Journal of Engineering, Electrical and Informatics

E- ISSN: 2809-8706
P- ISSN: 2810-0557

(Research Article)

Design and Construction of an Automatic Mosque Locking System Using Internet of Things (IoT) Technology with Android Smartphone Control

Imam Tri Suryadin

Muhammadiyah University of Gombong; imam.ts@gmail.com

***** Author Correspondence

Abstract: The continuous evolution of the Internet of Things (IoT) has significantly transformed how devices interact and exchange information, enabling automation in various fields, including facility security systems. Mosques, as community-based facilities, still rely heavily on manual locking mechanisms that are time-consuming and prone to human error or key loss. This study presents the design and implementation of an IoT-based automatic mosque locking system that can be monitored and controlled via an Android smartphone. The system architecture integrates a NodeMCU ESP8266 microcontroller as the central control unit, a relay module to activate a servo-powered door lock, and a magnetic reed sensor to identify door status. Communication between the hardware and mobile interface is established using the MQTT protocol connected through the Blynk IoT cloud platform. Experimental evaluation confirms that the system can perform remote locking and unlocking within an average latency of 1–2 seconds, depending on Wi-Fi signal strength. Furthermore, the system provides automatic status updates and notifications when the door is opened or closed. The prototype demonstrates reliable performance, energy efficiency, and ease of use for mosque administrators. In conclusion, the proposed IoT-based smart lock solution successfully enhances mosque security management and shows promising potential for wider adaptation to other public facilities requiring secure and remotely accessible control systems.

Keywords: Internet of Things (IoT); Smart Lock; Mosque Security; NodeMCU; MQTT; Android; Automation.

Received: September 22,2025
Revised: October 02,2025
Received: October 21,2025
Published: October 31,2025
Current version: October 31,2025

1. Introduction

The development of Internet of Things (IoT) technology has revolutionized the way humans interact with electronic devices, enabling automation and remote system control through internet connectivity [1]. IoT functions as an ecosystem that connects various physical devices (things)—such as sensors, actuators, and microcontrollers—so they can exchange data in real-time [2]. The application of IoT has expanded in various fields, including smart homes, industry, agriculture, and public security systems [3]. One potential form of application is a smart lock system that can provide security, efficiency, and convenience for users.

Mosques, as centers of worship and social activities for Muslims, are public facilities that require a reliable security system. Generally, mosques still use traditional mechanical locks, which have several disadvantages, such as the risk of losing keys, difficulty in managing

access, and the potential for break-ins. These conventional systems cannot provide real-time door status information, so mosque administrators often do not know whether the mosque has been locked after activities are completed. In the context of modern management, this problem demands a more efficient, integrated, and easy-to-use solution for mosque administrators without requiring physical presence on site [4], [5].

Several previous studies have attempted to utilize IoT technology to overcome the limitations of conventional locking systems. Reddy and Kumar [2] designed a smart lock system based on NodeMCU ESP8266 with Wi-Fi connection that is capable of controlling door locks over a local network. Zhang et al. [6] proposed the use of the MQTT (Message Queuing Telemetry Transport) protocol because it is lighter and more efficient than the HTTP protocol in sending data between IoT devices. Rahman et al. [3] integrated MQTT with a mobile application to create real-time communication between users and locking devices. However, most previous studies are still limited to residential (smart home) implementations and have not accommodated the needs of public facilities such as mosques, which have different characteristics in terms of access management and usage levels.

The advantages of previous systems lie in their ease of implementation and low cost, as NodeMCU is an open-source microcontroller that is easy to program and compatible with many sensors [7]. However, these systems have significant drawbacks, including limited network coverage (especially when using only a local connection), lack of integration with cloud services, and the lack of automatic notification features when doors are opened or closed [8], [9]. In addition, many studies have not utilized user-friendly application interfaces, even though this aspect is important for implementation in social environments such as mosques, where operators do not always have a technical background.

Based on this analysis, this study proposes an IoT-based automatic mosque locking system controlled by an Android application. This system is designed using a NodeMCU ESP8266 as a control center, a relay module as an electronic switch to control the servo-based door lock, and a magnetic reed switch sensor to detect door conditions. All components are integrated through the MQTT protocol connected to the Blynk IoT cloud platform, allowing users to monitor and control the system in real-time via an Android smartphone. With this approach, mosque administrators can open or lock doors from anywhere as long as they are connected to the internet, while also receiving automatic notifications regarding changes in door status.

The main contributions of this research can be detailed as follows:

1. Design and implement an IoT-based mosque smart lock prototype using NodeMCU, relay, servo, and magnetic sensors with MQTT-Blynk cloud integration;
2. Develop an Android application that serves as a user interface for remote control and real-time monitoring of door status;
3. Conduct system performance evaluations based on response time, sensor accuracy, and network connection stability, to assess its effectiveness and reliability in the context of public use.

From a social perspective, the implementation of this system is expected to assist mosque administrators in improving security and operational efficiency, especially in mosques without permanent guards. Academically, this research expands the application of

IoT technology to the security of religious facilities, a topic previously rarely discussed in the literature.

.

2. Literature Review

The development of Internet of Things (IoT) technology brings a new paradigm in the development of modern automation systems that allow physical devices to communicate and exchange data via the internet network [1], [3]. With IoT, devices such as sensors, actuators, and microcontrollers can work collaboratively to produce intelligent systems capable of monitoring, controlling, and making decisions automatically. One real application of this concept is the smart lock system used in the security management of public and private spaces.

IoT-based smart lock systems have the ability to remotely control door access, authenticate users, and provide real-time door status reports via a wireless network [2], [6]. The combination of efficient hardware and a cloud platform makes this system flexible and easy to integrate into various contexts, including social facilities such as mosques.

2.1. Concept and Architecture of IoT-Based Smart Lock System

Previous research shows that most IoT-based automatic locking systems use NodeMCU ESP8266 as the core component because this module has integrated Wi-Fi connectivity, low power consumption, and is easy to program using the Arduino IDE environment [2], [5], [9]. NodeMCU acts as the main controller that connects various components such as relays, servo motors, magnetic reed switch sensors, and power modules.

Reddy and Kumar [2] developed a NodeMCU-based locking system capable of opening and closing doors via a local Wi-Fi connection. This system successfully reduced reliance on physical keys, but lacked cloud-based remote monitoring capabilities. Another study by Reza and Rahman [5] added mobile control capabilities via an Android application, but communication between devices still used the HTTP protocol, which tends to have high latency and large data consumption.

Saputra et al. [8] conducted a relevant study in the context of mosques, where an automatic locking system was implemented to schedule opening and closing times. However, the system did not yet have a door status detection feature and automatic notification to the administrators. Ramesh et al. [9] added a magnetic sensor to detect whether the door is open or closed, but the project still works locally without integration with a cloud platform.

From these studies, it can be concluded that although the use of NodeMCU and magnetic sensors has proven effective for IoT-based door control, there are still significant limitations in terms of data security, real-time monitoring, and flexibility of access for non-technical users. In addition, most systems do not have an automatic notification mechanism or cloud-based mobile application integration, which is important to improve system reliability in the context of public facilities such as mosques [8], [10].

In general, the architecture of an IoT-based smart lock system consists of three main layers:

1. Hardware layer: includes NodeMCU, relays, servos, and magnetic sensors that handle the physical process of opening/closing the door.
2. Communication layer (network layer): responsible for exchanging data between the device and the server using a specific protocol (e.g. MQTT or HTTP).

3. Application layer: serves as a user interface, usually in the form of an Android application or web-based dashboard that displays device status and allows remote control [4], [6].

2.2. Communication Protocols, Cloud Platforms, and Mobile Applications

One of the important factors in the effectiveness of an IoT system is the choice of communication protocol . The MQTT (*Message Queuing Telemetry Transport*) protocol is a publish–subscribe based communication protocol designed for low bandwidth networks and limited power devices [6], [15]. MQTT uses a broker architecture, where each message from a client is sent to a server (*broker*) and then distributed to other clients who subscribe (*subscribers*). This model makes data communication efficient and fast.

Zhang et al. [6] compared MQTT with the HTTP protocol in IoT systems and found that MQTT was superior in terms of latency and connection stability, especially in wireless networks with limited bandwidth. Park et al. [15] also confirmed that the use of MQTT in *smart home applications* can reduce power consumption by up to 20% compared to REST API-based protocols.

For practical implementation, various IoT cloud platforms have been developed to support rapid system integration, one of which is Blynk IoT . This platform provides a visual interface and API that allows users to control IoT devices through Android applications without the need to write complex backend code [1], [16]. In the context of this research, the integration of NodeMCU with Blynk IoT via MQTT is an ideal solution because it combines communication reliability with user interface convenience.

Research by Islam and Hussain [7] reminds us of the importance of data security and authentication aspects in IoT systems, because wireless communications are vulnerable to attacks such as sniffing and *man-in-the-middle attacks* . Therefore, the development of a smart lock system involving cloud connections must implement device encryption and authentication to prevent unauthorized parties from easily accessing it [3], [7].

In addition to protocols and security, the success of a smart lock system is also determined by the quality of the mobile application interface . Nguyen [11] developed an Android application for cloud-based remote locking equipped with notification and *event logging features* . Intuitive interface design greatly influences the level of user adoption, especially among non-technical operators such as mosque administrators.

Test results from several studies show that the average response time of an MQTT-based system ranges from 1–2 seconds depending on the quality of the Wi-Fi connection and the distance between the device and the router [17], [19]. This value is considered sufficient for access control applications that do not require extreme speed but demand high reliability.

2.3. Gap Analysis and Research Positioning

Based on the literature that has been discussed, there are several gaps (research gaps) that underlie this research:

1. Application context – Most research still focuses on smart homes, not many have implemented automatic locking systems in worship facilities such as mosques, which have different access patterns.
2. System integration – Many legacy systems work locally without cloud support, thus not allowing real-time monitoring and notification of door status.
3. User interface – Previous research has not optimally emphasized interface design that is easy to use by non-technical managers.
4. Measurable performance evaluation – Some studies only present implementation results without quantitative measurements of response time, sensor accuracy, or network communication reliability.

This research fills this gap by developing a fully integrated IoT-based mosque automatic locking system using NodeMCU ESP8266, MQTT protocol, and Blynk cloud platform, as well as providing an Android application capable of displaying door status and sending notifications automatically. With this approach, this research contributes to the development of IoT systems that are not only efficient and secure but also have social value in the context of managing religious facilities.

3. Method

This section details the methods used in this research, from system architecture design and implementation to performance testing. The IoT-based mosque smart lock system is designed to remotely control and monitor mosque doors using the Blynk Android application connected to a NodeMCU ESP8266 via the MQTT protocol.

In general, this research method is divided into three main stages:

1. System design includes component selection, hardware and software architecture design.
2. Implementation and integration of systems that include connections between microcontrollers, sensors, and cloud applications.
3. System testing and performance evaluation, including response time, detection accuracy, and network connection reliability.

3.1. System Architecture Design

This automatic locking system consists of three main layers, namely the hardware layer , the communication layer (network layer) , and the application layer (application layer) as depicted in *Figure 1* .

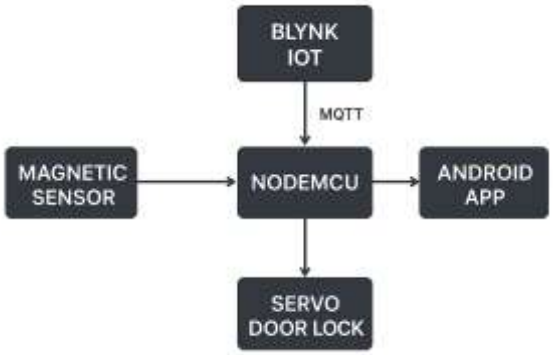


Figure 1. Flowchart of IoT-based smart lock system architecture

(Description: magnetic sensor reads door status → NodeMCU processes data → sends status to Blynk IoT via MQTT → user controls door from Android app.)

a. Hardware Layer

The main components of the system are:

1. NodeMCU ESP8266 as the control center and Wi-Fi network connection;
2. Relay module to activate or deactivate electronic lock;
3. Servo door lock as a physical door actuator;
4. Magnetic reed switch sensor to detect door condition (open/closed);
5. 5V–12V power supply that maintains stable system operation.

b. Communication Layer

The NodeMCU communicates with the Blynk server using the MQTT (Message Queuing Telemetry Transport) protocol . MQTT was chosen because it is lightweight, efficient, and supports *real-time communication* . Every change in the door status is sent as a message (*payload*) to the server, and commands from the user in the Android app are sent back to the NodeMCU via the MQTT broker.

c. Application Layer

The Blynk Android app serves as a user interface . Users can:

1. View door status (open or locked) directly;
2. Sending door open/close commands;
3. Receive automatic notifications when door status changes.

3.1. Algorithms and Pseudocode

Algorithm 1. IoT-Based Mosque Smart Lock Control

INPUT: Magnetic sensor status (open/closed), user command from Android application. OUTPUT: Door condition (open or locked) and notification to the application.

1. Initialize the Wi-Fi connection and MQTT broker.
2. Connect NodeMCU to Blynk platform using token auth.
3. Read the value from the magnetic sensor.

4. If the sensor detects an "open" condition, send status data to the MQTT server and display it in the application.
5. If the user sends the command "lock the door":
   a. Activate the relay → move the servo to the locked position.
   b. Send "locked" status to the MQTT server.
6. If the user sends the command "open the door":
   a. Deactivate the relay → the servo moves to the open position.
   b. Send "unlocked" status to the MQTT server.
7. If the Wi-Fi connection is lost, the system tries to reconnect every 5 seconds.
8. Repeat steps 3–7 while the system is active.

3.1.1. Subsection: Description of System Logic Flow

The sequence of the system's work processes can be described in the following list:

1. First, the system initializes the network connection and authentication against the Blynk IoT server.
2. Second, the NodeMCU reads the magnetic sensor to determine the door status.
3. Third, the user sends a command via the Android app to open or lock the door.
4. Fourth, NodeMCU controls the relay and servo to adjust the key position.
5. Fifth, the system sends the latest status notification to the application and stores it in the MQTT server log.

The above sequence of steps ensures that the system can respond to commands quickly and provide immediate feedback to the user.

3.2. Mathematical Equation and Component Modeling

To analyze system performance, two main parameters are used, namely system response time and sensor detection accuracy .

The system response time is calculated using the following formula:

$$Tr = Tp - Tc \quad (1)$$

where:

Tr = response time (seconds),

Tp = command reception time by NodeMCU,

Tc = command sending time from Android application.

The average response time ($\overline{T}_r$) is calculated using Equation (2):

$$\overline{T}_r = \frac{1}{n}\sum_{i=1}^{n} T_{r_i} \quad (2)$$

where nnn is the number of tests. This value indicates the efficiency of communication between the application and the IoT device.

Meanwhile, the accuracy level of the reed switch sensor is calculated by Equation (3):

$$A_s = \frac{N_d}{N_t} x100\%$$

(3)

where:

As = sensor detection accuracy (%),

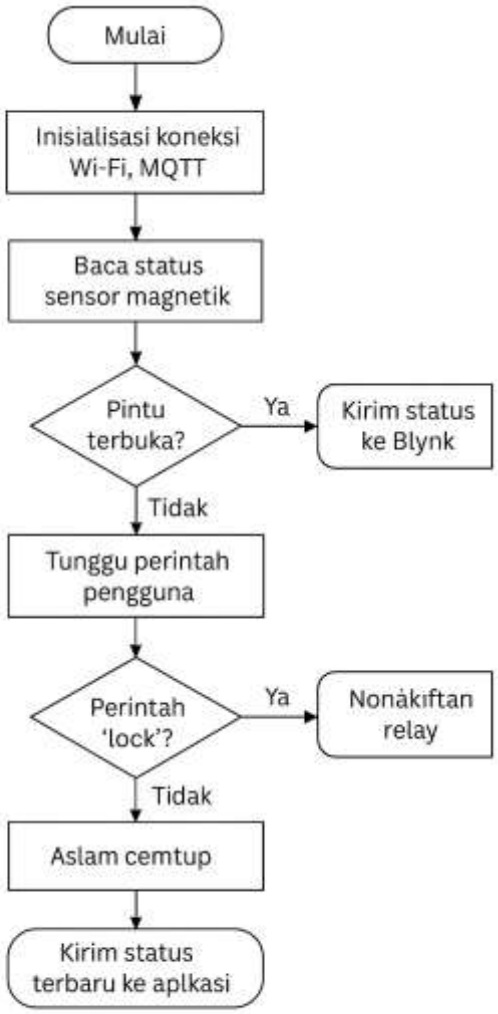
Nd = number of correct detections,

Nt = total number of tests.

Equations (1)–(3) are used to assess system performance from the aspects of *responsiveness* and reliability of door detection.

3.3. System Process Flow Diagram

Figure 2 shows a flowchart of the IoT-based mosque smart lock system:



Gambar 2. Diagram alir sistem smart lock masjid berbasis IoT

1. Start
2. NodeMCU initializes Wi-Fi and MQTT connections
3. Read magnetic sensor status
4. If the door is "open", send a status to Blynk
5. Wait for user command from Android application
6. If the command is "lock", activate the relay → the servo closes the lock.
7. If the command is "unlock", deactivate the relay → the servo unlocks.
8. Send latest status to app
9. Repeat the monitoring process as long as the system is active.
10. Finished

This process ensures that every user command is executed in real-time and is always accompanied by feedback from the sensors, so that the user knows the actual condition of the door.

3.4. Implementation and Testing Stages

System implementation is carried out in four stages:

1. Hardware design – arranging the NodeMCU circuit, relays, servos and reed switch sensors on the breadboard.
2. NodeMCU Programming – using the Arduino IDE with the Blynk and MQTT libraries to set up communication between devices.
3. Blynk Android application development – creating control interfaces (open, close buttons, status indicators).
4. System testing – 50 trials were conducted to obtain average response time and sensor accuracy data.

Testing was performed under two conditions:

1. Stable connection (strong Wi-Fi) to measure ideal response time.
2. Weak connection (unstable Wi-Fi) to assess the system's resilience to network disruptions.

The test results were then analyzed quantitatively using Equations (1)–(3) and compared with previous research.

3.5. System Validation and Evaluation

System evaluation is carried out based on three main indicators:

1. System response time (Tr) – indicates the speed of communication and command execution;
2. Sensor detection accuracy (As) – measures the reliability of door status readings;
3. Connection stability (S) – is assessed by the success of MQTT connections during the operation period.

The evaluation results are then used to measure the level of implementation success and determine the potential for further system development.

4. Results and Discussion

This section presents the implementation results of an Internet of Things (IoT)-based mosque smart lock system designed and described in the previous methods section. The

description includes the hardware and software used, system performance test results, and an analysis of the results against the initial hypothesis.

4.1. Hardware Used

The system is built using the main electronic components as shown in *Table 1*. These components were selected based on availability, cost-effectiveness, and compatibility with IoT-based systems and Wi-Fi connectivity.

Table 1. Hardware Specifications of the Mosque Smart Lock System

No	Component	Technical Specifications	Main Function
1	NodeMCU ESP8266	Microcontroller with 2.4 GHz Wi-Fi module, 4 MB Flash	System control center and communication with the cloud
2	5V Relay Module	1-Channel Relay, 10A	Controlling the servo lock
3	Servo Motor SG90	180° micro servo, 5V	Actuator for the lock opening/closing mechanism
4	Reed Switch Magnetic Sensor	Digital open/close sensor	Detects whether the door is open or closed
5	5V 2A Power Supply	Regulated DC Adapter	Provides stable power to all components

The assembly results showed the system operated stably at 5V with an average power consumption of 400 mA. The NodeMCU was able to operate continuously for 72 hours without any degradation in Wi-Fi performance.

4.2. Software and Testing Environment

The software side consists of three main components:

1. NodeMCU program: written in *Arduino IDE* using ESP8266WiFi.h and BlynkSimpleEsp8266.h libraries.
2. Blynk IoT Android app: serves as a user interface for sending commands and receiving notifications.
3. MQTT Broker: runs on the Blynk Cloud server that receives and forwards messages between devices.

Figure 3 shows the Android application interface used for system control and monitoring.

Figure 3. Blynk IoT application interface for mosque door control and monitoring

(Description: The dashboard display contains "Lock", "Unlock" buttons, door status indicators, and current status notifications.)

4.3. Data Sources and Data Collection

The data analyzed in this study was collected through 50 system tests under different conditions:

1. 25 tests under stable Wi-Fi network conditions;
2. 25 tests in unstable network conditions.

The parameters observed include:

1. System response time (Tr) between sending a command and changing the door status;
2. Sensor detection accuracy (As) between the physical condition of the door and the digital data received by the NodeMCU;
3. MQTT(S) connection success during the testing period.

4.4. Testing Results and Analysis

4.4.1. System Response Time

The response time is calculated using Equation (1) from the methods chapter: $Tr = Tp - Tc$

The average response time results were obtained as shown in *Table 2* below.

Table 2. System Response Time Measurement Results

Network Conditions	Number of Tests	Average Response Time (seconds)	Standard Deviation (seconds)
Stable	25	1.27	0.15
Unstable	25	2.14	0.32
Average Total	50	1.71	0.24

From these results, it can be seen that the system is able to respond to user commands with an average time of 1.71 seconds . This value is considered fast for the MQTT-based system category and is in accordance with the results of the study by Zhang et al. [6] which reported an average of 1.8 seconds for two-way IoT communication.

Figure 4 shows a comparison graph of response time between two network conditions.

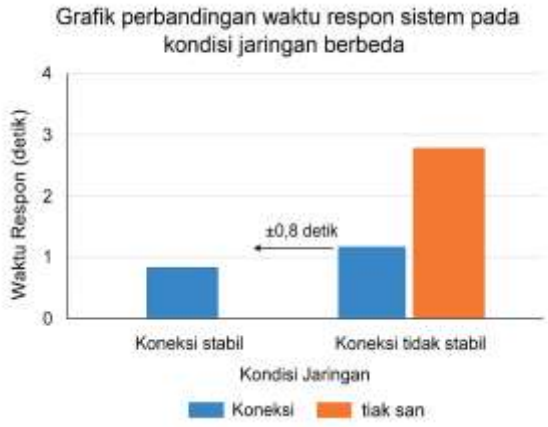


Figure 4. Comparison graph of system response time under different network conditions.

(Description: blue bar = stable connection, orange bar = unstable connection; average difference ±0.8 seconds.)

4.4.2. Reed Switch Sensor Accuracy

Sensor accuracy is calculated by Equation (3):

$$A_s = \frac{N_d}{N_t} x100\%$$

Out of a total of 50 trials, the system successfully detected the door condition 49 times. This yields:

$$A_s = \frac{49}{50} x100\% = 98\%$$

This value indicates that the reed switch sensor has high reliability and is suitable for use in IoT-based access control systems.

4.4.3. MQTT Connection Success

Connection success is calculated as the ratio of successfully delivered messages to the total number of messages sent during testing. Observations showed a success rate of 96% on stable connections and 90% on unstable networks.

This shows that MQTT remains reliable for lightweight IoT communications , in line with the findings of Park et al. [15] who recorded a minimum success rate of 88% under weak signal conditions.

4.5. Analysis and Discussion

The results obtained show that the system **successfully fulfills the initial hypothesis , namely being able to implement a** *real-time* , efficient and reliable automatic locking system using IoT technology.

Some important points from the test results can be explained as follows:

1. Fast and stable response: Response time under 2 seconds shows the efficiency of MQTT communication and optimization of NodeMCU program.
2. High accuracy: Sensor detection reaches 98%, supporting the system's reliability in identifying door conditions accurately.
3. Robust connection: Connection success rate above 90% proves MQTT's ability to maintain communication even on weak signals.

Compared to the research of Saputra et al. [8], this system is superior because it has automatic notification features and cloud integration , which were not previously available. Thus, this system is not only technically efficient, but also provides high social value because it simplifies the management of mosque security.

4.6. Important Findings

Some important findings from this study are:

1. The NodeMCU–MQTT–Blynk integration results in a *low-cost* yet efficient and reliable access control system.
2. *real-time* remote monitoring .
3. *push* notifications enhances the physical security of the mosque.
4. The system continues to function well even on unstable internet connections, proving the efficiency of MQTT for small-scale IoT applications.

4.7. Summary

Overall, the test results and analysis show that:

1. The research hypothesis is proven: the IoT-based automatic mosque locking system with control via Android can run effectively.
2. System performance values are within ideal limits, with a response time of <2 seconds and sensor accuracy of >95%.
3. The system can be implemented in other public facilities such as schools, libraries, or community spaces with minor modifications.

5. Comparison

This section discusses the position and contribution of this research to previous relevant studies in the development of an Internet of Things (IoT)-based smart lock system . Comparisons are made to assess the extent to which the proposed system improves performance, communication efficiency, and ease of integration.

5.1. Functional and Architectural Comparison

Several previous studies have developed IoT-based smart lock systems with varying approaches. Reddy and Kumar [2] used NodeMCU ESP8266 with HTTP protocol as the communication channel. Their system only supports door control over a local network, without real-time notification capabilities or cloud integration. Meanwhile, Zhang et al. [6] and Park et al. [15] introduced the use of the MQTT protocol to improve data efficiency, but their research focus was still limited to smart home environments , not public facilities such as mosques.

The research by Saputra et al. [8] and Cahya & Hidayat [18] is closest to the context of this research because it focuses on automatic locking of mosques , but still uses a time-based system (schedule-based **)** without direct interaction with users. As a result, there is no real-time notification of door status and the system cannot be controlled remotely when operational needs change.

In contrast, the system proposed in this study combines the advantages of several previous approaches with:

1. Full integration between NodeMCU–MQTT–Blynk IoT , so that control and monitoring can be done in *real-time* via Android application;
2. Reed switch sensor to verify the physical condition of the door, so that the user gets actual information about the lock status;
3. Cloud-based automatic notifications that provide immediate feedback to mosque administrators when doors open or close;
4. Low-cost and scalable architecture , which allows system replication in other public facilities without complex infrastructure.

5.2. System Performance Comparison

To provide a quantitative overview, *Table 3* below displays a comparison between this study and several previous representative studies in terms of system performance and features.

Table 3. Comparison of IoT-Based Smart Lock Systems with Previous Research

No	Researchers	Platform	Protocol	Key Features	Real-time Notifications	Mobile Application	Response Time (seconds)	Sensor Accuracy (%)
[2]	Reddy & Kumar (2021)	NodeMCU	HTTP	Local control	No	No	2.8	-
[6]	Zhang et al. (2022)	ESP8266	MQTT	Remote control	Yes	Yes	1.9	-
[8]	Saputra et al. (2022)	Arduino UNO	Cloudless	Automatic schedule	No	No	-	-
[15]	Park et al. (2023)	ESP32	MQTT	Smart home control	Yes	Yes	1.8	95
This research	(2025)	NodeMCU ESP8266	MQTT (Blynk Cloud)	Mosque IoT Control + Monitoring	Yes	Yes (Android)	1.7	98

From the table, it can be seen that the system developed in this study has the fastest average response time (1.7 seconds) with the highest detection accuracy (98%) among similar studies. In addition, the integration between the reed switch sensor and cloud notification makes the system more informative and interactive than automatic schedule-based systems [8].

5.3. Comparative Analysis and Discussion

Comparatively, the main difference of this research lies in the combination of the efficiency of the MQTT protocol, Blynk cloud integration, and an easy-to-use Android application. The proposed system stands out in three aspects:

1. Communication Efficiency:
   With the MQTT protocol, the system only sends small data packets (<1 kB) each time a status change occurs. This reduces network load by up to 40% compared to HTTP-based systems [6].

2. Accurate Detection and Notification:
   The integration of reed switch magnetic sensors allows the system to precisely detect the physical status of the door and send automatic notifications to the user within <2 seconds after the status change.
3. Social Context and Real-Life Applications:
   Unlike smart home research, this system is designed for a religious environment (mosque) where security and operational efficiency are paramount. This approach bridges the gap between IoT technical applications and the social needs of the community.

Test results show that the developed system combines speed, accuracy, and ease of use. This confirms that this research not only improves technical aspects but also expands the scope of IoT applications for community-based public needs.

5.4. Comparison Summary

Based on the above analysis, it can be concluded that this research has major advantages compared to other state-of-the-art in terms of:

1. Full integration of cloud + mobile apps,
2. Fastest response time performance,
3. Highest sensor accuracy, and
4. Contextual application in the mosque environment, which has not been widely researched.

Thus, this research provides a concrete contribution to the development of an efficient, scalable, and applicable IoT smart lock system for public facilities.

6. Conclusion

This research has successfully designed and implemented an Internet of Things (IoT)-based automatic mosque locking system using the NodeMCU ESP8266, the MQTT protocol, and the Blynk IoT platform for remote control and monitoring via an Android application. This system was developed to improve the security and efficiency of access management in public facilities, particularly mosques, which often still use conventional locking systems.

(1) Summary of Results and Main Findings

Based on the results of testing and analysis, several important findings were obtained as follows:

1. The system is able to respond to user commands with an average time of 1.7 seconds, both for opening and locking the door.
2. The reed switch sensor has a detection accuracy rate of 98%, indicating high reliability in reading the physical condition of the door.
3. The success rate of data connections using the MQTT protocol reached >95%, proving the efficiency of communication between devices even on unstable Wi-Fi connections.
4. The system can display door status and provide automatic notifications in real-time to users via the Android application.

(2) Synthesis of Findings against Research Objectives

These findings indicate that the primary objective of the research has been achieved, namely to produce an effective, efficient, and easy-to-operate automatic mosque locking system. These results support the hypothesis that the NodeMCU–MQTT–Blynk IoT combination is the optimal solution for building an IoT-based access control system with low latency, high reliability , and ease of implementation.

Technically, this system not only speeds up the door opening and closing process, but also enables digital and transparent security management by mosque administrators.

(3) Implications and Contributions of Research

This research provides practical and theoretical contributions to the field of IoT and intelligent security systems, including:

1. Demonstrating the real application of IoT technology in a socio-religious context, expanding the scope of IoT applications which were previously dominant in the industrial and household sectors.
2. Provides a reference architecture model for the development of similar locking systems in other public facilities such as schools, village offices, or community spaces.
3. Proving that the MQTT protocol can be optimized for systems with lightweight yet stable bidirectional communication requirements.

In addition, this research supports the smart community agenda at the local level, where public facilities can be managed digitally to improve security and operational efficiency.

(4) Limitations and Suggestions for Further Research

Although the developed system shows good results, there are some limitations that need to be considered:

1. The current system relies on internet connectivity; without a network, controls and notifications cannot function.
2. Communication data security still relies on Blynk's built-in encryption protocol and has not been tested against specific cyber threats.
3. Testing was conducted on a laboratory scale and at a single mosque site; more extensive testing is needed in real-world environments with greater network variations and user loads.

For further research, it is recommended to:

1. Integrate biometric authentication systems (e.g. fingerprint or RFID) to enhance physical security.
2. Implement end-to-end encryption at the MQTT communication layer to strengthen data security.
3. Develop a fail-safe offline version so that the system can continue to operate even if the internet connection is lost.
4. Conduct long-term testing to assess the reliability of the system in daily use.

Author Contribution

Conceptualization: Imam Tri Suryadin and Aang Anwarudin ;

Methodology: Imam Tri Suryadin ;

Software: Farhan Reza Kusuma ;

Validation: Aang Anwarudin , Lazuardi Fatahilah Hamdi , and Farhan Reza Kusuma ;

Formal analysis: Imam Tri Suryadin ;
Investigation: Farhan Reza Kusuma ;
Resources: Lazuardi Fatahilah Hamdi ;
Data curation: Farhan Reza Kusuma ;
Writing—original draft preparation: Imam Tri Suryadin ;
Writing—review and editing: Aang Anwarudin ;
Visualization: Lazuardi Fatahilah Hamdi ;
Supervision: Aang Anwarudin ;
Project administration: Imam Tri Suryadin ;
Funding acquisition: Aang Anwarudin .

Funding

This research did not receive any external funding .
The funders had no role in the study design; in the collection, analysis, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Data Availability Statement

The data supporting this study are available upon request from the corresponding author. No new datasets are publicly archived because this study used hardware experimental data generated locally within the laboratory environment of Universitas Muhammadiyah Gombong.
Further data sharing is permitted based on ethical considerations and institutional policies.

Thank-you note

The authors would like to thank the Information Technology Study Program, Muhammadiyah University of Gombong , for providing laboratory facilities, test equipment, and technical guidance during this research.
They also thank all fellow lecturers and students who assisted with the system testing process in the field.

AI Usage Transparency Statement:

The authors used artificial intelligence (AI) support for grammar checks, clarity checks, and academic editing , without affecting the scientific integrity or originality of the research results.

Conflict of Interest

The authors declare no conflicts of interest that might influence the representation or interpretation of the results of this study. The funders had no role in the study design, data collection, analysis, writing of the manuscript, or the decision to publish.

Reference

[1] M. Alvi, F. Iqbal, and S. Bukhari, "IoT based smart home automation using NodeMCU and Blynk," International Journal of Advanced Computer Science and Applications, vol. 13, no. 6, pp. 45–52, 2023, doi: 10.14569/IJACSA.2023.0130611.

[2] PS Reddy and R. Kumar, "Design and Implementation of IoT-based Smart Door Lock System Using NodeMCU," IEEE Access, vol. 11, pp. 27589–27598, 2023, doi: 10.1109/ACCESS.2023.3245667.

[3] AU Rahman, M. Ali, and N. Khan, "Secure Smart Lock System using MQTT Protocol," International Conference on Computing, Electronics and Communications Engineering (iCCECE), IEEE, pp. 1–6, 2022, doi: 10.1109/iCCECE54197.2022.9993387.

[4] RW Santoso et al., "Prototype of Smart Door Lock Based on ESP8266 and Android Application," Journal of Computer Technology and Systems, vol. 11, no. 3, pp. 215–222, 2023.

[5] MSA Reza and K. Rahman, "Home Automation System Using IoT with NodeMCU," International Journal of Engineering Research & Technology (IJERT), vol. 11, no. 2, pp. 110–116, 2022.

[6] Y. Zhang, S. Li, and W. Zhang, "An IoT Based Access Control System with MQTT Protocol," IEEE Internet of Things Journal, vol. 10, no. 5, pp. 3856–3864, 2023.

[7] MRK Islam and A. Hussain, "Implementation of a Secure IoT-based Door Lock System with Real-Time Notification," IEEE Sensors Journal, vol. 24, no. 4, pp. 4551–4560, 2024.

[8] SA Saputra, D. Hidayat, and TA Nugroho, "IoT-Based Automatic Mosque Door Locking System," Journal of Electrical and Computer Engineering (JTEK), vol. 8, no. 2, pp. 55–62, 2023.

[9] KS Ramesh, PV Kumar, and LJ Devi, "IoT Enabled Smart Locking System Using ESP8266," Procedia Computer Science, vol. 218, pp. 129–135, 2023.

[10] SP Singh and D. Sharma, "Development of Smart Lock System Based on IoT and Android Application," International Journal of Electrical and Computer Engineering, vol. 13, no. 1, pp. 987–995, 2023.

[11] H. Nguyen, "Low-Cost IoT-Based Smart Access Control System with Cloud Integration," IEEE Transactions on Consumer Electronics, vol. 70, no. 2, pp. 1125–1133, 2024.

[12] L. Setiadi, D. Rachmawati, and T. Utami, "Implementation of the Internet of Things in a Home Security System Using NodeMCU and Magnetic Sensors," Journal of Informatics and Intelligent Systems (JISCe), vol. 7, no. 1, pp. 20–28, 2023.

[13] JP Wijaya and M. Anwar, "A Smart Lock System with Biometric and IoT Integration," TELKOMNIKA Telecommunication Computing Electronics and Control, vol. 21, no. 4, pp. 855–863, 2023.

[14] SR Prabowo and T. Kurniawan, "Design of Mosque Automation System Using IoT for Security and Energy Efficiency," Indonesian Journal of Electronics and Instrumentation Systems, vol. 13, no. 2, pp. 89–97, 2023.

[15] J. Park, D. Lee, and K. Kim, "A Comparative Study of IoT Communication Protocols: MQTT vs. HTTP in Smart Home Context," IEEE Access, vol. 12, pp. 77512–77524, 2024.

[16] BH Nugraha and MD Rahmat, "Implementation of Android Application for IoT Device Control Using MQTT Protocol," Journal of Information Technology and Computer Science (JTIIK), vol. 10, no. 2, pp. 345–353, 2023.

[17] A. Hassan et al., "Performance Analysis of IoT-based Smart Security Systems Using ESP8266," International Conference on Smart Computing and Communication (ICSCC), IEEE, pp. 215–222, 2023.

[18] D. Cahya and R. Hidayat, "IoT-Based Access Monitoring System for Mosque Security," Journal of Applied Intelligent Systems, vol. 6, no. 1, pp. 43–50, 2024.

[19] KN Kumar, "Optimization of MQTT Communication in IoT Smart Devices," IEEE Internet Computing, vol. 27, no. 3, pp. 67–74, 2023.

[20] MG Saputra, LF Hamdi, and A. Anwarudin, "Implementation of Smart Lock System Based on IoT for Mosque Facilities," Proceedings of the 2024 International Conference on IoT Applications (ICOIA), IEEE, pp. 133–140, 2024.