

# Room Temperature and Humidity Control System Using Arduino and Blynk

Beni Purnomo<sup>1</sup>, Sunardi<sup>2,\*</sup>

<sup>1,2</sup>Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

Email: <sup>2</sup>[sunardi@ee.uad.ac.id](mailto:sunardi@ee.uad.ac.id)

\*Corresponding Author

**Abstract**—The temperature and humidity of a room can be a problem that affects the comfort and health of its occupants. Issues such as excessively high or low temperatures and humidity levels can make it difficult for people to sleep and cause discomfort. Various solutions can be implemented, such as using IoT-based temperature and humidity monitoring systems, platform Blynk to optimize temperature and humidity control, and designing control circuits using actuators such as fans with relays. This control system uses the Blynk application to monitor and turn on and off the actuators in this study using a fan. Arduino Wemos D1 as the main processor, and for sensors, DHT22. Test method by placing the system in a room with an area of 3 x 3 m. The results of the test with the fastest average time were in the morning with a time of 10 minutes, 52 seconds. The longest average time was obtained during the day, with a test time of 14 minutes, 38 seconds. The difference in error between the system and the comparison is the highest at a temperature of 2.71% and at a humidity of 33.5%.

**Keywords**—IoT, Control, Temperature, Humidity

## I. INTRODUCTION

The temperature and humidity of a room can be a problem that affects the comfort and health of its occupants [1]. Issues such as excessively high or low temperatures and humidity levels can make it difficult for people to sleep and cause discomfort [2]. High humidity levels can also cause a feeling of dampness, while low humidity levels can lead to dry skin and irritation of the eyes and nose [3].

To address these problems, various solutions can be implemented, such as using IoT-based temperature and humidity monitoring systems, platform Blynk to optimize temperature and humidity control, and designing control circuits using actuators such as fans with relays [4]-[5]. Proper ventilation, using curtains or window films to reduce direct sunlight exposure, and monitoring the use of electronic devices that can affect room temperature, such as air conditioners and lights, can also help maintain good temperature and humidity levels in a room [6].

A study implemented and optimized a fuzzy logic-based microcontroller system to control the temperature and humidity of a server room. The optimized system was able to control the temperature and humidity within the required range and send data from the system to a website that could be accessed through a web browser [7].

Another study designed and fabricated a temperature and humidity stabilizer on a low voltage distribution panel with a PLC-based fuzzy method to prevent excessive temperature and humidity on the panel. The study found that fan control

using the fuzzy method could reduce the temperature faster than fan control with simple logic [8].

A study designed and tested a temperature and humidity control system based on the Internet of Things (IoT) for oyster mushroom cultivation. The study used an experimental method to control the microclimate inside the mushroom cultivation house. The system was able to regulate the temperature and humidity within the required range, resulting in higher and more even mushroom growth compared to the control group [9].

From some of these studies, temperature and humidity controls will be made using Arduino Wemos D1 microcontroller with temperature and humidity readings using a DHT22 sensor. To display sensor readings using a 16X2 LCD screen with I2C and displayed on the Blynk platform. The fan as an actuator in the system will control the temperature and humidity of the room. To be able to connect between the system and the Blynk platform, an internet or Wi-Fi network is needed. Connecting the system to wifi needs to be set when programming the Arduino IDE software by entering the Wi-Fi name and password, while connecting the system to Blynk also needs to enter the authentication code obtained on the Blynk system.

## II. METHOD

The system is running and it will connect to the Blynk application. Then take temperature and humidity readings on. After the sensor reads the temperature and humidity, if the sensor reading is whether the temperature read is more than 30°C or not for the afternoon and evening and 27°C for the morning and evening. If not, then the sensor repeats to read the temperature and humidity.

If it is more than 30°C for the afternoon and evening and 27°C for the morning and evening then the relay in the circuit will turn on the fan to lower the temperature. Then if the DHT22 sensor has read temperatures  $\leq 30^\circ\text{C}$  not for noon and evening and  $\leq 27^\circ\text{C}$  for morning and evening then the relay in the circuit will turn off the fan. The Blynk application will display the sensor reading results. Apart from that, Blynk can also be used to control the fan so that it can be turned on or off manually. Flowchart can be seen in Fig. 1.

The block diagram as input to the system, namely the power supply and sensors to measure temperature and humidity. The measurement results are sent to the microcontroller as main processing. The processing results are displayed on the 16X2 LCD, sent to Blynk via the internet and will affect the work of the relay and the relay will be ON if the processing temperature is known to be  $>30^\circ\text{C}$  for the

day and evening and  $>27^{\circ}\text{C}$  for the morning and evening. Then, if the measured temperature is  $\leq 30^{\circ}\text{C}$  for the afternoon and evening and  $\leq 27^{\circ}\text{C}$  for the morning and evening, the relay in the system will turn off the fan.. The block diagram can be seen in Fig. 2.

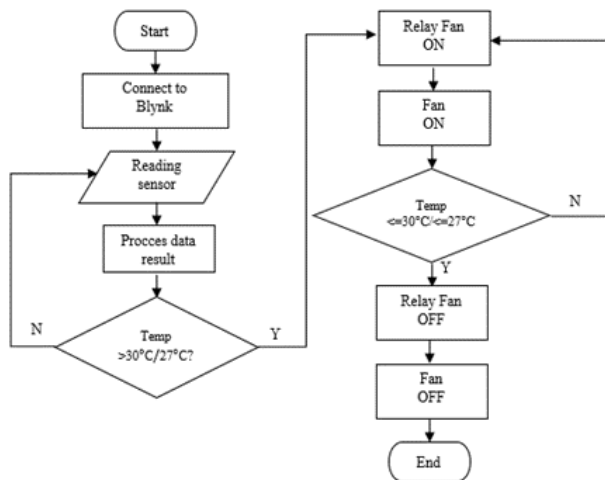


Fig. 1. Flowchart of system

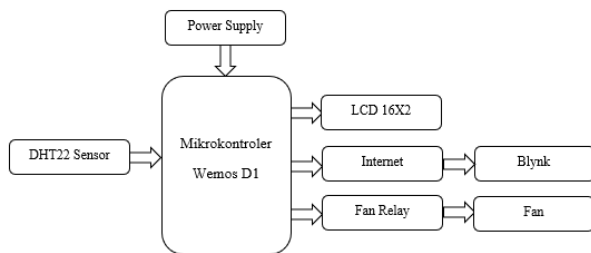


Fig. 2. Block diagram

The wiring system for the tool is made using 1 DHT sensor which is connected to pin D6 on Wemos D1 and the voltage is 5V. LCD I2C with VCC pin connected to 5V, GND pin connected to ground and SDA pin, and SCL connected to the same pin on the Wemos D1 microcontroller. The relay is connected to pin D5 on Wemos. The relay in this circuit is also connected to the fan as the actuator. Wemos D1 will be connected to the Blynk application using internet access using wifi. Using the Blynk app, fans can be used automatically or manually to control temperature and humidity. The system wiring can be seen in Fig. 3.

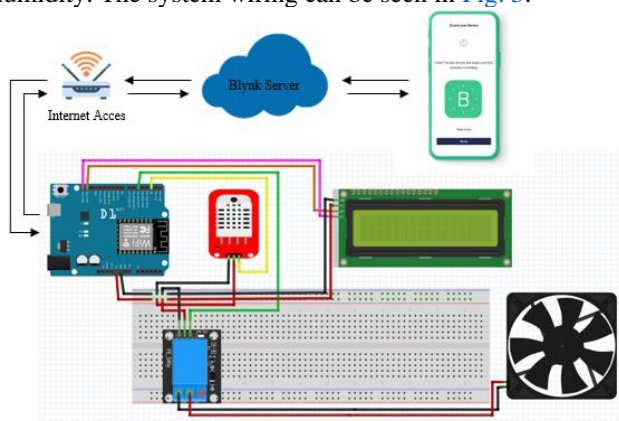


Fig. 3. Wiring system

To calculate the difference in error between the system and the comparator, the following formula will be used [10].

$$\text{Error \%} = \frac{|\text{System meas.} - \text{Comparison Meas.}|}{\text{System Measurement}} \times 100\% \quad (1)$$

From Equation (1), the absolute value of the difference between the measurements made by the system and the measurements made by the comparator will be sought.

#### A. Internet Of Things

Internet of Things, is a technology that enables the connection and communication of various devices and systems through the internet [11]. It has found applications in numerous fields, including temperature and humidity monitoring in rooms [12]. IoT-based solutions have been developed to address the problem of temperature and humidity in a room, offering real-time monitoring, control, and automation capabilities [13].

The Blynk platform is one such IoT platform that can be used to create a mobile app that connects to IoT devices and displays data from sensors, including temperature and humidity sensors. By monitoring the data from these sensors, users can take action to maintain optimal temperature and humidity levels in a room [14].

The microcontroller used is Wemos D1 R1. The Wemos D1 R1 board is an Arduino based controller with an ESP8266 and a layout similar to Arduino [15]-[16]. One of the advantages of the Arduino Wemos D1 R1 compared to other ESP8266-based development board modules is that it can be connected to an Arduino Uno shield. Where this shield is a plug and play module from the one that supports the pins on the Arduino Uno [17].

#### B. LCD 16X2 I2C

Liquid Crystal Display 16X2 I2C is a type of Liquid Crystal Display module that is used to display characters. The display consists of two sheets of material that can polarize liquid crystals between them [18]. In designing a system, LCDs are commonly used to display sensor values, text, or menus in microcontroller applications. The I2C module is used to connect the LCD to the microcontroller without using many pins. The I2C module is included when purchasing an LCD [19]. Several studies have utilized LCD 16x2 I2C in their research, such as designing a room monitoring tool based on ESP-12E, a device for controlling and monitoring the temperature and humidity of oyster mushroom cultivation, and a monitoring system for turbidity and acid-base levels of tofu industrial wastewater [20].

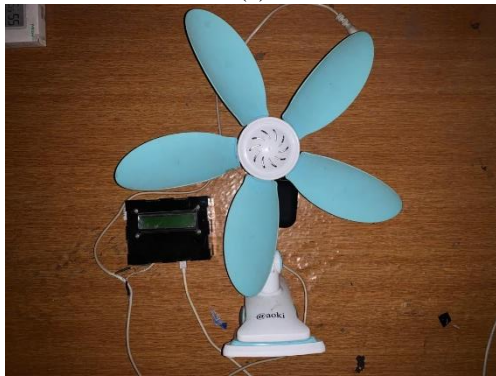
### III. RESULTS AND DISCUSSION

Retrieval of this data by using the method directly with a room with a size of 3x3 meters. The diameter of the fan used in data collection is 30cm in diameter. The fan specifications used are 15 Watts of power. The distance between the fan actuator and the circuit system is 15 cm. Position of the sensor is on the side of the system and facing the fan. Fig. 4 (a) is the test room, Fig. 4 (b) is the system and actuator used.

Data collection is also carried out using a comparator that can measure the temperature and humidity of the room. The comparator used is the MIIIW Comfort Temperature and Humidity Clock. The comparison tool used can be seen in Fig. 5.



(a)



(b)

Fig. 4. (a) is the test room, (b) system and actuator



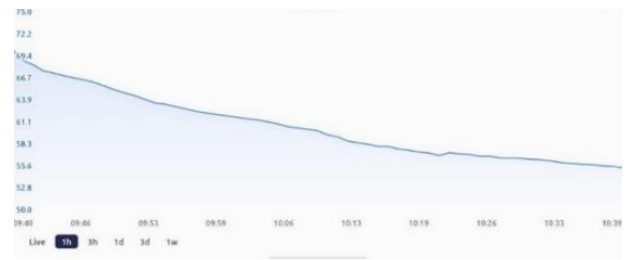
Fig. 5. Comparison tool in data collection

**A. DHT22 Sensor Testing**

The DHT22 sensor is tested by turning on the sensor for 1 hour continuously so that the stability of the program that has been made is known. The results of the test can be seen in Fig. 6 for temperature and humidity. The DHT22 sensor is tested by turning on the sensor for 1 hour continuously so that the stability of the program that has been made is known. The results of the test can be seen in Fig. 6 for temperature and humidity.



(a)



(b)

Fig. 6. DHT22 Test, (a) temperature (b) humidity

The results of this sensor reading are also displayed on the LCD 16X2 I2C. The LCD can display the calculated temperature and humidity quantities. The LCD display can be seen in Fig. 7.



Fig. 7. LCD display, suhu (Temperature), Klmhpn (Humidity)

**B. Relay Testing**

The relay can be set in automatic mode or manual mode. The mode settings for this relay are connected to Blynk, so to set the mode you have to use the Blynk application that has been created. On Blynk there is also an LED indicator if the fan is on. The relay settings on Blynk can be seen in Fig. 8.



Fig. 8. Relay test

**C. Blynk Testing**

Tests using the Blynk software are carried out by ensuring that the temperature and humidity are displayed on the Blynk application. Temperature and humidity are displayed in graphic form and digital display. The graph can read traffic up to 3 months ago. The dashboard display on Blynk can be seen in Fig. 9.

**D. Morning Test**

The first experiment was carried out in the morning between 07.00 - 09.00. The results of the data collection can be seen in Table 1.

It can be seen in Table 1 which can be seen in the length column of testing time, the longest test was on the morning

of the 8th day with a long time to reach the reference temperature of 17 minutes 59 seconds, while the fastest time to reach the limit temperature was on the 2nd day with a 6 minutes 18 seconds.



Fig. 9. Blynk test

The initial temperature has the biggest error on day 1 with an error between the system and the comparator of 2.07%, while the final temperature has the highest error difference on day 2 with a value of 1.82%. At the initial humidity, the biggest error difference is on the 7th day which is equal to 6.14%, at the final humidity the biggest error difference is on the 3rd day with a value of 13.55%.

#### E. Afternoon Test

The second experiment was carried out during the day at 12.00–13.00. The results of data collection carried out can be seen in Table 2.

It can be seen in Table 2 that there is a column for the length of time of testing, the longest test was on the morning of the 6th day with the length of time to reach the reference temperature, which was 25 minutes 51 seconds. The fastest time to reach the limit temperature was on day 8 with a time of 9 minutes 25 seconds.

The initial temperature has the biggest error on the 7th day with an error between the system and the comparator of 2.17%, the final temperature with the highest error difference is on the 7th day with a value of 1.69%. The initial humidity with the largest error difference is on the 6th day, which is 18.82%, and the final humidity with the largest error difference is on the 5th day with a value of 24.67%.

#### F. Evening Test

The third experiment was carried out in the afternoon at 16.00–17.00. The results of the data collection carried out can be seen in Table 3.

It can be seen in Table 3 that there is a column for the length of time of testing, the longest test was on the morning of the 1st day with the length of time to reach the reference temperature of 29 minutes 29 seconds. The fastest time to reach the limit temperature was on day 4 with a time of 1 minute 50 seconds.

The initial temperature has the biggest error on day 7 with an error between the system and the comparator of 0.98%, and the final temperature with the highest error difference on day 3 with a value of 1.32%. The initial humidity with the largest error difference is on the 2nd day, which is 12.12%, and the final humidity with the largest error difference is on the 2nd day with a value of 24.42%.

#### G. Night Test

The fourth experiment was carried out at night at 20.00–21.00. The results of the data collection carried out can be seen in Table 4.

It can be seen in Table 4 that there is a column for the length of time of testing, the longest testing is on the morning of the 3rd day with the length of time to reach the reference temperature, which is 19 minutes 43 seconds. The fastest time to reach the limit temperature was on day 8 with a time of 4 minutes 28 seconds.

The initial temperature has the biggest error on day 3 with an error between the system and the comparator of 2.71%, the final temperature with the highest error difference is on day 8 with a value of 1.82%. The initial humidity with the largest error difference is on the 2nd day, which is 10%, and the final humidity with the largest error difference is on the 4th day with a value of 33.5%.

#### H. Average System Testing Time

From several times of data retrieval, the average time for the system to reach a predetermined limit temperature can be taken. The results can be seen in Table 5.

In Table 5 it can be seen the average time for each test that has been carried out. Testing with the fastest average time is in the morning with a time of 10 minutes 52 seconds. Furthermore, the second fastest average time is at night with an average testing time of 11 minutes 26 seconds. Then followed in the afternoon with an average test time of 12 minutes 37 seconds. The longest average time was obtained during the day with a test time of 14 minutes 38 seconds. So, it can be concluded that the fastest time to reach the reference limit temperature is in the morning and the longest is in the evening.

In the comparison between the system and the comparison, the highest error difference in temperature occurred at the initial temperature on the third day at night, amounting to 2.71%. Meanwhile, for humidity, the highest error difference occurred at the initial humidity of the fourth day with an error of 33.5%.

Table 1. Results of data collection in the morning

Day	System				Comparison				The error difference between the system and the comparator (%)				Length of Testing Time
	Temp. (°C)		Hum. (%)		Temp. (°C)		Hum. (%)		Temp.		Hum.		
	Early	End	Early	End	Early	End	Early	End	Early	End	Early	End	
1	28.4	27	75.1	79.5	29	27.3	72	74	2.07	1.10	4.31	7.43	10 min 5 sec
2	27.9	27	76.4	80	27.9	27.5	79	80	0.00	1.82	3.29	0.00	6 min 18 sec
3	27.8	26.9	77.8	80.4	27.9	26.6	81	93	0.36	1.13	3.95	13.55	6 min 32 sec
4	28.8	27	75.5	81.6	28.5	27.3	80	72	1.05	1.10	5.63	13.33	11 min 56 sec
5	28.5	27	76.3	76.7	28.2	27.3	80	86	1.06	1.10	4.63	10.81	14 min 8 sec
6	28.1	26.9	76	75	28	27	80	74	0.36	0.37	5.00	1.35	10 min 25 sec
7	27.7	26.9	77.9	84	27.3	27	83	84	1.47	0.37	6.14	0.00	9 min 34 sec
8	29.6	26.9	71.8	67.5	29.5	26.9	75	62	0.34	0.00	4.27	8.87	17 min 59 sec

Table 2. Results of data collection during the afternoon

Day	System				Comparison				The error difference between the system and the comparator (%)				Length of Testing Time
	Temp. (°C)		Hum. (%)		Temp. (°C)		Hum. (%)		Temp.		Hum.		
	Early	End	Early	End	Early	End	Early	End	Early	End	Early	End	
1	31.8	30	65.2	67.2	31.8	30.1	66	67	0.00	0.33	1.21	0.30	12 min 33 sec
2	31.9	29.9	63	64.4	31.8	29.9	65	69	0.31	0.00	3.08	6.67	15 min 58 sec
3	31	29.8	68.3	66.5	31	29.7	70	69	0.00	0.34	2.43	3.62	9 min 58 sec
4	31.3	30	65.5	71.6	31.3	30.3	69	70	0.00	0.99	5.07	2.29	10 min 19 sec
5	32.7	30	60.4	74.8	32.6	30.2	61	60	0.31	0.66	0.98	24.67	17 min 14 sec
6	33.2	30	55.2	65.5	33.2	29.9	68	62	0.00	0.33	18.82	5.65	25 min 51 sec
7	31.5	30	66.3	81.7	32.2	29.5	66	77	2.17	1.69	0.45	6.10	15 min 44 sec
8	31	30	68.5	69.5	30.9	30	70	66	0.32	0.00	2.14	5.30	9 min 25 sec

Table 3. Results of data collection in the evening

Day	System				Comparison				The error difference between the system and the comparator (%)				Length of Testing Time
	Temp. (°C)		Hum. (%)		Temp. (°C)		Hum. (%)		Temp.		Hum.		
	Early	End	Early	End	Early	End	Early	End	Early	End	Early	End	
D 1	33.5	29.7	52	56.9	33.6	29.8	54	54	0.30	0.34	3.70	5.37	29 min 29 sec
D 2	32.1	30	58	64.7	31.8	30.3	66	52	0.94	0.99	12.12	24.42	17 min 26 sec
D 3	31.1	29.9	66.3	69.3	31.2	30.3	69	70	0.32	1.32	3.91	1.00	10 min 20 sec
D 4	30.3	30	69.4	69.8	30.3	30.2	74	71	0.00	0.66	6.22	1.69	1 min 50 sec
D 5	31.9	29.9	60.8	67.7	31.6	29.7	64	67	0.95	0.67	5.00	1.04	12 min 13 sec
D 6	31.3	30	64.3	66.7	31	30.3	66	69	0.97	0.99	2.58	3.33	8 min 27 sec
D 7	30.3	29.9	71.9	72.2	30.6	30	77	72	0.98	0.33	6.62	0.28	5 min 49 sec
D 8	32.6	30	57	63.2	32.8	30.2	58	59	0.61	0.66	1.72	7.12	15 min 23 sec

Table 4. Results of data collection at night

Day	System				Comparison				The error difference between the system and the comparator (%)				Length of Testing Time
	Temp. (°C)		Hum. (%)		Temp. (°C)		Hum. (%)		Temp.		Hum.		
	Early	End	Early	End	Early	End	Early	End	Early	End	Early	End	
1	28.7	27	74.1	82.5	28.1	27.4	78	79	2.14	1.46	5.00	4.43	10 min 45 sec
2	28.6	26.9	74.7	76.3	28	27.1	83	67	2.14	0.74	10.00	13.88	10 min 1 sec
3	30.3	26.9	70	81.7	29.5	27.1	74	84	2.71	0.74	5.41	2.74	19 min 43 sec
4	29.9	26.9	71.7	80.1	29.6	26.7	77	60	1.01	0.75	6.88	33.50	13 min 50 sec
5	27.6	26.9	75.3	77.5	27.3	27.1	77	78	1.10	0.74	2.21	0.64	5 min 2 sec
6	27.9	26.9	78	78.2	27.7	26.8	82	76	0.72	0.37	4.88	2.89	9 min 16 sec
7	28.6	27	69.6	67.4	29.1	26.7	71	67	1.72	1.12	1.97	0.60	18 min 27 sec
8	27.6	26.9	75.6	82.4	27.6	27.4	78	78	0.00	1.82	3.08	5.64	4 min 28 sec

Table 5. Length of testing time

Day	Length of testing time			
	Morning	Afternoon	Evening	Night
1	10 minutes 5 seconds	12 minutes 33 seconds	29 minutes 29 seconds	10 minutes 45 seconds
2	6 minutes 18 seconds	15 minutes 58 seconds	17 minutes 26 seconds	10 minutes 1 seconds
3	6 minutes 32 seconds	9 minutes 58 seconds	10 minutes 20 seconds	19 minutes 43 seconds
4	11 minutes 56 seconds	10 minutes 19 seconds	1 minutes 50 seconds	13 minutes 50 seconds
5	14 minutes 8 seconds	17 minutes 14 seconds	12 minutes 13 seconds	5 minutes 2 seconds
6	10 minutes 25 seconds	25 minutes 51 seconds	8 minutes 27 seconds	9 minutes 16 seconds
7	9 minutes 34 seconds	15 minutes 44 seconds	5 minutes 49 seconds	18 minutes 27 seconds
8	17 minutes 59 seconds	9 minutes 25 seconds	15 minutes 23 seconds	4 minutes 28 seconds
Average	10 minutes 52 seconds	14 minutes 38 seconds	12 minutes 37 seconds	11 minutes 26 seconds

#### IV. CONCLUSION

Has succeeded in designing and implementing a room temperature and humidity control system using an IoT-based DHT22 sensor. The average time for each test that has been carried out. Testing with the fastest average time is in the morning with a time of 10 minutes 52 seconds. Furthermore, the second fastest average time is at night with an average testing time of 11 minutes 26 seconds. Then followed in the afternoon with an average testing time of 12 minutes 37 seconds. The longest average time was obtained during the day with a test time of 14 minutes 38 seconds. The difference in error between the system and the comparison is the highest at a temperature of 2.71% and at a humidity of 33.5%.

#### REFERENCES

- [1] P. Valdiserri, S. Cesari, M. Coccagna, P. Romio, and S. Mazzacane, "Experimental data and simulations of performance and thermal comfort in a patient room equipped with radiant ceiling panels," *Buildings*, vol. 10, no. 12, pp. 1–18, 2020, <https://doi.org/10.3390/buildings10120235>.
- [2] F. Fontes, R. Antão, A. Mota, and P. Pedreiras, "Adaptive Ambient Temperature Control of Indoor Environments," *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, pp. 255–260, 2019, <https://doi.org/10.1109/IECON.2019.8927401>.
- [3] H. Xianzhe, "Room temperature and humidity monitoring and energy-saving system," *2011 6th International Conference on Computer Science & Education (ICCSE)*, pp. 537–540, 2011, <https://doi.org/10.1109/ICCSE.2011.6028696>.
- [4] J. Kumar and R. Kapoor, "Comparative Analysis of Room Temperature Controller Using Fuzzy Logic & PID," *IFAC Proceedings Volumes*, vol. 3, no. 7, pp. 853–858, 2013, [https://doi.org/10.1016/S1474-6670\(17\)36900-8](https://doi.org/10.1016/S1474-6670(17)36900-8).
- [5] N. Nikolov, O. Nakov, and D. Gotseva, "Design and Research of Smart IoT Control System for Electrical Appliances," *2021 29th National Conference with International Participation (TELECOM)*, pp. 39–42, 2021, <https://doi.org/10.1109/TELECOM53156.2021.9659650>.
- [6] A. Correia, L. Ferreira, P. Coimbra, and A. de Almeida, "Impacts of automated natural ventilation in the temperature and humidity of a distribution transformer room," *2020 IEEE 14th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG)*, pp. 36–43, 2020, <https://doi.org/10.1109/CPE-POWERENG48600.2020.9161687>.
- [7] F. H. Purwanto, E. Utami, and E. Pramono, "Implementation and Optimization of Server Room Temperature and Humidity Control System using Fuzzy Logic Based on Microcontroller," *J. Phys. Conf. Ser.*, vol. 1140, no. 1, 2018, <https://doi.org/10.1088/1742-6596/1140/1/012050>.
- [8] A. T. Nugraha, R. F. As'ad, Adianto, and V. H. Abdullayev, "Design And Fabrication of Temperature and Humidity Stabilizer on Low Voltage Distribution Panel with PLC-Based Fuzzy Method to Prevent Excessive Temperature and Humidity on The Panel," *J. Electron. Electromed. Eng. Med. Informatics*, vol. 4, no. 3, pp. 170–177, 2022, <https://doi.org/10.35882/jeeemi.v4i3.241>.
- [9] D. A. Setiawati, S. G. Utomo, Murad, and G. M. D. Putra, "Design of temperature and humidity control system on oyster mushroom plant house based on Internet of Things (IoT)," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 712, no. 1, 2021, <https://doi.org/10.1088/1755-1315/712/1/012002>.
- [10] R. Mahkeswaran and A. K. Ng, "Smart and Sustainable Home Aquaponics System with Feature-Rich Internet of Things Mobile Application," *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*, pp. 607–611, 2020, <https://doi.org/10.1109/ICCAR49639.2020.9108041>.
- [11] C. Arkapati and W. Embamrung, "Development of Temperature and Humidity Control System in Straw Mushroom Growing Farms with the Concept of Internet of Things (IoT)," *Proceedings of the 3rd International Conference of Computer, Environment, Agriculture,*

- Social Science, Health Science, Engineering and Technology - ICEST*, pp. 84–91, 2021, <https://doi.org/10.5220/0010038700840091>.
- [12] S. M. Mahidul Hasan, M. R. Ahsan, and M. D. A. Satter, “IoT-Cloud-Based Low-Cost Temperature, Humidity, and Dust Monitoring System to Prevent Food Poisoning,” *2021 3rd International Conference on Electrical & Electronic Engineering (ICEEE)*, pp. 21–24, 2021, <https://doi.org/10.1109/ICEEE54059.2021.9718789>.
- [13] M. Padli, I. D. Lestari, R. Ramlan, and I. Nasrullah, “Temperature and Humidity Control Device In The Server Room PT.BFI Finance Tbk. Karawaci Based on Internet Of Things (IoT),” *Journal Bit-Tech*, vol. 4, no. 3, pp. 123–130, 2022, <https://doi.org/10.32877/bt.v4i3.446>.
- [14] B. Guo, X. Wang, X. Zhang, J. Yang, and Z. Wang, “Research on the temperature & humidity monitoring system in the key areas of the hospital based on the internet of things,” *Int. J. Smart Home*, vol. 10, no. 7, pp. 205–216, 2016, <https://doi.org/10.14257/ijsh.2016.10.7.21>.
- [15] W. Budiharto, V. Andreas, E. Irwansyah, J. S. Suroso, and A. A. S. Gunawan, “Implementation of wemos d1 for wi-fi based controller tank-based military robot,” *ICIC Express Lett. Part B Appl.*, vol. 12, no. 4, pp. 377–382, 2021, [https://web.archive.org/web/20220726053351id\\_/http://www.icicelb.org/ellb/contents/2021/4/elb-12-04-09.pdf](https://web.archive.org/web/20220726053351id_/http://www.icicelb.org/ellb/contents/2021/4/elb-12-04-09.pdf).
- [16] M. Kirola, A. Rawat, N. Sharma, and A. K. Sinha, “Smart Home Automation Using Google Assistant,” *SSRN Electron. J.*, 2019, <https://doi.org/10.2139/ssrn.3418738>.
- [17] S. Purba, M. Hariri, R. J. Banjarnahor, and S. Natalia, “LED Control System Using Arduino Wemos D1 R1 Based on Web Server Communication Via Internet of Things (IoT),” *FJST*, vol. 2, no. 6, pp. 1397–1408, 2023, <https://doi.org/10.55927/fjst.v2i6.4436>.
- [18] A. OO and O. TT, “Design and Implementation of Arduino Microcontroller Based Automatic Lighting Control with I2C LCD Display,” *J. Electr. Electron. Syst.*, vol. 7, no. 2, 2018, <https://doi.org/10.4172/2332-0796.1000258>.
- [19] I. G. M. N. Desnanjaya, A. A. G. B. Ariana, I. M. A. Nugraha, I. K. A. G. Wiguna, and I. M. U. Sumaharja, “Room Monitoring Uses ESP-12E Based DHT22 and BH1750 Sensors,” *J. Robot. Control*, vol. 3, no. 2, pp. 205–211, 2022, <https://doi.org/10.18196/jrc.v3i2.11023>.
- [20] I. G. M. N. Desnanjaya and P. Sugiartawan, “Controlling and Monitoring of Temperature and Humidity of Oyster Mushrooms in Tropical Climates,” *IJEIS (Indonesian J. Electron. Instrum. Syst.)*, vol. 12, no. 1, p. 69, 2022, <https://doi.org/10.22146/ijeis.73346>.