Smart Techno (Smart Technology, Informatic, and Technopreneurship)
Volume 08, Number 01, 2026 pp. 121-131
E-ISSN : 2541-0679
Open Access: https://lppm.primakara.ac.id/jurnal/index.php/smart-techno/

Design and Development of an Internet of Things (IoT)-Based Smart Bell Using ESP32-CAM and Telegram

Puji Utami Rakhmawati[1*], Dinda Mareta Syahnaryanti[2], Rizdania[3], Sumantri[4]
[1,2,3] Department of Computer Science, Universitas PGRI Wiranegara, Pasuruan, Indonesia
[4] Department of Informatics Engineering, Universitas Nurul Jadid, Probolinggo, Indonesia

ARTICLE INFO
Article history:
Initial submission
10-12-2025
Received in revised form
22-01-2026
Accepted 22-01-2026
Available online 27-02-2026

Keywords:
ESP32-CAM, Internet of Things, Smartbell, Telegram

DOI:
https://doi.org/10.59356/smart-techno.v8i1.191

ABSTRACT

When guests or couriers arrive, homeowners must go to the front door to identify them due to the limitations of conventional doorbells, which only produce sound. If the homeowner is not at home, they may miss the arrival of guests, and couriers might leave packages elsewhere, posing a risk of loss since there is no real-time notification system. To address this issue, a Smartbell based on the Internet of Things (IoT) was designed using the ESP32-CAM module as an image capture device, integrated with Telegram to provide homeowners with real-time visual information. This study applied the prototype method, which consists of stages such as requirements identification, system design, coding, functionality and time testing, as well as system evaluation before implementation. The test results show that the Smartbell successfully performed as expected. In the simulation, when the bell button was pressed, the buzzer sounded, and the ESP32-CAM camera automatically captured an image and sent it to Telegram in real-time. Since the Smartbell was successfully connected to the Telegram bot, it can be operated remotely. Testing with a Wi-Fi network resulted in an average response time of 0.74 seconds, while using a cellular data network achieved 0.54 seconds. With a response time of less than one second from the integration of ESP32-CAM and Telegram, this system supports the homeowner’s needs as a remote and real-time guest monitoring solution.

1. INTRODUCTION

Since ancient times, doorbells have served as an important communication tool for homeowners. One notable example is the electric bell invented by Joseph Henry, an American scientist who was the first to develop the world’s electric doorbell (Alicia, 2024). Electric doorbells are capable of providing quick and easy notifications of visitors or couriers in modern times. However, conventional electric doorbells can only produce sound without providing any visual information. This limitation forces homeowners to rely on direct communication to verify the identity of visitors. In certain situations, particularly when homeowners are not at home, this doorbell system becomes less effective. This is especially problematic during scheduled package deliveries, as homeowners may miss information from couriers or need to reschedule deliveries. Therefore, additional technology is required in doorbell systems to integrate camera functionality with other devices in order to provide digital information.

With the rapid development of the internet, it has become an essential supporting infrastructure for daily activities. In addition to connecting users, the internet also enables automatic communication between devices. This concept is known as the Internet of Things (IoT) (Basabilik, 2021). The term “Internet of Things” consists of two main components: “Internet,” which serves as the primary connector and manager of connectivity, and “Things,” which refer to physical objects or devices (Selay et al., 2022). The Internet of Things (IoT) is a communication paradigm that refers to the idea of connecting everyday objects to the internet. These objects are equipped with microcontrollers and transceivers to enable communication and are configured with protocol stacks that allow objects to interact with one another to achieve common goals without human intervention (Poongothai, 2018). IoT is also recognized as one of the five fundamental digital technologies in the implementation of Industry 4.0 in

* *Corresponding Author: Puji Utami Rakhmawati: tammyglory@gmail.com*

Indonesia, alongside artificial intelligence, wearables (augmented reality and virtual reality), advanced robotics, and 3D printing (Press et al., 2019).

The Internet of Things enables physical objects to be connected to the internet, allowing them to be monitored and controlled remotely. This technology is particularly beneficial as it can be integrated with smartphones. One practical application of IoT technology is remote monitoring, especially in residential environments. To address the aforementioned issues, a prototype innovation of a conventional electric doorbell is developed by integrating Internet of Things technology, resulting in a Smartbell system. In this system, when the doorbell button is pressed, a signal is sent to a buzzer as an audible indicator, while the ESP32-CAM captures an image and sends a notification to a Telegram bot. The device requires wireless network configuration to connect the ESP32-CAM to the internet. Through this system, homeowners can identify visitors without approaching or opening the door. This solution enhances comfort and efficiency, particularly for users with busy schedules who are frequently away from home.

The objective of this study is to design a Smartbell prototype equipped with a camera capable of capturing images in real time, to integrate the Smartbell system with a Telegram bot, and to evaluate the Smartbell system in terms of notification delivery speed.

2. LITERATURE REVIEW

Research conducted by Puji Utami Rakhmawati et al., entitled "IoT-based Intelligent System for Laboratory Facility Control with Blynk" (pp. 304–331), aimed to develop an intelligent laboratory system based on Internet of Things technology using the Blynk application. This study employed the prototype method. The findings indicated that the system operated successfully according to its objectives, demonstrated through testing results. The system was able to automatically control laboratory devices via Blynk with a response time of approximately 1 second, and all four installed relays successfully controlled electronic equipment (Rakhmawati et al., 2024).

Research by Rizki Amelia Lestari, entitled "Design and Development of an IoT-Based Automatic Home Doorbell with Face Recognition Using an Android Application and ESP32-CAM" (pp. 100–108), focused on developing an IoT-based doorbell prototype integrated with face recognition to enable homeowners to identify visitors even when they are far from the door. This study also adopted the prototype method. The results showed that the PIR sensor was capable of detecting visitors at distances of up to 200 meters, while the ESP32-CAM detected faces within a range of 40–80 cm. The system exhibited a functional error rate of 16.6%, indicating that the prototype performed effectively (Lestari et al., 2024).

Another study conducted by Swingly R. Runtuwene et al., entitled "IoT-Based Home Security Application Using the Prototype Method" (pp. 26–37), aimed to develop a system capable of providing information when someone enters a house without the owner's knowledge, thereby enhancing home security. This study utilized the prototype method. The results suggested that the prototype method is an effective solution for improving home security, as the integration of sensors and devices enabled homeowners to monitor conditions and receive real-time notifications (Runtuwene et al., 2024).

Furthermore, research conducted by Purnomo et al. concluded that the design and development of an IoT-based controller for managing electronic devices via an Android smartphone can be achieved by integrating a Telegram Bot application as a control interface connected to a NodeMCU microcontroller. This controller is capable of sending commands to the PZEM-004T module to monitor electrical power consumption and executing Telegram Bot commands to activate or deactivate electronic devices (Saputro et al., 2024).

Based on previous studies, it can be concluded that the prototype method has been widely applied in similar IoT-based systems and has been successfully implemented in both laboratory and residential environments. In contrast to prior research, this study introduces novelty by utilizing a physical doorbell button as the primary trigger, integrated with an ESP32-CAM and the Telegram application for real-time notification delivery.

3. METHOD

3.1. Research and Development (R&D) Model

This study was conducted to design an Internet of Things–based Smartbell system using the ESP32-CAM, which is capable of automatically sending images via Telegram when the doorbell button is pressed by a visitor. To achieve this objective, the research was carried out through a structured sequence of stages. The research stages are illustrated in a research flowchart to provide a clear overview of the procedures undertaken prior to system development, as shown in Figure 1. The description of each research stage is as follows:

1. Problem Identification

Observations were conducted on conventional electric doorbells, which are limited to producing sound and are considered less effective when homeowners are not present. Based on these observations, a solution was proposed in the form of developing a Smartbell system integrated with the Telegram communication platform and the ESP32-CAM module as an image capture device.

2. Literature Review

Relevant scientific sources were analyzed to understand related concepts. The results of the literature review were used to adopt appropriate methods, program structures, ESP32-CAM configurations, and Telegram integration, which served as guidelines for system design.

3. Research Design

This stage began with the selection of the hardware and software components to be used. After identifying the required components, a system workflow was designed in the form of a program flowchart to facilitate the Smartbell development process.

4. System Development

This stage involved the implementation of the previously designed system. It started with assembling the hardware components, followed by uploading the program using the Arduino IDE. After successful uploading, each system function was tested to ensure proper operation.

5. Testing

Functional testing was conducted by performing actions such as pressing the doorbell button. The time duration from image capture by the ESP32-CAM to image delivery and reception via the Telegram application was then measured.

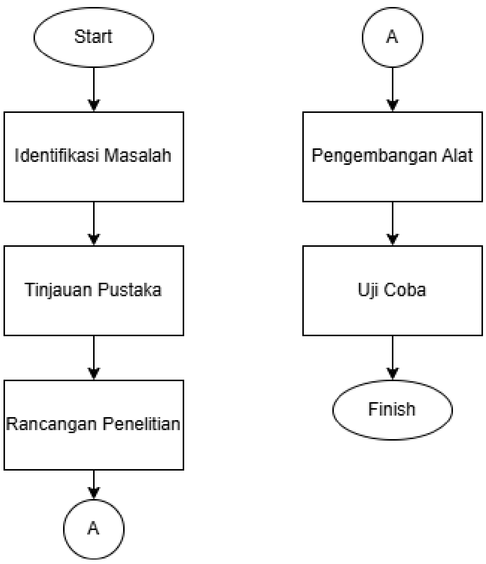

Figure 1. Research process flowchart

3.2. Research and Development Procedure

This study employs the prototype development method. The prototype is developed prior to the final version, either as an initial step in the development process or as a reference before large-scale production (Sugiharti & Mujiastuti, 2023). Figure 2 illustrates the prototype development stages applied in this study. These stages ensure that the Smartbell design process is structured and facilitate iterative improvements at each development cycle.

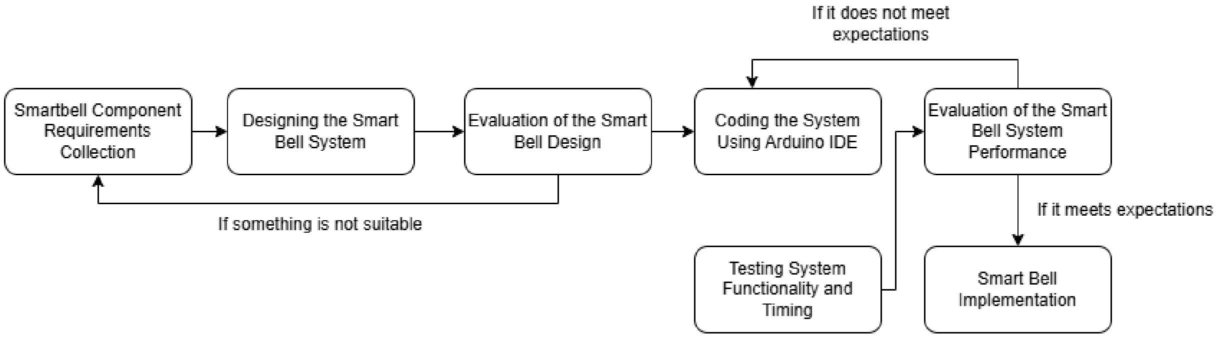

Figure 2. Prototype method

The steps involved in developing the Smartbell system using the prototype method include the following:

1. **Smartbell Component Requirements Collection**
   This stage involves collecting the component requirements for the Smartbell system through literature studies of scientific journals and tutorials from YouTube. The component requirements are divided into two categories: hardware components to be assembled into the device and software components that support the operation of the Smartbell system.

2. **Smartbell Design Development**
   a. Hardware Design, This stage includes assembling each component, such as the doorbell button, buzzer, ESP32-CAM, and shield, which are interconnected using AWG 24 cables. In addition, a Telegram bot account is created using the BotFather service.
   b. Program Flowchart, The flowchart is used to illustrate the logical workflow of the Smartbell system.

3. **Smartbell Design Evaluation**
   This process evaluates the design to ensure that all components are properly interconnected and function according to the planned design.

4. **System Coding Using Arduino IDE**
   This stage is a critical part of the system development process to ensure that the device operates according to the intended instructions. Programming is performed by uploading the code to the ESP32-CAM using the Arduino IDE software.

5. **Smartbell Functionality and Time Testing**
   This stage is conducted to ensure that all Smartbell features operate as expected. Black-box testing is employed to focus on system functionality. Additionally, time testing is performed ten times under different network conditions to measure the duration of image capture and transmission using a stopwatch.

6. **Smartbell System Performance Evaluation**
   This stage is conducted after uploading the code to the ESP32-CAM via the Arduino IDE to assess the success and performance of the implemented program.

7. **Smartbell Implementation**
   This final stage involves the deployment and implementation of the Smartbell system.

4. RESULT AND DISCUSSION

1. **Smartbell Component Requirements Collection**
   The detailed components used in this study are as follows:
   a. Hardware Components
      - Doorbell button as the input device
      - ESP32-CAM and shield as the main module board
      - Active buzzer as the audible output device
      - AWG 24 cables as interconnecting wires between components
      - Micro USB cable as the power supply source
   b. Software Components
      - Arduino IDE as the C++-based programming environment
      - Telegram Bot as the user output and notification medium

2. **Smartbell Design Development**
   This design stage consists of the following components:
   a. Hardware Design
      - **Doorbell Button Assembly**

In the doorbell button assembly, an AWG 24 cable is connected to the doorbell screw terminals and linked to the ESP32-CAM to supply electrical current, as illustrated in Figure 3. In addition to providing power, the opposite end of the doorbell button is connected to the ground (GND) pin and GPIO13 pin of the ESP32-CAM. This configuration enables the transmission of a signal to the ESP32-CAM as an indication that the doorbell has been pressed. When the button is pressed, the switch connects the GPIO pin to ground, allowing the microcontroller to detect a logic low (LOW) signal.

Figure 3. Bell button assembly

- **Buzzer Assembly**

The buzzer used in this system is an active buzzer with two pins of different lengths. During installation, the positive (+) pin is connected to a black AWG 24 cable, while the negative (-) pin is connected to a red AWG 24 cable, as shown in Figure 4.

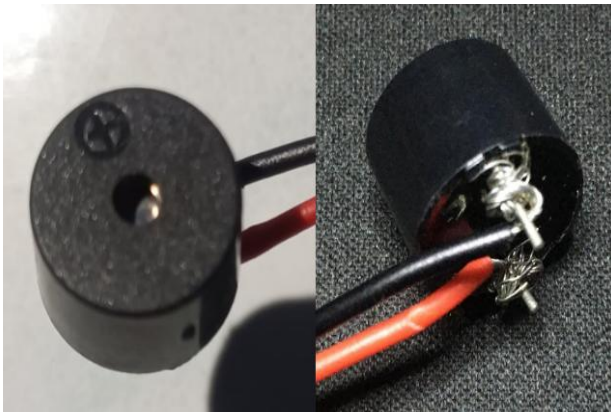

Figure 4. Buzzer active assembly

During the installation process, the AWG 24 cables are tightly wrapped around the buzzer pins and then soldered to ensure a secure electrical connection. After being attached to the buzzer, the opposite ends of the cables are connected to the ESP32-CAM, with the red cable connected to the Ground (GND) pin and the black cable connected to GPIO15.

Figure 5 presents the Smartbell system circuit configuration. The ESP32-CAM serves as the main module and is integrated with its shield into a single assembly. The shield used is the ESP32-CAM-MB type, which is specifically designed for the ESP32-CAM module and therefore does not require an additional breadboard. The buzzer and doorbell button are connected using AWG 24 cables, while power is supplied via a mini USB cable.

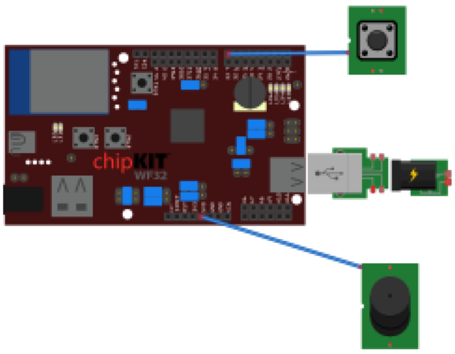

Figure 5. ESP32-CAM and Shield

Figure 6 shows that the Ground (GND) pin and GPIO13 pin are used as the connection path for the doorbell button. Through this configuration, the doorbell button is powered by the ESP32-CAM and can operate properly. Meanwhile, the Ground (GND) pin and GPIO15 pin are used as the connection path for the buzzer, allowing the buzzer to receive power from the ESP32-CAM and subsequently produce sound. AWG cables function as interconnecting components that are mounted on the shield by inserting the exposed copper ends into the female pin sockets of the shield, which are then secured by the male pins of the ESP32-CAM.

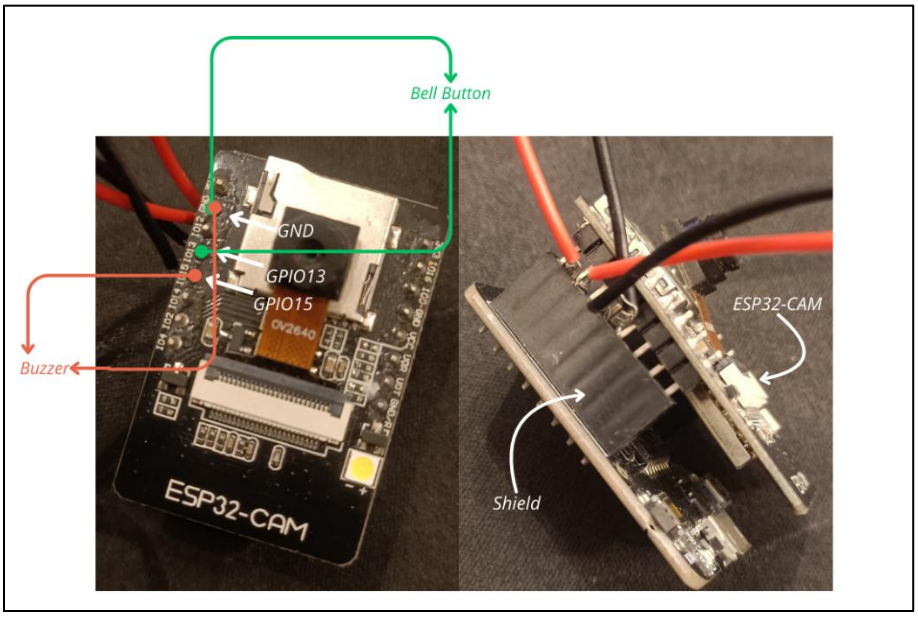

Figure 6. Component integration process

The final assembled circuit is shown in Figure 7, which illustrates that the doorbell button and buzzer components are connected to the ESP32-CAM using AWG 24 cables. The ESP32-CAM is integrated with the shield into a single unit, enabling the device to be powered through the micro USB connection provided by the shield.

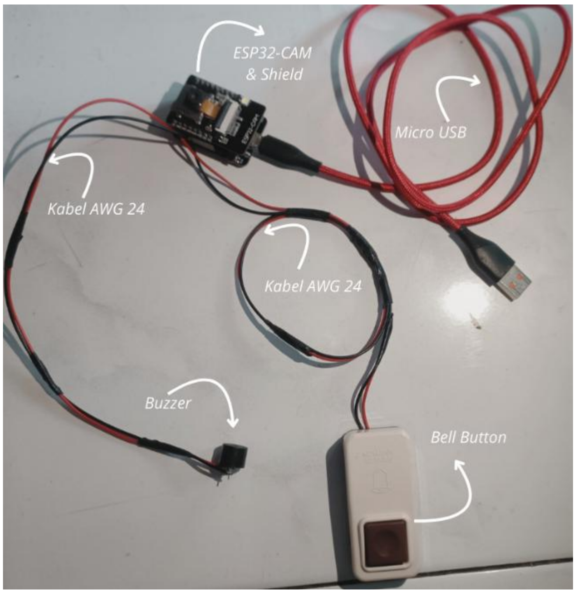

Figure 7. Final result of the smartbell device

- **Telegram Bot Creation**

In the Smartbell system, a new Telegram bot is utilized, as shown in Figure 8. The bot is created by searching for BotFather in the Telegram application, sending the /start command, and then selecting the /newbot option. The user is then prompted to assign a bot name, with the requirement that the username must end with the suffix "bot." Once the bot is successfully created, an API key is provided, which is subsequently used in the Arduino IDE. Next, the Telegram user ID associated with the target account is identified. After obtaining both the API key and the user ID, these parameters are programmed into the Arduino IDE to enable communication between the ESP32-CAM and the Telegram bot.

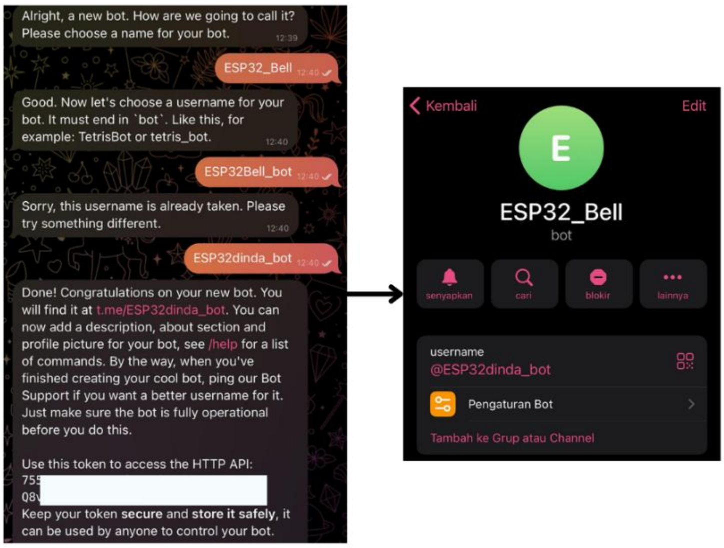

Figure 8. Creating a bot with BotFather

b. Program Flowchart

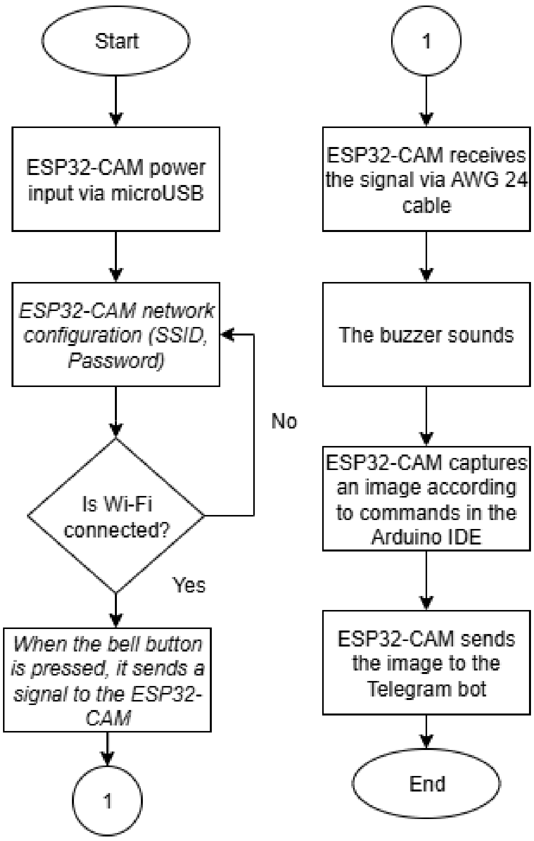

Figure 9. Flowchart program

Figure 9 illustrates the program workflow implemented in the Smartbell system. A detailed explanation of the workflow is provided as follows:

- In the first stage, the ESP32-CAM is supplied with electrical power through a micro USB connection connected to a USB charger adapter, power bank, or laptop.

- After the power indicator is activated, the Wi-Fi network configuration of the ESP32-CAM is performed by entering the SSID and password into the Arduino IDE program code.
- The system then verifies whether the ESP32-CAM is successfully connected to the Wi-Fi network. If the connection fails, the process returns to the network configuration stage and retries the connection. Conversely, if the connection is successful, the system is ready for operation.
- When the doorbell button is pressed, an electrical signal is sent to the ESP32-CAM.
- The ESP32-CAM receives the electrical signal through the connected AWG 24 cables. This signal triggers predefined commands that have been programmed into the ESP32-CAM using the Arduino IDE.
- Based on this signal, the buzzer is activated to emit a sound according to the configured settings.
- After the buzzer stops, the ESP32-CAM executes the command to capture an image.
- In the final stage, once the image is captured, the ESP32-CAM sends the image to the user in real time via the Telegram Bot API as an image output.

3. Smartbell Design Evaluation

This stage is carried out by rechecking that the AWG 24 cables used as interconnecting wires between components are securely installed and by ensuring that the selected pins on the ESP32-CAM are correctly assigned to avoid conflicts with other pin functions.

4. System Coding Using Arduino IDE

The main program workflow includes adding the ESP32-CAM library, configuring the Wi-Fi connection, and integrating the Telegram Bot API to enable the device to transmit data.

5. Smartbell Functionality and Time Testing

Subsequently, Smartbell functionality testing is conducted using the black-box testing method. This testing is performed by simulating the use of the Smartbell system and observing each output produced to verify that the system operates as expected.

Table 1. Functionality Testing Results

No	Trial	Working Technique	Hope	Result	Description
1	Doorbell	The doorbell is pressed	The buzzer sounds	Accurate	Buzzer active
2	Camera	After the buzzer stops	The camera takes and sends pictures	Accurate	Image successfully captured
3	Image delivery	Check the Telegram app	Image sent to Telegram bot	Accurate	Enter Telegram notifications

The test results presented in Table 1 indicate that the Smartbell system operates as expected. This is demonstrated by the successful sequential responses of each system component.

Table 2. Results of Duration Testing Using Wi-Fi

Test Number-	Smartphone Network	Duration (Seconds)
1	Wi-Fi	0,85
2	Wi-Fi	0,76

3	Wi-Fi	0,57
4	Wi-Fi	0,65
5	Wi-Fi	0,78
6	Wi-Fi	0,71
7	Wi-Fi	0,90
8	Wi-Fi	0,65
9	Wi-Fi	0,85
10	Wi-Fi	0,68

Meanwhile, the test results presented in Table 3 show that data transmission received by a smartphone via a cellular data network with speeds of 10-50 Mbps achieved an average delivery time of 0.54 seconds, with the fastest time of 0.49 seconds and the longest time of 0.63 seconds. Based on these results, it can be concluded that the fastest data transmission was achieved using a cellular data network, with a minimum time of 0.49 seconds, compared to a Wi-Fi network, which recorded a fastest time of 0.57 seconds. Therefore, users are able to receive Telegram notifications in real time with a latency of less than 1 second. The generated images are configured with a fixed resolution, as shown in Figure 10.

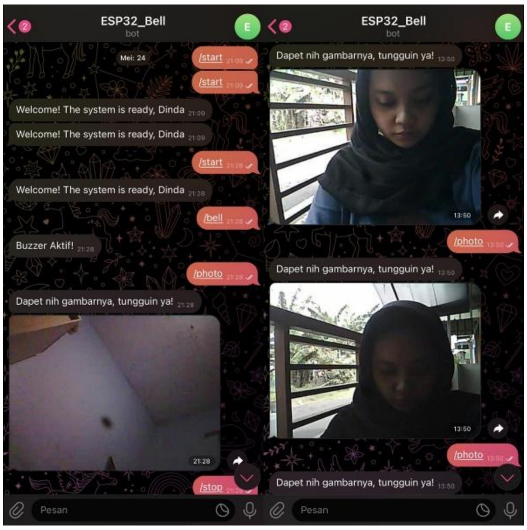

Figure 10. Camera Capture Results

6. **Smartbell System Performance Evaluation**

At this stage, the program code was successfully uploaded, enabling the Smartbell system to operate as expected, as shown in Table 1. When the doorbell button is pressed, the buzzer is activated to emit sound, and the camera captures an image that is subsequently sent to the Telegram bot in real time with a delivery time of less than 1 second.

7. **Smartbell Implementation**

The Smartbell system has not yet been deployed in real-world conditions, as it has only been developed as a prototype. Therefore, the current Smartbell system is used in a limited scope for testing and simulation purposes.

8. **Advantages and Limitations of the Smartbell System**

One advantage of the Smartbell system is the use of the Telegram application as an image output medium without requiring any additional applications. Moreover, the use of the ESP32-CAM module contributes to cost and space efficiency, as the module integrates both a camera and a Wi-Fi module. Conversely, a limitation of the Smartbell system lies in the constraints of the built-in ESP32-CAM camera, which has a maximum resolution of only 2 megapixels. This limitation affects the quality of visual information received by users, indicating the need for further development to improve image clarity. In addition, the system relies heavily on an internet connection; if the ESP32-CAM is not connected to the internet, the system cannot operate.

5. CONCLUSION

Based on the results of the design, implementation, and testing phases, it can be concluded that the Smartbell system functions in accordance with the intended objectives. The Smartbell operates properly, as evidenced by the successful activation of the buzzer when the doorbell button is pressed, followed by image capture by the camera and the transmission of the captured image via Telegram notifications. This study also successfully integrates the Smartbell system with a Telegram bot capable of rapidly transmitting captured images in real time, as well as enabling remote system interaction.

The duration testing results indicate that the Smartbell system does not require a long time to transmit images, whether using a Wi-Fi network or a cellular data network. The experimental results show that the average image transmission time using a Wi-Fi network is 0.74 seconds, while the average transmission time using a cellular data network is 0.54 seconds. These results demonstrate that the Smartbell system is capable of delivering real-time notifications with a latency of less than 1 second across different types of internet connections.

6. REFERENCES

Alicia. (2024). When Were Doorbells Invented? A Look at Their History.

Basabilik, P. (2021). RANCANG BANGUN SISTEM PEMANTAU KEDATANGAN TAMU BERBASIS INTERNET OF THINGS (IOT). 9(2), 110–116.

Lestari, R. A., Informatika, T., Studio, A., Kunci, K., Badan, M., Statistik, P., Otomatis, B. R., Pir, S., Plasida, S., & Ir, S. (2024). R ANCANG & B ANGUN B EL R UMAH O TOMATIS B ERBASIS I O T D AN F ACE. 29, 100–108.

Poongothai, M. (2018). Design and Implementation of IoT Based Smart Laboratory. 2018 5th International Conference on Industrial Engineering and Applications (ICIEA), 169–173. https://doi.org/10.1109/IEA.2018.8387090

Press, A. I., Wagyana, A., Artikel, I., Elektro, J. T., & Jakarta, P. N. (2019). Jurnal Ilmiah Setrum. 8(1), 238–247.

Rakhmawati, P. U., Rizdania, R., & Muladi, M. N. (2024). IoT-based Intelligent System for Laboratory Facility Control with Blynk. Journal of Information System Research (JOSH), 6(1), 304–311. https://doi.org/10.47065/josh.v6i1.5908

Runtuwene, S. R., Kumajas, S. C., Sangkop, F. I., Informatika, T., & Manado, U. N. (2024). Aplikasi IoT Keamanan Rumah Menggunakan Metode Prototype. 26–37.

Saputro, M. P. A., Kautsar, I. A., Fitrani, A. S., & Sidoarjo, U. M. (2024). Pengembangan pengendali perangkat elektronik dengan nodemcu melalui bot telegram. 7, 210–217. https://doi.org/10.37600/tekinkom.v7i1.1237

Selay, A., Andgha, G. D., Alfarizi, M. A., Bintang, M. I., Falah, M. N., Khaira, M., & Encep, M. (2022). Karimah Tauhid, Volume 1 Nomor 6 (2022), e-ISSN 2963-590X. 1, 860–868.

Sugiharti, T. I., & Mujiastuti, R. (2023). PEMBUATAN PROTOTYPE APLIKASI MIMOPAY DENGAN. 13(3), 191–198.