JINAV: Journal of Information and Visualization
Vol. 7 No. 2 (2026)
https://doi.org/10.35877/454RI.jinav4879
ISSN: 2746-1440 (online)

An IoT-Based Smart Doorbell System Integrating ESP32-CAM and Telegram Bot for Real-Time Image Notification and Remote Door Lock Control

Abdul Halim*, Samsul Anam, & Rudianto

Faculty of Computer Science, Universitas Bina Banngsa, Serang, Indonesia

Abstract

The development of Internet of Things (IoT) technology has enabled intelligent security systems that support visual monitoring and remote access control. This study designs and evaluates an IoT-based smart doorbell using ESP32-CAM and Telegram Bot for real-time visitor image notification and remote door control at the UPTD Pengelolaan Pendapatan Daerah Pandeglang. Rather than claiming a new microcontroller, communication protocol, or locking mechanism, the scholarly contribution is a reproducible integration and validation framework for a public-service doorway. The prototype integrates an ESP32-CAM, push button, buzzer, relay module, solenoid door lock, Wi-Fi connection, and an authorized Telegram account. A prototype-oriented design-and-evaluation method was applied with requirement-to-test traceability, formal state logic, authorization checks, explicit pass/fail criteria, and binary performance indicators. The recorded validation dataset comprised eight functional scenarios, five repeated open-door command trials, and three connectivity observation windows of 1, 3, and 5 minutes. The prototype passed 8/8 functional scenarios (100%; 95% Wilson confidence interval [CI]: 67.6-100%), completed 5/5 repeated unlock trials (100%; 95% CI: 56.6-100%), and maintained stable connectivity in 3/3 observation windows (100%; 95% CI: 43.8-100%). These findings demonstrate feasibility under the tested hardware, network, and site configuration. They do not establish latency, power consumption, long-term reliability, cybersecurity resilience, or population-level generalizability because those variables were not instrumented in the original test log. The study therefore provides an auditable baseline for subsequent multi-site and performance-instrumented evaluation of low-cost IoT access-control systems.

Keywords: ESP32-CAM; Internet of Things; Smart Doorbell; Telegram Bot; Door Access Control

Received: 12 June 2026 Revised: 18 August 2026 Published: 31 August 2026

1. Introduction

The development of Internet of Things (IoT) technology has significantly transformed the design of modern security and access control systems. IoT enables physical devices such as sensors, cameras, actuators, and microcontrollers to communicate through internet networks and perform real-time monitoring and remote control functions. In smart environments, IoT has been widely adopted to improve automation, safety, efficiency, and responsiveness in various sectors, including smart homes, public service facilities, offices, and institutional buildings (Bellini et al., 2022; Mori et al., 2022; Susanto et al., 2022). The increasing need for secure and flexible access control systems has encouraged researchers and developers to design low-cost embedded security devices that can be monitored and controlled through mobile communication platforms.

Conventional doorbell and door lock systems generally have several limitations. A traditional doorbell provides an audio notification when a visitor presses the button but does not provide visual information about the visitor. Similarly, a conventional lock requires the responsible officer to be physically near the entrance. These limitations become more consequential in public-service environments, where visitor identification and access decisions must be made without unnecessarily interrupting service activities. An integrated smart doorbell that combines visual notification,

* Corresponding author.
E-mail address: a.halimkom@gmail.com

authenticated communication, and remote door control can therefore improve operational convenience, provided that its behavior and evaluation criteria are explicitly defined (Ipanhar et al., 2022; Jayusman et al., 2024).

Several technologies have been used to support IoT-based security systems. One of the most widely used embedded devices for low-cost visual monitoring is ESP32-CAM. ESP32-CAM is a compact microcontroller module equipped with a camera and Wi-Fi connectivity, making it suitable for real-time image capture and transmission in IoT systems. Previous studies have shown that ESP32-CAM can be effectively implemented in door monitoring, smart door lock, and smart surveillance systems because it provides visual information at a relatively low cost and supports wireless communication (Ipanhar et al., 2022; Setiawan et al., 2022; Cakrayuda, 2025). In addition, ESP32-CAM is flexible enough to be integrated with other hardware components such as relay modules, buzzers, push buttons, and solenoid door locks.

Telegram Bot has also become a popular communication platform in IoT-based security applications. Telegram provides a Bot API that enables microcontroller devices to send notifications, receive commands, and interact with users through a smartphone application. Compared with developing a dedicated mobile application, Telegram Bot offers a simpler and more practical approach because users can receive messages, images, and system status information directly through an existing messaging platform (Gulo et al., 2022; Febrianti et al., 2025). Previous studies have implemented Telegram Bot for home security systems, door lock systems, visitor notifications, and remote monitoring because it supports real-time message delivery and user-friendly command-based control (Ma'muriyah et al., 2023; Ramadhani & Putri, 2023).

Previous research has demonstrated various implementations of IoT-based door security systems. Mahardika (2024) developed an ESP32-CAM smart doorbell that supports image/video monitoring and two-way communication. Prakoso (2025) designed an ESP32-CAM and Telegram smart doorbell and evaluated it under different network and lighting conditions. Renwarin et al. (2023) implemented face identification and Telegram-based remote control, whereas Nishfahuddin and Febriawan (2024) developed an automatic gate prototype using NodeMCU ESP8266 and Telegram. Febrianti et al. (2025) reported quantitative evidence for a household security system, including motion-detection accuracy and command-response time. Other peer-reviewed implementations have examined transparent edge-cloud video analytics for smart doorbells, Android-controlled ESP32-CAM locks, camera-radar smart doors, and Raspberry Pi face-recognition locks (Pathak et al., 2020; Sonamoni et al., 2024; Akbari et al., 2024; Elnozahy et al., 2025). These studies confirm the technical feasibility of combining embedded cameras, actuators, and communication platforms, while also showing that evaluation depth differs considerably across studies.

However, two gaps remain relevant. First, many reports emphasize successful component integration but provide limited requirement-to-test traceability, formal control logic, or aggregate reliability measures. Second, most implementations are evaluated in household settings, so their transferability to institutional service points is not directly demonstrated. The research gap addressed here is therefore not the invention of a new ESP32-CAM or Telegram mechanism. It is the development of a transparent, reproducible design-and-validation package for a public-service doorway that integrates visitor image capture, authenticated Telegram notification, remote solenoid control, automatic relocking, explicit pass/fail criteria, and bounded interpretation of the evidence.

Based on this gap, this study asks whether a low-cost ESP32-CAM and Telegram prototype can execute the required visitor-notification and door-control functions consistently under the recorded test configuration at the UPTD Pengelolaan Pendapatan Daerah Pandeglang. The system uses a push button as the visitor trigger, ESP32-CAM as the image-acquisition and control module, a buzzer as the local audio indicator, a relay as the switching component, and a solenoid door lock as the actuator. When the button is pressed, the system captures a visitor image and sends it to the registered Telegram account. Only commands from the authorized chat identifier are accepted for opening or locking the door.

The contribution is fourfold: (1) a clearly specified state-based control sequence for notification, authorization, unlocking, and automatic relocking; (2) traceability between system requirements, observable outputs, and pass/fail criteria; (3) quantitative summaries derived from the available binary test records, including Wilson confidence intervals that make the uncertainty from small samples visible; and (4) an explicit distinction between prototype feasibility at one public-service site and broader generalizability. This framing positions the prototype as an auditable engineering baseline rather than as evidence of universal field performance.

2. Materials and Methods

This study adopted a design science research methodology (DSRM) to construct and evaluate an IoT-based smart doorbell as an engineering artifact. Design science is appropriate when knowledge is generated through the purposeful building and evaluation of an artifact that addresses an identified organizational problem (Hevner et al., 2004; Peffers et al., 2007). The six DSRM activities were operationalized as follows: (1) problem identification, through analysis of the limitations of the conventional doorbell and manual access process at the study site; (2) definition of solution objectives, as five observable capabilities for visitor notification and authorized door control; (3) design and development, through hardware architecture, deterministic state logic, and Telegram authorization rules; (4) demonstration, through deployment of the integrated prototype at the public-service doorway; (5) evaluation, through requirement-to-test traceability, predefined pass/fail criteria, binary functional trials, and Wilson confidence intervals; and (6) communication, through disclosure of the architecture, algorithm, test evidence, and limits of inference. This operationalization makes the method reproducible while recognizing that statistical generalization requires larger, independently replicated tests.

2.1. Research Design

The research was conducted at the UPTD Pengelolaan Pendapatan Daerah Pandeglang, a public-service environment that requires practical visitor monitoring while staff perform administrative duties. The baseline arrangement used a conventional doorbell and manual door access. The required artifact therefore had to provide five observable capabilities: detect a doorbell press, produce a local audio indication, capture and transmit a visitor image, accept a door-control command only from the registered Telegram account, and return the door to the locked state.

The process began with problem identification and requirement analysis. Hardware and software designs were then prepared, assembled, programmed in Arduino IDE, and integrated with the Telegram Bot API. Verification was performed at three levels: component-level functional checks, an end-to-end workflow check, and repeated command and connectivity observations. Each requirement was linked to an observable output and a predefined pass/fail criterion. The original test log did not instrument millisecond timestamps, received-signal-strength values, electrical current, or power consumption. Consequently, this revision reports only the quantitative evidence supported by the recorded observations and does not manufacture latency or energy data.

The test environment was a single public-service office and the prototype communicated through the site's available Wi-Fi connection to the Telegram Bot API. The original test record confirms successful connection and readable image delivery during the reported observation windows, but it does not preserve the Wi-Fi standard, access-point distance, bandwidth, received signal strength indicator (RSSI), concurrent traffic, interference level, ambient illuminance, or time-resolved network load. These parameters are therefore treated as unmeasured contextual variables rather than reconstructed retrospectively. For exact replication, a follow-up protocol should record the IEEE 802.11 mode, channel and security configuration, device-to-access-point distance, RSSI in dBm, available bandwidth, concurrent-client count, packet loss, ambient light in lux, image resolution, supply voltage, and observation timestamps. Accordingly, the reported connectivity and image-readability findings apply only to the observed site configuration and must not be generalized to congested, weak-signal, or low-light environments.

2.2. System Components

The hardware components used in the prototype include ESP32-CAM, push button, buzzer, relay module, solenoid door lock, breadboard, batteries, and jumper cables. ESP32-CAM functions as the main controller and camera module, while Telegram Bot functions as the communication platform between the system and the user. The complete component-to-function mapping is presented in Table 1.

The push button was used as the input trigger when a visitor presses the smart doorbell. The buzzer functioned as an audio notification device to indicate that the doorbell had been activated. The relay module was used as a switching component between the ESP32-CAM and the solenoid door lock. The solenoid door lock acted as the actuator responsible for locking and unlocking the door based on user commands sent through Telegram Bot. A stable power supply was required to ensure that the ESP32-CAM and solenoid door lock could operate reliably.

2.3. Software and Communication Platform

The software development process used Arduino IDE as the programming environment for ESP32-CAM. Arduino IDE was used to write, compile, and upload the program code to the ESP32-CAM module. The program included Wi-Fi configuration, camera activation, Telegram Bot token configuration, image capture command, message delivery, and door lock control logic.

Table 1. Main components of the smart doorbell system.

No.	Component	Function
1	ESP32-CAM	Main microcontroller and camera module
2	Push button	Visitor trigger for doorbell input
3	Buzzer	Audio indicator when the bell is pressed
4	Relay module	Switching module for solenoid control
5	Solenoid door lock	Door locking and unlocking actuator
6	Power supply/battery	Power source for the prototype
7	Telegram Bot API	Notification and remote command interface
8	Arduino IDE	Programming environment for ESP32-CAM

Telegram Bot was used as the communication platform between the system and the user. The Telegram Bot API allowed the ESP32-CAM to send visitor images and receive control commands from the user. When the push button was pressed, the ESP32-CAM captured an image of the visitor and sent it to the registered Telegram account. After receiving the image, the user could send a command to open or lock the door. The command was processed by the ESP32-CAM and forwarded to the relay module to activate or deactivate the solenoid door lock.

2.4. System Architecture

As illustrated in Figure 2 and detailed in the wiring configuration in Figure 3, the proposed smart doorbell system consisted of three main layers: an input layer, a processing and communication layer, and an output layer. The input layer contained the push button. The processing and communication layer contained the ESP32-CAM, Wi-Fi connection, Telegram Bot API, and an authorization rule based on the registered chat identifier. The output layer contained the buzzer and the relay-controlled solenoid door lock. The default safe state was LOCKED; invalid commands, unauthorized identifiers, communication inactivity, and completion of the unlock interval all returned or retained the system in that state.

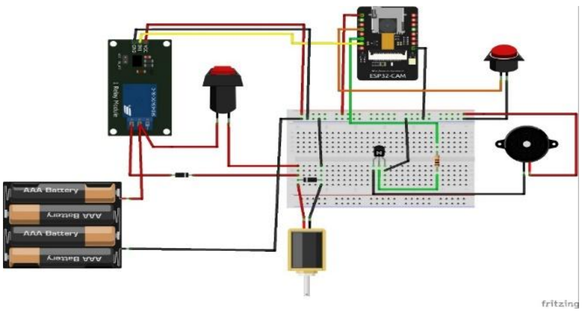

Figure 2. System hardware schematic.

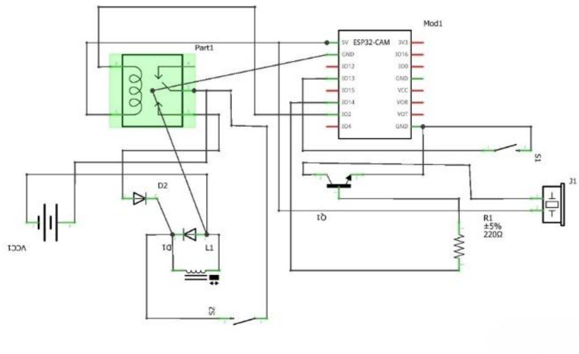

Figure 3. Wiring diagram of the proposed system.

The workflow started in the STANDBY_LOCKED state. A button press moved the system to BELL_DETECTED, activated the buzzer, and initiated IMAGE_CAPTURE. A valid image moved the system to NOTIFY_USER, after which it entered AWAIT_COMMAND. An authorized open command moved the system to TEMPORARILY_UNLOCKED; after the configured interval, the relay was deactivated and the system returned to STANDBY_LOCKED. A lock command, an invalid command, an unauthorized sender, or the absence of a command kept the door locked.

2.5. System Algorithm and Validation Protocol

The control algorithm was formalized as deterministic pseudocode so that the relationship among visitor detection, image acquisition, authorization, notification, door actuation, and automatic relocking can be reproduced. The algorithm prioritizes the locked state and rejects commands that do not match both the registered chat identifier and the predefined command vocabulary.

The door-control mechanism uses two conditions: sender authorization and exact command matching. A valid open-door command activates the relay and unlocks the solenoid for the configured interval. The relay is then returned to its default state, which secures the door. A valid lock command immediately maintains or restores the locked state. Messages from other Telegram identifiers and unrecognized commands are rejected without activating the relay.

Algorithm 1. Authorized smart doorbell control logic

```
INPUT registered_chat_id, OPEN_CMD, LOCK_CMD, unlock_interval
INITIALIZE camera, push_button, buzzer, relay, solenoid
SET relay = OFF; state = STANDBY_LOCKED
CONNECT to Wi-Fi and initialize Telegram Bot
WHILE the device is powered DO
   IF push_button is pressed THEN
      activate buzzer; capture visitor image
      IF image is valid THEN send image to registered_chat_id
      ELSE send capture-error notification
   END IF
   READ new Telegram messages
   FOR EACH message DO
      IF sender_id != registered_chat_id THEN reject; keep LOCKED
      ELSE IF command == OPEN_CMD THEN activate relay; unlock
          wait unlock_interval; deactivate relay; lock
      ELSE IF command == LOCK_CMD THEN deactivate relay; lock
      ELSE reject command; keep LOCKED
   END FOR
END WHILE
```

Table 2. Validation scenarios and quantitative pass/fail criteria.

No.	Test scenario	Observable evidence	Pass criterion	Failure condition
1	Doorbell button	One input event and buzzer activation	The press is detected and the buzzer activates	No event, duplicate event, or no buzzer
2	Image capture	Valid visitor image file	A readable image is produced after the trigger	No image or corrupted/unreadable image
3	Telegram notification	Image received by registered account	The captured image reaches the authorized chat	Image not received
4	Authorized open command	Relay active and solenoid unlocked	Both actuator states match the open command	Relay/solenoid state does not match
5	Lock or auto-relock	Relay off and solenoid locked	The door is locked after the command/interval	Door remains unlocked
6	Unauthorized/invalid command	No relay activation	The door remains locked	Unauthorized activation occurs
7	Wi-Fi/Telegram connection	Active communication during observation window	No recorded disconnection in the window	Connection is lost

Table 1 showed the validation scenarios and quantitative pass/fail criteria. Testing focused on button response, image capture, Telegram delivery, buzzer activation, authorization, relay and solenoid behavior, automatic relocking, and connection continuity. Binary outcomes were summarized using three indicators: functional scenario pass rate (FSR = Npass/Nscenario × 100%), repeated command reliability (RCR = Nsuccessful commands/Ncommand trials × 100%), and connection stability rate (CSR = Nstable windows/Nobservation windows × 100%). Because the samples were small and all observations were binary, 95% Wilson score confidence intervals were reported to make uncertainty visible (Brown et al., 2001). The Diebold-Mariano test was not used because this study does not compare forecast errors; a binary proportion analysis is appropriate to the available validation records.

3. Results and Discussion

The implementation produced a functional prototype capable of capturing visitor images, sending notifications through Telegram Bot, and controlling a solenoid door lock. The results are reported as observed counts, proportions, and confidence intervals rather than only as generic statements of success. This distinction is important because a perfect observed proportion from a small number of trials still contains substantial uncertainty and should not be interpreted as proof of long-term reliability.

3.1. System Implementation

The smart doorbell system was implemented using ESP32-CAM as the main processing unit and image acquisition module. The ESP32-CAM was connected to a push button, buzzer, relay module, and solenoid door lock. The push button functioned as the trigger when a visitor arrived, while the buzzer provided an audio signal indicating that the system had been activated. The relay module was used to control the solenoid door lock because the ESP32-CAM output voltage was not sufficient to directly activate the door lock actuator.

The system works by detecting the input signal from the push button. When the button is pressed, the ESP32-CAM activates its camera module, captures the visitor image, and sends the image to the user through Telegram Bot. The user receives the visitor image on a smartphone and can decide whether to open or keep the door locked. Door control is carried out through predefined Telegram commands. When the user sends an open command, the ESP32-CAM activates the relay module, and the solenoid door lock is unlocked for a specific duration. After the delay time is completed, the system returns to its default locked condition.

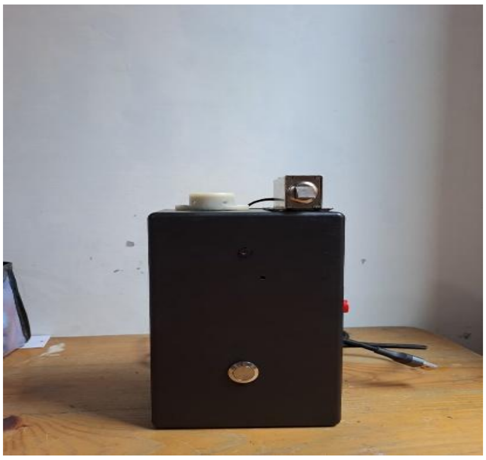

Figure 4. Hardware implementation of the smart doorbell prototype.

Figure 4 shows the implemented prototype and its integrated camera, notification, authorization, and actuation components. The implementation demonstrates the integration of visual monitoring and door access control in one IoT-based prototype. Its engineering value lies in placing the camera, notification channel, authorization rule, and actuator within one traceable workflow. Similar studies show that ESP32-CAM can support low-cost door monitoring and security applications (Setiawan et al., 2022; Cakrayuda, 2025; Mahardika, 2024). The present study extends this line of work by documenting explicit state transitions and linking each requirement to a test criterion in a public-service context.

3.2. Telegram Bot Notification and Door Control

Telegram Bot was implemented as the main communication interface between the smart doorbell system and the user. The use of Telegram Bot provides several advantages, including real-time message delivery, simple command-based interaction, and no requirement to develop a separate mobile application. Once the ESP32-CAM captures a visitor image, the image is transmitted directly to the registered Telegram account. This process allows the user to receive visual information about the visitor even when the user is not near the door.

The door control mechanism was also integrated into Telegram Bot. The user can send commands such as opening or locking the door, and the ESP32-CAM processes the command by activating or deactivating the relay connected to the solenoid door lock. This implementation confirms that Telegram Bot can function not only as a notification medium but also as a remote control platform for IoT-based access systems. Similar findings were reported by Gulo et al. (2022), Febrianti et al. (2025), and Ramadhani and Putri (2023), who demonstrated that Telegram Bot can be used effectively in IoT-based home security and door control systems.

The Telegram notification and door-control interface is presented in Figure 5. The main advantage of using Telegram Bot is its accessibility. Users do not need to install a dedicated application because Telegram is already available on smartphones and supports bot-based interaction. This makes the proposed system more practical and user-friendly for public-service environments, including office buildings and institutional facilities.

3.3. Functional Testing Results

Functional testing evaluated whether the developed prototype matched the predefined criteria in Table 3. The eight scenarios covered input detection, image capture, Telegram delivery, buzzer activation, open and lock commands, Wi-Fi connectivity, and command processing. As presented in Table 3, all eight scenarios passed, giving an observed functional scenario pass rate of 8/8 or 100%. This result establishes coverage of the specified functions under the tested configuration; it does not by itself establish reliability under prolonged, adverse, or multi-site conditions.

Figure 5. Telegram Bot interface for visitor notification and door control.

Table 3. Functional test results against predefined criteria.

No.	Test scenario	Expected evidence	Observed evidence	Outcome
1	Push button	Input event detected	1/1 event detected	Pass
2	Visitor image capture	Readable image generated	1/1 readable image generated	Pass
3	Telegram image notification	Image received by authorized account	1/1 image received	Pass
4	Buzzer activation	Audio indication produced	1/1 activation observed	Pass
5	Open-door command	Relay active; solenoid unlocked	Expected states observed	Pass
6	Lock-door command	Relay off; solenoid locked	Expected states observed	Pass
7	Wi-Fi connection	Device connected	Connection established	Pass
8	Telegram command response	Authorized command executed	1/1 command executed	Pass

3.4. Quantitative Performance Analysis

The available records support quantitative analysis of pass proportions, repeated command execution, workflow-stage completion, and short observation-window connectivity. They do not support an average end-to-end response time, unlock latency, packet-loss rate, current draw, or energy estimate because no synchronized timestamps or electrical measurements were recorded. Table 4 therefore reports only defensible aggregate indicators. Wilson intervals are included to prevent the observed 100% proportions from being interpreted as certainty. For example, the 5/5 repeated-command result has a lower 95% bound of 56.6%, illustrating the need for substantially more trials before making a strong reliability claim. Future benchmarking should follow instrumented ESP32-CAM studies that record network, frame-rate, voltage, and temperature variables (Nowroz et al., 2025).

Table 4. Aggregate quantitative validation indicators.

Indicator	Evidence	Observed estimate	95% Wilson CI	Interpretation
Functional scenario pass rate	8/8 scenarios	100%	67.6-100%	All specified functions passed once; sample remains small
Workflow-stage completion	9/9 stages	100%	70.1-100%	The recorded end-to-end sequence completed
Repeated unlock-command reliability	5/5 trials	100%	56.6-100%	No failure observed, but precision is limited
Connection stability rate	3/3 windows	100%	43.8-100%	Stable in 1-, 3-, and 5-minute windows only

The image-notification and unlock functions were additionally observed in five repeated open-door trials. In every trial, the Telegram command corresponded with an active relay and an unlocked solenoid, producing a repeated-command reliability of 5/5 (100%). The image provided sufficient visual information for basic visitor observation. However, the test record did not include image-delivery timestamps, resolution-level quality scores, packet loss, or lighting measurements. The evidence therefore supports functional feasibility, not a claim of low latency or standardized image quality. This interpretation is consistent with the need for explicit ESP32-CAM benchmarking reported in the literature (Ipanhar et al., 2022; Prakoso, 2025; Nowroz et al., 2025).

Table 5. Repeated authorized open-door command trials.

Trial	Telegram command	Relay state	Solenoid state	Outcome
1	Open Door	Active	Unlocked	Pass
2	Open Door	Active	Unlocked	Pass
3	Open Door	Active	Unlocked	Pass
4	Open Door	Active	Unlocked	Pass
5	Open Door	Active	Unlocked	Pass

Table 6. Recorded connection stability observation windows.

Observation window	Recorded connection state	Outcome
1 minute	Stable	Pass
3 minutes	Stable	Pass
5 minutes	Stable	Pass

The solenoid-control logic behaved as specified in all five repeated open-door trials. After an authorized command, the relay activated and the solenoid unlocked; the design then returned the actuator to the default locked state. The recorded connection was also stable in all three observation windows, giving 3/3 or 100%. These short windows are useful as a basic continuity check but are not an endurance test. They do not quantify performance during congestion, weak signal strength, internet outage, Telegram API delay, power fluctuation, or prolonged operation.

3.5. Discussion

The findings show that the prototype satisfied the functions specified for the tested doorway: visitor triggering, local indication, image acquisition, authorized Telegram notification, remote unlocking, and return to the safe locked state. The aggregate results are stronger than a purely qualitative "Success" statement because they expose the number of observations and the uncertainty associated with small samples. At the same time, the broad Wilson intervals demonstrate why the results should be interpreted as feasibility evidence rather than as proof of mature operational reliability.

Comparison with related IoT door-security studies is summarized in Table 7. Existing systems demonstrate useful capabilities such as Blynk-based door control, face identification, video monitoring, PIR detection, and Telegram-based actuation. The closest quantitative household study reported 95% motion-detection accuracy, an average command response of approximately 2 seconds, and 93% overall effectiveness (Febrianti et al., 2025). The present prototype cannot claim superior timing because latency was not instrumented. Its distinct contribution is instead the combination of image notification, authorized remote solenoid control, automatic relocking, explicit state logic, and auditable test criteria in a public-service deployment context.

The academic significance is therefore methodological and contextual rather than component-level novelty. The study provides a traceable artifact description that other researchers can replicate, challenge, and extend. The architecture and state logic are transferable to offices with a comparable entrance workflow, Wi-Fi access, and authorized staff account. This is analytical generalizability: the design principles may transfer, but the numerical results cannot be assumed to transfer without replication at other sites and under other network and environmental conditions.

Table 7. Comparison with selected IoT door-security studies.

Study	Main capability	Remote door control	Reported evaluation evidence	Setting
Setiawan et al. (2022)	ESP32-CAM Wi-Fi door lock with Blynk	Yes	Functional implementation	Household
Renwarin et al. (2023)	Face identification and Telegram smart lock	Yes	Functional implementation	Household
Mahardika (2024)	Image/video smart doorbell and two-way communication	Not reported	Qualitative functional result	Household
Febrianti et al. (2025)	PIR, Telegram, relay, and solenoid lock	Yes	95% detection; about 2 s average response; 93% overall	Household
Prakoso (2025)	ESP32-CAM smart doorbell via Telegram	Not reported	Network/lighting tests; low average delay reported	Household
Proposed study	Image notification, authorized control, and auto-relock	Yes	8/8 functional; 5/5 repeated unlock; 3/3 connection windows	Public-service office

Accordingly, the study does not claim novelty from merely combining ESP32-CAM and Telegram. Its contribution lies in treating the integration as an evaluable public-service access-control artifact: the safe default state is defined, authorization is explicit, transitions are formalized, requirements are linked to observable evidence, and results are summarized with denominators and confidence intervals. This framing establishes the practical and scholarly significance of the study by providing a reproducible baseline for more rigorous IoT access-control research.

The small samples also impose a strict limit on the practical meaning of the all-success observations. If the next trial failed, the repeated-unlock estimate would change from 5/5 (100%) to 5/6 (83.3%), the functional-scenario estimate from 8/8 (100%) to 8/9 (88.9%), and the connectivity-window estimate from 3/3 (100%) to 3/4 (75.0%). Thus, the reported percentages are descriptive of the recorded trials and not stable reliability parameters. A confirmatory evaluation should predefine a substantially larger trial count, include failure and recovery scenarios, extend connectivity monitoring to hours or days, and replicate the protocol across network, lighting, and institutional conditions.

The unmeasured variables have direct operational implications. Unknown notification and unlock latency may delay staff response and encourage repeated commands; weak or congested Wi-Fi may prevent timely image delivery; unmeasured power demand may cause brownouts or unintended restarts when the camera, relay, and solenoid operate together; and the absence of endurance testing leaves maintenance intervals and availability uncertain. In a public facility, cybersecurity is particularly consequential: disclosure of the bot token or authorized chat credentials could permit unauthorized control, while replay, flooding, or denial-of-service conditions could impair availability. Deployment beyond the present feasibility baseline should therefore require latency and power instrumentation, connection-loss and fail-safe tests, protected credential storage and rotation, network segmentation, audit logging, rate limiting, and penetration testing before the system is entrusted with routine access-control duties (Allen et al., 2024; Vardakis et al., 2024).

Several limitations remain. First, evaluation was conducted on one prototype at one public-service site, so the findings do not establish multi-site generalizability. Second, the repeated samples were small, as reflected by the wide confidence intervals. Third, the original log did not record end-to-end latency, unlock latency, RSSI, packet loss, image-quality scores, power consumption, or internal temperature. Fourth, the connectivity observations lasted only up to five minutes and therefore do not represent endurance testing. Fifth, security evaluation was limited to chat-identifier and command matching; token compromise, replay attempts, denial-of-service conditions, and network attacks were not tested. Sixth, visitor identification remained manual and no user-experience study was performed. Future work should use synchronized timestamps, power meters, controlled network attenuation, longer runs, larger trial counts, multiple institutional sites, penetration testing, and user evaluation. Face recognition, encrypted credential management, cloud visitor logs, and video streaming should be evaluated only after the baseline reliability and security measurements are established.

4. Conclusion

This study designed and evaluated an IoT-based smart doorbell that integrates ESP32-CAM, a push button, buzzer, relay module, solenoid door lock, Wi-Fi, and Telegram Bot for visitor image notification and remote door control at the UPTD Pengelolaan Pendapatan Daerah Pandeglang. The study's contribution extends beyond a simple assembly of established components. The artifact is documented through explicit requirements, deterministic state logic, an authorized-command rule, predefined pass/fail criteria, repeated observations, and quantitative summaries with confidence intervals. Under the recorded configuration, the prototype passed 8 of 8 functional scenarios, completed 5 of 5 repeated authorized unlock trials, executed all 9 stages of the recorded end-to-end workflow, and maintained connectivity in the 1-, 3-, and 5-minute observation windows. Each observed proportion was 100%; however, the Wilson intervals remained wide because of the small number of observations. The evidence therefore supports technical feasibility at the tested site, not a claim of universal reliability or superiority over other smart doorbell systems. The broader academic significance is the provision of a reproducible validation baseline for public-service access-control research, including a clear safe state, authorization behavior, automatic relocking, and requirement-to-test traceability. The design may be analytically transferable to other institutional entrances with comparable workflows, but the numerical results require independent replication. The current study did not measure notification delay, unlock latency, packet loss, signal strength, image quality, energy consumption, long-term availability, cybersecurity resilience, or user experience. Future evaluation should instrument these variables with synchronized logs and power measurements, increase the number and duration of trials, introduce controlled network and lighting conditions, test multiple public-service sites, conduct security attack scenarios, and involve intended users. Such work is necessary before the prototype can be characterized as an operationally reliable and generalizable access-control solution.

Acknowledgements

The authors would like to express their sincere gratitude to Universitas Bina Bangsa, especially the Faculty of Computer Science and the Computer Science Study Program, for providing academic support throughout this research. The authors also thank the UPTD Pengelolaan Pendapatan Daerah Pandeglang for providing the research context and supporting the implementation of the proposed smart doorbell system. Appreciation is also extended to the supervisors, lecturers, and all parties who provided guidance, feedback, and assistance during the completion of this study.

References

Akbari, Y., Al-Binali, A., Al-Mohannadi, A., Al-Hemaidi, N., Elharrouss, O., & Al-Maadeed, S. (2024). A new framework for smart doors using mmWave radar and camera-based face detection and recognition techniques. Sensors, 24(1), 172. https://doi.org/10.3390/s24010172

Allen, A., Mylonas, A., Vidalis, S., & Gritzalis, D. (2024). Security evaluation of companion Android applications in IoT: The case of smart security devices. Sensors, 24(17), 5465. https://doi.org/10.3390/s24175465

Arduino. (n.d.). Getting started with Arduino IDE 2. https://docs.arduino.cc/software/ide-v2/tutorials/getting-started-ide-v2/

Arya, S. (2021). Internet of Things (IoT) based video analytics: A use case of smart doorbell. arXiv. https://arxiv.org/abs/2105.06508

Aydin, O., & Erhan, I. I. (2021). Video or image transmission security for ESP-EYE IoT device used in business processes. arXiv. https://arxiv.org/abs/2107.08321

Bellini, P., Nesi, P., & Pantaleo, G. (2022). IoT-enabled smart cities: A review of concepts, frameworks and key technologies. Applied Sciences, 12(3), 1607. https://doi.org/10.3390/app12031607

Bhering, F., Passos, D., Obraczka, K., & Albuquerque, C. (2024). Network performance estimator with applications to route selection for IoT multimedia applications. SIMULATION, 100(1), 23-37. https://doi.org/10.1177/00375497231156618

Brown, L. D., Cai, T. T., & DasGupta, A. (2001). Interval estimation for a binomial proportion. Statistical Science, 16(2), 101-133. https://doi.org/10.1214/ss/1009213286

Cakrayuda, L. (2025). SICEMOT: Sistem keamanan cerdas berbasis ESP32-CAM, sensor gerak, dan notifikasi Telegram. Jurnal Informatika dan Teknik Elektro Terapan, 13(2), 1176-1185. https://doi.org/10.23960/jitet.v13i2.6420

Domínguez-Bolaño, M., Campos, O., Barral, V., Escudero, C. J., & García-Naya, J. A. (2021). An overview of IoT architectures, technologies, and existing open-source projects. Internet of Things, 15, 100316. https://doi.org/10.1016/j.iot.2021.100316

Elnozahy, S. S. F. A., Pari, S. C., & Liang, L. C. (2025). Raspberry Pi-based face recognition door lock system. IoT, 6(2), 31. https://doi.org/10.3390/iot6020031

Espressif Systems. (2024). ESP32 series datasheet. Espressif Systems. https://www.espressif.com

Febrianti, A., Azizah, B. N., & Septiani, E. S. (2025). Pemanfaatan Telegram Bot pada sistem keamanan rumah berbasis IoT dengan mikrokontroler ESP32. Prosiding Seminar Nasional Teknologi Informasi dan Bisnis, 532-539. https://doi.org/10.47701/jxxvgb60

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29(7), 1645–1660. https://doi.org/10.1016/j.future.2013.01.010

Gulo, M., Suherdi, D., & Yetri, M. (2022). Sistem keamanan rumah berbasis Internet of Things menggunakan Telegram dan NodeMCU. Jurnal Sistem Komputer Triguna Dharma, 1(4). https://doi.org/10.53513/jursik.v1i4.5579

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. MIS Quarterly, 28(1), 75-105. https://doi.org/10.2307/25148625

Ipanhar, A., Wijaya, T. K., & Gunoto, P. (2022). Perancangan sistem monitoring pintu otomatis berbasis IoT menggunakan ESP32-CAM. Sigma Teknika, 5(2), 333–350. https://doi.org/10.33373/sigmateknika.v5i2.4590

Islam, M. M., Nooruddin, S., Karray, F., & Muhammad, G. (2022). Internet of Things device capabilities, architectures, protocols, and smart applications in healthcare domain: A review. arXiv. https://arxiv.org/abs/2204.05921

Itahara, S., Nishio, T., & Yamamoto, K. (2021). Packet-loss-tolerant split inference for delay-sensitive deep learning in lossy wireless networks. In 2021 IEEE Global Communications Conference (GLOBECOM) (pp. 1-6). IEEE.

Jayusman, D., Armanto, & Martadinata, A. T. (2024). Sistem keamanan rumah berbasis IoT menggunakan sensor PIR. Resolusi: Rekayasa Teknik Informatika dan Informasi, 5(1), 58–66. https://doi.org/10.30865/resolusi.v5i1.2144

Ma'muriyah, N., Yulianto, A., & Tymoden. (2023). Perancangan sistem smart door dengan notifikasi Telegram. Jurnal Mahajana Informasi, 8(2). https://doi.org/10.51544/jurnalmi.v8i2.4642

Mahardika, A. T. (2024). Rancang bangun smart doorbell berbasis Internet of Things menggunakan ESP32-CAM. Jurnal Digital Teknologi Informasi, 7(2), 6-13. https://doi.org/10.32502/digital.v7i2.9299

Mori, M., Jat, D. S., Barik, R. K., & Rathore, V. S. (2022). IoT technologies in smart environment: Security issues and future enhancements. Environmental Science and Pollution Research. https://doi.org/10.1007/s11356-022-20132-1

Nishfahuddin, A., & Febriawan, D. (2024). Rancang bangun prototipe sistem keamanan gerbang rumah otomatis menggunakan NodeMCU ESP8266 dengan kendali Telegram. Journal of Information System Research (JOSH), 6(1), 330–337. https://doi.org/10.47065/josh.v6i1.6025

Nowroz, S. T., Saleh, N. M., Shakur, S., Banerjee, S., & Amsaad, F. (2025). A benchmark reference for ESP32-CAM module [Preprint]. arXiv. https://arxiv.org/abs/2505.24081

Pathak, T., Patel, V., Kanani, S., Arya, S., Patel, P., & Ali, M. I. (2020). A distributed framework to orchestrate video analytics across edge and cloud: A use case of smart doorbell. In Proceedings of the 10th International Conference on the Internet of Things (Article 23, pp. 1-8). ACM. https://doi.org/10.1145/3410992.3411013

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. Journal of Management Information Systems, 24(3), 45-77. https://doi.org/10.2753/MIS0742-1222240302

Podder, P., Mondal, M. R. H., Bharati, S., & Paul, P. K. (2021). Review on the security threats of Internet of Things. arXiv. https://arxiv.org/abs/2101.05614

Prakoso, S. B. (2025). Rancang bangun smart doorbell berbasis Internet of Things (IoT) menggunakan ESP32-CAM via Telegram. YASIN, 5(5), 5058-5067. https://doi.org/10.58578/yasin.v5i5.7237

Puche Rondon, L., Babun, L., Aris, A., Akkaya, K., & Uluagac, A. S. (2022). Survey on enterprise Internet-of-Things systems (E-IoT): A security perspective. Ad Hoc Networks, 125, 102728. https://doi.org/10.1016/j.adhoc.2021.102728

Ramadhani, S., & Putri, D. P. (2023). Design of a home door security system based on NodeMCU ESP32 using a magnetic reed switch sensor and Telegram Bot application. Sinkron: Jurnal dan Penelitian Teknik Informatika, 8(4), 2059–2068. https://doi.org/10.33395/sinkron.v8i4.12688

Renwarin, A. Y. W., Susilo, B., & Widodo, B. (2023). Smart door lock menggunakan identifikasi wajah dan Bot Telegram sebagai kendali jarak jauh berbasis Internet of Things (IoT). Lektrokom: Jurnal Ilmiah Teknik Elektro, 6(2), 1–11. https://doi.org/10.33541/lektrokom.v6i2.5259

Setiawan, D., Jaya, H., & Nurarif, S. (2022). Implementasi ESP32-CAM dan Blynk pada WiFi door lock system menggunakan teknik duplex. Journal of Science and Social Research, 5(1), 159–164. https://doi.org/10.54314/jssr.v5i1.807

Sonamoni, J. S., Sikdar, R., Akib, A. S. M. A. S., Islam, M. S., Sourov, S., Al Ahasan, M. A., Islam, M., Habib, M. A., & Mridha, M. F. (2024). IoT-based smart remote door lock and monitoring system using an Android application. Engineering Proceedings, 76(1), 85. https://doi.org/10.3390/engproc2024076085

Susanto, F., Prasiani, N. K., & Darmawan, P. (2022). Implementasi Internet of Things dalam kehidupan sehari-hari. Jurnal Imagine, 2(1), 35–40. https://doi.org/10.35886/imagine.v2i1.329

Telegram. (n.d.). Telegram Bot API. https://core.telegram.org/bots/api

Vardakis, G., Hatzivasilis, G., Koutsaki, E., & Papadakis, N. (2024). Review of smart-home security using the Internet of Things. Electronics, 13(16), 3343. https://doi.org/10.3390/electronics13163343

Younas, M. I., Ali, M., Khan, S., & Kim, H. (2023). Toward QoS monitoring in IoT edge devices driven by artificial intelligence. Sensors, 23(21), 8867. https://doi.org/10.3390/s23218867