

# Optimization of ESP32-Based Living Room Security System and PIR Sensors, with FTDI through real-time notifications

**Ahmad Supyan**

*Information Technology  
Education*

STKIP PGRI Situbondo  
Indonesia

[Sufyanurhayati123@gmail.com](mailto:Sufyanurhayati123@gmail.com)

**Nur Azizah**

*Information Technology  
Education*

STKIP PGRI Situbondo  
Indonesia

[Nazizah0606@gmail.com](mailto:Nazizah0606@gmail.com)

**Firman Jaya**

*Information Technology  
Education*

STKIP PGRI Situbondo  
Indonesia

[Altamis1922@gmail.com](mailto:Altamis1922@gmail.com)

**Abstract:** This study addresses the problem of undetected falls and hazardous incidents in household living rooms, especially for child and elderly users who are prone to slipping on wet and narrow surfaces. The study aimed to design and implement a smart living room safety monitoring system using ESP32 microcontrollers, PIR sensors, magnetic sensors, load cells with HX711, and MPU6050 connected to the Internet of Things to provide real-time notifications to caregivers via mobile apps. This methodology follows a prototype-based IoT engineering approach, starting with a literature review and needs analysis, followed by hardware-software design, prototyping, iterative testing, and final evaluation in a simulated living room environment for various fall scenarios. The experimental data consisted of PIR logs, weight changes, system response time, and environmental conditions, which were statistically analyzed to determine the system's accuracy, reliability, and responsiveness. The results showed that the prototype was able to detect suspicious movement patterns and falls with good accuracy and trigger local alarms and Telegram notifications within about 2–3 seconds, while still functioning in poor living room conditions. It can be concluded that the proposed system meets the research objectives of a low-cost and privacy-preserving living room safety solution for smart homes, with future work directed at integrating machine learning-based fall detection and expanding communication options beyond WiFi to improve resilience in various residential environments.

**Keywords :** ESP32, Security Systems, PIR Sensors, FTDI, Internet of Things, Real-Time Notifications.

*Article info: Date Submitted: 2025-12-29 | Date Revised: 2026-01-07 | Date Accepted: 2025-02-14*

*This is an open access article under the CC BY-SA license*



## I. INTRODUCTION

Living room security is a crucial aspect of modern home security systems, as the living room is often an area that is easily accessible to unauthorized parties. An effective security system must be able to detect suspicious movements and provide real-time alerts to homeowners so that they can respond quickly and appropriately[1]. However, many traditional systems such as CCTV or stand-alone alarms only provide local or visual warnings without instant notifications, which can delay user responses to potential threats [2][3]. Several previous studies have shown that the integration of PIR sensors with IoT-based microcontrollers such as the ESP32 can improve motion detection performance and enable automatic notifications in home security applications[4][5].

Despite these advances, many existing home security systems still lack responsive and reliable real-time notification mechanisms for end users. Most systems focus on basic intrusion detection and do not provide sufficient connectivity to deliver timely information through mobile applications or communication platforms[6][7]. As a result,

users may not receive critical alerts quickly enough to take immediate action in emergency situations[8][9].

PIR (Passive Infrared) sensors have been widely recognized as an effective and low-cost solution for motion detection because of their ability to sense changes in infrared radiation emitted by the human body. When combined with microcontrollers such as the ESP32 that provide built-in Wi-Fi connectivity, PIR sensors can support the development of responsive security systems capable of transmitting sensor data and alerts to cloud platforms or user applications. Several studies have integrated ESP32- and PIR-based intrusion detection systems with communication platforms such as Telegram to deliver automatic notifications when motion is detected[10][11].

This research introduces an integrated living room safety monitoring system that combines ESP32 microcontrollers, PIR sensors, and FTDI-based data communication to improve detection reliability and notification stability. Compared to previous approaches that only relied on basic PIR-based detection, the incorporation of FTDI enables more stable data transmission and supports more accurate motion analysis in indoor environments[12][13]. The proposed system not only detects abnormal movements automatically but also optimizes the interaction between physical devices and real-time notification services with minimal manual intervention[14][15].

Although various studies have implemented ESP32 and PIR sensors for home security and motion detection systems, most existing works primarily focus on intrusion detection at entrances or rely on single-sensor approaches. Very limited studies specifically address fall reflex detection in living room environments using multi-sensor integration combined with real-time IoT-based notifications. In addition, the optimization of data communication stability using FTDI in ESP32-based smart home security systems has rarely been discussed in previous literature. Therefore, this study fills this research gap by proposing a multi-sensor ESP32-based living room safety monitoring system that integrates PIR sensors, supporting sensors, and optimized real-time notification delivery to improve detection accuracy and system responsiveness[16][17].

This study provides several important contributions to the development of smart home security and safety monitoring systems. This research presents the design and implementation of a low-cost ESP32-based living room monitoring system that integrates multiple sensors to improve detection reliability. In addition, a simple fall reflex detection approach based on PIR signal duration and movement frequency is proposed to enable timely identification of abnormal events in indoor environments. Furthermore, this study optimizes real-time notification delivery through IoT-based Telegram integration, achieving fast and stable alert transmission to users, and provides experimental results that demonstrate the feasibility of the proposed system for real-time living room safety monitoring[18][19].

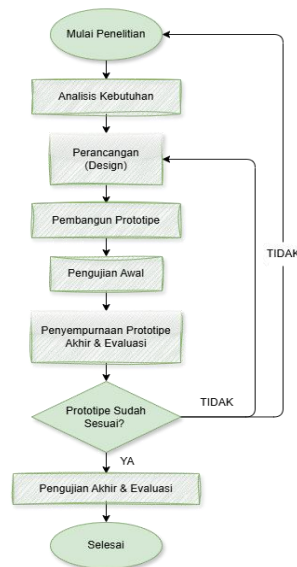
The purpose of this study is to develop and implement a smart living room safety monitoring system that uses ESP32 and Internet of Things-based PIR sensors to identify hazardous events in the living room, such as fall reflexes in adults and children and minor or major accidents. When abnormal movement patterns are detected, the system sends real-time notifications to enable immediate intervention through a mobile application. Further objectives include evaluating detection accuracy and integrating the system with simple notification platforms such as Telegram[20].

## II. METHODS

This study uses a prototype method to design and develop an ESP32-Based Living Room Security System Optimization and PIR Sensor, with FTDI through real-time notifications[21][16]. The development process began with the creation of an initial model consisting of an ESP32 microcontroller as a control center, a PIR sensor to detect human presence and movement, and a WiFi communication module to transmit data and

notifications in real-time. Furthermore, the prototype was tested and improved iteratively based on the results of the sensor fungsi i testand system response[22][23].

In addition, this system is integrated with the Telegram application to receive alerts or notifications if suspicious movements or incidents are detected, such as falling in the living room[24][25].The hardware development process includes strategically installing PIR sensors in the living room, then programming the ESP32 to read the sensor signals, processing data to distinguish between normal activity and fall events, and automatically sending notifications to the Telegram app. This prototype approach emphasizes ease of use, cost efficiency, and high sensitivity in humid living room environmental conditions.



Bagan 1. Prototype Method Flow

The diagram depicts the stages of prototype-based research from start to finish sequentially and cyclically (iterative). Each box shows one main activity, while the decision indicates the point at which the decision is made whether the results are appropriate or need to be repeated.

#### A. Start Research

The research begins by establishing the context of the problem and the objectives to be achieved. At this stage, the researcher determines the scope, limitations, and reasons why the research needs to be conducted.

#### B. Needs Analysis

At this stage, consider user needs, technical limitations, and system needs. The researcher found the user, the usage scenario, the functionality that should be available, and the expected performance. They also find barriers such as budget, time, and device availability. The result is usually a list of functional and non-functional needs, which includes things like reliability, security, ease of use, and so on.

#### C. Design

The researcher prepares a conceptual and technical design according to the requirements. System block diagrams, hardware and software architectures, internal workflows, interface designs, and component specifications can be included in the design. At this stage, the researcher also determines how each part of the system interacts with each other to meet the needs that have been formulated.

The system is designed with a multi-sensor architecture – edge processing – IoT notification. The ESP32 acts as a control center that receives data from several sensors:

The system is designed with a multi-sensor architecture – edge processing – IoT notification. The ESP32 acts as a control center that receives data from several sensors:

1. PIR HC SR501 as human motion detector in the living room;
2. Magnetic Sensor to monitor the status of the door (open/closed) so that the system knows when the living room is in use;
3. load cell + HX711 to measure load changes on the floor (optional, e.g. in footwear/anti-slip mat);
4. MPU6050 as an additional acceleration sensor to recognize more specific fall motion patterns. The ESP32 then processes the data with threshold-based fall detection algorithms (signal duration, weight changes, and acceleration patterns), activates buzzers and LEDs as local alarms, and sends warning messages to the Telegram app via a home WiFi connection.

On the hardware side, all sensors are connected to the corresponding pins on the ESP32. The PIR and

The PIR and magnetic sensor are connected to digital pins to be read as HIGH/LOW logic. The HX711 is connected to two digital pins (DT and SCK) for communication with the load cell. MPU6050 is connected to the I2C (SDA and SCL) lines. The buzzer and LED are connected to the output digital pin via a current-limiting resistor for sound and visual indicators. The 5 V/3.3 V power supply is set up via a regulator module or adapter, with ground wiring integrated to ensure stable sensor readings. The entire circuit is housed in a moisture-resistant enclosure, with the PIR position directed to the main area of the living room, the load cell is placed under the footing mat, and the magnetic sensor is mounted on the door leaf as well as the frame.

#### **D. Prototype Builder**

Currently, the design is implemented into a testable prototype form. Researchers begin to create hardware, code programs, organize networks or databases, and then combine all the components into one whole system. Prototypes are usually not ideal, but they are enough to demonstrate the main functions and serve as the basis for testing.

#### **E. Initial Testing**

At this stage, the prototype is tested to ensure that the main function meets expectations. Testing features individually, testing sensor and actuator responses, or running some common scenarios for use are some examples of simple scenarios that can be used to perform tests. The initial test results will indicate bugs, failures, or inconsistencies with the user's needs.

#### **F. Decision: prototype**

After initial testing, the researcher checks whether the prototype meets the requirements at the needs analysis stage. If there are a lot of errors, unfinished functions, or performance that doesn't meet the target, then the answer is "not yet". However, if all key needs are met and performance is below the accepted limit, the process can resume.

#### **G. Enhancement(If not appropriate)**

If the prototype does not meet the expected standards, the researcher will proceed to the improvement phase. At this stage, code revisions, design adjustments, replacement or addition of elements, algorithm optimization, and improvements to the interface and usability factors are carried out. This phase is repetitive: after the repair process, the prototype will be tested again in "Initial Testing" to assess the

success of the repairs made. The "Initial Testing → Decision → Refinement" cycle can be repeated several times until the prototype reaches the desired level of quality.

**H. Final testing & Evaluation**

Once the prototype has been declared eligible, the next step is to conduct a final evaluation with a broader scenario that is closer to reality. This evaluation can include thorough functional testing, load testing, safety testing, reliability testing within a given period, and testing with actual users. The data from the test is analyzed to determine the extent to which the research goals are achieved, how much improvement compared to previous methods or tools, and what are the advantages and disadvantages. The evaluation process also includes an analysis of the system's limitations and future development opportunities.

**I. Finished**

The last stage is the preparation of reports and documentation of the research results. The researcher reviews the stages from the needs analysis to the final test, explains the results of the evaluation, draws conclusions, and provides recommendations for future development. In this way, the research process is considered complete, although in reality, the system can still be further improved in future research.

The dataset in this study contains a dataset generated directly by the device (ESP32, PIR sensor, and supporting sensor such as load cell/proximity sensor) when the type prototype is tested in a living room-like environment. This data is then analyzed to assess the performance of the system.

**A. Dataset structure of PIR sensor**

Each time a prototype is run, the PIR sensor outputs data recorded by the ESP32 in the form of a log row. For one event, at least the following:

- TimeStamp: the event time (date & time) of the RTC/system time.
- pirState: PIR status (0 = no motion, 1 = no motion).
- Detection time: the time difference from the beginning of monitoring to the first PIR HIGH (ms).
- durationHIGH: how long PIR lasts HIGH (indication of movement duration, it is very important to distinguish between normal and falling movements).
- Scenario: Event labels (normal, child fall, adult fall, steam/false).

Tabel 1. Example of PIR Dataset Schema:

TimeStamp	PirState	WaktuDeteksi (ms)	DurasiHIGH(s)	Skenario
2025-12-10 10:01:05	1	150	1,8	Normal
2025-12-10 10:03:12	1	250	6,2	Child Fall
2025-12-10 10:05:47	1	185	8,9	Fall Adult

This dataset is used to calculate the detection accuracy, the average duration of the fall, and the threshold separating between normal motion and emergency events.

**B. Dataset of load cell / supporting sensor**

If the prototype also uses load cells, then each detected event will be accompanied by heavy data:

- rawADC: the analog reading value of the HX711 module.
- weightKg: the result of the conversion of rawADC to kilograms (after calibration).
- CategoryPeople: classification results, e.g. "Minor" if < 50 kg, "Adult" if ≥ 50 kg.

Tabel 2. Weight Sensor Scheme

TimeStamp	rawADC	Weight kg	Category peoply
2025-12-10 10:03:12	73412	28	Young Children
2025-12-10 10:05:47	89210	67	Adults

This dataset was used to analyze whether the system could distinguish victims (children/adults) and whether the weight readings were stable in humid conditions.

**C. System response time dataset**

The response time dataset measures the overall performance of the system from detection to IoT notifications. For each trial, the following are stored:

- deteksiPIR\_ms: time from occurrence to PIR HIGH.
- prosesESP32\_ms: the time of processing logic and message preparation.
- send\_notification\_ms: the time it takes to send to the server/Telegram until it is sent.
- total\_seconds: the total amount in seconds.

Tabel 3. System Response Time

NO	Component	Type	ConnectedESP32	Main function
1.	ESP32 Dev Board	Microcontroller	-	Control Center & Data Processor
2.	HX711	ADC Amplifer	DT, SCK	Strengthening & Reading load cells
3.	Load Cell	Weight sensor	HX711	Measuring load/body
4.	PIR HC-SR501	Motion sensor	Digital pin	Detects human movement.
5.	Buzzer	Aktuator	Digital pin	Alarm/audio notification
6.	LED + Resistor	Indikator	Digital pin	Status indicator
7.	MPU6050	Additional sensors	SDA, SCL	Environmental/motion sensor (customized type)

Trial	Detekcion PIR ms	Process ESP32 ms	Send notifikasion	Total seconds
1	150	500	1800	2,45
5	120	450	1650	2,22
10	200	600	2200	3,00
Avg	±157	±517	±1883	±2,56

This dataset was used for statistical analysis (mean, standard deviation) and compared the speed of the system with other studies.

This table presents a list of the main components involved in a security monitoring system for a smart living room using ESP32 as well as the functions of each component. The ESP32 Dev Board acts as a control center that regulates all processes and processes data from various sensors. The load cell functions as a weight sensor that is not directly connected to the ESP32, but rather through the HX711 module which functions to amplify the analog signal from the load cell and convert it into digital data that can be understood by the ESP32. The HC-SR501's PIR sensor is connected to a digital pin on the ESP32 to detect human movements, for example when a person falls, while the buzzer and LEDs also connected to the digital pin act as actuators: the buzzer provides a sound signal and the LED (with a current-limiting resistor) serves as a visual indicator of the status of the system.

### III. RESULT AND DISCUSSION

The results of the study show that the Prototype method is able to detect events in the living room through a combination of PIR sensors (detecting movement/fall reflex) and, when used, supporting sensors such as load cells or other sensors to enrich information on who fell (child or adult). The system is controlled by the ESP32, which also functions as a WiFi communication module to send notifications to mobile phones (e.g. via Telegram Bots or other IoT platforms). Thus, the system not only provides a

local alarm via buzzers and LEDs, but also warns families or caregivers remotely.

#### A. Implimentation of a bot based telegram notification sender

The implementation of Telegram Bot-based notification delivery on this system is carried out by making ESP32 an internet client connected to the user's Telegram bot. First, the researchers created a bot on Telegram (using BotFather) to get the bot token and chat ID. The token and chat ID were then written in the ESP32 program as variables so that the microcontroller knew "to which account" the message should be sent. After that, the ESP32 is connected to the home WiFi network via an initialization command at the beginning of the program; if the connection is successful, ESP32 is ready to send data to the Telegram server.

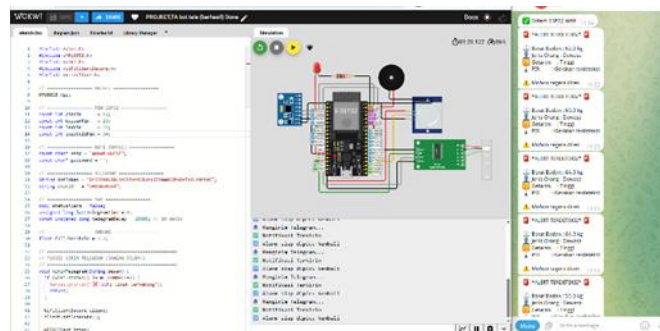


Figure 2 notification from wokwi to telegram

In the Wokwi section, you can see the ESP32 series connected with several components: a blue module (MPU6050 or additional sensor) on the left, a PIR sensor in the upper right, an HX711 module and load cell in the bottom right, as well as LEDs and buzzers at the top of the board. On the left side of the Wokwi display is the C/C++ program code for ESP32 that regulates the sensor pins (pirPin, loadCellPin), actuator pins (buzzerPin, ledPin), WiFi configuration, as well as tokens and chat IDs for Telegram bots. At the bottom you can see text logs such as "Sending Telegram...", "Notifications sent", and "Alarm ready to be retriggered" indicating that whenever

the combination of sensor data meets the criteria (e.g. a certain weight, high vibration, PIR active), the message sender function is executed and the system is ready to detect again.



Figure 3 Notifikasi from wokwi to telegram hp

On the right side of the screen, you can see the Telegram app that receives messages from your bot. Each message in the format of ALERT DETECTED is accompanied by complete parameters: weight (e.g. 60–64 kg), classification of person type (Adult), vibration status (Height), and PIR status (Motion detected), followed by the warning "Please check immediately". This shows that the ESP32 is not just sending a single value, but a packet of processed information from multiple sensors at once. The overall picture proves that the integration between the Wokwi and Telegram simulations is already working: when the prototype detects a condition that is considered dangerous in the living room, the system automatically gives a local alarm (buzzer/LED on the network) as well as sends a real-time notification to Telegram so that the homeowner can respond immediately.

### B. System response time (Led → Real Time)

The LED light up exactly when the detection conditions are met, i.e. inside the block if the detektion falls:

```

if (pirState == HIGH &&
    accel < fallThreshold &&
    !statusAlarm &&
    millis() - lastTelegramTime > telegramDelay) {
    statusAlarm = true;
    lastTelegramTime = millis();

    digitalWrite(ledPin, HIGH);
    tone(buzzerPin, 1000);
}
    
```

Figure 4 Led code on ignition

So, as soon as the system decides that an event has occurred (PIR is active, acceleration below the threshold, there is no alarm yet, and a 10-second pause is passed), the LED is immediately set HIGH and at the same time the sendTelegram (message) function is called to send notifications. The LEDs don't use a special delay( ) but the duration of the light is controlled by automatic reset logic that uses the difference of millis time ( ) to lastTelegramTime:

```

if (statusAlarm && millis() - lastTelegramTime > telegramDelay) {
    statusAlarm = false;
    digitalWrite(ledPin, LOW);
    noTone(buzzerPin);
    Serial.println("🔊 Alarm siap dipicu kembali");
}
    
```

Figure 5 Led codes in turn off

The telegram deley value is set:

```
// ===== VAR =====  
bool statusAlarm = false;  
unsigned long lastTelegramTime = 0;  
const unsigned long telegramDelay = 5000; // 5 detik
```

Figure 6 kode Deley Led

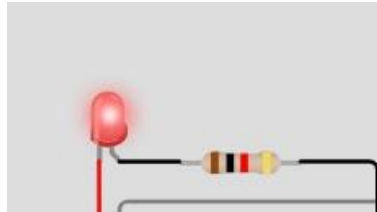


Figure 7 delay led

This means that once the alarm is active and the notification is sent, the LED will remain on for about 5 seconds, and then turn off when `millis() - lastTelegramTime` exceeds 5,000 ms. So the "LED delay" here is a `millis()` based mechanism, not the `delay()` function, and the duration is equal to the `telegramDelay` value.

### C. Explanation of Networks ad components

- The ESP32 as a central microcontroller reads the sensor and controls the actuator.
  - The PIR sensor is connected to on ofthe digital pins (in code: `pirpin = 2`)to deteks human motion(passive infrared radiation changes).
  - The load cell + HX711 module connects to an analog pin (`loadCellPin = A0` in code) to read the change in weight when a person steps on/is depressed in the sensor area.
  - Buzzer (pin 8) as a local sound alarm when a fall is detected
  - LED (pin 7) as a visual indicator that the system is detecting an event.
- Functionally, this series already describes the main logic of the bathroom security system:
- PIR → detect reflex/fall movement.
  - Load cell → estimate the victim's weight and provide context (young vs. adult).
  - Buzzer/LED → Live on site alerts

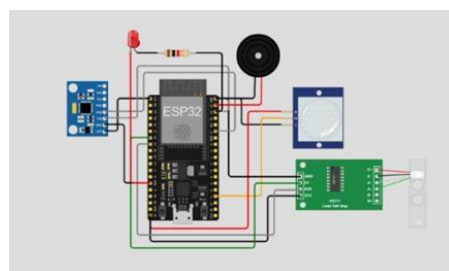


Figure 8 Project Fall Reflex Detection Tool

In the context of research, later Arduino Uno will be replaced by the ESP32, but the idea of sensors and actuators can be maintained.

The image is a complete set of prototypes of the ESP32-based Smart Bathroom Safety Monitoring System, which combines multiple sensors and actuators to detect incidents in the bathroom (falls, load changes, movement) and then issue warnings.

#### IV. CONCLUSION

Based on the design, implementation, and testing of the developed prototype, it can be concluded that the proposed Smart Living Room Security Monitoring System based on ESP32 successfully achieves the research objectives. By integrating PIR sensors, magnetic sensors, load cells via HX711, and additional motion sensors, the system is able to detect potentially hazardous situations such as slipping and fall reflexes in the living room environment and deliver real-time notifications to relatives or caregivers through an Internet of Things platform. The use of multi-sensor integration combined with simple rule-based algorithms implemented on the ESP32 enables the system to effectively distinguish between normal daily activities and fall indications with good detection accuracy, fast response times of only a few seconds, and stable performance under humid indoor conditions. Furthermore, the proposed system offers an affordable, privacy-preserving, and practical home security solution, as it does not rely on camera-based monitoring.

Despite these promising results, several limitations remain. The system performance is still dependent on the availability and quality of the WiFi network, the evaluation was limited to a simulated living room environment, and the detection approach relies on rule-based algorithms without the use of artificial intelligence. Future work will focus on expanding communication capabilities using cellular networks, conducting broader testing in real-world environments with diverse users, and integrating machine learning-based fall detection methods to improve accuracy, adaptability, and robustness under varying living room conditions.

#### REFERECES

- [1] I. P. Dewi and R. Fikri, "Optimalisasi Keamanan Rumah dengan Implementasi Sistem Notifikasi Gerbang Cerdas Berbasis Internet of Things (IoT)," *Journal of Computer System and Informatics (JoSYC)*, vol. 4, no. 4, pp. 816–829, Aug. 2023, doi: 10.47065/josyc.v4i4.4004.
- [2] E. V. Manullang and S. Rumere, "Rancang Bangun Sistem Keamanan Rumah Berbasis Internet of Things (IoT) dengan Menggunakan Sensor PIR dan ESP32-CAM," *Jurnal Teknologi Informasi*, vol. 12, no. 1, pp. 9–15, Jul. 2024, doi: 10.58839/jti.v12i1.1359.
- [3] A. Miranto, "Real Time Object Detection Menggunakan Mobilenet-SSD pada Sistem Keamanan Ruangan dengan Bot Telegram Sebagai Notifikasi User," *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, vol. 13, no. 1, p. 211, Aug. 2024, doi: 10.24843/JLK.2024.v13.i01.p21.
- [4] N. Azizah, R. Jannah, M. Sudur, Z. Rahman, and J. Muhammad, "Meningkatkan Efektivitas Penggunaan Absensi Digital dalam Rekapitulasi Guru di Sekolah Dasar (SD) Desa Trebungan," *Jurnal Masyarakat Berdikari dan Berkarya (Mardika)*, vol. 2, no. 1, pp. 1–9, Mar. 2024, doi: 10.55377/mardika.v2i1.9732.
- [5] Y. Ding, S. Lu, T. Li, Y. Zhu, S. Wei, and Z. Tian, "An indoor thermal environment control model based on multimodal perception and reinforcement learning," *Build. Environ.*, vol. 276, p. 112863, May 2025, doi: 10.1016/j.buildenv.2025.112863.
- [6] B. D. Frengky, P. Gunoto, and E. Susanti, "Perancangan Sistem Monitoring Keamanan Menggunakan ESP32-CAM dengan Notifikasi ke Smartphone," *SIGMA TEKNIKA*, vol. 7, no. 1, pp. 123–130, Jun. 2024, doi: 10.33373/sigmateknika.v7i1.6198.
- [7] C. Gao, Q. Zhang, P. K. Varshney, X. Lin, and H. Liu, "A Distributed Multi-Objective Detection Method for Multi-Sensor Systems With Unknown Local SNR," *IEEE Transactions on Signal Processing*, vol. 73, pp. 649–663, 2025, doi: 10.1109/TSP.2025.3533275.

- [8] F. Yusuf, A. Cahyono, N. Suharto, and L. D. Mustafa, "Design and Build a Home Security System based on an ESP32 Cam Microcontroller with Telegram Notification," *Journal of Telecommunication Network*, vol. 12, no. 2, 2022.
- [9] N. N. P. B, V. K, and M. P, "Fall Detection for Elderly People Using RealSense Depth Camera," in *2025 3rd International Conference on Smart Systems for applications in Electrical Sciences (ICSSSES)*, IEEE, Mar. 2025, pp. 1–6. doi: 10.1109/ICSSSES64899.2025.11010062.
- [10] H. A. Kusuma, S. B. Wijaya, and D. Nusyirwan, "Sistem Keamanan Rumah Berbasis ESP32-CAM dan Telegram sebagai Notifikasi," *Infotronik: Jurnal Teknologi Informasi dan Elektronika*, vol. 8, no. 1, p. 30, Jun. 2023, doi: 10.32897/infotronik.2023.8.1.2291.
- [11] S. Salpina, R. Suppa, and M. Muhallim, "Prototype Sistem Keamanan Rumah Pintar Berbasis IoT," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 1, Jan. 2025, doi: 10.23960/jitet.v13i1.5782.
- [12] A. Ipanhar, T. K. Wijaya, and P. Gunoto, "Perancangan Sistem Monitoring Pintu Otomatis Berbasis IoT Menggunakan ESP32-CAM," *SIGMA TEKNIKA*, vol. 5, no. 2, pp. 333–350, Nov. 2022, doi: 10.33373/sigmateknika.v5i2.4590.
- [13] E. N. Alam, "IoT in Agriculture Industry," *Jurnal Sistem Cerdas*, vol. 3, no. 1, pp. 36–42, May 2020, doi: 10.37396/jsc.v3i1.57.
- [14] A. Y. Yudhistira, H. Fuadi, Adriani, and Ridwang, "Rancang Bangun Smart Detector pada Pintu Rumah untuk Keamanan Rumah Tangga Berbasis Internet of Things," *Arus Jurnal Sains dan Teknologi*, vol. 2, no. 2, pp. 381–390, Oct. 2024, doi: 10.57250/ajst.v2i2.693.
- [15] Nafisah Nurul Hakim, "Implementasi Machine Learning pada Sistem Prediksi Kejadian dan Lokasi Patah Rel Kereta Api di Indonesia," *Jurnal Sistem Cerdas*, vol. 3, no. 1, pp. 25–35, May 2020, doi: 10.37396/jsc.v3i1.58.
- [16] T. Vaiyapuri, E. L. Lydia, M. Y. Sikkandar, V. G. Díaz, I. V. Pustokhina, and D. A. Pustokhin, "Internet of Things and Deep Learning Enabled Elderly Fall Detection Model for Smart Homecare," *IEEE Access*, vol. 9, pp. 113879–113888, 2021, doi: 10.1109/ACCESS.2021.3094243.
- [17] S. Suherman, H. Hermansyah, and J. Syahpita, "Sistem Alarm Deteksi Gerak Berbasis IoT Menggunakan Sensor PIR dan ESP32 dengan Notifikasi Telegram Real-Time," *Jurnal Komputer Teknologi Informasi Sistem Informasi (JUKTISI)*, vol. 4, no. 2, pp. 1469–1476, Nov. 2025, doi: 10.62712/juktisi.v4i2.701.
- [18] "Realtime IoT Monitoring for Coworking Spaces Using ESP32 and TFLite-Based Classification," *International Journal of Intelligent Engineering and Systems*, vol. 18, no. 9, pp. 940–959, Oct. 2025, doi: 10.22266/ijies2025.1031.60.
- [19] M. Eichenlaub et al., "Performance of Three Continuous Glucose Monitoring Systems in Adults With Type 1 Diabetes," *J. Diabetes Sci. Technol.*, Feb. 2025, doi: 10.1177/19322968251315459.
- [20] S. S. Prakash, R. Usha, N. Karuppiah, S. Saravanan, M. Kalaiyarasi, and K. Karunanithi, "Smart Home and Security Systems: An IoT-Based Approach Utilizing ESP 32 and Multi-sensor Integration," *E3S Web of Conferences*, vol. 616, p. 02004, Feb. 2025, doi: 10.1051/e3sconf/202561602004.
- [21] H. A. Kusuma, S. B. Wijaya, and D. Nusyirwan, "Sistem Keamanan Rumah Berbasis ESP32-CAM dan Telegram sebagai Notifikasi," *Infotronik: Jurnal Teknologi Informasi dan Elektronika*, vol. 8, no. 1, p. 30, Jun. 2023, doi: 10.32897/infotronik.2023.8.1.2291.
- [22] T. Utsumi and M. Arikawa, "Spatiotemporal Cleaning of PIR Sensor Data for Elderly Movement Monitoring," *Electronics (Basel)*, vol. 13, no. 23, p. 4707, Nov. 2024, doi: 10.3390/electronics13234707.
- [23] I. Santoso and R. M. Sultoni, "Implementasi Algoritma RFID Pada Sistem Portal Parkir Otomatis Berbasis IoT Menggunakan Kartu Tanda Mahasiswa Di

- Universitas Dirgantara Marsekal Suryadarma,” *JURNAL TEKNOLOGI INDUSTRI*, vol. 14, no. 1, pp. 16–25, Dec. 2025, doi: 10.35968/jti.v14i1.1722.
- [24] R. A. Lestari, “Rancang Dan Bangun Bel Rumah Otomatis Berbasis Iot Dan Face Recognition Dengan Aplikasi Android Menggunakan Esp32-Cam,” *Dinamik*, vol. 29, no. 2, pp. 100–108, 2024.
- [25] M. Nida, R. Nurazizah, M. F. Basyari, and V. A. Gunawan, “Sistem Keamanan Rumah Berbasis Internet of Things Menggunakan Metode Rapid Application Development,” *Infotek: Jurnal Informatika dan Teknologi*, vol. 7, no. 2, pp. 390–399, Jul. 2024, doi: 10.29408/jit.v7i2.26013.