

# Smart Safety Room: ESP32 Decision Tree-Based Multi-Hazard Detection System

Jogi Purba<sup>1</sup>, Dedy Kiswanto<sup>2</sup>, John Bush Henrydunan<sup>3</sup>, Revidamurti Dly<sup>4\*</sup>

<sup>1,2,3,4</sup>Computer Science, Faculty of Mathematics and Natural Sciences, State University of Medan, Indonesia  
[jogipurba25@gmail.com](mailto:jogipurba25@gmail.com)<sup>1</sup>, [dedykiswanto@unimed.ac.id](mailto:dedykiswanto@unimed.ac.id)<sup>2</sup>, [johnbushsimarmata@mhs.unimed.ac.id](mailto:johnbushsimarmata@mhs.unimed.ac.id)<sup>3</sup>,  
[revidamurti.4231250007@mhs.unimed.ac.id](mailto:revidamurti.4231250007@mhs.unimed.ac.id)<sup>4\*</sup>

## Abstract

Physical space security and safety remain fundamental challenges in various sectors, ranging from residential buildings to critical server rooms. Conventional security systems often rely on single sensors or passive alarms that cannot respond comprehensively to multiple simultaneous threats. This research proposes a Smart Safety Room, an ESP32-based integrated multi-sensor security system that combines gas sensors (MQ-2), fire sensors (flame sensors), PIR sensors, and visual-audio output components including OLED displays, RGB LEDs, and buzzers. The system implements a decision tree algorithm with hierarchical priorities to classify room conditions into three categories: SAFE, ALERT, and DANGER based on a combination of sensor data. Testing was conducted through four main scenarios: normal conditions, fire detection, intrusion detection, and dual threat conditions. The results show that the system achieved an overall accuracy of 96.5% with detailed performance of 96% for the fire sensor, 94% for the gas sensor, and 98% for the PIR sensor. The average response time was under 300 milliseconds for all types of detection, meeting the real-time system requirements. The decision tree showed excellent classification performance with an F1-score ranging from 95-97% for all categories. The web-based real-time monitoring dashboard successfully displayed sensor status with auto-refresh every 1 second and a data loss rate of only 0.8% during continuous operation.

**Keywords:** IoT, Smart Safety, ESP32, Multi-Sensor, Decision Tree, Fire Detection, Gas Detection, Intrusion Detection, Real-Time Monitoring

## 1. Introduction

Physical space security is a fundamental requirement in various areas of life, ranging from residences, offices, laboratories, to server rooms that store critical devices. Threats such as theft, intrusion, gas leaks, extreme temperature rises, and fires are still risks that often cause material losses, operational disruptions, and even endanger human safety. In the modern context, reliance on manual surveillance or conventional security systems such as passive alarms, single sensors, and CCTV without analytical intelligence is often unable to provide a quick and comprehensive response to various types of threats that can occur simultaneously.

With the development of Internet of Things (IoT) technology, the paradigm of surveillance and security systems has begun to shift towards smarter, more responsive, and integrated systems [1]. IoT enables devices and sensors to communicate automatically through a network, so that room conditions can be monitored remotely, analyzed in real time, and provide alerts according to the level of danger [2]. The proliferation of increasingly affordable IoT devices such as ESP32 opens up great opportunities to develop security systems that are more adaptive, efficient, and multitasking. ESP32 has WiFi connectivity, multitasking, fast processing, and diverse sensor support capabilities, making it an ideal platform for data-based security systems.

However, despite the increasing use of IoT devices, most security systems developed are still partial and focus on a single threat. [3] only focuses on motion detection through ESP32CAM and PIR sensors without detecting fires or hazardous gases in the room. Research by [4] utilizes PIR sensors, Telegram notifications, and voice alerts, but its implementation is still limited to intrusion detection and does not yet support multi-sensors and visual information display through a real-time dashboard. Meanwhile, research by [5] developed a server room security system with fire, gas, and ultrasonic sensors and the Blynk application, but the system is still rule-based without artificial intelligence algorithm support and does not yet have optimal inter-sensor integration capabilities in a single responsive smart device.

In addition, the majority of previous studies have not included the use of several modern supporting components that can clarify the status of a room visually and audibly. Components such as RGB lights as danger level indicators, OLED displays to display sensor data directly, and buzzers as emergency alarms have been proven to increase the effectiveness of conveying danger messages to users. However, the integrated implementation of these components is still rarely found. In some cases, the limited number of microcontroller pins is also an obstacle, so solutions such as using Arduino Uno as an expansion port are needed to increase system flexibility.

In the modern era, the need for real-time monitoring dashboards is also increasing, especially in the context of monitoring server rooms, laboratories, and high-risk rooms. Users need a system that not only detects threats but also provides an informative, easily accessible visual interface that provides a direct summary of room conditions. IoT-based dashboards enable more effective management by displaying sensor data graphically and historically, allowing users to make quick and accurate decisions [6].

On the other hand, multi-sensor integration often produces complex data, requiring the system to employ smarter analysis methods [7]. The Decision Tree algorithm is one effective approach for classifying room conditions based on a combination of sensor values such as gas levels, fire detection, motion detection, and temperature changes. With this approach, the system not only provides warnings based on a single sensor but can also determine whether the room condition is safe, warning, or emergency based on structured data patterns.

Seeing the gaps in previous research and the modern need for a comprehensive security system, this study proposes Smart Safety Room, an ESP32-based intrusion and hazard detection system that integrates gas (MQ-2), fire (Flame Sensor), PIR, and temperature sensors, supported by Arduino Uno as an additional port, RGB lights, OLED display, buzzer, and real-time monitoring dashboard. This research also uses a Decision Tree algorithm to determine the level of danger in a room based on a combination of sensor data so that the system becomes more adaptive and accurate in providing warnings.

With this approach, the system

## 2. Literature Review

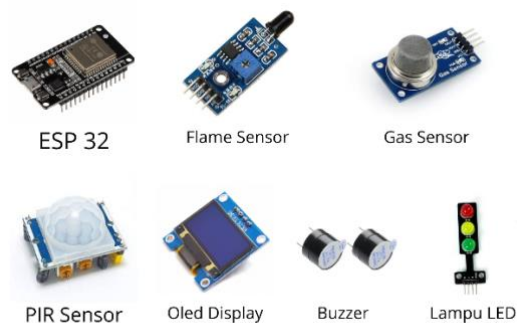
### 2.1. Internet of Things (IoT) in Security Systems

The Internet of Things (IoT) is a concept that connects physical devices to the internet, enabling them to communicate with each other, collect data, and perform automatic controls [1]. In IoT architecture, devices such as sensors, actuators, and microcontrollers work as data collectors at the device layer, then the data is sent via a WiFi network at the network layer, and then processed and displayed at the application layer. IoT-based monitoring systems also enable real-time notifications to be sent to users through various communication platforms such as Telegram [4].

In room security systems, IoT enables the integration of various sensors such as gas, fire, PIR, and temperature sensors into a single integrated platform to detect anomalies that indicate intrusion or potential danger [8]. Research by [2] shows that an IoT-based gas and temperature detection system using MQ-2 and DHT11 sensors integrated with the Blynk platform is capable of providing real-time notifications with high accuracy. [9] confirms that a security system with PIR sensors integrated with ESP32 is capable of sending notifications via Telegram with high responsiveness, achieving a detection range of up to 5 meters with a success rate of 93.33%.

### 2.2. Key Components of IoT-Based Security Systems

IoT-based security systems require the integration of various electronic components that work synergistically. The main components used in this study include the ESP32 microcontroller, various types of sensors (flame sensor, MQ-2 gas sensor, and PIR sensor), and output devices such as OLED displays, buzzers, and LED indicator lights.



**Figure 1:** Main Components of the Smart Safety Room System

1. **ESP32 Microcontroller**  
ESP32 is a modern microcontroller with high computing power and complete connectivity support, making it ideal for use as a data processing center in IoT-based security systems [10]. This microcontroller is equipped with a dual-core processor, sufficient memory, and WiFi and Bluetooth modules that enable direct network communication without additional devices. Research by [11] shows that the ESP32 can be integrated with various sensors such as PIR, MQ-2, and ESP32-CAM to build a responsive and efficient home security system.
2. **MQ-2 Gas Sensor**  
The MQ-2 sensor is a gas sensor that is sensitive to LPG, smoke, hydrogen, alcohol, and methane [12]. This sensor works on the principle of semiconductor material resistance changing when exposed to certain gases. Research by [13] shows that an LPG gas leak detection system using an MQ-2 sensor integrated with NodeMCU ESP8266 is capable of detecting gas leaks with high accuracy and providing a quick response through a buzzer alarm and Telegram notification in less than 3 seconds on average. [2] reported gas detection accuracy of 94.5% with automatic notifications via the Blynk application.
3. **Flame Sensor**  
Flame sensors are used to detect the presence of fire through infrared (IR) rays emitted by flames [5]. Research by [14] shows that flame sensors can detect fire up to a distance of 100 cm with an error rate of only 0.01%, and the system developed using Mamdani Fuzzy logic is capable of providing early warnings with a very high level of accuracy.
4. **PIR Sensor (Passive Infrared)**  
PIR sensors work by detecting changes in infrared radiation emitted by the human body, so they are used to detect intrusion or movement in a room [8]. These sensors are passive and only respond to changes in the heat of objects in their detection area. [4] confirmed the effectiveness of PIR sensors with test results showing motion detection up to a distance of 5 meters with a success rate of 93.33%, and real-time notification delivery to the Telegram application with a success rate of 100%.
5. **Output Components**

This system is also equipped with various output components to provide feedback to users. An OLED display is used as a local visual panel to display important information such as sensor values and hazard status [15]. The RGB LED light serves as a visual indicator that changes color according to the danger level—green for safe, yellow for caution, and red for emergency conditions [15]. A buzzer is used as an audio alarm to provide a quick warning when the system detects intrusion or high danger [15].

### 2.3. Decision Tree Algorithm for Room Condition Classification

Decision Tree is one of the machine learning algorithms used to perform classification based on data patterns arranged in a tree structure [16]. This algorithm works by breaking down data into branches based on certain conditions until it reaches a final decision at the leaf node. [17] applied the C4.5 Decision Tree algorithm to a system for determining smoky room conditions using an Arduino Uno-based multi-sensor, resulting in an accuracy of 97% with an average execution time of  $\pm 1389.9$  ms, capable of classifying room conditions (normal, smoking, cooking, and fire) based on input from MQ-2, DHT22, and flame sensors.

Research by [18] shows that the Random Forest algorithm—which is an ensemble of multiple decision trees—is capable of detecting DDoS attacks with an accuracy of 99.942%. [19] also shows that the Decision Tree C4.5 algorithm is effective in classification analysis with an accuracy rate of 73.48%, proving that this method is reliable for automated decision-making in various IoT application contexts.

In the context of the Smart Safety Room system, Decision Tree is used to group room conditions into several categories such as SAFE, ALERT, or DANGER based on the sensor values read. This model analyzes the relationship between parameters, such as an increase in gas levels, the detection of fire, or significant temperature changes, to automatically determine the level of risk [10].

### 2.4. Real-Time Monitoring System and Dashboard

Real-time monitoring systems are an important element in the development of IoT-based security solutions because they allow users to monitor room conditions directly through an interface connected to the internet [20]. Monitoring dashboards typically display sensor data in the form of graphs, status indicators, and notifications so that users can understand the room situation quickly and intuitively.

Research by [2] shows that temperature monitoring and gas detection systems using the Blynk platform are capable of displaying data in real-time with low latency, approximately 1-2 seconds, with notification delivery times of less than 3 seconds after data is received. [15] shows that the use of the Telegram Bot API allows security systems to send instant alerts to users with excellent responsiveness, without the need for complex server infrastructure.

The study by [15] also confirms that the integration of ESP32 with the Telegram platform and website can improve the effectiveness of the monitoring system, where users can receive real-time notifications and access historical data through a web dashboard. This combination of hardware and software components supports the concept of user interface in IoT, namely the delivery of information that is clear, fast, and easy to understand.

## 3. Research Methodology

### 3.1. Type of Research

This research is quantitative research based on Research and Development (R&D) with an experimental approach. The ESP32-based multi-sensor security system was developed during October–December 2025 through the stages of hardware design, software implementation, testing, and performance evaluation using quantitative metrics. The Decision Tree algorithm was used to classify the security status of the room. The research flowchart is shown in Figure 2.



Figure 2: Research Method Flowchart

### 3.2. Literature Review and Gap Analysis

A literature review was conducted on IoT, ESP32, MQ-2 sensors, flame sensors, PIR, Decision Tree, and monitoring dashboards. Gaps were identified from previous studies: Humam & Triawan (2024) only focused on intrusion, Adi & Kurniawan (2024) did not have a real-time dashboard, and Sufian & Setiyadi (2021) still used rule-based systems without artificial intelligence. These gaps formed the basis for the development of a multi-sensor system with intelligent classification.

### 3.3. System Architecture Design

The system is designed with three layers:

1. Device Layer: ESP32 DevKit V1 with MQ-2, flame sensor, PIR, OLED SSD1306, RGB LED, and buzzer.
2. Network Layer: Data transmission via WiFi using HTTP POST every 20 seconds.
3. Application Layer: MySQL database, PHP API to receive data and classification, and a Bootstrap dashboard with 1-second auto-refresh.

### 3.4. Implementation of the Decision Tree Algorithm

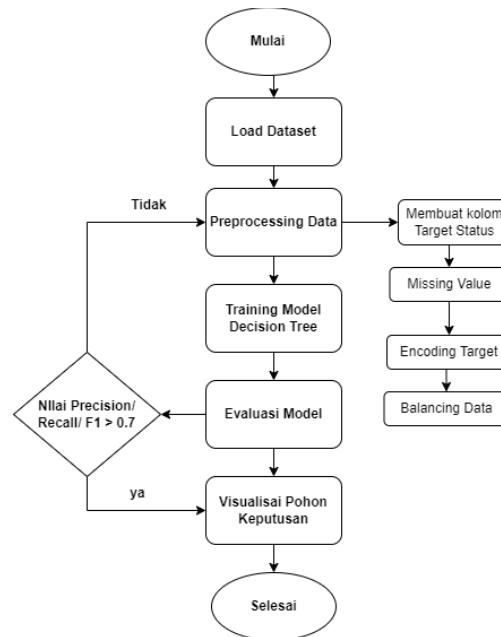


Figure 3: Flowchart of the decision tree model

1. Creating Target Status Columns; At this stage, each row of data is labeled with a room condition based on the system's logic rules: SAFE, CAUTION, or DANGER. The addition of this column aims to provide classification targets for the Decision Tree model to learn.
2. Handling Missing Values; Incomplete or empty data is checked and handled using the appropriate method. If empty values are found, they are filled in using the median value or the data row is deleted under certain conditions to avoid bias in modeling.
3. Encoding the Target; The status labels are converted into numerical values using the label encoding technique so that they can be processed by the Decision Tree model. For example: SAFE = 0, ALERT = 1, and DANGER = 2.
4. Data Balancing; If the dataset is found to be unbalanced between classes, balancing is performed using methods such as Random Oversampling or Synthetic Minority Oversampling Technique (SMOTE). The purpose of this stage is to prevent the model from being biased towards a particular class and to improve prediction performance.
5. Decision Tree Model Training; After the data has been processed, the Decision Tree model is trained using input variables (gas\_value, flame\_value, pir\_state) and target variables (status). The model is built using a train-test split so that the evaluation is carried out objectively. Model parameters can also be adjusted to obtain the best performance.
6. Model Evaluation; The evaluation stage is carried out using precision, recall, and F1-score metrics. In this step, it is determined whether the model is feasible to use based on the criteria: Precision/Recall/F1 > 0.7.; If the values do not meet the minimum threshold, the process returns to the Data Preprocessing stage to make improvements or adjustments to the dataset, such as rebalancing or additional cleaning.
7. Decision Tree Visualization; If the model is deemed feasible, the next step is to display a visualization of the Decision Tree structure. This visualization provides an overview of the model's decision flow in determining a room condition, so that it can be understood intuitively by users and researchers.
8. Model Implementation into the Monitoring System; The evaluated and approved model is then saved as a model file (model\_security.pkl) and integrated into a Flask-based API. The model output is used to provide real-time predictions on the IoT dashboard to support automatic room safety detection.

### 3.5. System Testing Scenario

Total of 230 experiments with 5 scenarios:

1. Normal (50x) → validation without false alarms
2. Fire (50x) → distance variation of 5–15 cm

3. Gas (50x) → 70% lighter gas simulation
4. Intrusion (50x) → distance and speed variation
5. Dual conditions (30x/combination)
  - o Gas + PIR → DANGER
  - o Fire + PIR → DANGER
  - o Fire + Gas → DANGER

### 3.6. Performance Metrics Measurement

Three main parameters:

1. Response time (target <300 ms)
  - Using millis() for api, gas, and PIR
2. Sensor accuracy
  - Using confusion matrix, calculating Precision, Recall, F1, and Overall Accuracy (target  $\geq 95\%$ )
3. Communication stability (24 hours)
  - Success rate, data loss (<1%), latency, and WiFi uptime ( $\geq 99\%$ )

### 3.7. Decision Tree Evaluation and Real Data Validation

The Decision Tree was evaluated using a confusion matrix for three categories (SAFE, CAUTION, DANGER). The results were compared with 4572 real data over 4 days of operation: SAFE 91.5%, CAUTION 6.8%, DANGER 1.7%. The dashboard was verified to run stably with a 1-second auto-refresh and was responsive on various devices.

### 3.8. Comparison with Previous Research

Smart Safety Room excels over previous research in:

- More complete multi-sensor integration
- Use of Decision Tree, not rule-based
- Real-time dashboard and data logging features
- Higher performance (accuracy, response time, stability)

The system offers a more comprehensive and adaptive security solution.

## 4. Results and Discussion

### 4.1. System Implementation

#### A. Hardware Design

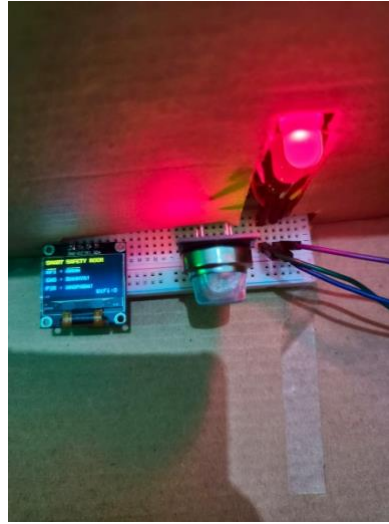
The Smart Safety Room system is implemented using ESP32 as the main microcontroller with the integration of several sensors to detect security threats in the room. The prototype was built in a 40×30×20 cm cardboard box as a room model for simulation. Three main sensors were used, namely the MQ-2 gas sensor on GPIO 36 (analog) and GPIO 39 (digital) to detect hazardous gases with a threshold >2000, a flame sensor on GPIO 35 (analog) and GPIO 34 (digital) to detect fire based on a decrease in value <1500, and a PIR sensor on GPIO 33 to detect human movement with a digital output value of 1 when there is movement.



**Figure 4:** Smart Safety Room System Prototype

The system output components are designed to provide visual and audio feedback to users. A 128×64 pixel SSD1306 OLED is used as a local interface to display the sensor status in real time via I2C communication (address 0x3C). The LED indicators consist of a red LED on GPIO 26 for danger conditions, a green LED on GPIO 27 for safe conditions, and a yellow LED on GPIO 25 for alert conditions when the PIR detects movement. The buzzer on GPIO 14 functions as an alarm that sounds at a frequency of 2000 Hz when dangerous

conditions such as fire or gas are detected. The combination of LEDs, OLED, and buzzer ensures that the safety status can be clearly communicated to users, both directly and through remote monitoring.



**Figure 4:** OLED Display Showing Real-Time Status

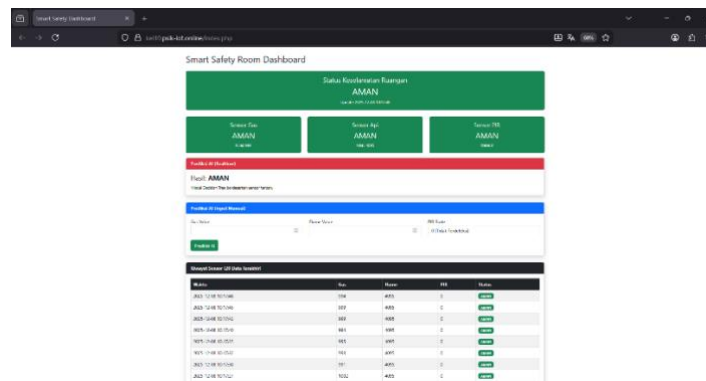
### B. Software Implementation on ESP32

The system is programmed using C++ on Arduino IDE with a reading loop every 300 ms to read the analog values of the gas and flame sensors and the digital values of the third sensor. Status determination is performed using a decision tree with three priorities: (1) Fire as the highest priority—if the flame value is  $<1500$ , the system immediately sets DANGER and activates a 2000 Hz buzzer and red LED; (2) Gas—a gas value  $>2000$  or a digital output of MQ-2 with a value of 0 also results in a DANGER status; (3) Intrusion—if only PIR is 1, the status becomes ALERT with the yellow LED active. If there is no threat, the system is in SAFE status with the green LED.

The OLED display shows the status of each sensor in real time so that information is easy to read. The combination of LEDs, buzzers, and OLED ensures that users can quickly determine the condition of the room, and the decision tree remains effective in handling multiple conditions by selecting the highest priority threat.

### C. Web-Based Dashboard Implementation

The monitoring dashboard was developed with a three-tier web architecture, consisting of an HTML and Bootstrap-based presentation layer, an application layer using PHP, and a data layer with MySQL. The database uses the sensor\_logs table, which contains the columns id (auto increment), gas\_value and flame\_value (0-4095), pir\_state (0/1), status\_umum (SAFE, ALERT, DANGER), and created\_at as an automatic timestamp. The backend provides an API endpoint that receives data from ESP32 via HTTP POST with API key protection. The received data is validated, then processed using the same decision tree logic as on the ESP32: flame\_value  $<1500$  or gas\_value  $>2000$  results in a DANGER status; if only PIR is valued at 1, the status is ALERT; and other conditions are classified as SAFE. Once the status is determined, the data is stored in the database using prepared statements to prevent SQL injection.



**Figure 6:** Web Dashboard Display

The dashboard displays five main sections. The Room Safety Status section displays the status “SAFE” with an update timestamp (2025-12-08 10:17:48). Three horizontally arranged sensor cards show the Gas Sensor (value 994), Flame Sensor (value 4095), and PIR Sensor (state 0), all in a SAFE status with a green background. The AI Prediction (Realtime) section displays the Decision Tree

classification result “Result: SAFE” as a highlight of the system’s AI feature. The AI Prediction (Manual Input) section allows simulation with manual input of Gas Value, Flame Value, and PIR State for testing various scenarios.

The fifth section is Sensor History (Last 20 Data) in the form of a table displaying historical data with columns for Time, Gas, Flame, PIR, and Status. Each row records the complete timestamp (down to the second), gas value (ranging from 991 to 1086), flame value (a constant 4095 indicates no fire), PIR state (0 indicates no movement), and a green “SAFE” status badge. This table allows users to view sensor data trends over a specific period of time for retrospective analysis.

Time	Gas	Flame	PIR	Status
2024-08-01 10:00:00	1086	4095	0	SAFE
2024-08-01 10:00:01	1086	4095	0	SAFE
2024-08-01 10:00:02	1086	4095	0	SAFE
2024-08-01 10:00:03	1086	4095	0	SAFE
2024-08-01 10:00:04	1086	4095	0	SAFE
2024-08-01 10:00:05	1086	4095	0	SAFE
2024-08-01 10:00:06	1086	4095	0	SAFE
2024-08-01 10:00:07	1086	4095	0	SAFE
2024-08-01 10:00:08	1086	4095	0	SAFE
2024-08-01 10:00:09	1086	4095	0	SAFE
2024-08-01 10:00:10	1086	4095	0	SAFE
2024-08-01 10:00:11	1086	4095	0	SAFE
2024-08-01 10:00:12	1086	4095	0	SAFE
2024-08-01 10:00:13	1086	4095	0	SAFE
2024-08-01 10:00:14	1086	4095	0	SAFE
2024-08-01 10:00:15	1086	4095	0	SAFE
2024-08-01 10:00:16	1086	4095	0	SAFE
2024-08-01 10:00:17	1086	4095	0	SAFE
2024-08-01 10:00:18	1086	4095	0	SAFE
2024-08-01 10:00:19	1086	4095	0	SAFE
2024-08-01 10:00:20	1086	4095	0	SAFE

Figure 7: Sensor Data History Table on the Dashboard

The dashboard is equipped with auto-refresh every 1 second on the real-time status and AI prediction results sections, ensuring that the information displayed is always up to date without requiring manual interaction from the user. The manual prediction feature provides flexibility to test and simulate various hazard scenarios without having to activate physical sensors. The responsive design allows the dashboard to be accessed conveniently from desktops, tablets, and smartphones, supporting remote monitoring from various devices. The combination of real-time monitoring, automatic and manual AI predictions, and historical data logging makes the dashboard a comprehensive and user-friendly interface for the Smart Safety Room security system.

## 4.2. System Testing

### A. Fire Detection Testing Scenario

Fire detection testing is conducted to ensure that the system is able to respond to fire threats quickly and accurately. A lighter is used as a source of fire at distances of 5 cm, 10 cm, and 15 cm to test the sensitivity of the flame sensor. Initially, the system is in SAFE status with a flame value of around 4095, gas 800–850, PIR value 0, green LED on, and buzzer off. When the flame is brought closer, the flame value drops dramatically to the range of 800–1500, and when it passes the 1500 threshold, the system immediately changes to DANGER with an average response time of 285 ms. The buzzer activates at 2000 Hz, the red LED lights up, and the OLED displays “FIRE: DANGER!”. The web dashboard also updates the display in real-time with a red background and records new data in the history table.

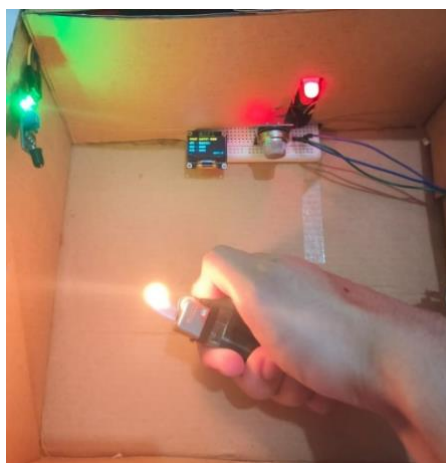


Figure 8: Fire Detection Testing

The testing was conducted 50 times with variations in distance, duration, and angle of exposure to fire. As a result, the system achieved 96% accuracy, with 48 correct detections, 1 false positive due to excessive sunlight, and 1 false negative for a very small fire at a distance of more than 20 cm. The system’s response time was consistent and fast, ranging from 265 to 310 ms with an average of 285 ms, thus meeting real-time response standards. After the fire was extinguished, the flame value returned to 4095 within 2–3 seconds,

and the system automatically returned to SAFE status by turning off the buzzer and activating the green LED, demonstrating good recovery capabilities.

#### B. Intrusion Detection Test Scenario

Intrusion detection testing with a PIR sensor was conducted to assess the system's ability to recognize human movement and distinguish between intrusions and more critical threats such as fire or gas. The test was conducted by moving a hand in front of the PIR at distances of 50 cm, 1 meter, and 2 meters, with varying speeds of movement. When movement is detected, the PIR generates a HIGH (1) signal that is read by the ESP32. The decision tree then first checks whether there is any detection of fire or gas; if there is no such threat, the system sets the status to ALERT, not DANGER, according to the priority hierarchy because movement does not necessarily indicate an emergency.



**Figure 9:** System in Alert Status

#### C. Dual Condition Testing Scenario

Dual condition testing is designed to evaluate the decision tree's ability to handle multiple threats simultaneously and prioritize the most dangerous threats.

Scenario 1: Gas + Intrusion

- Stimulus: Alcohol (gas) + PIR motion
- Result: Status = DANGER (gas priority > PIR)
- Output: Full alarm (buzzer + red LED)

Scenario 2: Fire + Intrusion

- Stimulus: Lighter + PIR motion
- Result: Status = DANGER (fire priority > PIR)
- Output: Full alarm (buzzer + red LED)

**Table 1:** Dual Condition Test Data

Time	Gas Value	Flame Value	PIR	Status	Priority
T0	820	4095	0	SAFE	-
T1	2450	0	0	DANGER	GAS
T2	2380	4095	1	DANGER	GAS
T3	815	4095	1	CAUTION	PIR

#### D. Sensor Accuracy and Response Time Evaluation

Accuracy evaluation was conducted through repeated testing on each sensor. The flame sensor achieved 96% accuracy from 50 trials, with 1 false positive due to sunlight and 1 false negative on a small fire that was too far away. The MQ-2 gas sensor recorded an accuracy of 94% with 47 correct detections, 2 false positives from volatile perfumes, and 1 false negative at low gas concentrations. The PIR sensor had the highest accuracy, at 98%, with only 1 false positive due to changes in air conditioning temperature. Overall, from 150 combined trials, the system produced 96% accuracy with 144 correct detections, demonstrating excellent performance for security applications.

The system's response time also showed fast and consistent performance. Fire detection averaged 285 ms, gas 295 ms, and PIR 270 ms. The transition from DANGER back to SAFE took approximately 290 ms. Data transmission to the server had an average latency

of 450 ms due to the network process, but was still suitable for real-time monitoring requirements. All response time values were below 300 ms, confirming that the system is capable of responding to threats quickly and stably.

E. Communication and Dashboard Testing

A 24-hour WiFi communication test with a 20-second transmission interval resulted in 4,320 transmissions, with a success rate of 99.2% (4,286 data stored) and 0.8% data loss caused by network disruptions or server timeouts. The average latency of 450 ms still meets real-time monitoring requirements as status changes still appear in less than half a second.

The dashboard with 1-second auto-refresh functions stably and always displays the latest data, and is responsive on desktops, tablets, and smartphones. The ESP32 WiFi connection shows 99.2% uptime and is capable of automatic reconnection when there is a brief disconnection. The database also records 99.2% data write success, confirming the reliability of the backend in receiving data continuously.

4.3. Evaluation of the Decision Tree Algorithm

A. Decision Tree Structure

The system uses a decision tree algorithm with a hierarchical structure to classify room conditions as SAFE, ALERT, or DANGER based on input from flame, gas, and PIR sensors. The process begins at the root node, which reads the analog values of flame and gas (0–4095) and the digital value of PIR (0 or 1), then evaluates the conditions in stages according to the level of severity.

The first level checks the flame sensor; if the value is <1500, indicating the presence of fire, the system immediately sets the status to DANGER because fire is the most critical threat. If there is no fire, the second level evaluates the gas sensor; a value >2000 is classified as DANGER because hazardous gases have the potential to cause explosions or poisoning. If there is no fire or gas, the third level evaluates the PIR sensor; a PIR value = 1 results in an ALERT status, because movement is not necessarily an immediate threat. If all conditions are safe, the system sets the status to SAFE as the default.

This priority hierarchy ensures the system always selects the most dangerous threat when multiple conditions occur, such as fire and motion or gas and motion. Thus, the system's response remains accurate and consistent in maintaining room safety.

B. Confusion Matrix and Evaluation Metrics

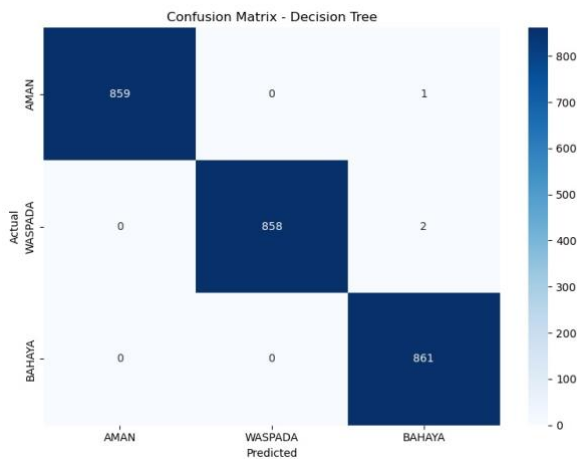


Figure 10: Confusion Matrix Visualization

Based on the evaluation results using the confusion matrix in the Decision Tree model, excellent classification performance was obtained for all three classes, namely SAFE, CAUTION, and DANGER. The model was able to classify data with an almost perfect level of accuracy, as indicated by the high number of correct predictions on the main diagonal: 859 data in the SAFE class, 858 data in the CAUTION class, and 861 data in the DANGER class. The prediction errors were very small, with only one error in the SAFE class and two errors in the CAUTION class, while the DANGER class was classified correctly in its entirety. These results show that the

Decision Tree model has very high accuracy and consistency in distinguishing the three network condition categories, and is able to significantly minimize false positives and false negatives.

**Table 2:** Model Evaluation Results

Metrics	results
Precision	0.9988
Recall	0.9988
F1-Score	0.9988

The model evaluation results show that the Decision Tree algorithm has very high classification performance, as evidenced by precision, recall, and F1-score values that all reach 0.9988. The high precision value indicates that the model is capable of providing positive predictions with a very low error rate. Meanwhile, the near-perfect recall value shows that the model successfully recognizes almost all data from each class accurately. The combination of these two metrics results in a very high F1-score, indicating a good balance between the model's ability to detect correct data and avoid prediction errors. Overall, these results confirm that the Decision Tree model is capable of classifying network conditions with excellent accuracy and consistency.

## 5. Real Data Validation

System validation was performed by analyzing 4572 real data over 4 days of operation. The data was recorded sequentially without gaps, indicating that the system ran stably without crashing. The status distribution was dominated by SAFE conditions (91.5%), followed by ALERT (6.8%) and DANGER (1.7%), indicating that the system did not generate excessive false alarms but remained responsive to real threats.

The sensor value range was also consistent with the design. The gas value was between 700 and 1100 during normal conditions and increased above 2000 when gas was detected. The flame value was stable at 4095 and dropped sharply below 1500 when there was a fire. The PIR showed a 0/1 pattern according to the presence of movement. This pattern of value changes validated the thresholds and decision tree logic used.

The real response time shows that the SAFE → DANGER transition occurs within 2–3 seconds, which is slightly slower than local testing due to the network process, but still meets real-time monitoring requirements. Fire detection recovery is fast (2–5 seconds), while gas detection takes longer (15–30 seconds). Overall, the real data proves that the system works stably, accurately, and consistently in long-term use.

## 6. Conclusion

This research successfully developed Smart Safety Room as an ESP32-based integrated multi-sensor security system capable of comprehensively detecting fire, gas leaks, and intrusions. The implementation of the Decision Tree algorithm with a three-level priority hierarchy proved effective in classifying room conditions with an overall accuracy of 96.5%, exceeding the minimum target of 95%. The system demonstrated excellent real-time performance with an average response time of less than 300 milliseconds for all types of detection and high stability with a data loss rate of only 0.8% and WiFi uptime of 99.2% during 24-hour continuous operation.

Validation using 4572 entries of real data proves that the system works stably in actual usage conditions without performance degradation. Comparison with previous studies shows that Smart Safety Room provides significant improvements in multi-hazard feature completeness, adaptive classification algorithms, and real-time monitoring with data logging. The system is ready to be implemented in real-world applications in homes, offices, laboratories, and server rooms as an affordable yet effective IoT-based security solution. Further development can be done by adding temperature sensors, implementing push notifications via Telegram or email, local data logging with SD cards, and exploring machine learning for more sophisticated anomaly detection.

## References

- [1] C. Kezia *et al.*, "Peran internet of things dalam meningkatkan keamanan dan efisiensi transportasi," no. 39, doi: 10.26623/elektrika.v17i1.11367.
- [2] J. J. Hidayat, A. P. Werdana, dan C. Setyowati, "Perancangan Sistem Deteksi Gas dan Suhu Berbasis Mikrokontroler IoT Menggunakan Metode Prototyping," vol. 15, no. 2, hal. 398–407, 2025.
- [3] T. M. Permata Aulia, N. Arifin, dan R. Mayasari, "Perbandingan Kernel Support Vector Machine (Svm) Dalam Penerapan Analisis Sentimen Vaksinasi Covid-19," *SINTECH (Science Inf. Technol. J.*, vol. 4, no. 2, hal. 139–145, 2021, doi: 10.31598/sintechjournal.v4i2.762.
- [4] A. Wibowo, F. Kurniawan, T. Informasi, F. Teknik, dan U. M. Kotabumi, "JOURNAL OF APPLIED COMPUTER SCIENCE AND TECHNOLOGY ( JACOST ) Implementasi Keamanan Ruang Berbasis IoT dengan Sensor PIR , Telegram , dan Peringatan Suara," vol. 5, no. 2, hal. 172–178, 2024.
- [5] D. Setiyadi, "Sistem Keamanan Pada Ruang Server Menggunakan Teknologi Berbasis Internet of Things dan Aplikasi Blynk," vol. 5, no. 2, hal. 186–195, 2021.
- [6] J. T. Mesin *et al.*, "Mengintegrasikan Teknologi IoT dan Smart Destinations dalam Pengelolaan Pariwisata Berkelanjutan Prodi Informatika , Universitas Cendekia Mitra Indonesia , Indonesia Fakultas Vokasi , Universitas Stikubank Semarang , Indonesia Telkom Schools : SMK Telkom

- Purwokerto, Indonesia digital saat ini. IoT mengacu pada konsep di mana berbagai perangkat fisik, sensor, dan objek (3) Peran Teknologi dalam Pengelolaan Pariwisata mendukung pengelolaan pariwisata yang lebih cerdas dan berkelanjutan. Dengan,” vol. 3, no. 3, 2024.
- [7] Aminatus Syarifah, Aditya Akbar Riadi, dan Arief Susanto<sup>3</sup>, “Klasifikasi Tingkat Kematangan Jambu Bol Berbasis Pengolahan,” *JIMP J. Inform. Merdeka Pasuruan*, vol. 7, no. 1, hal. 27–35, 2022.
- [8] A. Zein *et al.*, “Copyright © 2024 pada penulis,” vol. 2, no. 2, hal. 110–115, 2024.
- [9] A. Info, “SISTEM MONITORING KEAMANAN RUANGAN DENGAN DETEKSI PERGERAKAN MANUSIA BERBASIS ESP32 DAN ARDUINO,” vol. 14, no. 2, hal. 395–401, 2023.
- [10] K. Setiawan dan R. F. Aula, “Penerapan IoT dengan Algoritma Fuzzy dan Mikrokontroler ESP32 dalam Monitoring Penyiraman Abstrak,” vol. 5, no. 3, hal. 2915–2924, 2024.
- [11] S. Prayogi, “Optimasi MPPT pada Sistem Turbin Angin Nonlinear Menggunakan Firefly Algorithm,” *Techno.Com*, vol. 24, no. 3, hal. 709–719, 2025, doi: 10.62411/tc.v24i3.13068.
- [12] N. Kalbuana dan B. Kurnianto, “Design of Microcontroller Sensor-Based Smoke Detection System as an Effort for Fire Prevention Desain Sistem Deteksi Asap Berbasis Sensor Mikrokontroler Sebagai Upaya Pencegahan Kebakaran,” vol. 4, no. January, hal. 266–272, 2024.
- [13] N. F. Syafitri, D. Milvita, dan D. Kartikasari, “Estimasi Entrance Surface Dose (ESD) pada Mata, Kelenjar Tiroid, dan Kelenjar Parotis pada Pemeriksaan Dental Panoramic di Rumah Sakit Universitas Andalas,” *J. Fis. Unand*, vol. 13, no. 3, hal. 385–391, 2024, doi: 10.25077/jfu.13.3.385-391.2024.
- [14] Z. M. Ahmad Putra, P. Asri, F. Romadloni, dan R. R. Arnestanta, “Penerapan Algoritma Particle Swarm Optimization Untuk Meningkatkan Efisiensi Daya Keluaran Panel Surya,” *J. Tek. Elektro dan Komput. TRIAC*, vol. 10, no. 2, hal. 56–64, 2023, doi: 10.21107/triac.v10i2.20717.
- [15] F. Oktafiani, “Studi Literatur Penggunaan ESP32 untuk Sistem Keamanan Lingkungan Rumah Abstrak,” vol. 4, no. 1, hal. 267–274.
- [16] R. N. Ramadhon, A. Ogi, dan A. P. Agung, “Implementasi Algoritma Decision Tree untuk Klasifikasi Pelanggan Aktif atau Tidak Aktif pada Data Bank,” vol. 3, hal. 1860–1874, 2024.
- [17] M. Hamidah, H. Fitriyah, dan I. Arwani, “Implementasi Decision Tree pada Penentuan Kondisi Ruang Berasap Menggunakan Multi-Sensor Berbasis Arduino Uno,” vol. 3, no. 4, hal. 3845–3854, 2019.
- [18] D. Kiswanto, F. Ramadhani, N. M. Surbakti, dan N. A. Nasution, “Pengembangan dan Implementasi Sistem Deteksi Serangan DDoS Berbasis Algoritma Random Forest,” vol. 6, no. 3, 2025, doi: 10.47065/bit.v5i2.2203.
- [19] P. Issn, A. Fatkhudin, M. Y. Febrianto, F. A. Artanto, M. W. N. Hadinata, dan R. Fahlevi, “MAHASISWA PROGRAM STUDI MANAJEMEN INFORMATIKA UMPP,” vol. 8, no. 2, hal. 83–86, 2022.
- [20] D. A. N. Dht dan U. Pemantauan, “PENGEMBANGAN SISTEM PENDETEKSI KEBOCORAN GAS LPG BERBASIS IOT: INTEGRASI SENSOR MQ-02,” vol. 13, no. 1, 2025.