Bulletin of Network Engineer and Informatics
E-ISSN 2986-8017 | P-ISSN 2987-4858
https://bufnets.tech

Vol. 4 No 1 – June 2026
DOI: 10.59688/737748

IOT-BASED SMART HELMET PROTOTYPE FOR FIELD SECURITY SAFETY IN OIL PALM PLANTATIONS

Rasyid Abdullah Habib Syaban[1], Andi Prayogi[1], Muhammad Akbar Syahbanapane[2]

[1] Department of Information Systems and Technology, Faculty of Science and Technology, Institut Teknologi Sawit Indonesia, Jl. William Iskandar No. 78, Medan, North Sumatra, Indonesia
[2] Department of Information Technology and Computer Education, Faculty of Engineering, Universitas Negeri Medan, Jl. William Iskandar Pasar V, Medan, North Sumatra, Indonesia

e-mail: rasyidabdullah@itsi.ac.id

Received : 21 May 2026	Accepted : 13 June 2026	Published : 21 June 2026

Abstract

Oil palm plantation security officers face serious risks including occupational accidents, physical assaults during Fresh Fruit Bunch (FFB) theft incidents, and communication limitations in vast and remote field areas. Existing surveillance systems remain manual, lacking real-time position tracking, anomaly detection, and visual verification capabilities. This research develops an IoT-Based Smart Helmet Prototype integrating a GPS NEO-M8N module for real-time location tracking, an IMU MPU6050 sensor for four-level impact classification (NONE/LOW/MEDIUM/HIGH), and dual OV3660 cameras on two ESP32-S3 CAM WROOM N16R8 microcontrollers for simultaneous front and rear RTSP video streaming. All data is transmitted over a 4G modem to a VPS running Mosquitto MQTT broker, MediaMTX media server, Redis cache, Node.js REST API, and a Next.js web dashboard with interactive Leaflet mapping. Functional testing at the Institut Teknologi Sawit Indonesia (ITSI) Experimental Plantation demonstrated a GPS coordinate error of less than 0.0015%, IMU impact classification consistency of 98% across 50 trials, streaming latency of 3–5 seconds via HLS, MQTT average latency of 28 ms, and panic button response under 1 second. The system provides a viable IoT solution for improving field security monitoring in oil palm plantations.

Keywords: *Internet of Things, Smart Helmet, ESP32-S3, GPS Tracking, IMU Impact Detection, Oil Palm Plantation Security*

1. INTRODUCTION

The development of the Internet of Things (IoT) has proven capable of increasing monitoring effectiveness[1]and reducing occupational accident rates through real-time early warning systems in various high-risk sectors [2]. In Indonesia, as the largest palm oil producer with a land area reaching 16.8 million hectares [3], , manual monitoring across vast areas with minimal signal coverage is no longer adequate.This results in high occupational accident rates, where 60.5 percent (224,000 cases) in 2023 occurred in this sector [4],Additionally, security risks stemming from Fresh Fruit Bunch (FFB) theft[5],directly threaten the physical safety of field officers [6].These vulnerabilities are exacerbated by weak security management systems [7], low compliance with the use of Personal Protective Equipment [8], and communication limitations that significantly delay incident response.Collectively, these conditions highlight a critical and unresolved gap in field safety and security management for oil palm plantation workers.

Efforts to address occupational safety challenges in high-risk industries have led to the development of IoT-based smart helmet technologies, particularly in sectors such as coal mining, where helmets are used to detect hazardous gases, temperature, humidity, and worker positioning in underground environments[9],However, these solutions cannot be directly adapted for oil palm plantations due to three fundamental limitations. First, existing smart helmets commonly rely on GPS NEO-6 modules, which support a limited number of satellite connections and struggle to maintain positional accuracy beneath dense agricultural canopies; in contrast, the NEO-M8N module adopted in this study supports multi-constellation satellite reception, enabling reliable location tracking even under the heavy foliage conditions typical of oil palm fields. Second, previous systems relied on local WiFi mesh networks and mobile hotspots for data transmission, which confines their operational range to fixed infrastructure zones and renders them unsuitable for vast, remote plantation areas; this study addresses that limitation by utilizing an independent 4G SIM-based modem, enabling fully mobile and continuous connectivity without reliance on any fixed network. Third, the sensing focus of existing mining helmets centered on gas detection and environmental monitoring does not address the primary threats faced by plantation field officers, namely physical impact from assault during FFB theft incidents and occupational fall accidents, which require dedicated IMU-based impact classification. To date, no IoT-based wearable system has been purpose-built to overcome these combined hardware, connectivity, and sensing limitations in the agricultural plantation context.

Therefore, this study proposes and develops an IoT-based Smart Helmet Prototype specifically engineered for oil palm plantation field officers. The system integrates a NEO-M8N GPS module for real-time location tracking, an MPU6050 IMU sensor for four-level impact classification, and dual OV3660 cameras on ESP32-S3 CAM WROOM N16R8 microcontrollers for simultaneous front and rear visual surveillance. Data from the helmet is transmitted via the MQTT protocol over a 4G modem to a supervisor dashboard supported by VPS infrastructure, MediaMTX, and an SQLite database. The system employs a dual microcontroller Master-Slave architecture via UART, with an adaptive SD Card backup mechanism for offline resilience. This research is expected to contribute practically through a more responsive occupational safety system, and academically by enriching the literature on agricultural wearable devices while opening pathways for future Artificial Intelligence (AI) integration in proactive incident prediction.

2. RESEARCH METHODOLOGY

2.1 Research Type and Approach

This research utilizes an experimental approach with the Prototyping method. This approach focuses on developing an IoT hardware-based system through an iterative cycle of design, assembly, testing, and refinement until it meets all functional requirements. The prototyping cycle in this study comprises eight sequential stages: problem identification, literature review, requirements gathering, prototype design, prototype implementation, functional testing, analysis of test results, and documentation and conclusion.

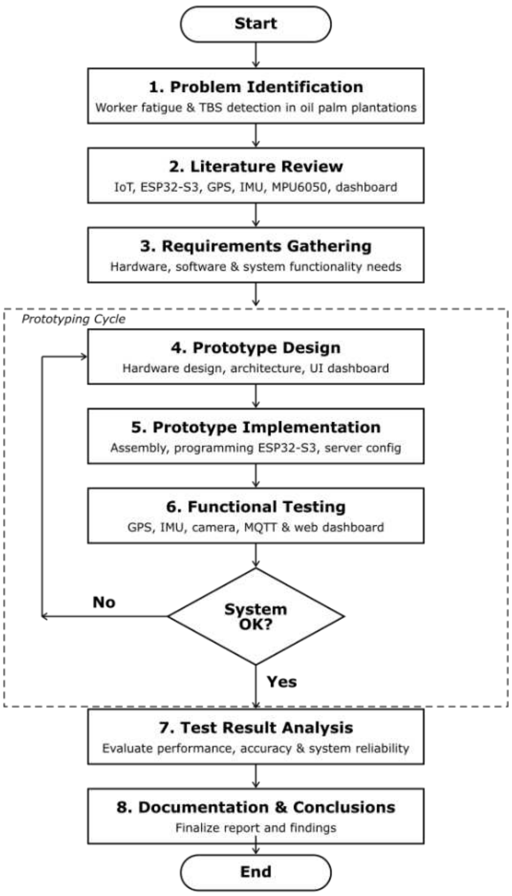

Figure 1. research stages
[Source: Author, 2026]

2.2 Population and Testing Sample

The population in this study is the entire IoT-based safety monitoring system on wearable devices. The testing sample was determined purposively based on functional verification requirements, including:

- Location Testing (GPS): 10 location points with variations in canopy cover conditions (open to semi-closed).
- Impact Testing (IMU): 50 testing scenarios distributed into 5 scenario levels (each repeated 10 times) to obtain sufficient statistical data for calculating the average error and system consistency.

2.3 Instruments and System Architecture

The instrument development refers to the three main layers of IoT architecture—perception, network, and application layers—which enable automatic and real-time data collection and command execution [10].

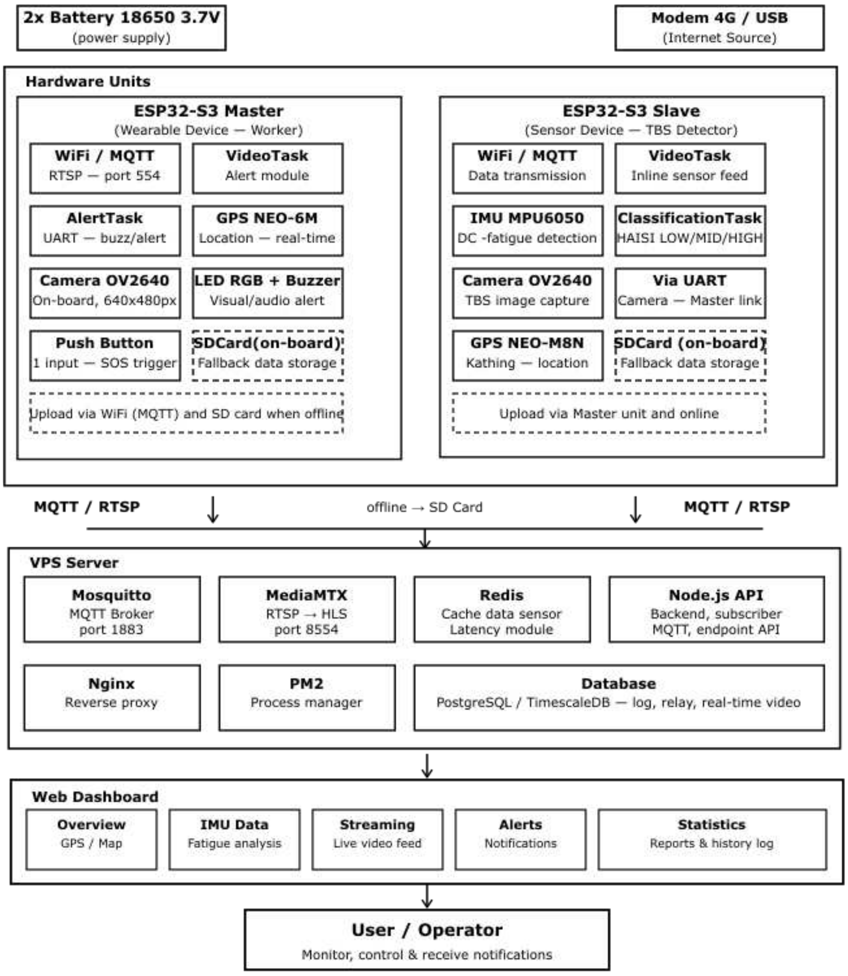

Figure 2. Block Diagram
[Source: Author, 2026]

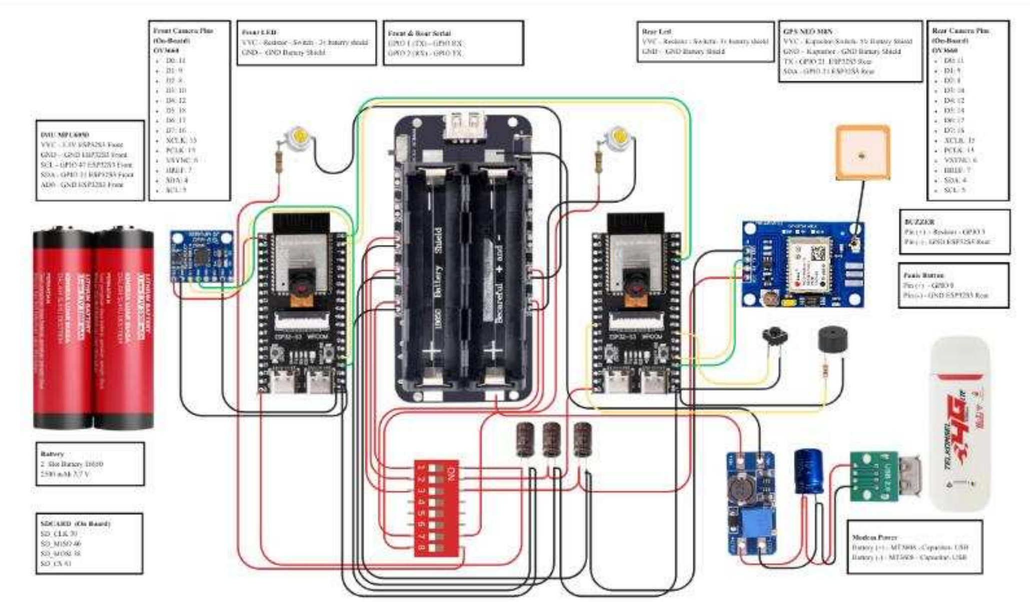

Figure 3. hardware wiring
[Source: Author, 2026]

2.3.1 Hardware (Perception and Network Layers)

- Microcontroller: 2 units of ESP32-S3 CAM WROOM N16R8 configured with a parallel architecture (Master and Slave) communicating via the UART protocol.
- Location Module: NEO-M8N GPS, This selection is based on its capability to provide high-accuracy coordinates with updates up to 10 Hz [11]. which is deemed capable of supporting stable navigation and location tracking processes across various environmental conditions [12]. This module addresses the urgency of tracking officers' routes during incidents.
- Motion Sensor: MPU6050 IMU (6-axis acceleration and orientation). This sensor has proven reliable for classifying emergency conditions such as falls or physical impacts due to resistance with extremely low angle errors [13].
- Visual: 2 units of OV3660 cameras (3 megapixels) for recording field incidents
- Power & Transmission: A 4G USB modem as the data transmission line and 2 lithium-ion 18650 battery cells (2,500 mAh).
- Indicators & Inputs: The on-board LED acts as a connectivity indicator, the buzzer serves as an internal alarm for critical events (falls or severe impacts), and the push button functions as a manual trigger, providing an efficient setup for visual and auditory feedback [14].

2.3.2 Software (Network and Application Layers)

- **Firmware:** Visual Studio Code with the PlatformIO extension (Arduino Framework). VSC offers a cross-platform environment with comprehensive software development features that enhance productivity [15] while PlatformIO supports multi-board programming for devices like the ESP32 by providing automatic compilation, project management, and extensive library integration [16].
- Communication Protocol: Eclipse Mosquitto as the MQTT Broker. MQTT was chosen because its publish-subscribe mechanism has proven superior to HTTP in maintaining persistent connections with minimal network overhead [17].
- Data Processing & Server: Node.js (Express) for the REST API, MediaMTX for RTSP to HLS streaming, SQLite for the log database, and Next.js for the dashboard, RTSP is capable of reducing transmission latency and improving video stability[18]**.**
- Caching: Redis is used as an in-memory cache because it has been proven to drastically increase sensor data access speeds and reduce response latency [19].

2.4 Data Collection Techniques

Data collection was carried out through two complementary approaches:

- Library Research: Reviewing literature to formulate the theoretical foundation and IoT architecture, as well as to justify the selection of hardware and software components based on real security incidents in plantations.
- Field Research: Observing plantation environmental conditions and conducting structured testing on the prototype directly in the field to obtain raw system performance data.

2.5 Data Analysis Techniques

The collected data were analyzed quantitatively using the Black-Box Testing method to evaluate the functionality of five main subsystems. The analysis utilizes descriptive statistics (average, minimum/maximum values, consistency percentage):

- GPS Accuracy: Measured using the error percentage by comparing the coordinates captured by the helmet against the reference coordinates from Google Maps at the same time.
- IMU Impact Detection: Evaluated based on the classification consistency of four impact levels (*NONE, LOW, MEDIUM, HIGH*) against predetermined angle and acceleration thresholds
  The resultant acceleration magnitude is computed from the raw three-axis accelerometer readings of the MPU6050 using the following formula:

$$g_{max} = \frac{\sqrt{a_x^2 + a_y^2 + a_z^2}}{9.81 m/s^2}$$

  Where a_x, a_y, and $a_z$represent the raw acceleration components on the X, Y, and Z axes respectively (in m/s²), and 9.81 m/s² is the gravitational acceleration constant used to normalize the result into units of g. Theclassification logic evaluates the peak g-force value (g_{max}) recorded within a 100 ms sampling window (IMU_SAMPLE_WINDOW = 10 samples at 10 ms intervals) and maps it to one of four impact levels according to the boundary thresholds defined in below.

Table 1. IMU MPU6050 Impact Classification Thresholds
[Source: Author, 2026]

No	Classification	Threshold	Condition
1	NONE	g_{max} <3.0g	No impact detected
2	LOW	3.0g≤g_{max} <5.0g	Minor impact
3	MEDIUM	5.0g≤ g_{max} <8.0g	Moderate impact
4	HIGH	g_{max} ≥8.0g	Severe impact

- MQTT Communication Latency: Calculated based on the timestamp difference between the sending time on the ESP32 Serial Monitor and the receiving time on the Node.js backend log.
- Camera Performance: Evaluated based on HLS streaming latency measurements and the calculation of connection recovery time during signal disruptions.
- Panic Button Function: Tested for the system's consistency and speed in transmitting emergency signals to the centralized web dashboard.

3. RESULT AND DISCUSSION

3.1 System Implementation

The implementation of the IoT smart helmet system encompasses hardware assembly and web dashboard development. The hardware integrates two ESP32-S3 CAM WROOM N16R8 units using a Master-Slave architecture. The Master unit, mounted at the rear of the helmet, houses the NEO-M8N GPS module, a buzzer, and a panic button. The Slave unit, mounted at the front, houses the MPU6050 IMU sensor. The system is powered by two 18650 lithium-ion batteries and utilizes a 4G USB modem for network connectivity.

Figure 4. smart helmet prototype
[Source: Author, 2026]

The monitoring dashboard was successfully developed using Next.js and deployed on a NevaCloud Virtual Private Server (VPS). It features an intuitive user interface for supervisors, providing real-time GPS tracking on an interactive map, dual-camera video streaming, sensor data monitoring, and automated incident logging.

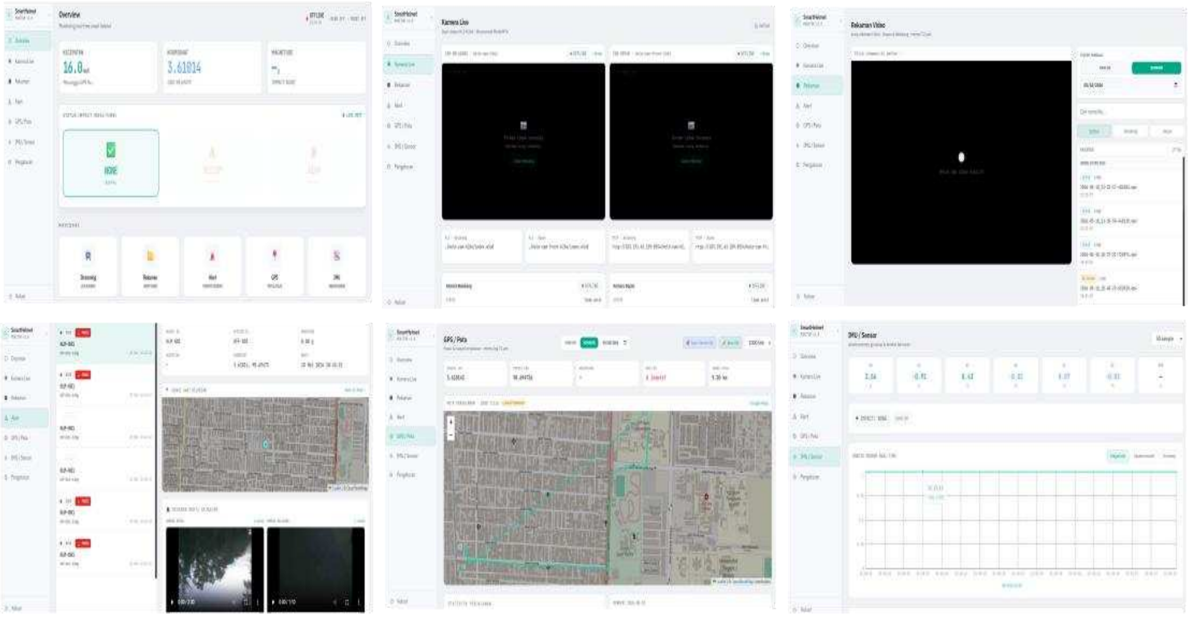

Figure 5. helmet monitoring dashboard
[Source: Author, 2026]

3.2 Testing Analysis Results

The system testing was conducted structurally at the Kebun Praktek Institut Teknologi Sawit Indonesia (ITSI), evaluating five main subsystems. All subsystems successfully passed the functional tests with the following performance recapitulation

Table 2. NEO-M8N GPS accuracy test results
[Source: Author, 2026]

No	Experimental Value (Helmet)		Theoretical Value (Google Maps)		Error (%)		Status
	Latitude	Longitude	Latitude	Longitude	Lat	Lon	
1	3.61776	98.71077	3.61776	98.71077	0.000000	0.000000	PASSED
2	3.61747	98.71078	3.61747	98.71078	0.000000	0.000000	PASSED
3	3.61726	98.71115	3.61728	98.71115	0.000553	0.000000	PASSED
4	3.61756	98.71113	3.61751	98.71123	0.001382	0.000101	PASSED
5	3.61791	98.71126	3.61791	98.71126	0.000000	0.000000	PASSED
6	3.61732	98.71140	3.61729	98.71140	0.000829	0.000000	PASSED
7	3.61762	98.71145	3.61760	98.71147	0.000553	0.000020	PASSED
8	3.61785	98.71153	3.61788	98.71151	0.000829	0.000020	PASSED
9	3.61811	98.71142	3.61810	98.71140	0.000276	0.000020	PASSED
10	3.61709	98.71160	3.61711	98.71159	0.000553	0.000010	PASSED
Average Error					**0.000498**	**0.000017**	**PASSED**

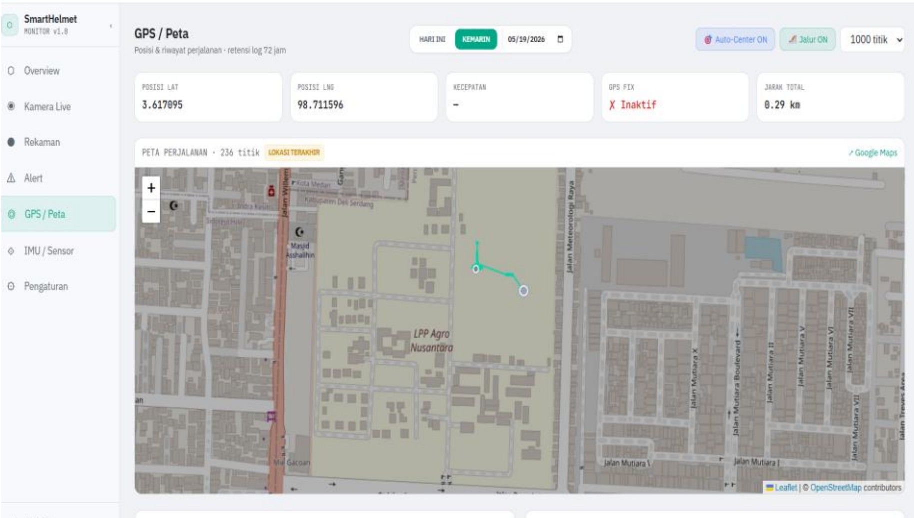

Figure 6. GPS polyline test view
[Source: Author, 2026]

GPS NEO-M8N Accuracy: Tested across 10 locations, the module achieved an average latitude error of 0.000498% and a longitude error of 0.000017%, equating to a positional difference of less than 1 meter. All 10 points passed (10/10 consistency).

Table 3. IMU MPU accuracy test results
[Source: Author, 2026]

No	Scenario	Expected Level	Average Peak (g)	Consistency	Status
1	Walking normally on patrol route	NONE	1.002 g	10/10	PASSED
2	Riding patrol vehicle (dirt road)	NONE / LOW	2.800g	9/10	PASSED
3	Light punch	MEDIUM	6.087 g	10/10	PASSED
4	Hard punch / assault simulation	HIGH	9.465 g	10/10	PASSED
5	Dropping helmet from heights	HIGH	10.965 g	10/10	PASSED

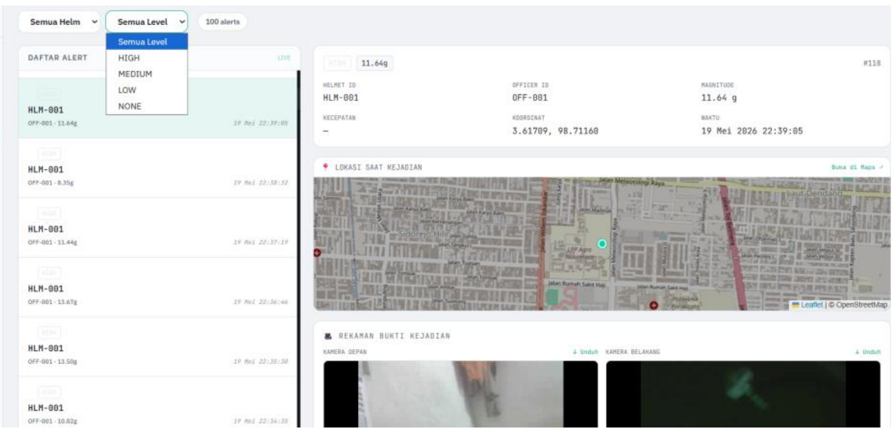

Figure 7. Imu mpu warning display
[Source: Author, 2026]

IMU MPU6050 Impact Detection: Tested through 50 trials across 5 scenarios, the sensor achieved an overall classification consistency of 98%. Walking scenarios averaged 1.002 g (10/10 NONE). Patrol vehicle scenarios averaged 2.800 g (9/10 NONE/LOW). Light punches averaged 6.087 g (10/10 MEDIUM), and hard punches averaged 9.465 g (10/10 HIGH). Drop tests averaged 10.965 g (9 HIGH, 1 MEDIUM).

Table 4. Camera Test Results
[Source: Author, 2026]

No	Function	Scenario	Recorded Result	Status	Description
1	Streaming	Simultaneous dual-camera streaming	Latency: 3–5 seconds	PASSED	HLS via MediaMTX
2	Streaming	Connection disconnection and reconnection	Recovery time: 26 seconds	PASSED	Auto-reconnect
3	Server Recording	Video recorded to server while WiFi active	MP4 file: Available	PASSED	fmp4 recording
4	Server Recording	Video playback from dashboard	Playable: Yes	PASSED	Playback successful
5	SD Card Recording	Video recorded to SD Card while WiFi offline	SD file: Available	PASSED	Offline backup

Camera Streaming and Recording: Simultaneous streaming of two OV3660 cameras operated with a 3–5 second latency using the HLS protocol. The system successfully auto-reconnected within 26 seconds following a deliberate disconnection. Server-side auto-recording (fMP4 format) and offline SD Card backups functioned properly.

Table 5. MQTT Communication Test Results
[Source: Author, 2026]

No	MQTT Topic	Data Sent	Interval	Min (ms)	Max (ms)	Average (ms)	Status
1	helm/gps	GPS Coordinates	1 second	10	50	30	PASSED
2	helm/sensor	IMU Data (peak g)	80 ms	10	50	30	PASSED
3	helm/alert	Impact Notification	Upon detection	10	50	30	PASSED
4	helm/status	Online/Offline Status	Upon change	10	30	20	PASSED
		Overall Average				**28 ms**	**PASSED**

MQTT Communication: The average latency across all topics was 28 ms (minimum 10 ms, maximum 50 ms). The helm/status topic recorded the lowest average latency (20 ms) due to smaller payload sizes

Table 6. Panic Button Test Results
[Source: Author, 2026]

No	Scenario	Expected Result	Response Time	Status	Description
1	Single press — emergency signal (WiFi active)	Panic alert + GPS coordinates appear on dashboard; yellow LED turns on	<1second	PASSED	10/10 consistency
2	Quick double press (< 500 ms) — SD Card upload	Purple LED turns on; CSV files & videos uploaded; SD Card cleared upon completion	~3 minutes	PASSED	Depending on file size in SD Card

Panic Button: The emergency signal and GPS coordinate transmission responded in less than 1 second (10/10 consistency). The double-click SD Card upload mode successfully transferred files and cleared local storage in approximately 3 minutes

Based on the testing conducted at the ITSI Practice Plantation, all IoT smart helmet subsystems were declared PASSED and functioned according to specifications.

Table 7. Recapitulation Of Test Results
[Source: Author, 2026]

No	Subsystem	Primary Metric	Result	Status
1	GPS NEO-M8N	Average coordinate error	< 0,001%	PASSED (10/10)
2	IMU — Normal Walking	NONEdetection consistency	1,002 g	PASSED (10/10)
3	IMU — Patrol Vehicle	NONE/LOW detection consistency	2,800 g	PASSED (9/10)
4	IMU — Light Punch	MEDIUM detection consistency	6,087 g	PASSED (10/10)
5	IMU — Hard Punch	HIGH detection consistency	9,465 g	PASSED (10/10)
6	IMU — Fall from Heights	HIGH detection consistency	10,965 g	PASSED (10/10)
7	Camera (Streaming & Recording)	Latency / Connection recovery time	3–5 sec / 26 sec	PASSED (5/5)
8	MQTT Communication	Average latency	28 ms	PASSED (4/4)
9	Panic Button	Emergency signal response time	< 1 second	PASSED (10/10)

Figure 8. Backend Server Logs
[Source: Author, 2026]

3.3Discussion

The following discussion elaborates on the performance and practical implications of each subsystem evaluated during field testing. The GPS NEO-M8N demonstrated exceptional reliability for tracking plantation security personnel, with an average error translating to less than 1 meter of positional difference. The slight deviation observed at one of the test points indicates a multipath signal phenomenon caused by dense palm oil canopies reflecting the satellite signals before they reached the antenna. This highlights a practical limitation where dense vegetation can cause temporary, localized accuracy degradation, suggesting the future need for IMU-based dead reckoning integration.

The IMU MPU6050 proved highly capable of classifying dynamic impulsive impacts, achieving a 98% overall consistency. The single deviation during the patrol vehicle scenario (registering an unexpected MEDIUM level at 5.1 g, classified as an outlier) does not signify a system failure; rather, it confirms that extreme jolts from severely uneven dirt roads can momentarily produce acceleration profiles that exceed the LOW threshold, mimicking light physical impacts. Furthermore, the drop test scenarios confirmed the sensor's sensitivity to kinetic energy changes, displaying a positive correlation between the drop height and the peak acceleration.

For visual monitoring, the 3–5 second streaming latency via the HLS protocol serves as an acceptable tradeoff for a 4G-based IoT security system. This latency could not be further minimized using ultra-low latency protocols like WebRTC due to the hardware constraints of the microcontroller. The ESP32-CAM natively outputs MJPEG frames, whereas the WebRTC standard strictly requires H.264 or VP8 video encoding for browser compatibility. Consequently, the system relies on a VPS-side transcoding pipeline—utilizing FFmpeg to convert the raw MJPEG stream into an H.264 RTSP format, which is then served as HLS by MediaMTX to ensure playback stability on standard web browsers like Google Chrome. This transcoding overhead is the primary technical factor contributing to the observed latency.

To complement these findings and demonstrate the system's relative efficacy, Table 8 presents a performance comparison matrix evaluating our proposed prototype against prominent wearable tracking baselines found in recent IoT literature, specifically those designed for industrial and mining sectors.

Table 8. Performance Comparison with Recent IoT Wearable Systems.
[Source: Author, 2026]

No	System / Authors	Primary Network Architecture	Primary Sensors	Data Transmission / MQTT Latency	Streaming Capability	Operational Context
1	Smart Helmet for Mining (Sulistiyanto*et al.*, 2025)	Localized WiFi Mesh	Gas (MQ-series), Temperature	Local gateway processing (High variance based on node distance)	None	Confined underground / Fixed infrastructure
2	Industrial Wearables (Naranjo *et al.*, 2025)	Bluetooth / Local WiFi	IMU, Heart Rate, SpO2	~50-100 ms (BLE to localized gateway)	None	Factory floors / Bounded facilities
3	Proposed Smart Helmet (This Study)	Modem SIM 4G	GPS NEO-M8N, IMU MPU6050	28 ms (Average) direct to VPS	Dual RTSP/HLS (3–5s latency)	Vast, remote agricultural plantations

As shown in the comparative matrix, existing smart helmets commonly rely on localized mesh networks or mobile hotspots that restrict their operational range to fixed infrastructural zones. Our proposed system overcomes this by maintaining a highly responsive average MQTT latency of 28 ms over an independent 4G network. This confirms that the Mosquitto Broker efficiently handles real-time sensor data transmission even in remote areas without relying on localized gateways.

Furthermore, the robust 26-second automatic reconnection and offline SD Card backup mechanisms provide a highly relevant architectural contribution for plantation environments prone to cellular signal fluctuations. Coupled with a panic button response time of under 1 second, the system ensures that supervisors can receive and react to emergency notifications instantaneously, proving its practical viability and reliability for real-world deployment in high-risk agricultural sectors.

4. CONCLUSION

This research successfully designed and validated an IoT-based smart helmet prototype for palm oil plantation security personnel. The system, which integrates an ESP32-S3 CAM (Master-Slave), a NEO-M8N GPS, an MPU6050 IMU, and dual cameras, demonstrated high performance during testing at the ITSI Practice Plantation. The results showed that the GPS accuracy has a margin of error of less than one meter, the IMU impact detection consistency reached 98%, MQTT latency was only 28 ms, the panic button response time was under 1 second, and the camera streaming latency was 3–5 seconds with automatic connection recovery.

Theoretically, this study proves the effectiveness of a multi-layered IoT architecture (a combination of MQTT, RTSP/HLS, and Redis) for wearable devices in areas with limited network infrastructure. Practically, this system modernizes security management through centralized monitoring of location, physical condition, and visual situations. Its main scientific contribution is the introduction of the first smart helmet designed specifically for the palm oil sector, successfully integrating three critical functions—GPS tracking, IMU detection, and dual cameras—equipped with an SD Card failsafe in a single device.

For future research, it is recommended to adopt the WebRTC protocol to reduce video latency and implement a temporal filter in the IMU algorithm to suppress false alarms caused by rough patrol roads. Additionally, dead reckoning integration is necessary to address GPS blind spots under dense vegetative canopies. Broader operational scale testing and the integration ofArtificial Intelligence (AI) are also recommended to realize a proactive incident prediction system in the future.

ACKNOWLEDGMENTS

The author expresses the deepest gratitude to the thesis advisors for their patient guidance, and to the examiners for their constructive feedback that significantly elevated the quality of this work. Special thanks are also extended to the author's beloved parents for their endless prayers and unwavering support, and to colleagues and peers whose collaboration and technical support were instrumental during the fieldwork and system testing phases.

REFERENCES

[1] R. A. Sugianto, A. Prayogi, and H. F. S. Simbolon, “Mobile Robot Detects Pests And Diseases In Palm Oil Nurseries,” *journal of technology informatics and engineering*, vol. 3, no. 1, pp. 38–43, Apr. 2024, doi: https://doi.org/10.51903/jtie.v3i1.159.

[2] J. E. Naranjo, C. A. Mora, D. F. Bustamante Villagómez, M. G. ManchenoFalconi, and M. V. Garcia, “Wearable Sensors in Industrial Ergonomics: Enhancing Safety and Productivity in Industry 4.0,” *sensors*, vol. 25, no. 5, pp. 2–43, Feb. 2025, doi: 10.3390/s25051526.

[3] Badan Pusat Statistik, *Statistik Tanaman Perkebunan Tahunan Indonesia (Kelapa Sawit, Kopi, Kakao, Karet, Teh dan Komoditas Perkebunan Unggulan) 2024*, vol. 1. Jakarta: Badan Pusat Statistik, 2025. Accessed: May 21, 2026. [Online]. Available: https://www.bps.go.id

[4] BPJS Ketenagakerjaan, “BPJS Ketenagakerjaan Gelar ToT K3 untuk Tekan Kecelakaan di Perkebunan Sawit.” Accessed: Apr. 28, 2026. [Online]. Available: https://www.bpjsketenagakerjaan.go.id/berita/29100/BPJS-Ketenagakerjaan-Gelar-ToT-K3-untuk-Tekan-Kecelakaan-di-Perkebunan-Sawit

[5] Kasriadi and ANTARA, “PT AGU alami kerugian miliaran rupiah akibat pencurian TBS sawit.” Accessed: Apr. 28, 2026. [Online]. Available: https://kalteng.antaranews.com/berita/425110/pt-agu-alami-kerugian-miliaran-rupiah-akibat-pencurian-tbs-sawit

[6] N. Yenti and A. Dawar, “Penyidikan Tindak Pidana Pencurian Buah Kelapa Sawit Di Wilayah Hukum Polsek Sangir Jujuan,” *Jurnal Normative : Jurnal Ilmiah Hukum*, vol. 11, no. 1, pp. 66–78, 2023, doi: https://ojs.unitas-pdg.ac.id/index.php/normatif/article/view/908.

[7] Rasiani *et al.*, “Analisa Risiko Kecelakaan Kerja pada Karyawan Panen Kelapa Sawit Afdeling Golf PT. Subur Abadi Wana Agung,” *Buletin LOUPE (Laporan Umum Penelitian)*, vol. 20, no. 02, pp. 145–155, Dec. 2024, doi: https://doi.org/10.51967/buletinloupe.v20i02.3177.

[8] J. Nisa, R. Anwar, and Yuanita, “Evaluasi Tingkat Kepatuhan Penerapan Instruksi Penggunaan APD Pada Karyawan Panen Di Perkebunan Kelapa Sawit,” *JURNAL AGRIMENT*, vol. 7, no. 1, pp. 52–59, 2022, doi: 10.51967/jurnalagriment.v7i1.597.

[9] Sulistiyanto, N. Hadits, and S. Widoretno, “Helm Cerdas Pendeteksi Gas Berbasis IoT untuk Keselamatan Pekerja Tambang,” *AKIRATECH : Journal of Computer and Electrical Engineering*, vol. 2, no. 2, pp. 40–47, Jul. 2025, doi: https://doi.org/10.63935/akiratech.v2i2.135.

[10] A. Nugroho and A. Sulistianto, “Perancangan dan Pembuatan Dry Cabinet Menggunakan Konsep Internet of Things (IoT),” *AKIRATECH : Journal of Computer and Electrical Engineering*, vol. 2, no. 1, pp. 33–39, Mar. 2025, doi: https://doi.org/10.63935/akiratech.v2i1.102.

[11] C. A. P. Kirana, W. Djatmiko, and M. Yusro, “Sistem Keamanan Sepeda Motor Menggunakan Aplikasi Smartphone Android Berbasis IoT Menggunakan Modul ESP32 CAM dan Modul ESP32,” *Risenologi: Jurnal Sains, Teknologi, Sosial, Pendidikan, dan Bahasa*, vol. 8, no. 2, pp. 59–67, Dec. 2023, doi: 10.47028/j.risenologi.2023.8i2.514.

[12] L. M. Laurent and R. Pramudita, “Penerapan Internet of Things berbasis GPS untuk Optimalisasi Respons Keamanan Jalan Tol,” *TeknoIS: Jurnal Ilmiah Teknologi Informasi dan Sains*, vol. 15, no. 2, pp. 192–204, Jul. 2025, doi: https://doi.org/10.36350/jbs.v15i2.312.

[13] F. Idris, D. T. P. Yanto, A. Aswardi, and M. Yuhendri, “Rancang Bangun Pemantauan Pergerakan Orang Lanjut Usia Berbasis Mikrokontroller,” *JTEIN: Jurnal Teknik Elektro Indonesia*, vol. 4, no. 2, pp. 601–608, 2023, doi: 10.24036/jtein.v4i2.450.

[14] N. Fauza *et al.*, “Rancang Bangun Prototipe Detektor Hujan Sederhana Berbasis Raindrop Sensor Menggunakan Buzzer Dan Led,” *Jurnal Kumparan Fisika*, vol. 4, no. 3, pp. 163–168, 2021, doi: 10.33369/jkf.4.3.163-168.

[15] G. F. Sunarko, Y. Lawa, and H. Christianto, “Implementation Of Visual Studio Code Integrated PHPMyAdmin And MYSQL Programming In Chemical Education Laboratory Information System Undana,” in *Prosiding Seminar Nasional Pendidikan dan Sains Kimia (SNP-SK) FKIP-Undana*, kupang: Fakultas Keguruan dan Ilmu Pendidikan, Universitas Nusa Cendana, 2023, pp. 63–104. Accessed: May 21, 2026. [Online]. Available: https://conference.undana.ac.id/WNPSK/article/view/535

[16] I. P. O. Wisnawa, I. P. W. Prasetia, and C. A. S. Lahallo, “Arsitektur Internet of Things (IoT) Berskala Industri Dengan Fitur Auto Provisioning,” *TIERS Information Technology Journal*, vol. 2, no. 2, pp. 24–30, Dec. 2021, doi: 10.38043/tiers.v2i2.3312.

[17] A. Dwi Azahra and K. Monika Dian Pertiwi, “Deteksi Serangan DoS pada IoT Berbasis MQTT Menggunakan XGB dan PSO,” *Jurnal Algoritma*, vol. 22, no. 2, Dec. 2025, doi: 10.33364/algoritma/v.22-2.2623.

[18] K. Kariyamin, I. Riadi, and H. Herman, “PERFORMANCE ANALYSIS OF REAL TIME STREAMING PROTOCOL (RTSP) AND REAL TIME TRANSPORT PROTOCOL (RTP) USING VLC APPLICATION ON LIVE VIDEO STREAMING,” *Jurnal Teknik Informatika (Jutif)*, vol. 4, no. 4, pp. 769–778, Aug. 2023, doi: 10.52436/1.jutif.2023.4.4.698.

[19] I. Nur Ramadhan and G. Saraswati, “Penerapan Database Redis Sebagai Optimalisasi Pemrosesan Kueri Data Pengguna Aplikasi SIRESMA Berbasis Laravel,” *Technomedia Journal*, vol. 8, no. 3, pp. 64–77, Dec. 2024, doi: 10.33050/tmj.v8i3.2152.