Integra: Journal of Integrated Mathematics and Computer Science
Journal Homepage: https://integrajimcs.com/index.php/integra/index

integra
Journal of Integrated Mathematics and Computer Science

Research Paper

IoT-Based Air Conditioner Monitoring and Control at PT XYZ: A Prototype Approach Utilizing Node MCU ESP8266, Relay Modules, and DHT11 Temperature Sensors

Thalia Gemi Innaya[1], Akmal Junaidi[1*], Fadli Alfikri[2], Didik Kurniawan[1], Muhammad Iqbal[3]

[1] *Department of Computer Science, Faculty of Mathematics and Natural Science, Universitas Lampung, Lampung, 35145, Indonesia*
[2] *Industrial Partner, PT XYZ, Bandar Lampung, Lampung, 35241, Indonesia*
[3] *Statistics and Data Science Study Program, Universitas Negeri Semarang, Semarang, Central Java, 50229, Indonesia*
[*] **Corresponding author**: akmal.junaidi@fmipa.unila.ac.id

Keywords
IoT, Air Conditioner, Node MCU ESP8266, DHT11, Relay Module

Abstract
This study presents the development and implementation of an IoT-based monitoring and control system for air conditioners (AC) at PT XYZ. Leveraging the capabilities of Node MCU ESP8266, relay modules, and DHT11 temperature sensors, the proposed system enables real-time monitoring of AC status and room temperature. It also facilitates control over individual and multiple AC units, thereby enhancing operational efficiency. The system's design focuses on energy efficiency, aiming to reduce unnecessary power consumption by ensuring that AC units are active only when needed. The system includes features for monitoring the AC's on/off status and providing historical temperature data through graphical charts. This real-time data display allows users to track temperature trends and make informed decisions regarding AC operation. The prototype approach employed in this research involves several stages: communication with stakeholders, planning, modeling, prototype development, deployment, and iterative feedback. Testing results indicate that the system effectively meets the research objectives by providing accurate temperature readings and reliable AC control. Compared to existing solutions, this system offers enhanced functionality and integration, contributing to energy savings and improved management of AC units. This research contributes to the field by addressing gaps identified in previous studies and demonstrating the practical application of IoT technology in energy management for air conditioning systems.

Received: 18 May 2025, Accepted: 12 July 2025
https://doi.org/10.26554/integrajimcs.20252237

1. INTRODUCTION

In today's era, the integration of Internet of Things (IoT) technologies has revolutionized various fields, including energy management and automation. The need for efficient energy utilization and effective control systems is increasingly critical in industrial settings. This study focuses on developing an IoT-based system for monitoring and controlling air conditioners (AC) at PT XYZ, aiming to enhance energy efficiency and provide real-time temperature management.

The primary objectives of this research are threefold. Firstly, the study aims to develop a comprehensive system capable of monitoring and controlling AC units at PT XYZ using IoT technology. This includes real-time monitoring of AC status and temperature, as well as control over the operation of individual and multiple AC units. Secondly, the research seeks to improve energy efficiency by ensuring that AC units operate only when necessary, thus reducing unnecessary energy consumption and operational costs. Lastly, the system aims to provide real-time temperature monitoring across various areas within PT XYZ, offering accurate and up-to-date temperature data for better decision- making and energy management.

Previous research has laid a foundation for this study by exploring various aspects of IoT-based AC monitoring and control system. The system for monitoring and controlling server room data had been investigated [1, 2], while Navarrete-Sanchez et al [3] observed the IoT-based classroom temperature monitoring and using Raspberry Pi and ESP32, and Septiasari and Firdausy in 2021 implemented a system utilizing NodeMCU ESP8266 for AC temperature monitoring and control, showcasing the effectiveness of real-time temperature regulation through a mobile application [4]. A remote-controlled AC management system, which addressed issues of AC usage when unattended and emphasized energy efficiency was investigated Gamaliel and Arliyanto [5], an automating AC maintenance and monitoring, incorporating various sensors to optimize AC performance and extend its lifespan is observed by Diori et al [6]. For a larger area, Stoev et al [7] observed the application for temperature control in residential premises.

The prototype methodology adopted in this research involves communication with stakeholders, detailed planning, system modeling, prototype development, deployment, and iterative feedback. This approach ensures a robust and user-centric system, capable of meeting the energy management needs of PT XYZ while contributing to advancements in IoT applications for industrial energy efficiency.

2. METHODS

The research methodology used in creating an IoT-based AC monitoring and control application at PT XYZ consists of several stages. Figure 1 shows those stages.

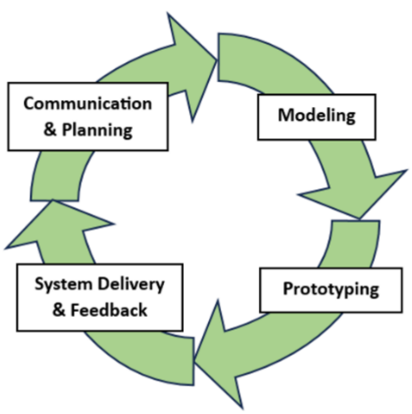

Figure 1. Prototype Development Life Cycle

The prototype research method involves several main stages: communication, planning, modeling, prototyping, system delivery, and feedback delivery. Communication and planning are the initial stages where system needs and goals are identified. Modeling involves the initial design of the system, followed by the creation of a prototype as an early representation of the proposed system. This prototype is then tested, refined, and refined through iterations based on user feedback until it meets the desired specifications [8].

In the development of IoT-based systems, such as AC monitoring and control systems, the prototype method is very useful because it allows developers to test basic functions such as device connectivity, sensor data collection, and user interfaces on a small scale before developing a more complex final version [9].

3. RESULTS AND DISCUSSION

3.1 Communication

Communication to obtain information verbally with the aim of obtaining data that can explain research problems. Based on the source of the problem in the use of Air Conditioner, namely by leaving the Air Conditioner on in an empty room and there is no activity that will have an impact on increasing electricity consumption, causing the cost of electricity bills to increase. In addition, the use of Air Conditioner that is not accompanied by good maintenance will cause damage and suboptimal performance of the Air Conditioner. Therefore, a centralized monitoring and control system is needed to turn the Air Conditioner on and off.

3.2 Planning

This stage is carried out to plan the system that will be used in detail. This system design includes hardware and software specifications that will be used in the system.

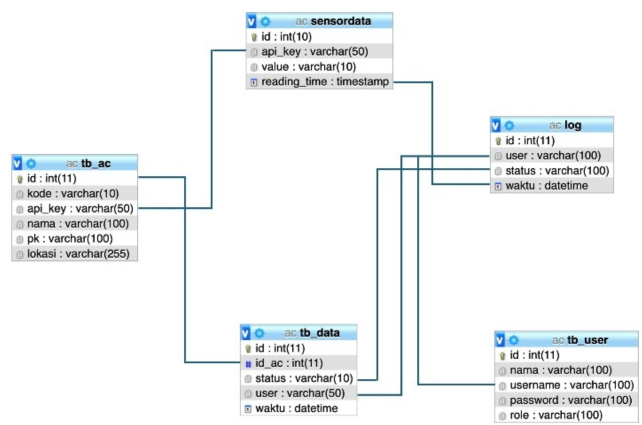

Figure 2. Entity Relationship Diagram of the System Database

3.3 Modeling

The next stage is the design or modeling of the system to be developed, such as the processes that occur. Modeling is done by designing activity diagrams and user interface designs.

3.3.1 Entity Relationship Diagram

Entity-Relationship Diagram (ERD) is a conceptual tool that represents the structure of a database. It shows the relationships between different entities within the system, aiding in the design of an organized database schema [10]. ERD plays a critical role in the system development lifecycle by providing a visual representation of the data model, ensuring that all relationships and dependencies are accurately captured before implementation [11]. Figure 2 displays the ERD of the system database.

3.3.2 UseCase Diagram

Use case diagram is a tool used in software engineering to describe the interaction between users or actors with the system. This diagram aims to identify important functions that must be owned by the system based on user needs [8]. Figure 3 displays the use case diagram of the system.

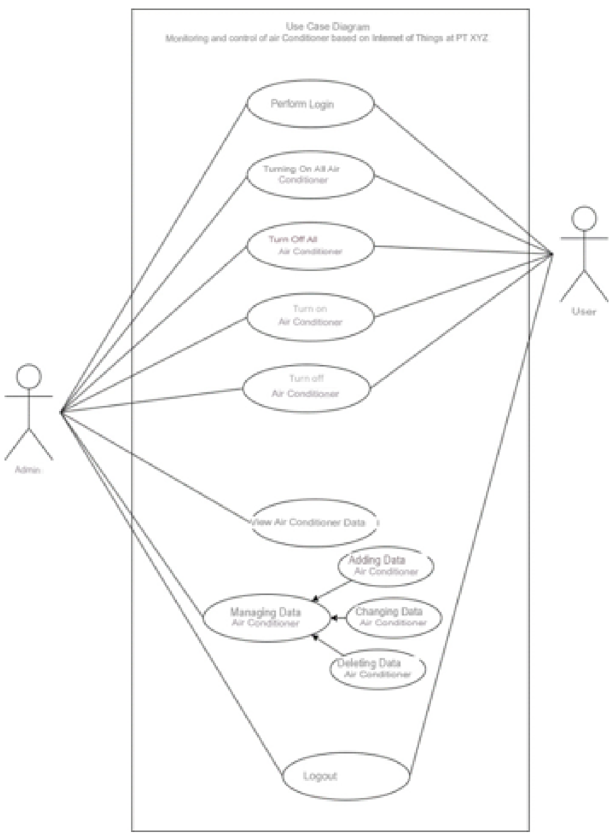

Figure 3. Use Case Diagram of the System

3.3.3 Activity Diagram

Activity diagrams can help system developers understand the business process or workflow of an application in a visual and easy-to-understand way. Activity diagrams usually consist of elements such as activities, transitions, decisions, and the beginning and end of the workflow. These elements are connected by arrows that indicate the sequence of activities [12].

3.4 Construction

Build a functional prototype of the designed system. At this stage, implement the selected features and integrate the required technologies. Develop software and hardware for the AC monitoring system, such as programming IoT sensors, setting up servers, and creating user interfaces.

3.4.1 Hardware Design Stage

This stage is carried out to design the hardware used in the system. The specifications of the devices to be used are as follows:

• NodeMCU ESP8266: The main module for data processing and internet connectivity. The ESP8266 Node MCU is a microcontroller module based on the ESP8266 chip. This module is very popular in Internet of Things (IoT) applications because of its ability to connect to WiFi networks easily. The ESP8266 Node MCU is equipped with ample memory and programmable GPIOs, allowing users to control devices or collect data over the internet [13, 14].

• DHT11 Temperature Sensor: To measure room temperature. DHT11 is a digital temperature and humidity sensor that is often used in various electronic projects. This sensor provides temperature measurement in the range of 0-50°C with an accuracy of ± 2°C and humidity in the range of 20-80% RH with an accuracy of ± 5% RH. DHT11 is very popular because of its ease of use and low cost [15, 16, 17, 18] .

• Relay Module: To control the AC device. The relay module is a device that functions as an electronic switch that can control high electric currents with small electrical signals from a microcontroller. This module consists of several relays that can be used to control electrical devices, such as lights or motors, using signals from a microcontroller such as the Node MCU ESP8266 [19, 20].

• Supporting tools: (Jumper cables, boxes, and power adapters / USB cables)

After determining the device specifications, the device modules are wired and connected. Figure 4 indicates the connection among the device modules. With this configuration, the modules are designed to monitor and control the AC not only from the office but also from a remote location.

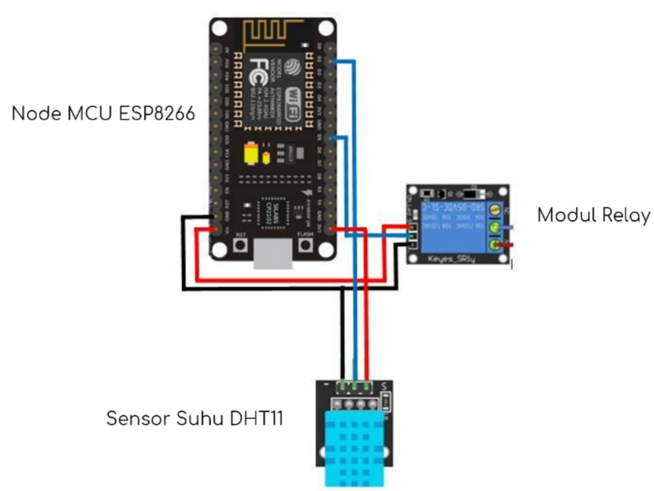

Figure 4. Hardware Design

Figure 5 is a series of AC monitoring and control hardware consisting of a DHT11 temperature sensor, NodeMCU ESP8266, and a relay module in the actual setting. The modules are physically connected using a jumper/dupont cable. All device modules have been assembled and placed in a project box or enclosure box. The project box or enclosure box functions as protection, security, and aesthetics. By using a project box or enclosure box, the hardware series is not only more durable and safe, but also neater and easier to manage and present.

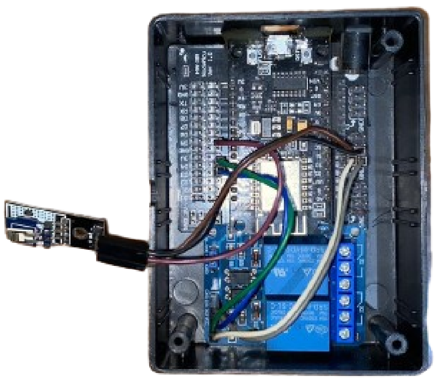

Figure 5. Actual Connectivity of Hardware

Configuration:
DHT11:
• Data Pin (DHT11) is connected to GPIO2 (D5) on NodeMCU.
• VCC (DHT11) is connected to 3.3V on NodeMCU.
• GND (DHT11) is connected to GND on NodeMCU.
Relay Module:
• Control Pin (Relay) is connected to GPIO5 (D2) on NodeMCU.
• VCC (Relay) is connected to 5V on NodeMCU or external power supply.
• GND (Relay) is connected to GND on NodeMCU.
Circuit Schematic:

a. NodeMCU ESP8266 is connected to the DHT11 sensor to read temperature and humidity data.
b. NodeMCU ESP8266 is connected to a relay module, which is then connected to the AC (Air Conditioner) unit.
c. NodeMCU ESP8266 is connected to a WiFi network to send data to the server and receive commands from the server.

Work Flow:
• DHT11 → Measures temperature and humidity → Sends data to NodeMCU ESP8266. • NodeMCU ESP8266 → Sends data to server → Server stores data and performs analysis. • Server → Sends commands to NodeMCU ESP8266 → NodeMCU controls the relay module to turn on/off the AC.

3.4.2 Software Design Stage

Code Function
• Sending Data to Server: Temperature and humidity data are sent to the server using an HTTP POST request with https.POST (httpRequestData).
• Retrieving Data from Server: Retrieving JSON data from the server with https.GET() and processing the data to get the AC status. AC Control: The status of the AC (ON/OFF) is checked and the relay control is carried out based on that status. If the status changes from OFF to ON or vice versa, the relay will be activated or deactivated.

3.4.3 Temperature Calibration

Temperature calibration is the process of adjusting a temperature measuring instrument so that the resulting measurements correspond to the values that should be based on international standards [21]. Figure 6 shows the temperature calibration between dht11 and thermometer.

The First Day

DHT11	Analogue Thermometer	Absolute Error
27.10	26.00	1.10
26.20	26.20	0
25.30	25.30	0
27.10	26.00	1.10
24.50	24.50	0
Average		0.44

The Second Day

DHT11	Analogue Thermometer	Absolute Error
28.50	28.50	0
28.50	28.50	0
28.60	28.60	0
28.90	28.90	0
28.90	28.90	0
Average		0

Figure 6. Temperature Calibration Between DHT11 and Thermometer

Temperature calibration between the DHT11 sensor and an analog thermometer tested every thirty minutes, with the AC on and the sensor module (DHT11, WiFi ESP8266, and relay module) installed, showed fairly consistent results, with a measurement error of 0.44°C on the first day and 0°C on the second day. This difference indicates that the DHT11 sensor has an adequate level of accuracy after the calibration process is carried out. The error on the first day is likely caused by external factors or initial instability of the sensor. However, on the second day, the results showing an error of 0°C indicate that the DHT11 sensor has achieved good consistency after adjustment. The average calibration error is 0.22°C. Thus, it can be concluded that the DHT11 sensor is reliable for applications that require temperature measurement with minimal error tolerance after proper calibration.

3.4.4 Website Design Stage

The features of this Internet of Things-based air conditioner monitoring and control website include:

a. Temperature monitoring, which displays the room temperature read from the DHT11 sensor.
b. AC control, which provides the option to turn the AC on or off via a web interface.
c. Relay status, which displays the current status indicating whether the AC is on or off.

Figure 7 shows the Internet of Things-based AC monitoring and control dashboard at PT XYZ. This system provides a summary of the status of all AC units at PT XYZ.
• The top contains the AC data feature: can view AC data in detail, add AC data, change AC data, and delete AC data.

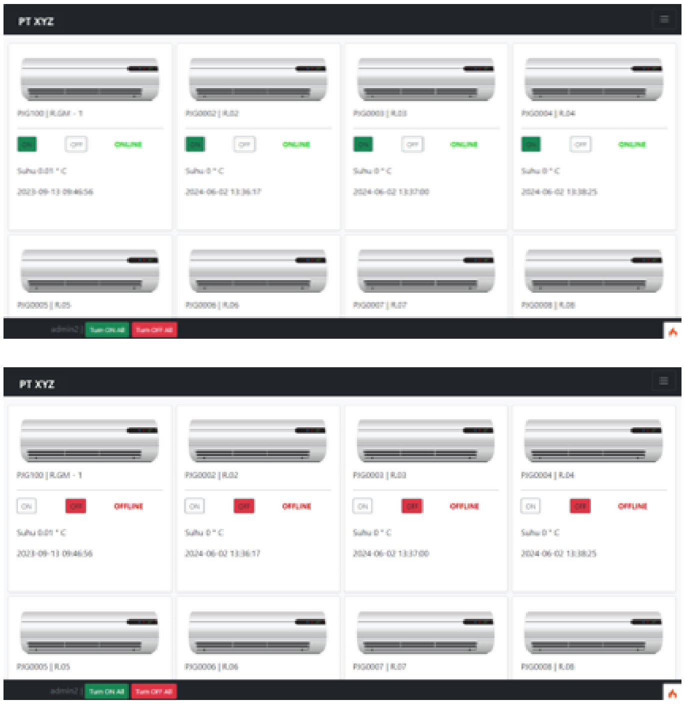

Figure 7. The System Dashboard

• The middle of the screen displays a list of all AC units connected to the IoT system. Each AC unit may be accompanied by specific information such as location (eg: meeting room, main office, production room), current temperature, operational status (on/off), and real-time.
• Remote control: On/Off button that allows users to turn all AC units on or off and turn each AC unit on or off.

Visual indicators such as green icons indicate that each AC is operating and visual indicators such as red icons indicate that each AC is not operating.

This IoT-based AC monitoring and control web display makes it easy to manage the air conditioning system at PT XYZ. With all AC units on and all AC units off, this display shows that the system is working optimally to maintain a comfortable environment for all employees.

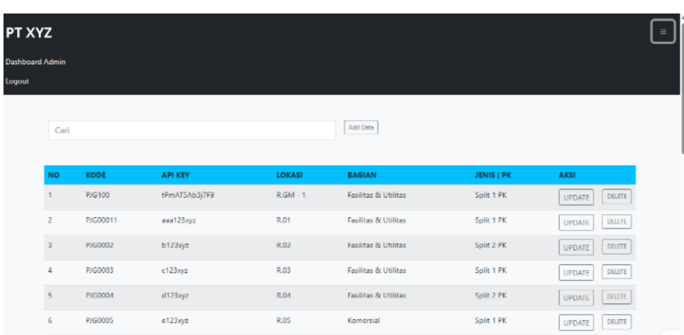

Figure 8. AC Data Website View

Figure 8 can only be accessed by admins. On the AC data page, users can view detailed AC data, add AC data, change AC data, delete AC data. This web display presents various important information related to the condition and operation of AC units at PT XYZ, where the data is taken and updated in real-time via Internet of Things (IoT) technology. The collected data is stored in a local database, which is placed on PT XYZ's internal server. Placing the database locally ensures that all data, including AC status, temperature, and operating history, is stored within the company's internal network. This provides several advantages, such as increased security because data does not pass through an external internet network, as well as faster accessibility for users within the local network. On the Internet Of Things-based AC monitoring and control data page at PT XYZ, the admin will see several main elements that make up the AC data, namely AC code, AC location, AC part, AC/PK type.

3.5 System Submission/feedback

Prototype Evaluation occurs at the submission and feedback stage, after the prototype is created, the prototype is submitted to the user to be tested and evaluated to ensure that it meets user requirements. The results of the prototype evaluation are given as feedback to the developer for further prototyping iterations. Several components in this information system prototype do not match user expectations, such as:
• In the AC data display displayed on the dashboard, a view chart feature and date selection are added so that users can easily view the history of AC usage based on the selected date.
• Converting SQL database to JSON
Figure 7 is a display of the system where the view chart button had been added. View chart AC is a history of AC temperature data. Users can select the date they want to display. SQL has been converted to JSON, because JSON can be used to present data in a more user-friendly format, for example, to create reports or interactive data views in web or mobile applications.

Figure 9 displays the sample of daily temperature change.

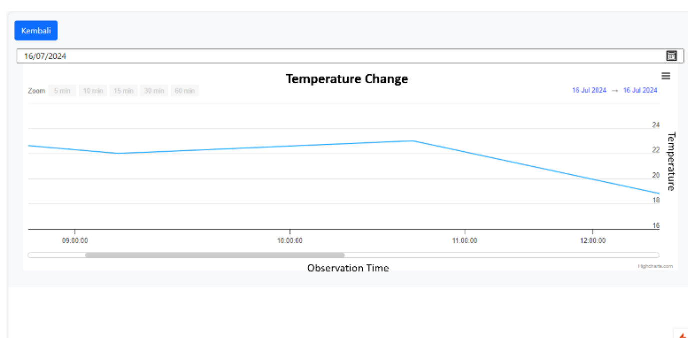

Figure 9. The Sample of Daily Temperature Change

3.6 Testing

IoT system testing generally involves evaluating its performance and reliability. Various testing methods for IoT systems, including functionality, performance, and reliability testing. The

TC ID	Test Class	Test Case	Expected Output	Status Accepted
TCH01	Connection	Connection between NodeMCU ESP8266 and DHT11 sensor	Connection successfully established	Valid Accepted
TCH02		Connection between Node MCU ESP8266 and modul relay	Connection successfully established	Valid Accepted
TCH03		Connection between NodeMCU ESP8266 and server	Connection successfully established	Valid Accepted
TCH04	Monitoring	Temperature sensor	Temperature data is read accurately	Valid Accepted
TCH05	Monitoring	AC Status	AC status is read accurately	Valid Accepted
TCH06	Control	Turn on the AC	AC turns on according to command	Valid Accepted
TCH07	Control	Turn off the AC	AC turns off according to command	Valid Accepted

Figure 10. Hardware Testing

Tes Case ID	Test Class	Test Case	Expected Output	Achieved Output	Status Accepted
TC001	AC Control	Turn on a specific unit	The AC unit will turn on and the status will be updated on the dashboard.	A notification appears "are you sure? You want to turn on AC?" when the user presses the "yes, turn on AC" button Then the AC unit will turn on and the status will be updated on the dashboard. The AC remains in its previous status when the user presses the cancel button	Valid Accepted
TC002		Turn off a specific unit	The AC unit will turn off and the status will be updated on the dashboard.	A notification appears "are you sure? You want to turn off AC?" when the user presses the "yes, turn off AC" button Then the AC unit will turn off and the status is updated on the dashboard. The AC remains in its previous status when the user presses the cancel button	Valid Accepted
TC003		Turn on all AC units	All AC units will turn on and the status will be updated on the dashboard.	A notification appears "are you sure? You want to turn on all AC?" when the user presses the "yes, turn on all AC" button. Then all AC units will turn on and the status is updated on the dashboard. The AC remains in its previous status when the user presses the cancel button.	Valid Accepted
TC004		Turn off all AC units	All AC units will turn off and the status will be updated on the dashboard.	A notification appears "are you sure? You want to turn off all AC?" when the user presses the "yes, turn off all AC" button. Then all AC units will turn off and the status is updated on the dashboard. The AC remains in the previous status when the user presses the cancel button.	Valid Accepted
TC005	Manage AC data (admin only)	Admin manages data (add, edit, delete)	Admin can view AC data in detail, add AC data, edit AC data, and delete AC data.	Admin can view AC data in detail, add AC data, edit AC data, and delete AC data.	Valid Accepted
TC0010	Chart	Viewing the AC usage chart.	The AC usage chart is displayed with the correct data and according to the	The AC usage chart is displayed with the correct data and according to the selected date	Valid Accepted

Figure 11. The Functionality Testing

method used in this study follows the guidelines suggested by Smith to ensure the AC system functions as expected [22].

The Internet of Things-based Air Conditioner monitoring and control system can be tested using NodeMCU ESP8266, DHT11 sensor, and relay. NodeMCU ESP8266 can be used to collect temperature and humidity data from the DHT11 sensor, and the relay can be used to control devices such as air conditioners.

System testing is carried out to prevent the emergence of bugs or errors that may arise during the system development process, as well as to identify weaknesses that may exist in the system. Figure 10 shows the results of hardware testing, while Figure 11 displays the functionality testing.

From Figure 10 it can be seen that the IoT-based AC monitoring and control system works well, starting from the connection between components to the temperature monitoring and AC control functions.

From Figure 11, there are some information regarding the system.

• Test Case ID: Each test case has a unique ID, such as TC001, TC002, and so on.

• Test Case: Describes the test scenario. For example, "Turn off a specific unit" or "Turn off all AC units" means the test is performed to turn off the AC individually or all ACs.

• Expected Output: The expected result of the test scenario. For example, in the "Turn off a specific unit" test, the expected output is that the AC will turn off and its status will be updated on the dashboard.

• Actual Output: The result obtained from the actual test. This

describes whether the system behaves as expected. For example: "The AC unit will turn off and the status will be updated on the dashboard".
• Status: Indicates whether the test was successful or failed. If the actual result is as expected, the status is "Valid" or "Accepted". The table illustrates the results of testing the features of the AC control and monitoring system, with all tests providing valid results as expected.

4. CONCLUSIONS

This study successfully designed and implemented an IoT-based AC monitoring and control system at PT XYZ, which increases user flexibility and convenience in remotely adjusting the temperature and increases energy efficiency through real-time optimization of AC usage. This system also ensures accurate and up-to-date temperature information in various areas. For further system development, it is recommended to use the latest IoT technology, improve data security protocols through stricter encryption and authentication, conduct periodic evaluations to ensure optimal performance and identify areas for development.

5. ACKNOWLEDGEMENT

The authors would like to thank Komputasi Dasar and Rekayasa Perangkat Lunak Lab, Department of Computer Science, Universitas Lampung, for the support given.

REFERENCES

[1] Agung Pradana and Nurfiana. Rancang bangun monitoring dan kontrol suhu ruang server menggunakan perangkat mobile berbasis internet of things (IoT). *Vol. 4*, pages A93–A98, 2019.
[2] M. Asep Rizkiawan, Harry Ramza, and Agus Sofwan. Internet of things (iot) based temperature and humidity detector prototype in the uhamka data center room. *Indonesian Journal of Artificial Intelligence and Data Mining*, 7(4):110–120, 2024.
[3] M. A. Navarrete-Sanchez, Re. Olivera-Reyna, Ro. Olivera-Reyna, R. J. Perez-Chimal, and J. U. Munoz-Minjares. Iot-based classroom temperature monitoring and missing data prediction using raspberry pi and esp32. *Journal of Robotics and Control*, 6(1):234–245, 2025.
[4] Lara Septiasari and Muhammad Fahriza Firdausy. Sistem kontrol dan monitoring air conditioner berbasis internet of things (IoT). Technical report, 2021.
[5] Fritz Gamaliel and P. Yudi Dwi Arliyanto. Implementasi sistem monitoring dan kontrol air conditioner menggunakan internet of things. *Jurnal Informatika dan Teknik Elektro Terapan*, 11(3):311–318, 2023.
[6] Gohi Diori, Dhea Amelia Rianjani, Gilang Maulana, Tyara Zhafirah Tamzil, Maykel Manawan, and Agus Sukandi. Sistem otomatisasi dan monitoring perawatan berkala AC (Air Conditioner) berbasis arduino yang terintegrasi iot (internet of things). In *Prosiding Seminar Nasional Teknik Mesin, Politeknik Negeri Jakarta*, pages 184–193, 2019.
[7] I. Stoev, S. Zaharieva, and A. Borodzhieva. Internet of things (IoT) application for temperature control in residential premises. In *21st International Symposium INFOTEH-JAHORINA (INFOTEH)*, pages 1–5, East Sarajevo, Bosnia and Herzegovina, 2022.
[8] R. S. Pressman. *Software Engineering: A Practitioner's Approach*. McGraw-Hill, New York, 7th edition, 2010.
[9] L. Xu, W. He, and S. Li. Internet of things in industries: A survey. *IEEE Transactions on Industrial Informatics*, 10(4):2233–2243, November 2014.
[10] T. Connolly and C. Begg. *Database Systems: A Practical Approach to Design, Implementation, and Management*. Pearson, Harlow, England, 6th edition, 2015.
[11] S. Bagui and R. Earp. *Database Design Using Entity Relationship Diagrams*. CRC Press, Boca Raton, FL, 2nd edition, 2011.
[12] M. Fowler. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley, Boston, MA, 3rd edition, 2004.
[13] NodeMCU Team. Nodemcu documentation. https://nodemcu.readthedocs.io/en/latest/. Accessed: July 2025.
[14] H. Jang. A review of esp8266 wi-fi module based applications. *Journal of Embedded Systems*, 5(3):45–51, 2018.
[15] Adafruit Learning System. Adafruit dht11 tutorial. https://learn.adafruit.com/dht. Accessed: July 2025.
[16] DHT11 datasheet. https://components101.com/sites/default/files/component_datasheet/DHT11.pdf. Accessed: July 2025.
[17] Arduino Team. DHT11 documentation on arduino website. https://www.arduino.cc/en/Reference/DHT11. Accessed: July 2025.
[18] T. Y. Hsu. An overview of DHT11 sensor and its applications in environmental monitoring. *Sensors and Actuators B: Chemical*, 244:112–120, 2017.
[19] Songle srd-05vdc-sl-c relay datasheet. https://components101.com/sites/default/files/component_datasheet/SRD-05VDC-SL-C-Datasheet.pdf. Accessed: July 2025.
[20] R. A. Hightower. Relay module: Applications and circuit design. *International Journal of Electrical Engineering & Technology*, 8(1):78–85, 2016.
[21] W. R. Fenner and R. C. Voth. *Fundamentals of Temperature Calibration*. John Wiley & Sons, 2005.
[22] J. Smith. Methods for testing IoT systems: A comprehensive review. *Journal of Internet Technology*, 22(3):123–135, 2022.