PROtek : Jurnal Ilmiah Teknik Elektro
https://ejournal.unkhair.ac.id/index.php/protk/index

Volume 12, No, 3 September 2025
e-ISSN 2527-9572 / p-ISSN 2354-8924

Design and Implementation of a Community-Based Neighborhood Alarm System Using a PC-Hosted Telegram Bot

Dharmawan
Department of Electrical Engineering, Khairun University, Ternate, Indonesia.
Jl. Jusuf Abdulrahman, Gambesi, Kec. Ternate Selatan
Kota Ternate, Maluku Utara
dharmawan@unkhair.ac.id

Fahrizal Djohar
Department of Electrical Engineering, Khairun University, Ternate, Indonesia.
Jl. Jusuf Abdulrahman, Gambesi, Kec. Ternate Selatan
Kota Ternate, Maluku Utara
fahrizaldjohar@unkhair.ac.id

Hafid Saifudin
Department of Electrical Engineering, Khairun University, Ternate, Indonesia.
Jl. Jusuf Abdulrahman, Gambesi, Kec. Ternate Selatan
Kota Ternate, Maluku Utara
hafid@unkhair.ac.id

***Abstract* – Neighborhood security is still a major worry, especially in residential areas with high population densities where quick action is necessary to control emergencies and deter crime. Current security systems frequently depend on IoT devices dispersed throughout several locations or manual monitoring, which can raise costs and complexity. This study suggests creating and deploying a neighborhood alarm system that is controlled by a Telegram bot hosted on a PC. Through the system, sirens linked to an amplifier-speaker/TOA network placed in key locations throughout the neighborhood can be turned on and off by registered users. User authentication, access control, alarm activation, and Telegram-delivered real-time status updates are some of the main features. By using a centralized PC as the control center, the architecture eliminates the need for numerous IoT devices and streamlines maintenance. Based on experimental results, the system is a viable solution for enhancing community security and cooperative response in emergency situations because it achieves a fast response time, dependable operation under network conditions, and high user acceptance.**

***Keywords: Telegram, Bot, Community, Alarm, Siren*.**

I. INTRODUCTION

In an era marked by escalating security concerns, particularly regarding residential burglaries during peak holiday seasons, innovative and accessible security solutions are paramount [1]. Traditional home security systems, while effective, often involve substantial financial investment and complex installations, thereby limiting their widespread adoption [2]. The integration of Internet of Things devices with popular messaging platforms like Telegram offers a cost-effective and efficient alternative for enhancing community-level security and remote monitoring capabilities [3] [4]. This paper details the design and implementation of a community-based neighborhood alarm system, leveraging a PC-hosted Telegram bot to provide an accessible and robust security framework [5]. This system aims to bridge the gap between sophisticated security functionalities and ease of use, enabling real-time alerts and community-wide communication through a familiar interface [4]. The proposed system specifically addresses the rising incidence of residential theft by providing timely alerts and enabling collaborative responses among community members [6]. The system capitalizes on the ubiquitous nature of smartphones and the robust notification features of Telegram, offering a decentralized security solution that minimizes reliance on traditional infrastructure [7] [8]. By integrating various sensor technologies with a Telegram bot, this system offers a practical and affordable means to detect unauthorized intrusions and disseminate immediate alerts to a designated network of community members [9] [10].

Recent studies on IoT-based home security systems have primarily focused on automating alarms and notifications through microcontrollers such as ESP32, NodeMCU, or Raspberry Pi, combined with sensors including PIR motion, gas, and temperature detectors. Such a system, integrating features like PIR sensors and magnetic switches, provides comprehensive intrusion detection, similar to other multi-sensor approaches [9]. Furthermore, this approach integrates capabilities for gas leak and fire detection, thereby offering a more comprehensive hazard monitoring solution than typically found in singular intrusion detection systems [11].This multi-faceted security framework not only enhances individual home safety but also fosters a collective security posture within a neighborhood, facilitating rapid communal responses to potential threats. This design provides a significant advancement over systems focused solely on individual monitoring, such as those for smart keychains or single-household clusters, by establishing a distributed alert network [12] [13]. The system's reliance on a PC-hosted bot allows for flexible deployment and management, while its integration with Telegram ensures widespread accessibility and immediate notification

delivery to community members [14]. Such comprehensive monitoring extends beyond typical security provisions, incorporating environmental sensors to detect gas leaks and potential fires, thereby expanding its utility as a holistic safety system [15].

Therefore, the objective of this study is to design and implement a Community-Based Neighborhood Alarm System managed through a PC-based Telegram Bot. The system enables authorized residents to remotely trigger or deactivate an alarm siren installed throughout the neighborhood. The alarm signal is transmitted through the PC's soundcard output to an amplifier, which then drives the speaker or TOA network for wide-area coverage. Key design considerations include real-time responsiveness, false-alarm prevention, user authentication, and feedback confirmation to ensure successful alarm activation or termination. The performance evaluation focuses on response time, system reliability under network disruptions, and user satisfaction in an actual residential environment.

II. METHOD

The proposed Community-Based Neighborhood Alarm System adopts a centralized approach to ensure a simple yet efficient emergency response mechanism. Unlike the majority of existing systems that still rely on distributed IoT nodes—requiring regular maintenance and leading to increased cost and system complexity—this method utilizes a Telegram Bot hosted on a PC as the main control center. The PC-based system directly drives the amplifier–speaker/TOA network, triggers alarms, and provides instant feedback to users through Telegram.

The development process begins with the creation of a Telegram Bot using BotFather, which generates a unique authentication token. This token enables secure communication between the Telegram server and the PC application, allowing real-time command processing and response. The schematic representation of the proposed system is shown in Figure 1.

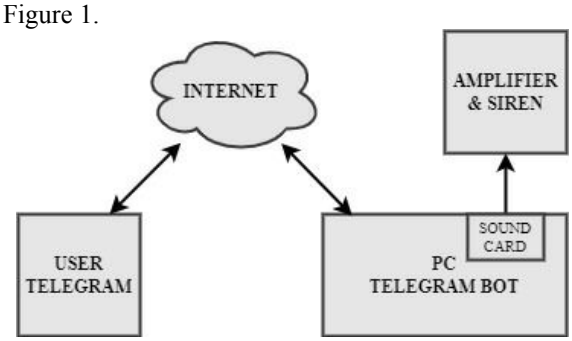

Figure 1. Schematic of Community-Based Neighborhood Alarm System

After the bot is created, a set of specific commands are defined to handle core system functions:

a. /panic – to activate the alarm siren,
b. /stopalarm – to deactivate the alarm, and
c. /register – to register a new user ID into the system.

This command structure is designed to be intuitive and user-friendly, allowing residents to quickly trigger or stop alarms during emergency situations.

The PC Bot application acts as the system's control hub. It continuously monitors incoming messages from Telegram, verifies the sender's Telegram ID, and executes the corresponding command if the user is authorized. When an alarm is triggered, the system transmits an audio signal through the PC's soundcard, which serves as the output interface to the amplifier. The amplifier then drives the speaker or TOA network installed across the neighborhood, broadcasting the alarm sound clearly and effectively. This direct soundcard-to-amplifier connection eliminates the need for additional IoT audio nodes, further reducing system complexity, latency, and maintenance requirements.

System testing and verification are crucial phases in this approach. During testing, registered user IDs are validated, and the alarm activation process is evaluated under simulated emergency scenarios—such as theft, fire, and suspicious activity. The response time is measured from the moment the user sends a Telegram command until the siren sound is heard, ensuring that the system meets rapid response standards required for emergency situations.

Finally, the results are analyzed to evaluate system responsiveness, reliability, and community engagement. User feedback is collected to assess usability and practicality in real-world contexts. Based on the findings, further recommendations are proposed to enhance neighborhood safety and promote the adoption of similar centralized systems in other residential areas.

III. RESULTS AND DISCUSSION

This section presents the experimental outcomes of the PC-based Telegram Bot application operating on an Internet-connected computer. Using the API token obtained from BotFather, the application successfully established a secure connection with the Telegram server. After initialization, the PC Bot continuously monitored incoming messages from registered users, confirming that the Telegram Bot and PC application communicated as designed. The main interface of the Panic Button Bot application is shown in Figure 2.

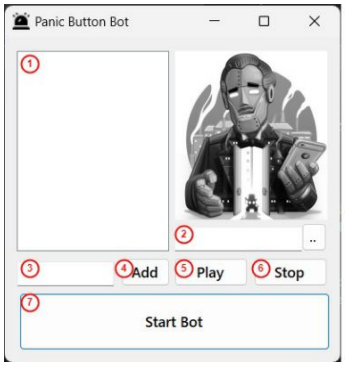

Figure 2. Display Panic Button Bot App

To verify user authentication and system security, an unregistered Telegram account was used for testing. When this account attempted to send commands such as /start, /panic, or /stopalarm, the PC Bot correctly detected that the user ID was unauthorized and returned the message:

"Account is not registered to use the Bot."

This response verified that unauthorized users could not access or trigger the alarm. The chat interface for an unauthorized user is presented in Figure 3.

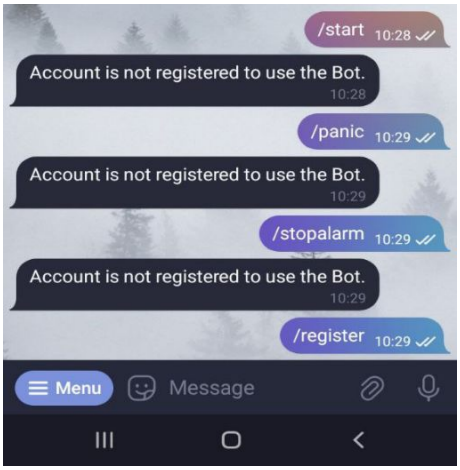

Figure 3. user ID not authorized

When the unregistered user entered /register, the PC Bot recorded the Telegram ID and displayed it on the administrator panel. After administrator approval, the user was officially added to the verified user list and granted access to all commands. This registration and approval mechanism effectively prevents misuse and false alarms. The registration process is shown in Figure 4.

Figure 4. Adding User ID

Once verified, the registered user initiated the /panic command. The Bot responded with the question:

"Are you in danger?"

A PANIC ALARM button appeared in the chat interface. Upon pressing this button, the PC Bot transmitted an audio signal via the PC's soundcard to the amplifier, which activated the speaker/TOA system distributed throughout the neighborhood.

Multiple trials confirmed that the alarm system produced a clear, loud siren across the area, demonstrating that the alarm activation sequence functioned correctly. The chat display and the alarm activation attempt by a registered user ID are shown in Figure 5.

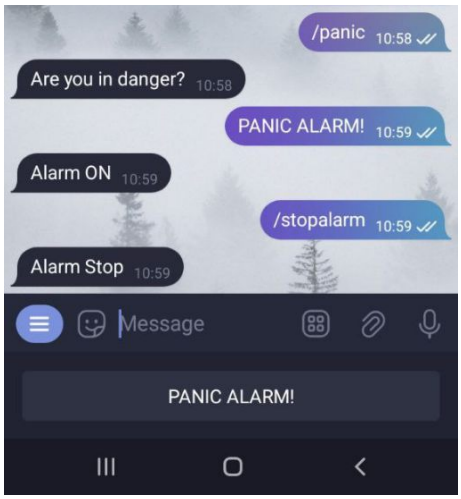

Figure 5. Testing by register user ID

Table 1 summarizes the quantitative testing results, including the number of successful commands, failed commands, and the average response time measured from the moment a Telegram command was sent until the siren was audible.

Table 1. system's response time trials

Test Scenario	Number of Trials	Succes	Failed	Respon (s)
Registration process	5	5	0	2.0
Authorized user – /panic	10	10	0	1.2
Authorized user – /stopalarm	10	10	0	0.8
Unauthorized user – /panic	5	0	5	0.8

The results demonstrate that the proposed centralized Telegram Bot–based system operates reliably for real-time emergency communication. The connection between the PC application and Telegram server remained stable throughout testing, and the response time of less than 2 seconds on average is suitable for rapid emergency alerting.

However, this centralized design also introduces certain risks and limitations. The system relies on continuous Internet connectivity; a network failure

would prevent message delivery and alarm activation. Similarly, PC hardware or software failures could interrupt service. To mitigate these issues, future improvements may include:

a. Implementing redundant backup communication channels (e.g., GSM SMS or local Wi-Fi trigger).
b. Adding cloud-based logging and automatic service restart features.
c. Deploying UPS power backup to maintain system operation during outages.

In summary, the centralized Telegram Bot alarm system effectively balances simplicity, reliability, and cost-efficiency. Quantitative data confirm its responsiveness, while qualitative observations highlight ease of use for community members. Future versions could further improve fault tolerance and scalability for larger residential areas.

IV. CONCLUSION

This study used a PC-hosted Telegram Bot to successfully design and implement a neighborhood alarm system. With the help of the suggested system, neighborhood residents will be able to actively monitor and respond to emergencies like fires, thefts, and other questionable activity. The system offers a straightforward yet efficient way to remotely set off alarms and instantly notify residents by utilizing Telegram's Bot API and a specially designed PC application.

Several scenarios were used to test the application created for this study, including interacting with unregistered users, registering and approving user IDs, and executing commands like /panic and /stopalarm. The findings showed that only verified users were able to sound the alarm since unregistered users were prevented from accessing the system. Users were able to send the /panic command, which activated the siren through the amplifier and speaker system, after registering and being approved by the administrator. They also received instant feedback verifying the alarm activation. Similar to this, the /stopalarm command successfully turned off the siren from the user's Telegram chat or straight from the PC Bot interface, demonstrating that the system could be managed locally and remotely.

All things considered, the system's main goals of improving neighborhood security using a community-based strategy were met. It provides a low-cost and easily accessible solution that can be used with readily available PCs and software rather than specialized IoT hardware. This strategy reduces emergency response times while enabling locals to work together to ensure safety. By incorporating advanced features like camera feeds or AI-based event detection for more thorough monitoring, enhancing the user interface for better usability, and supporting multiple neighborhoods, future work can investigate system scalability.

Even though the suggested system has achieved its goals, further research can examine a number of possible enhancements. Initially, the system can be improved to facilitate the deployment of multiple communities or neighborhoods, enabling multiple PC Bots to coordinate alarms and communicate across various regions for broader coverage. Second, a more user-friendly dashboard could be added to the PC Bot application's user interface, allowing administrators to instantly view system status, user activity logs, and alarm history.

The incorporation of multimedia notifications, such as sending pictures or brief videos taken by CCTV cameras whenever a panic alarm is set off, is another exciting avenue. Before acting further, residents and security staff could use this to visually verify the emergency's nature. To further speed up response times, AI-based event detection could be integrated to automatically sound alarms in the event that suspicious activities or incidents are noticed.

Last but not least, strengthening system security is crucial. This includes encrypting data exchanged between Telegram and the PC Bot and putting role-based access control in place to stop unwanted use. These improvements will increase the solution's resilience, scalability, and adaptability for practical uses in smart city settings.

V. ACKNOWLEDGMENTS

Since VB.Net offered a strong and adaptable programming environment for the creation of the PC Bot application utilized in this study, the authors would like to sincerely thank its creators and contributors. Implementing essential features like user ID verification, command handling, and alarm control logic was made much easier by the integrated development environment (IDE) and the large collection of.NET framework libraries.

Additionally, special thanks are given to the developers and maintainers of the Telegram Bot API libraries, whose open-source tools enabled the PC Bot and the Telegram platform to connect seamlessly. In order to overcome integration obstacles and guarantee dependable two-way communication between the system and end users, their thorough documentation and community support were crucial.

The authors also give credit to the Telegram messaging app, which provided the residents with an easy-to-use and widely available user interface. The suggested community-based alarm system's usability and adoption potential were greatly increased by its cross-platform compatibility and real-time messaging features.

Finally, the authors express their gratitude to every member of the community who took part in the system testing phase and offered input that enhanced the developed prototype's responsiveness and dependability.

REFERENCES

[1] M. N. Osman, M. H. F. Ismail, K. A. Sedek, N. A. Othman, and M. Maghribi, “A low-cost home security notification system using IoT and Telegram Bot: A design and implementation,” Journal of Computing Research and Innovation, vol. 7, no. 2, pp. 327–337, 2022. doi: 10.24191/jcrinn.v7i2.325

[2] . Netinant, T. Utsanok, M. Rukhiran, and S. Klongdee, “Development and assessment of Internet of Things-driven smart home security and automation with voice commands,” IoT, vol. 5, pp. 79–99, 2024. doi: 10.3390/iot5010005

[3] A. G. Ingole, “Home security system using IoT: A research paper,” International Journal for Research in Applied Science & Engineering Technology (IJRASET), vol. 12, no. 5, 2024. doi: 10.22214/ijraset.2024.62773

[4] C. F. Y. Aji, S. Nugroho, and M. L. Diana, “Design and build a home security system based on an ESP32-CAM microcontroller with Telegram notification,” Jurnal Jartel: Jurnal Jaringan Telekomunikasi, vol. 12, no. 2, pp. 58–64, 2024. doi: 10.33795/jartel.v12i2.296

[5] G. Perez-Aquise and F. E. Escobedo-Bailón, “Smart doorbell with Telegram notification for multifamily dwellings,” in Information Management and Big Data (SIMBig 2022), J. A. Lossio-Ventura, J. Valverde-Rebaza, E. Díaz, and H. Alatrista-Salas, Eds. Cham: Springer, 2023, vol. 1837, Communications in Computer and Information Science, pp. doi: 10.1007/978-3-031-35445-8_18

[6] H. J. Hadi, K. U. Nisa, and S. Harris, “Autonomous and Collaborative Smart Home Security System (ACSHSS),” arXiv preprint arXiv:2309.02899, 2023. [Online]. Available: https://arxiv.org/abs/2309.02899

[7] M. Idhom, A. Fauzi, R. Alit, and H. E. Wahanani, “Implementation system Telegram bot for monitoring Linux server,” in Proceedings of the International Conference on Science and Technology (ICST 2018), Atlantis Highlights in Engineering Series, Dec. 2018, pp. 1089–1093. doi: 10.2991/icst-18.2018.219

[8] R. B. Salikhov, V. Kh. Abdrakhmanov, and I. N. Safargalin, “Internet of Things (IoT) security alarms on ESP32-CAM,” Journal of Physics: Conference Series, vol. 2096, Proc. Int. Conf. on Automatics and Energy (ICAE 2021), Vladivostok, Russia, Oct. 7–8, 2021. doi: 10.1088/1742-6596/2096/1/012109

[9] V. Arinde and L. Idowu, “Multi-sensor intrusion detection system,” arXiv preprint arXiv:2406.05137, 2024. https://arxiv.org/abs/2406.05137

[10] M. E. Emetere, D. C. Okpala, M. M. Bakeko, and S. A. Afolalu, “Review on home security system in developing countries: Affordability or comfortability,” Journal of Computer Science, vol. 19, no. 4, pp. 415–430, Mar. 2023. doi: 10.3844/jcssp.2023.415.430

[11] A. M. S. V. S. Sushma, A. A. Priyanka, A. N. G. Lakshmi, N. Khatri, and H. Sharma, “Integrated IoT-based smart home security system: Real-time hazard detection and alerts using multi-sensor fusion,” in Proc. 2024 Int. Conf. on Intelligent & Innovative Practices in Engineering & Management (IIPEM), Singapore, 2024, pp. 1–6. doi: 10.1109/IIPEM62726.2024.10925672

[12] M. Gibran, “Analisis keamanan chatbot Telegram dan hardware pada penggunaan ESP32 sebagai smart keychain untuk notifikasi kehilangan,” JINTEKS, vol. 6, no. 3, pp. 588–594, Aug. 2024. doi: 10.51401/jinteks.v6i3.4225

[13] M. F. Wicaksono and M. D. Rahmatya, “Smart cluster housing monitoring with ESP32, ESP-Mesh and Django,” Journal of Engineering Research, ASSEEE Special Issue, 2024. doi: 10.36909/jer.ASSEEE.16099

[14] H. Zhang, R. Zhang, and J. Sun, “Developing real-time IoT-based public safety alert and emergency response systems,” Scientific Reports, vol. 15, p. 29056, 2025. doi: 10.1038/s41598-025-13465-7

[15] M. Babiuch and J. Postulka, “Smart home monitoring system using ESP32 microcontrollers,” in Internet of Things. IntechOpen, Aug. 18, 2021. doi: 10.5772/intechopen.94589