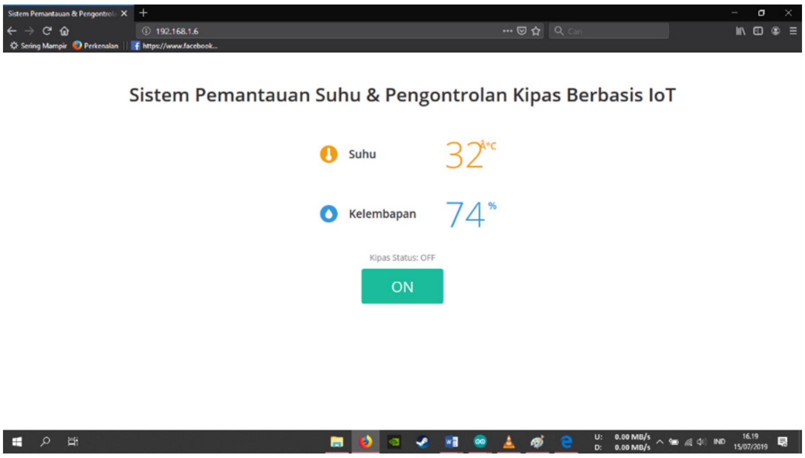
Suhu 32°C

Kelembapan 74%

Kipas Status: OFF

ON

Figure 10 Web Page Display Test



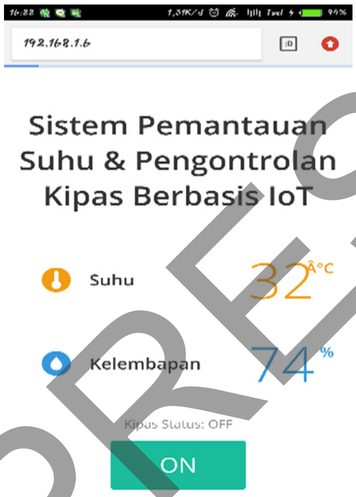
Sistem Pemantauan Suhu & Pengontrolan Kipas Berbasis IoT

Suhu 32Â°C

Kelembapan 74%

Kipas Status: OFF

Figure 11 Website Display on Computer



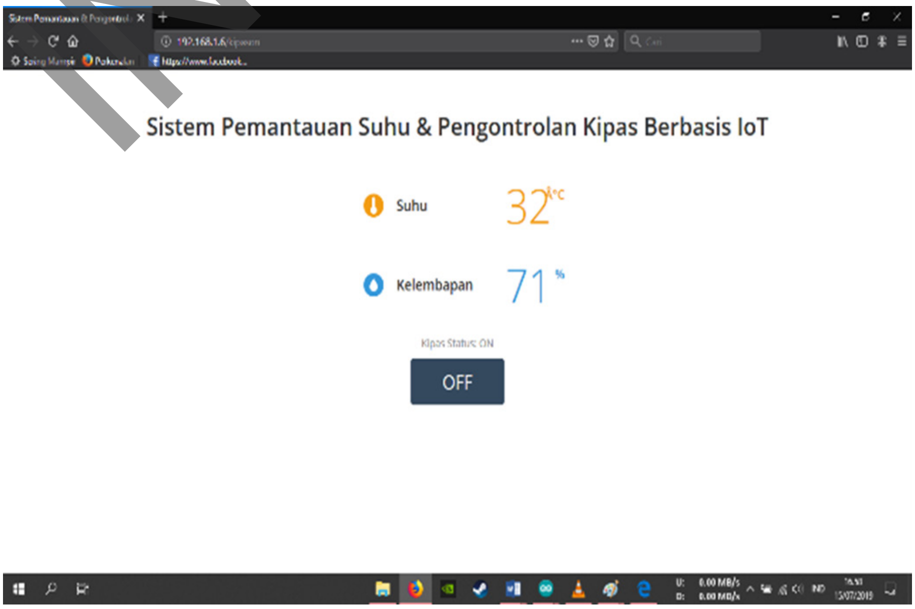
Sistem Pemantauan Suhu & Pengontrolan Kipas Berbasis IoT

Suhu 32Â°C

Kelembapan 74%

Kipas Status: OFF

Figure 12 Web Page Display on Mobile Phone



Sistem Pemantauan Suhu & Pengontrolan Kipas Berbasis IoT

Suhu 32Â°C

Kelembapan 71%

Kipas Status: ON

Figure 13 Disabling the Fan Through the Website

After the data are obtained, it affects the sensors in the input circuit so that the input circuit provides a voltage to the output pin in the form of temperature and humidity. Then, in testing the fan control with the L298N motor driver, it is active by pressing the button on the website page so that it can turn on the fan which makes the button on the website change, and the fan status becomes ON. Otherwise, if the button on the website is pressed again, the fan will turn off, which makes the button return to normal, and the fan status is OFF.

After implementing the tool, which consists of technical testing to determine the amount of voltage on the DHT22 sensor, the fan with the L298N motor driver, and functional testing to determine whether the initialization of each circuit functions and runs as expected or not, the result from the implementation is shown in Table 2. The description in Table 2 is Wi-Fi (Connected) means Wi-Fi has been connected to the AP and can access website pages. Meanwhile, Wi-Fi (Not Connected) means Wi-Fi is not connected to the AP (AP search not found) and cannot access website pages. Then, sensor DHT22 (Active) means that the sensor has been connected to the microcontroller to take temperature and humidity readings. The DHT22 (Off) means the sensor is inactive, so the sensor is not connected to the microcontroller and cannot take temperature and humidity readings. Next, the L298N motor driver is active if the fan can be controlled via the website page when the motor driver is active. In contrast, the L298N motor driver turns off if the fan cannot be controlled via the website page when the motor driver is not active.

A device generally consists of three parts: input, processor, and output. The input on this tool is a DHT22 sensor which provides input signals in the form of temperature and humidity readings to the data processor. Then, the data processor is an important part of the circuit because it processes data from the input received to be processed into output as needed. The data processor uses the NodeMCU ESP8266 V3. Next, the processed data are sent to a website page that is useful for displaying the output of the DHT22 sensor. The fan used to cool the temperature in the room can also be controlled via the website page by pressing the available button to turn it on or off.

The DHT22 sensor through readings of the surrounding environment is detected when there is a received input. The data obtained are sent to the microcontroller, and the output is displayed on the website page on computer or mobile phone. The fan to cool the temperature in the room can be controlled via the website page by pressing the available button to turn it on or off.

IV. CONCLUSIONS

The results of the tests carried out on this IoT-based system design can work in two outputs, namely monitoring the temperature and humidity displayed on the website page through reading the DHT22 sensor in the surrounding environment and controlling the fan that can be turned on or off by pressing the button on the websites. The DHT22 sensor can read temperature and humidity every two seconds so that the resulting data are stable to display. The L298N motor driver

Table 2 Overall System Test

No	**Input from DHT22**	**Wi-Fi**	**Website**			**Motor Driver**	**Fan**
			Temperature Celcius	**Humidity %**	**Fan**		
1	active	connected	21,5	82	on/off	active	on
2	active	connected	28,8	68	on/off	active	off
3	not active	connected	--	--	on/off	active	off
4	not active	connected	--	--	on/off	active	off
5	active	connected	33,9	62	off	not active	off
6	active	connected	20,7	76	off	not active	off
7	not active	connected	--	--	off	not active	off
8	not active	connected	--	--	off	not active	off
9	active	connected	27,3	72	on/off	active	on
10	active	connected	32,6	66	on/off	active	on
11	not active	not connected	--	--	off	not active	off
12	not active	not connected	--	--	off	not active	off
13	active	not connected	--	--	on/off	active	off
14	active	not connected	--	--	on/off	active	off
15	not active	not connected	--	--	off	not active	off

can adjust the speed of the fan with Pulse Width Modulation (PWM) using analog data ranging from 1 to 1.024. Moreover, the L298N motor driver requires an additional 9 V power supply to drive the 5 V fan because the motor driver filters the voltage on the fan. Fan control on the model is done manually. If it is applied to a certain room where the temperature and humidity values of the room are fixed, the fan control can be made automatically. Next, the website page to view the output of the monitoring and controlling system can be accessed using the IP address of the server that is connected to the client (user).

The suggestions for improvement from the research that has been done are in the AP. It is recommended to use a password so that no other users can enter monitoring or controlling the sensors in it. This IoT-based prototype system can be developed into various electronic devices such as fans, lights, tv, and others. Moreover, HTML functions are recommended to use Apache so that the web server can be accessed online. In future research, it is better to use applications than HTML. It is recommended to use the Blynk application so that it can be reached globally to access this system. The impact of the research is to provide another solution for the use of processors and applications used in automated temperature and humidity monitoring systems.

REFERENCES

Atique, S., Noureen, S., Roy, V., Subburaj, V., Bayne, S., & Macfie, J. (2019). Forecasting of total daily solar energy generation using ARIMA: A case study. In *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)* (pp. 114–119). IEEE. https://doi.org/10.1109/CCWC.2019.8666481

Ab Rahman, R., Hashim, U. R. A., & Ahmad, S. (2020). IoT based temperature and humidity monitoring framework. *Bulletin of Electrical Engineering and Informatics*, *9*(1), 229–237.

Adani, F., & Salsabil, S. (2019). Internet of things: Sejarah teknologi dan penerapannya. *Jurnal Online Sekolah Tinggi Teknologi Mandala*, *14*(2), 92–99.

Adhiwibowo, W., Daru, A. F., & Hirzan, A. M. (2020). Temperature and humidity monitoring using DHT22 sensor and Cayenne API. *Jurnal Transformatika*, *17*(2), 209–214.

Al Dahoud, A., & Fezari, M. (2018). *NodeMCU V3 for fast IoT application development*. Retrieved from https://galopago.github.io/assets/pdf/NodeMCUV3.pdf

Attubel, M., Siswanto, D., & Mukhsim, M. (2019). Sistem monitoring perawatan kendaraan berbasis Internet of Things (IOT). In *Conference on Innovation and Application of Science and Technology (CIASTECH)* (pp. 331–338).

Budiyanta, N. E., Tanudjaja, H., & Mulyadi, M. (2019). Rancang bangun robot line follower portable sebagai upaya minimalisasi sampah elektronik di ranah robotika. *TESLA: Jurnal Teknik Elektro*, *20*(2), 148–156.

Jacksi, K., & Abass, S. M. (2019). Development history of the world wide web. *International Journal of Scientific & Technology Research*, *8*(9), 75–79.

Mihai, B. (2016). How to use the DHT22 sensor for measuring temperature and humidity with the arduino board. *Acta Uiversitatis Cibiniensis–Technical Series*, *68*, 22–25.

Paul, A., & Jeyaraj, R. (2019). Internet of things: A primer. *Human Behavior and Emerging Technologies*, *1*(1), 37–47.

Puspasari, F., Satya, T. P., Oktiawati, U. Y., Fahrurrozi, I., & Prisyanti, H. (2020). Analisis akurasi sistem sensor DHT22 berbasis arduino terhadap thermohygrometer standar. *JFA (Jurnal Fisika Dan Aplikasinya)*, *16*(1), 40–45.

Putra, R. F. H., Lhaksmana, K. M., & Adytia, D. (2018). Aplikasi IoT untuk rumah pintar dengan fitur prediksi cuaca. *eProceedings of Engineering*, *5*(1), 1746–1760.

Putri, A. R., & Rahayu, P. N. (2020). Food warmer system based on DHT-22. *JAREE (Journal on Advanced Research in Electrical Engineering)*, *4*(1), 56–63.

Selvarasi, R., Aswini, & Pandian, A. A. (2019). Smart city based on street light and garbage monitoring by using IOT. *JASC: Journal of Applied Science and Computations*, *VI*(III), 55–64.

Shahid, N., & Aneja, S. (2017). Internet of things: Vision, application areas and research challenges. In *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, (pp. 583–587). IEEE

Utomo, M. A. P., Aziz, A., & Harjito, B. (2019). Server room temperature & humidity monitoring based on Internet of Thing (IoT). *Journal of Physics: Conference Series*, *1306*(1), 1–8.

Vinola, F., Rakhman, A., & Sarjana, S. (2020). Sistem monitoring dan controlling suhu ruangan berbasis Internet of things. *Jurnal Teknik Elektro dan Komputer*, *9*(2), 117–126.