The Development of an IoT-based Indoor Air Monitoring System Towards Smart Energy Efficient Classroom

Moeljono Widjaja[1], Dareen Halim[2], Rahmi Andarini[3]
[1] Department of Informatics, Universitas Multimedia Nusantara, Tangerang, Indonesia
[2] Department of Computer Engineering, Universitas Multimedia Nusantara, Tangerang, Indonesia
[3] Department of Engineering Physics, Universitas Multimedia Nusantara, Tangerang, Indonesia
[1] moeljono.widjaja@umn.ac.id, [2] dareen.halim@umn.ac.id, [3] rahmi.andarini@lecturer.umn.ac.id

Accepted on 03 May 2022
Approved on 16 June 2022

***Abstract*— Indoor air quality has become a crucial issue, specifically during COVID 19 pandemic. The good indoor air quality will lead to occupants' comfort condition, thus affecting their productivity. Indoor air temperature and relative humidity are two essential components of thermal comfort. This paper presents the development of a temperature and relative humidity monitoring system for the classroom using the Internet of Things (IoT). This system consists of three main components: logger nodes, a gateway logger, and an interconnected cloud server. The logger node (ESP8266 / ESP32 microcontroller and DHT22 sensor) is a device at the edge of the IoT system and is placed at the monitoring location. The logger gateway is built on a Raspberry Pi 4, which serves as an intermediate server. It receives periodic data (temperature and humidity) from the logger nodes through the publish-subscribe MQTT protocol and sends it to the MongoDB Atlas cloud database. The logger gateway saves all received logs into the SQLite database as temporary local storage and then uploads the data to the MongoDB Atlas cloud at a predefined interval. The MongoDB data is then displayed on a monitoring dashboard using MongoDB charts. The logger node with the DHT22 sensor has been adjusted using a linear model and successfully tested to monitor indoor and outdoor air conditions with satisfactory results. The recorded data has also been successfully modeled using the Gaussian Mixture Model and a simple Fuzzy model. These models can capture the dynamic of air condition in the room and predict the state of the cooling system.**

***Index Terms*— Fuzzy Model; Gaussian Mixed Model; Internet of Things; MQTT Protocol; Sensors.**

I. Introduction

Air conditioning systems are increasingly widespread for homes, office buildings, shopping centers, factories, campuses and schools. As a result, the consumption of electrical energy for the cooling system also increases from year to year. Often the use of the cooling system is inefficient due to several factors, including an adequate monitoring system is not yet available, so efficiency actions cannot be carried out optimally [1].

Instead of energy consumption issues, thermal comfort is also a crucial factor that must be achieved for a room with an air conditioning system (conditioned room) [2]. The good indoor air quality of a room will lead to the occupants' productivity [3]. In order to achieve thermal comfort conditions, it is crucial to maintain the indoor air parameter such as temperature, relative humidity, and CO2 level of the room [4].

Currently, there are various ways to monitor the condition of the room, either manually or automatically. Air conditioning units are usually equipped with a conventional room temperature controller (thermostat), but the tools that are commonly used are usually unable to record the measured data for further analysis. In order to maintain a comfortable room temperature, the system must be able to monitor the dynamic of air condition as affected by many factors, including the weather outside (exposure to sunlight through glass windows or rain), room occupancy rate, time of day (morning, afternoon or evening) and the impact of heat generated by electronic equipment [5], [6].

An Internet of Things (IoT) is a device connected through the Internet network so that it can be monitored and controlled remotely as long as an Internet network is available. This IoT technology is relatively affordable and quite easy to implement so that its use has been included in the daily activities of various industries; its application includes and is not limited to smart city applications [7], [8], smart campuses [9]–[11], smart grid, smart home [12], security [13] and smart building management system [14], [15]. There has been no standard as a reference in the application of IoT in the field of building management, so there are still plenty of opportunities for exploration of various methods and related technologies [16]. This study aims to develop a temperature and humidity monitoring system for classrooms based on the Internet of Things which offers more comprehensive measurements and broader coverage areas. The first stage of this research was to design the IoT-based monitoring system architecture. The development of this system was then continued by testing two IoT loggers: one to monitor indoor air conditions and the other to monitor outdoor air conditions. Prior to field testing, the two IoT loggers were adjusted so that the measurement results of the two loggers were identical when measuring the same air condition. The adjusted measurement was modelled linearly using the least-squares method. The saved data was then analysed and used to build a simple model using GaussianMixture and Fuzzy approaches to capture the dynamic of air condition as it is influenced by the outdoor weather and the air cooling system.

II. METHODOLOGY

The development of this IoT-based temperature/humidity monitoring system includes the design of the IoT system architecture, the setup of the MongoDB database in the Atlas cloud, the development of a dashboard monitoring system using MongoDB Charts, and the process of calibrating the sensors.

A. *IoT Architecture System*

In general, this IoT-based temperature/humidity monitoring system consists of three main components: (1) logger nodes, (2) logger gateways, and (3) cloud servers that are connected to each other as described in Figure 1.

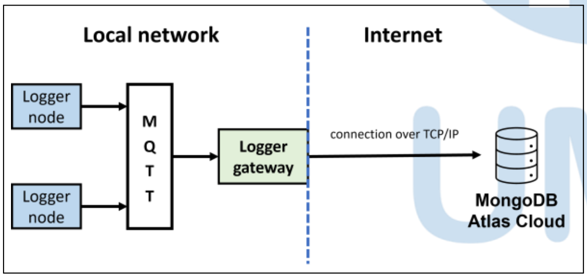

Fig. 1. IoT-based monitoring system architecture

The logger node is the far end device of the IoT system and is placed at the monitoring location. In this study, the logger device is composed of the ESP8266 / ESP32 microcontroller family is equipped with Wi-Fi connectivity and paired with a DHT-22 temperature and humidity sensor. The logger is connected via a local WLAN network (Wi-Fi technology) with the logger gateway. This local WLAN can be an ad-hoc network created by a logger gateway or a WLAN network originating from a public Wi-Fi router device.

The logger gateway is a Raspberry Pi 4, which functions as an intermediary server, which receives periodic temperature and humidity data from the logger node, and sends it to the MongoDB Atlas cloud database, which will be discussed in more detail later. Communication between the logger gateway and logger node is carried out through the publish-subscribe MQTT [17] protocol, where the logger node unit becomes the MQTT client and the logger gateway become the MQTT broker as well as the MQTT client. Communication between the logger gateway to the MongoDB Atlas is done via the MongoDB wire protocol over the TCP/IP standard socket connection. The intermediary server function in this research is designed in two versions, namely: NodeJS and Python script.

The query-driven communication flow between the logger gateway and the node can be seen in Figure 2. Two MQTT communication channels were provided, namely the 'control' channel and the 'log/data' channel. The logger gateway periodically (every 10 minutes in this study) sends a message to the 'control' channel, which will be received by all logger nodes. When receiving the control message, the logger node will respond by sending a message containing the loggerID and the current temperature/humidity to the 'log/data' channel.

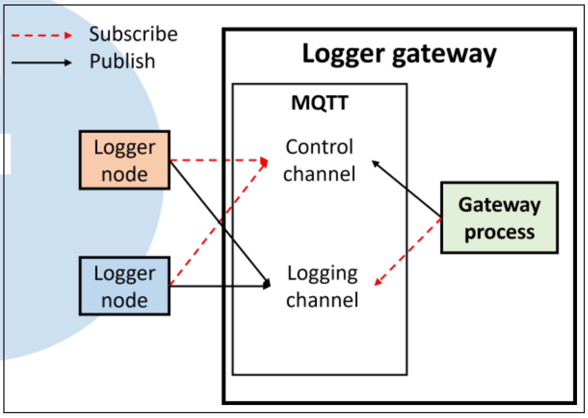

Fig. 2. The query-driven communication flow between logger gateway and node

The use of these two separate channels is carried out with the following considerations:

1) *Security*

When the system has been implemented with loggers on a large scale, the 'control' channel can be opened to the public so that all devices can join the channel to get information about when temperature and humidity logging should be carried out, or for development other more complex control schemes (including initiation when a new logger joins the network for the first time). Meanwhile, information regarding the 'log/data' channel can be provided only for 'trusted' loggers, so those other unverified devices cannot send fake logging info because they cannot access the 'log/data' channel. This scheme can be further developed so that the information dissemination process regarding the 'log/data' channel is integrated with the authentication process. For example, the logger can be registered with the system via a management dashboard. A secret 'log/data' channel and access token are provided by the dashboard upon successful registration. The logger then can access the given MQTT channel while providing access token [18]. This schematic is described in Figure 3.

2) *Efficiency*

Similar to the first point, a system that is implemented on a large scale will result in a large number of MQTT message transactions as well. If only one channel is used both for ’control’ and ’log/data’, control commands from the logger gateway will be mixed with data logging messages from the logger node. As a result, it will cause unnecessary processing on the logger node side [19]. In operation, the logger node must monitor the communication channel to get commands from the logger gateway, which in this study is the command to do logging. If the command is mixed with messages containing logging data from other nodes (a large number), then each node must filter the incoming messages only to recognize control messages from the gateway logger, thus affecting performance. Processing this large amount of messages may not be a problem for devices with more resources, but it can have an impact on microcontroller devices

Event #	Logger node	Channel ‘control’	Channel ‘log/data’	Logger gateway
1		Content: {Auth API info, log command}	Content: {existing logger’s logs}	Periodically publishing *Auth API* and *log command* to ‘**control**’
2	Join logger network, only knows channel ‘**control**’	...	...	...
3	Subscribe to ‘**control**’, get Auth API	...	...	...
4	Hit Auth API with proper credentials	...	...	Receive API request, authentication attempt
5	OK, subscribe to ‘**log/data**’	...	...	Auth successful, reply with channel ‘**log/data**’ information (*name*)
6	Log and publish to ‘**log/data**’ upon logger gateway command on ‘**control**’	...	...	
		Content: {Auth API info, log command}	Content: {existing logger’s logs}	

Fig. 3. Data transmission security scheme using ’control’ and ’log/data’ channels

The gateway logger then saves all received logs into an SQLite database as temporary local storage before being uploaded to the MongoDB Atlas cloud. In this research, we create two schemes related to local storage. We have tested both storage schemes, and both are working fine.

1) All log data received via MQTT is stored in the SQLite database first and will be uploaded to the MongoDB Atlas cloud at specified periods (if internet connection exists).

2) All log data received via MQTT will be directly uploaded to the MongoDB Atlas cloud. If during upload there is an interruption in the internet connection, the logs that failed to upload will be stored in the SQLite database first, and try to be uploaded again in the next logging period (10 minutes later) together with the new data.

B. MongoDB Atlas

The measurement data of several loggers is stored in a MongoDB database provided by MongoDB Atlas. MongoDB Atlas is a global cloud database service for modern applications [20]. Unpaid service is available with limited 512 MB storage capacity, sharing RAM, Highly available replica sets (provided three nodes), end-to-end encryption, automatic patches, and REST API.

The data stored in the MongoDB database is loggerID, timestamp (ts), temperature (temp) and relative humidity (hum) in JSON format (JavaScript Object Notation) as shown in Figure4.


```
{
  "_id": {
    "$oid": "605366675e8b6b713a449066"
  },
  "loggerID": "loggerA",
  "ts": {
    "$date": {
      "$numberLong": "1616073609494"
    }
  },
  "temp": {
    "$numberDouble": "29.4"
  },
  "hum": {
    "$numberDouble": "52.2"
  }
}
```


Fig. 4. A sample document from the MongoDB database

The MongoDB database automatically adds a unique identity _id for every record saved. Figure 5 shows an overview of the data stored in a MongoDB database. The report is based on a sample of 1000 documents. The total amount of data stored is 5925. The two loggerIDs are logger3 and logger4. The recorded air temperature was in the range of 27.7$^{\circ}C$ to 36.2$^{\circ}C$ with an average temperature of about 30$^{\circ}C$ and a skew-normal distribution with a long tail on the right side. Air humidity is in the range of 35.9%RH to 87.5%RH and it can be seen that there are two peaks of data distribution (local and global maximum) so it can be concluded that the data distribution is bi-modal. Data stored in the database were in the period from 5 to 26 November 2020.

ISSN 2355-3286

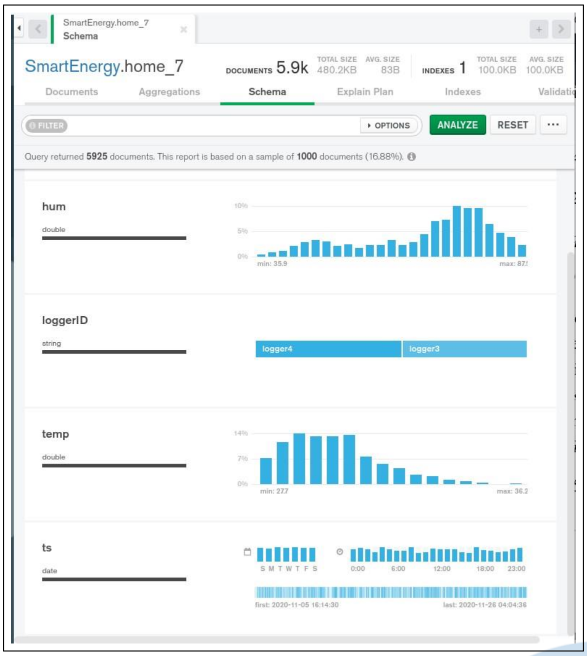

Fig. 5. Overview of data stored in the MongoDB database

C. MongoDB Chart

MongoDB charts are tools to visualise data in MongoDB Atlas. There are various types of charts available such as column/bar charts, line/area charts, combo charts, grid charts (heatmap and scatter), circular charts (donut and gauge), text charts (number, word cloud and top item), and geospatial charts (heatmap, scatter and chloropleth). It is easy to create a dashboard with a collection of charts using MongoDB charts. Charts update automatically as real-time data comes into the MongoDB database. The dashboard is updated in real-time and can be made interactive by enabling filters.

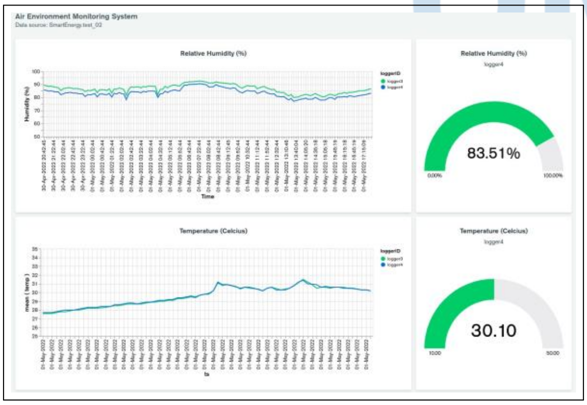

Fig. 6. Air condition monitoring dashboard

Figure 6 shows the dashboard of the air condition monitoring system. The dashboard shows the current measured humidity and temperature as gauge charts. It also shows the historical measurements of humidity and temperature as line charts.

D. DHT22 Sensor Adjustment

Two IoT loggers equipped with DHT22 temperature/humidity sensors generated different reading measurements when placed in the same room, so an adjustment process was required for both sensors. The process was carried out in a room equipped with air conditioning and windows so that it could condition a reasonably wide range of temperature and humidity. One of the DHT22 sensors will be used as a reference, for example, the sensor used in the loggerB. Under ideal conditions, a calibrated temperature/humidity sensor can be used as a reference.

The data from the temperature and humidity measurement during the adjustment process on two DHT22 sensors can be seen in Figure 7. The process was carried out from 8.10 pm local time on 18 March 2021 until the morning at 9.00 am on 19 March 2021.

Figure 7 shows that there is a difference between loggerA and loggerB in both temperature and humidity measurements. There is a fairly large gap (about 5%) seen in the results of air humidity measurements. The adjustment process was performed using the DHT22 sensor on the loggerB as a reference. The result of linear modelling for air temperature on loggerA is given in the following linear equation:

$$t_{adjusted} = 0.93 \times t_{measured} + 1.91 \quad (1)$$

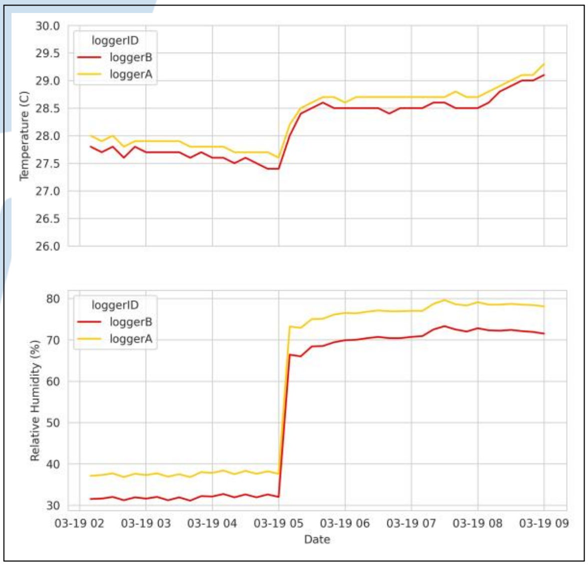

Fig. 7. Results of measurement of air temperature and humidity before the adjustment process

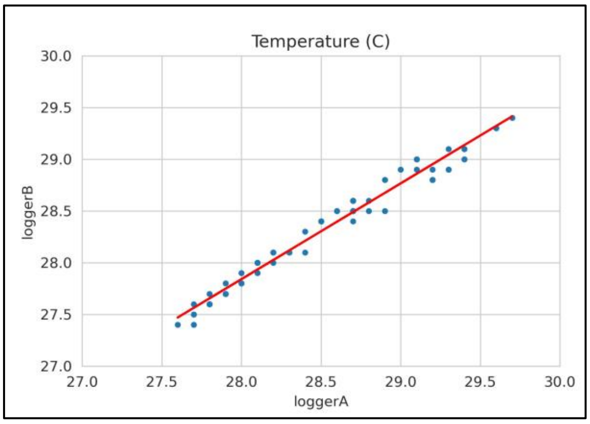

Fig. 8. The results of fitting air temperature data from loggerA and loggerB

Where $t_{adjusted}$ is the result of temperature measurement on loggerA that has been adjusted and $t_{measured}$ is the original result of temperature measurement on loggerA. This linear model is the result of fitting with the least-squares method on the measurement data, which is displayed as a scatter plot as shown in Figure 8.

Linear modelling was also carried out on the loggerA humidity measurement with the results in the form of the following linear equation:

$$h_{adjusted} = 0.98 \times h_{measured} - 5.06 \quad (2)$$

where $h_{adjusted}$ is the result of measuring humidity on an adjusted loggerA and $h_{measured}$ is the original result of measuring humidity on the loggerA. This modelling is the result of fitting with the *least-squares* method on the measurement data and displayed as a scatter plot as shown in Figure 8.

Table I shows the parameters and metrics of the regression models for temperature and relative humidity. A high correlation value (0.975656) for the temperature model indicates that the modelling results have high accuracy; in other words, the results of air temperature measurements with adjusted loggerA and loggerB are very similar. A very high correlation value (0.99985) for the relative humidity model indicates that the modelling results have very high accuracy; in other words, the results of air humidity measurements with adjusted loggerA and loggerB are almost identical.

TABLE I. THE PARAMETERS AND METRICS OF THE FITTED LINEAR MODELS (TEMPERATURE AND RELATIVE HUMIDITY)

Parameters of Regression Model	Temperature	Relative Humidity
Slope	0.92596934	0.982
Intercept	1.91308973	-5.06556
Metrics of Regression Model		
RMS error	0.006698	0.042672
R2 score	0.975656	0.99985

After adjusting the loggerA, the plot results for the same period can be seen in Figure 10. The results of temperature and humidity measurements during the adjustment process show that temperature and humidity measurements with loggerA and loggerB almost overlap in one plot. It indicates that loggerA and loggerB have been successfully ”calibrated” so that both loggers are ready to use.

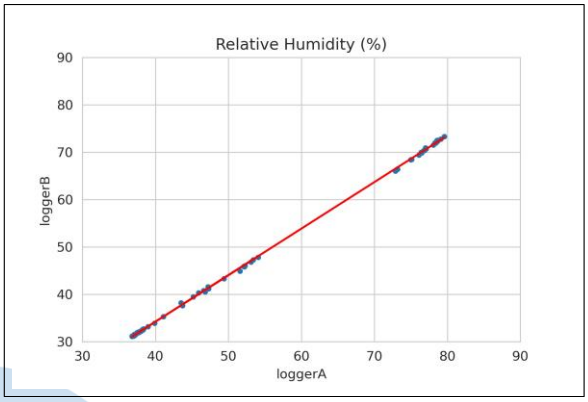

Fig. 9. The results of fitting air humidity data from loggerA and loggerB

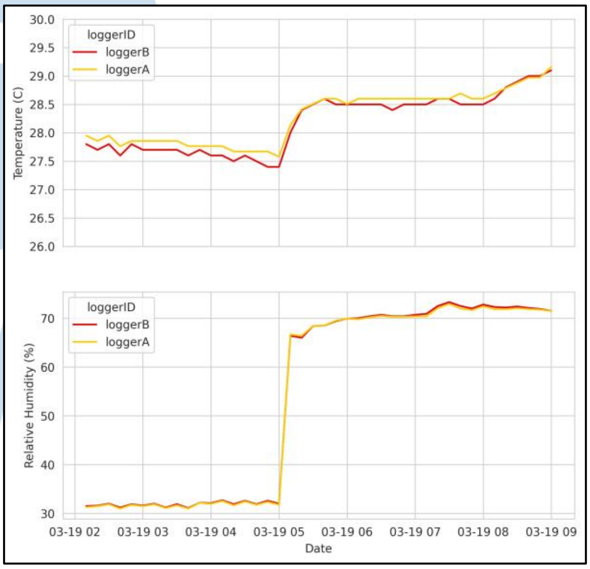

Fig. 10. The results of monitoring the temperature and humidity of the air after the adjustment process

III. Result and Discussion

The developed system has been successfully applied to monitor air conditions in the field. The system is ready to monitor the temperature/humidity of the air in an enclosed space and an open space. The data collected from the monitoring system is used to model the dynamic of air condition due to the outdoor weather condition and the effects of a cooling system.

A. *Experimental Testing*

Air temperature and humidity measurements were conducted at two locations: indoor and outdoor environments. The first location was a classroom. The loggerA and a mini server (Raspberry Pi 4) were placed in this classroom. LoggerA functioned to record the indoor air temperature and humidity. The second location was a corridor with considerable access to fresh air. The loggerB was located in this corridor. LoggerB served to monitor the outdoor air condition. The measurements began at around 11 am on 19 March 2021.

Monitoring of air conditions was carried out for six days consecutively, starting from 20 March 2021 to 25 March 2021 as seen in Figure 11. The air conditioning system was only turned on (as indicated by the value of the state equal to 1) for a few hours on 24 and 25 March 2021 specifically for this monitoring purpose. The air conditioning system is usually turned on when there is a class session in progress. However, there have been no classes scheduled in the building for almost two years due to the pandemic. When the cooling system was turned off, the temperature and humidity in the classrooms were relatively stable (a temperature range of about 28°C and a humidity of about 70%RH). The outdoor air conditions are pretty extreme, namely: the air temperature range is between 24°C to 33°C, and the humidity range is between 50%RH to 100%RH.

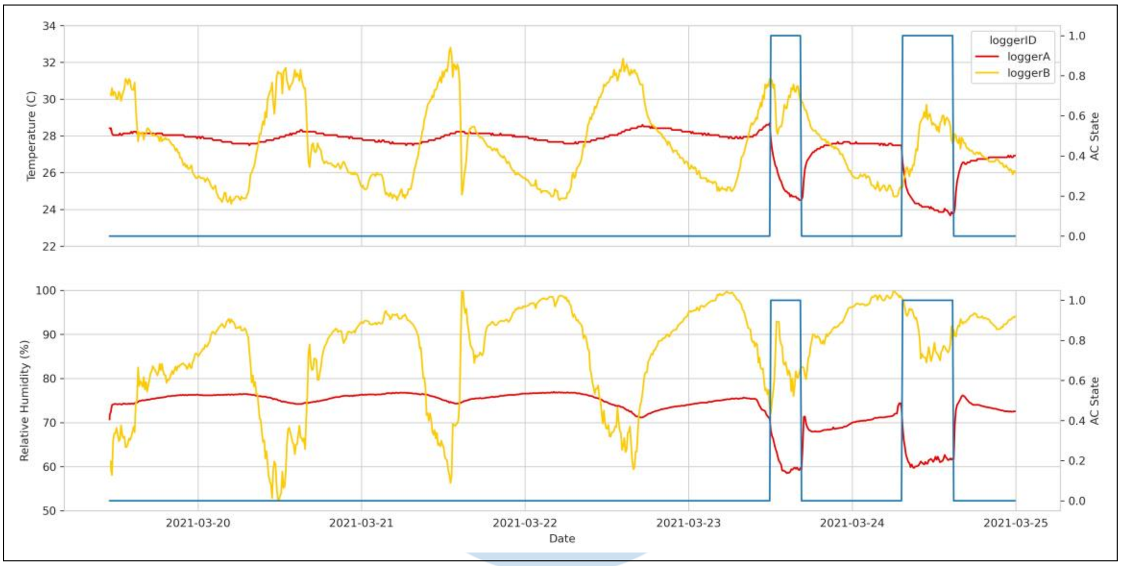

Fig. 11. The results of fitting air temperature data from loggerA and loggerB

When the air conditioning system was turned on, the air temperature quickly dropped to a range of 24°C when the outdoor temperature was in the range of 30°C. A decrease also occurred in the humidity in the classroom from the range of 75%RH to the range of 60%RH when the outdoor humidity was in the range of 90-90%RH.

The air condition in the classroom is relatively stable and isolated from the outdoor air condition because the building is equipped with a double skin facade and a heat-insulating wall that can regulate the intensity of light and reduce the heat from the sun that enters the room so that the room is bright enough and cool even when the air conditioner is turned off.

B. *Modeling Results*

The logged data were analyzed to understand the relationship between indoor and outdoor temperature/humidity and the effect of the cooling system on indoor air conditions. The distribution of temperature and humidity parameter values in each logger provides an overview of the distribution of each of these parameters.

Figure 12(a) shows that the distribution of data from loggerA (indoor temperature) is clustered in two centers: one is centered around 24◦C with low density, and the other is centered around 28°C with high density. The cluster with low intensity came from the measurement when the air conditioning was turned on for a short period. Meanwhile, the data from loggerB (outdoor temperature) is distributed more evenly over a more comprehensive range.

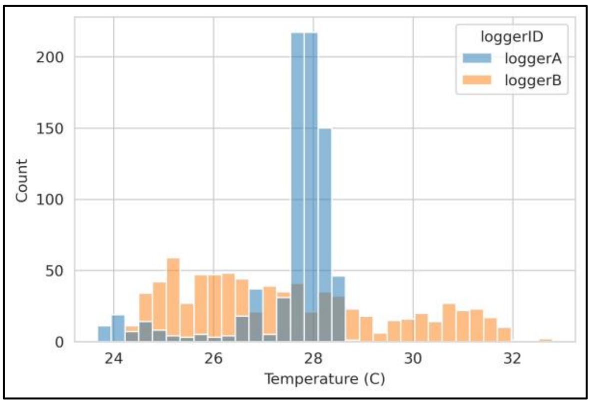

Fig. 12. Histogram of air temperature data recorded by loggerA (located indoors) and loggerB (located outdoors)

Figure 13(b) clearly shows that there are two peaks (local and global) in the humidity data from loggerA. If we fit two Gaussian distributions on the data using the *ExpectationMaximization* [21] algorithm, we will get two clusters with the following distribution:

- Gaussian distribution in low humidity cluster with μ = 68, σ =5.2, weight =0.3.
- Gaussian distribution in high humidity cluster with μ = 75, σ =1.1, weight =0.7.

The mean values of the low and high humidity clusters are 68%RH and 75%RH. The peak at the high humidity cluster is much higher than the peak at the low humidity cluster because during the monitoring process, the air conditioning system is turned on for a minimal time compared to the length of the measurement.

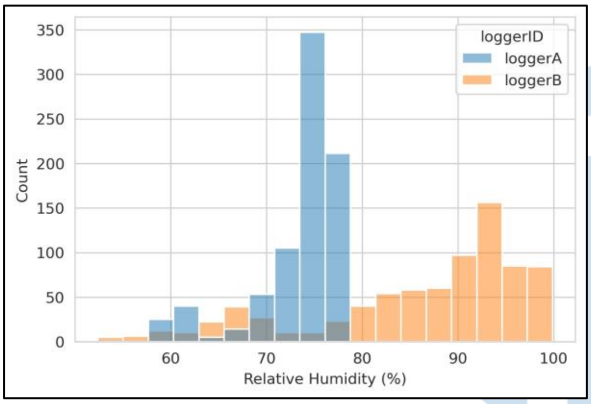

Fig. 13. Histogram of air humidity data recorded by loggerA (located indoors) and loggerB (located outdoors)

Figure 14 shows the Gaussian-Mixture model of the low and high humidity clusters. The model does not fit well to the data because Gaussian-Mixture models assume the data is distributed normally. Figure 14 clearly shows that the collected data is not normally distributed and is imbalanced between two variables. It could be due to the limited number of available data. As the number of measured data increases, the data would be expected to be normally distributed.

Alternatively, we could manually shape the fuzzy membership functions as proposed in [22] to fit the two clusters, as shown in Figure 15. It is shown that the membership functions of the low and high humidity can define the two distributions more distinctly. This fuzzy model can be used as an indicator to distinguish the state of the cooling system, either active orinactive. A simple fuzzy rule can be defined as follows: (a) if the relative humidity is low, then the cooling system is active, and (b) if the relative humidity is high, then the cooling system is inactive.

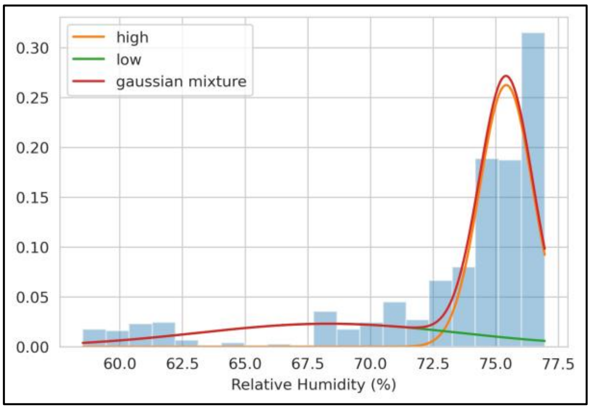

Fig. 14. Gaussian distribution modeling with two clusters (*low and high*) on humidity data from loggerA

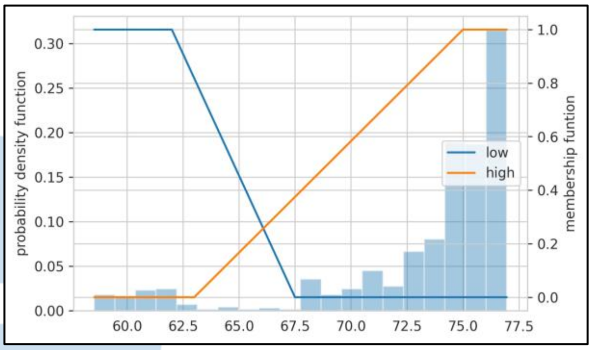

Fig. 15. Fuzzy Model with two clusters (*low and high*) on humidity data from loggerA

The data obtained can be used to predict the state of the cooling system: actively working or not. This model can be used to monitor unmeasured or hidden variables. It can be a cheaper and more effective solution than installing a current sensor in the air conditioning system.

IV. Conclusion

A temperature and relative humidity monitoring system for a conditioned classroom using the Internet of Things has been developed. The system has been successfully ”calibrated” and tested to simultaneously monitor indoor and outdoor air conditions. The monitoring results have been analyzed and used to construct a Gaussian-Mixture and a simple fuzzy model capturing the relationship between the indoor air conditions and the state of the air conditioning system. Further research can use the model as an input to a controller of the air conditioning system to achieve its optimal operation.

Acknowledgment

The authors would like to acknowledge the support given by Universitas Multimedia Nusantara for this research work.

References

[1] R. Andarini, M. Salehuddin, and C. O. Harahap, “Pengembangan Sistem Manajemen Energi di Kampus Universitas Multimedia Nusantara,” Universitas Multimedia Nusantara, Tangerang, Tech. Rep., 12 2019.

[2] M. Di Giulio, R. Grande, E. Di Campli, S. Di Bartolomeo, and L. Cellini, “Indoor air quality in university environments,” Environmental monitoring and assessment, vol. 170, no. 1-4, pp. 509–517, 11 2010. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/19941062/

[3] H. B. Awbi, “Ventilation for Good Indoor Air Quality and Energy Efficiency,” Energy Procedia, vol. 112, pp. 277–286, 3 2017.

[4] A. Asif, M. Zeeshan, and M. Jahanzaib, “Indoor temperature, relative humidity and CO2 levels assessment in academic buildings with different heating, ventilation and air-conditioning systems,” Building and Environment, vol. 133, pp. 83–90, 4 2018.

[5] K. M. Al-Obaidi, M. Ismail, and A. M. Abdul Rahman, “A study of the impact of environmental loads that penetrate a passive skylight roofing system in Malaysian buildings,” Frontiers of Architectural Research, vol. 3, no. 2, pp. 178–191, 6 2014.

[6] P. H. Shaikh, N. B. M. Nor, P. Nallagownden, I. Elamvazuthi, and T. Ibrahim, “A review on optimized control systems for building energy and comfort management of smart sustainable buildings,” Renewable and Sustainable Energy Reviews, vol. 34, pp. 409–429, 6 2014.

[7] J. Shah and B. Mishra, “IoT enabled environmental monitoring system for smart cities,” 2016 International Conference on Internet of Things and Applications, IOTA 2016, pp. 383–388, 2016.

[8] B. Hammi, R. Khatoun, S. Zeadally, A. Fayad, and L. Khoukhi, “IoT technologies for smart cities,” IET Networks, vol. 7, no. 1, pp. 1–13, 1 2018.

[9] A. Zhamanov, Z. Sakhiyeva, R. Suliyev, and Z. Kaldykulova, “IoT smart campus review and implementation of IoT applications into education process of university,” 2017 13th International Conference on Electronics, Computer and Computation, ICECCO 2017, vol. 2018Janua, pp. 1–4, 2018.

[10] K. Sharma and T. Suryakanthi, “Smart System: IoT for University,” Proceedings of the 2015 International Conference on Green Computing and Internet of Things, ICGCIoT 2015, pp. 1586–1593, 1 2016.

[11] R. Andarini, M. Widjaja, and D. K. Halim, “Evaluation of Indoor Air Condition of University Building in Tropics using Building Thermal Modeling,” 2021 2nd International Conference On Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS), pp. 52–57, 10 2021. [Online]. Available: https: //ieeexplore.ieee.org/document/9617192/

[12] L. Smirek, G. Zimmermann, and M. Beigl, “Just a Smart Home or Your Smart Home - A Framework for Personalized User Interfaces Based on Eclipse Smart Home and Universal Remote Console,” Procedia Computer Science, vol. 58, pp. 107–116, 2016.

[13] S. S. Vedaei, A. Fotovvat, M. R. Mohebbian, G. M. Rahman, K. A. Wahid, P. Babyn, H. R. Marateb, M. Mansourian, and R. Sami, “COVIDSAFE: An IoT-based system for automated health monitoring and surveillance in post-pandemic life,” IEEE Access, vol. 8, pp. 188538– 188551, 2020.

[14] D. Minoli, K. Sohraby, and B. Occhiogrosso, “IoT Considerations, Requirements, and Architectures for Smart Buildings-Energy Optimization and Next-Generation Building Management Systems,” IEEE Internet of Things Journal, vol. 4, no. 1, pp. 269–283, 2017.

[15] G. Lilis, G. Conus, N. Asadi, and M. Kayal, “Towards the next generation of intelligent building: An assessment study of current automation and future IoT based systems with a proposal for transitional design,” Sustainable Cities and Society, vol. 28, pp. 473–481, 1 2017.

[16] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” Computer Networks, vol. 54, no. 15, pp. 2787–2805, 10 2010.

[17] R. A Light, “Mosquitto: server and client implementation of the MQTT protocol,” The Journal of Open Source Software, vol. 2, no. 13, p. 265, 5 2017.

[18] D. K. Halim and S. Hutagalung, “Towards data sharing economy on Internet of Things: a semantic for telemetry data,” Journal of Big Data, vol. 9, no. 1, pp. 1–24, 12 2022.[Online]. Available: https://journalofbigdata.springeropen.com/articles/ 10.1186/s40537-021-00549-0

[19] A. van den Bossche, N. Gonzalez, T. Val, D. Brulin, F. Vella, N. Vigouroux, and E. Campo, “Specifying an MQTT Tree for a Connected Smart Home,” in International Conference On Smart homes and health Telematics (ICOST 2018). Springer, Cham, 2018, pp.236–246. [Online]. Available: https://link.springer.com/chapter/10.1007/ 978-3-319-94523-1 21

[20] “Learn More About MongoDB — MongoDB.” [Online]. Available: https://www.mongodb.com/discover-mongodb-atlas

[21] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum Likelihood from Incomplete Data via the EM Algorithm,” Journal of the Royal Statistical Society. Series B (Methodological), vol. 39, no. 1, pp. 1–38, 1977.

[22] M. Widjaja, A. Darmawan, and S. Mulyono, “Fuzzy classifier of paddy growth stages based on synthetic MODIS data,” in 2012 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2012 - Proceedings, 2012, pp. 239–244.