The Design a Real-Time Fire Alarm Prototype Based on Arduino IDE With Telegram Notifications

M. Zainul Muttaqin[1,*] , Dr. Yuyun Suprapto, S,SiT, MM[2] , Ajeng Wulansari, ST, MT[3]

[1&2] *Politeknik Penerbangan Surabaya, Air Navigation Engineering*
[3] *Politeknik Penerbangan Surabaya, Aircraft Engineering*
[*]*Corresponding author. Email: zainulmuttaqim25@gmail.com*

ABSTRACT

Fire is a disaster that can cause significant losses, both in terms of material and human safety. An early detection system is a crucial solution to minimize its impact. This research aims to design and build an Arduino IDE-based fire alarm prototype capable of providing real-time notifications via the Telegram application. This system uses a NodeMCU ESP8266 microcontroller integrated with MQ-2, MQ-8, MQ-9, and DHT11 sensors to detect smoke, hazardous gases, and high temperatures. The prototype is designed with local indicators such as a buzzer, LED, and LCD, as well as remote notification via a Telegram bot. The method used is a prototype model that includes needs identification, design, construction, testing, and evaluation. Test results show that the system is capable of detecting LPG, hydrogen, smoke, and temperature increases with good accuracy. Notifications can be sent to Telegram within a short time after the sensor detects a dangerous condition. This system is suitable for implementation as a learning tool in the laboratory and as a small-scale fire early warning system.

Keywords: *Fire Alarm, Arduino IDE, NodeMCU ESP8266, Internet of Things, Telegram.*

1. INTRODUCTION

Fire hazards remain a persistent threat in residential, industrial, and educational environments, causing significant material damage and posing serious risks to human safety [1]. In laboratory settings, especially those involving electrical equipment and combustible materials, the potential for fire incidents is higher due to the combination of heat sources, flammable gases, and human error. Within the Surabaya Aviation Polytechnic, the Air Navigation Laboratory operates with a range of electronic systems, test equipment, and experimental setups, creating conditions where early fire detection and rapid response systems are critical for both safety and operational continuity.

Recent advancements in electronics and the Internet of Things (IoT) have enabled the development of integrated fire alarm systems that are not only capable of detecting multiple fire-related indicators but also transmitting real-time alerts to remote users. This is particularly beneficial in environments where personnel may not always be physically present to monitor conditions, such as during off-hours or when multiple facilities require simultaneous supervision. By leveraging microcontrollers with wireless communication capabilities, such as the NodeMCU ESP8266, it becomes feasible to design low-cost yet highly functional systems that combine local alarm features with remote notification capabilities.

Traditional fire alarm systems often rely on single-type sensors, such as smoke or heat detectors, and are typically connected to local alarms only. While these systems can be effective in certain scenarios, they lack the ability to integrate various detection methods to reduce false alarms and increase reliability. Moreover, conventional systems rarely offer connectivity to modern communication platforms, limiting their effectiveness in situations where immediate remote notification is essential. IoT-enabled systems address these limitations by allowing real-time monitoring, remote control, and instant alert dissemination through widely used messaging applications such as Telegram.

In this research, a prototype fire alarm system is designed and implemented using the Arduino Integrated Development Environment (IDE) and NodeMCU ESP8266 as the primary controller. The system integrates multiple sensors, namely MQ-2 for detecting smoke and liquefied petroleum gas (LPG), MQ-8 for hydrogen gas detection, MQ-9 for carbon monoxide and flammable gases, and DHT11 for temperature and humidity monitoring. These sensors collectively provide a comprehensive detection capability that reduces the risk of false alarms by correlating multiple hazard indicators [2]. The prototype also includes a buzzer, LED indicators, and an LCD for immediate local alerts, while a Telegram bot delivers real-time notifications to designated users.

By combining multi-sensor detection, IoT connectivity, and real-time messaging, this research aims to bridge the gap between traditional fire alarm systems and modern, connected safety solutions. The outcomes are expected to demonstrate the feasibility of deploying such systems in educational laboratories, highlight the benefits of multi-parameter hazard detection, and provide a foundation for future developments in low-cost, network-enabled safety technologies.

2. METHODS

This research adopts the Prototype development model, which in this study consists of six sequential stages: initial needs identification, initial design planning, prototype creation, review and update, system development, and system testing [3]. This model is well suited for IoT-based safety systems as it allows progressive refinement and direct validation with users in real operating environments. In this research, the model guided the process of developing a real-time fire alarm prototype from understanding laboratory safety requirements to building, refining, and testing the final system.

1. Initial Needs Identification

The first stage focused on identifying the safety gaps in the Air Navigation Laboratory at Surabaya Aviation Polytechnic. Observations and discussions with laboratory staff revealed that there was no integrated system capable of detecting smoke, flammable gases, and elevated temperatures while also sending remote alerts. Manual monitoring methods were slow and prone to oversight, leading to delayed hazard response. This stage resulted in defining core requirements such as multi-sensor integration, instant local alerts, and real-time remote notifications via Telegram.

2. Initial Design Planning

In this stage, the system architecture was conceptualized, including both hardware and software components. The NodeMCU ESP8266 was selected as the central controller, integrated with MQ-2, MQ-8, and MQ-9 sensors for gas detection and a DHT11 sensor for temperature and humidity measurement. Buzzer, LED indicators, and an LCD were included for local alerts, while the Telegram bot API was chosen for remote communication [4]. Block diagrams, wiring schematics, and alarm threshold logic were designed to ensure effective hazard detection and reliable system operation.

3. Prototype Creation

The initial prototype was assembled according to the planned schematic. The NodeMCU was programmed using Arduino IDE with supporting libraries for sensor reading, LCD display, Wi-Fi connectivity, and Telegram message sending. Careful attention was given to stable wiring, proper sensor placement, and regulated power supply to ensure accurate readings and system reliability [5] .

4. Review and Update

After the first prototype was completed, it underwent an initial review to identify design flaws and performance limitations. Adjustments were made to the hardware arrangement, such as improving sensor mounting and optimizing wiring for better signal stability [6] Software updates included refining threshold settings, improving alarm logic, and adding notification interval control to prevent message flooding on Telegram.

5. System Development

In this stage, the refined design was developed into a more complete and stable version. The hardware structure was secured inside a protective casing, the wiring was organized for safety, and the software was finalized with optimized sensor calibration and improved alarm handling logic . Additional safety features, such as hysteresis in threshold detection and multi-sensor confirmation, were implemented to reduce false alarms [5].

6. System Testing

The final system was tested in realistic laboratory scenarios using the black-box method. Controlled hazard simulations included introducing smoke, LPG, hydrogen, and carbon monoxide, as well as increasing temperature. The system’s detection accuracy, alarm response time, and Telegram notification reliability were recorded [7]. The tests confirmed that the system met safety requirements, providing timely alerts both locally and remotely, and was well-suited for use as both a functional fire alarm and an educational tool for IoT applications.

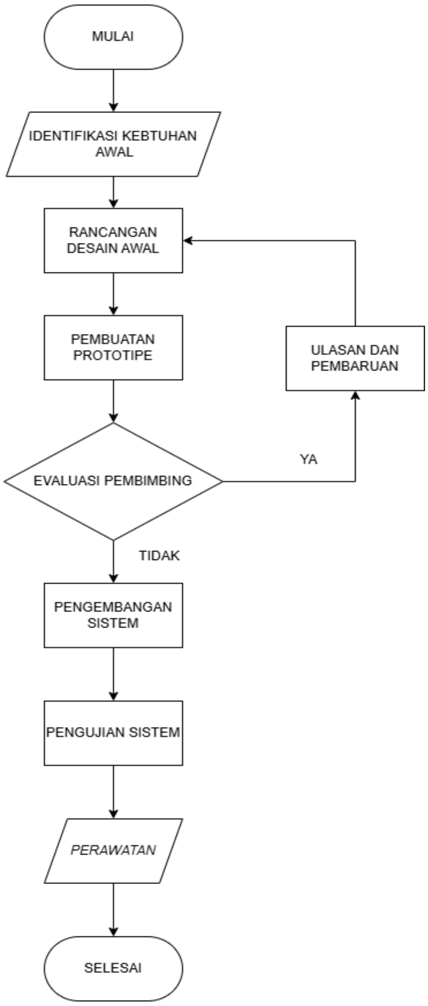

Figure 1 Research Flowchart

3. RESULTS AND DISCUSSION

The Results and Discussion chapter is a critical component of a scientific article, as it presents the main findings of the research and provides interpretation, comparison, and significance of those findings [8]. The purpose of this section is to demonstrate what was achieved during the research process, why the results matter, and how they relate to existing theories or prior studies. In this study, the results and discussions are organized according to the Prototype development model, which in this research consists of six stages: initial needs identification, initial design planning, prototype creation, review and update, system development, and system testing. Each stage contributed significantly to the overall objectives and final conclusions.

1. Initial Needs Identification

The first stage of the research focused on identifying the actual safety needs in the Air Navigation Laboratory of Surabaya Aviation Polytechnic. Field observations and interviews with laboratory staff revealed that the laboratory lacked an automated fire alarm system capable of detecting smoke, flammable gases, and high temperatures while sending real-time notifications. The existing method relied solely on manual monitoring, which could lead to delays in hazard response. The initial data analysis confirmed the urgency of developing an IoT-based fire alarm system integrating multiple sensors and remote alert capabilities [9]. This finding aligned with modern safety management principles that emphasize early detection, automated response, and accessibility of alerts.

2. Initial Design Planning

Based on the identified needs, the initial design was developed to address all functional requirements. The NodeMCU ESP8266 was chosen as the main controller for its compact design, integrated Wi-Fi, and low energy consumption. The sensing unit consisted of MQ-2 for smoke and LPG detection, MQ-8 for hydrogen gas, MQ-9 for carbon monoxide, and DHT11 for temperature and humidity monitoring [10]. A buzzer, LED indicators, and an LCD display were selected for local alerts, while Telegram Bot API integration provided remote notifications. The block diagram, wiring schematic, and alarm threshold logic were carefully planned to ensure accurate readings, stable communication, and reliable operation under varying environmental conditions.

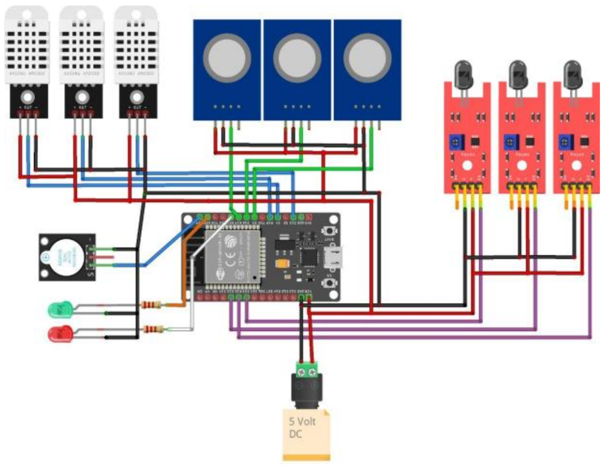

Figure 2 Integration of Components in Tool

3. Prototype Creation

At this stage, the first prototype was assembled according to the planned schematic. The NodeMCU was programmed using Arduino IDE, with libraries to handle sensor readings, LCD display output, Wi-Fi connectivity, and Telegram notifications. The hardware was arranged on a breadboard to allow easy adjustments during the initial trials. Sensor placement was optimized to ensure accurate detection, and the power supply was regulated to prevent fluctuations. The prototype successfully combined all planned components into a single, operational unit.

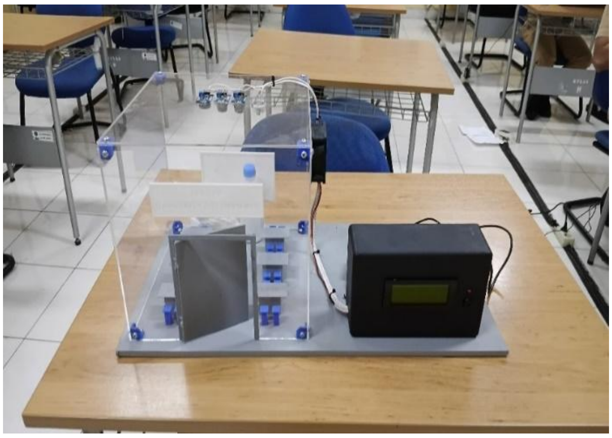

Figure 3 Display Fire Alarm Equipment

Figure 4 Display Notification on Telegram

4. Review and Update

The initial prototype underwent functional testing to identify weaknesses. During early trials, some false alarms occurred due to unstable sensor readings under fluctuating conditions. Adjustments were made to the software, including implementing hysteresis in threshold detection and multi-sensor confirmation logic to reduce false positives [11].The wiring layout was refined to minimize noise interference, and sensor mounting positions were adjusted for better airflow and detection accuracy. These updates improved the stability and reliability of the system

5. System Development

Following the refinements, the system was developed into its final hardware configuration. The components were transferred from a breadboard to a soldered PCB to ensure stability. The entire system was housed in a protective enclosure to safeguard against dust and accidental contact [12]. The final software included optimized sensor calibration, improved notification timing to avoid message flooding, and a user-friendly LCD status display. The result was a compact, durable, and reliable IoT-based fire alarm system ready for operational testing.

6. System Testing

The final stage involved systematic testing using the black-box method to evaluate the system's performance from a user's perspective. Six scenarios were tested:

Table 1 Testing Result

No.	Parameter	Success Indicator	Test Result
1.	NodeMCU Esp8266	Functions as the main controller, processing inputs and sending output control.	Test results indicate that the NodeMCU ESP8266 operates properly, capable of sending control signals as required.
2.	MQ-2 Sensor	Able to detect fire/smoke accurately.	Test results show that the sensor can detect smoke/gas and send a warning signal before a fire occurs.
3.	MQ2 Sensor	Able to detect harmful gases (CO) effectively.	Test results show that the sensor can detect harmful gases and send danger signals reliably.
4.	DHT11 Temperature	Able to detect temperature and humidity changes accurately.	Test results show accurate temperature and humidity readings, displayed clearly on the LCD.
5.	*Buzzer*	Activates when a fire or hazardous gas is detected.	Test results indicate buzzer activation when MQ-2 or MQ-9 sensors detect smoke or gas.
6.	LED	Green LED when safe, Red LED when danger is detected.	Test results indicate LEDs work correctly green for safe conditions, red for danger alerts.

From Table 1, the testing results of the IoT-based fire alarm prototype demonstrated that all core components performed according to their intended functions. The NodeMCU ESP8266, serving as the main controller, successfully processed inputs from various sensors and transmitted control signals to output devices without errors. The MQ-2 sensor accurately detected the presence of smoke or flammable gases and provided early warning signals prior to potential fire incidents. Similarly, the MQ-9 sensor effectively identified harmful gases such as carbon monoxide, transmitting hazard alerts with high reliability. The DHT11 temperature and humidity sensor produced precise readings, which were clearly displayed on the LCD screen to provide additional environmental monitoring data.

As an audible alarm, the buzzer activated consistently whenever the MQ-2 or MQ-9 sensors detected smoke or dangerous gases, ensuring immediate notification. The LED indicators also functioned as expected, with the green LED illuminating under safe conditions and the red LED activating during hazard detection, offering clear and instant visual feedback to users. Overall, the integration of these components resulted in a responsive and reliable fire alarm system capable of delivering both local and remote alerts, thereby enhancing safety monitoring in the target environment.

4. RESULTS AND DISCUSSION

1. This research successfully designed and implemented an advanced fire alarm system that utilizes the Internet of Things (IoT) platform, with the NodeMCU ESP-32 microcontroller as the core manager of the entire detection and communication process. The main advantage of this system lies in its ability to detect various fire and gas leak threats early through the strategic integration of several sensors, namely DHT11, MQ-2, MQ-8, and MQ-9. Test results show that the system is programmed to prioritize direct detection of fire, smoke, or gas as the main triggers for the alarm, rather than temperature changes. Temperature sensors, such as the DHT11, only function as additional data monitors and displays, so the system is designed to minimize false alarms that may arise from normal environmental temperature changes.
2. In addition to its strong detection capabilities, this research has also successfully implemented a highly effective and responsive notification system. When one of the threat detection sensors (fire, smoke, or gas) detects a hazard, the fire alarm system will automatically activate local warning mechanisms (e.g., buzzers and indicator lights, although not explicitly mentioned in the problem statement, these are typically included in alarm systems). Furthermore, and this is a key aspect of IoT integration, the system will send real-time notifications in the form of short messages to the Telegram app. The use of Telegram allows users or relevant parties to immediately receive hazard information wherever they are, provided they have an internet connection. The automatic activation of the alarm and the sending of notifications have been successfully tested and verified for functionality, proving that the system can provide instant and reliable warnings to users, which is crucial in handling emergency situations and mitigating the risks of fire or gas leaks.

REFERENCES

[1] I. Munadhif, Adianto, and A. A. Mustofa, "Sistem Pengendalian Penanganan Kebakaran Gedung Menggunakan Metode Fuzzy Control System for Handling Building Fire Using Fuzzy Method," *J. Ilm. Rekayasa*, vol. 11, no. 2, pp. 171–183, 2018.

[2] R. B. Nurrohim Oktaf Kharisma Harris Simaremare Ewi Ismaredah, P. Studi Teknik Elektro, F. Sains dan Teknologi Program Studi Teknik Elektro, F. Sains dan Teknologi UIN Sultan Syarif Kasim Riau UIN Sultan Syarif Kasim Riau UIN Sultan Syarif Kasim Riau UIN Sultan Syarif Kasim Riau Pekanbaru, and R. Pekanbaru, "Autonomus Call System Berbasis ESP32 Untuk Peringatan Dini Kebakaran Rumah," 2023.

[3] R. N. Fauzi, Y. Suprapto, and R. D. Puspita, "Rancangan Prototipe De-Icing Dengan Menggunakan Sensor Ds18b20 Berbasis Mikrokontroller Arduino Uno," *Pros. SNITP (Seminar Nas. Inov. Teknol. Penerbangan)*, vol. 3, no. 2, pp. 1–6, 2019, [Online]. Available: https://ejournal.poltekbangsby.ac.id/index.php/SNITP/article/view/396

[4] A. Hartono and A. Widjaja, "Prototipe Pendeteksi Kebakaran Menggunakan Sensor Flame, Sensor DHT11 Dan Mikrokontroller NodeMCU ESP8266 Berbasis Website," 2022.

[5] M Wahidin, A. Elanda, and S. S. Lie, "Implementasi Sistem Pendeteksi Kebakaran Berbasis IoT dan Telegram Menggunakan Nodemcu Pada Kantor Notaris Leodi Chanda Hidayat, S.H., M.Kn," *J. Interkom J. Publ. Ilm. Bid. Teknol. Inf. dan Komun.*, vol. 16, no. 2, pp. 1–8, 2021, doi: 10.35969/interkom.v16i2.104.

[6] C. G. I. Raditya, P. A. S. Dharma, I. K. A. A. Putra, I. B. K. Sugirianta, and I. B. I. Purnama, "Pendeteksi Kebocoran Gas dan Kebakaran Dini Menggunakan NodeMCU Berbasis Telegram," *Maj. Ilm. Teknol. Elektro*, vol. 21, no. 1, p. 13, Jul. 2022, doi: 10.24843/mite.2022.v21i01.p03.

[7] H. Jurnal, A. Anggarani, and T. Feri Efendi,

"Rancang Bangun Prototype Fire Alarm Berbasis Arduino IDE Secara Realtime Dengan NotifikasiTelegram," *Jurtikom*, vol. 1, no. 2, 2024, doi: 10.69714/tr6qwt56.
[8] M. Harahap and L. M. Siregar, "Mengembangkan Sumber dan Media Pembelajaran," *Educational*, no. January, pp. 2–3, 2021, doi: 10.13140/RG.2.2.19282.86721.
[9] A. Mahendra and B. Gumilar, "Prototype Deteksi Kebakaran Menggunakan Kamera DENGAN Notifikasi Message Telegram Berbasis IOT Di Bandara Mutiara Sis AL-Jufrie".
[10] P. Ramadhani and J. C. Chandra, "Prototipe Sistem Pendeteksi Kebakaran Berbasis Telegram Bot Dengan Sensor Gas MQ-2 Dan Flame NodeMCU," 2024.
[11] Zaenuar Erfandi, D. Hartanti, and J. Maulindar, "Implementasi Internet of Things (IoT) Untuk Sistem Pemantauan Kebakaran Dini Dengan Notifikasi Telegram dan Alarm," *Infotek J. Inform. dan Teknol.*, vol. 8, no. 1, pp. 86–93, Jan. 2025, doi: 10.29408/jit.v8i1.28248.
[12] R. Bintang Ganiezha, N. Pambudiyatno, R. Indrianto Sudjoko, P. Penerbangan Surabaya, and J. I. Jemur Andayani No, "Rancang Bangun Monitoring Lokasi Smoke Detector Berbasis IoT Menggunakan Aplikasi Android Perum LPPNPI Kantor Cabang Yogyakarta," *Pros. Semin. Nas. Inov. Teknol. Penerbangan Tahun 2023*.