

# Integrated Emergency Communication System for Disaster Areas Using Long Range

Mahadika Nafiz Luqman<sup>1\*</sup>, Muhammad Fikry<sup>2\*</sup>, Yesy Afrillia<sup>3\*</sup>

\* Teknik Informatika, Universitas Malikussaleh

[mahadika.220170110@mhs.unimal.ac.id](mailto:mahadika.220170110@mhs.unimal.ac.id)<sup>1</sup>, [muh.fikry@unimal.ac.id](mailto:muh.fikry@unimal.ac.id)<sup>2</sup>, [yesy.afrillia@unimal.ac.id](mailto:yesy.afrillia@unimal.ac.id)<sup>3</sup>

## Article Info

### Article history:

Received 2026-04-13

Revised 2026-04-26

Accepted 2026-04-30

### Keywords:

*Bluetooth Low Energy,  
Global Positioning System,  
Long Range,  
Natural Disaster,  
Multi-Hop.*

## ABSTRACT

Natural disasters disrupt communication infrastructure, hindering emergency response coordination. This study designs and evaluates an integrated emergency communication system combining LoRa for transmission, GPS for geolocation, and BLE for alternative interface. The system comprises a Field Device with a 9-state finite-state machine, a Beacon Network forming a linear multi-hop relay chain with heartbeat-based node failure detection, and a Headquarter Device connected to the Blynk platform for monitoring and notifications. A custom binary protocol with 8 message types uses packed structures. All performance was evaluated in urban area, except maximum communication direct range in urban area and rural area. PDR achieves 100% up to 1,000 m Line-of-Sight with an average end-to-end latency of 1.02 s. A single beacon relay extends communication range to 2000 m compared to maximum direct communication range, 1288 m in rural area and 1044 m in urban campus area. Outdoor GPS accuracy measures 0.945 m, while indoor accuracy 28.68 m due to building attenuation. The system successfully detected motion >5 m with 100% sensitivity within 5 s. Usability testing average completion times of 14.84 s via physical interface and 23.85 s via mobile application. BLE range reaches 16 m outdoors and 11 m indoors. Operational durations were 4.32 h for the Field Device, 9.18 h for the Beacon Network, and 8.03 h for the Headquarter Device, falling short of the 12-hour target, necessitating aggressive GPS duty cycling and Wi-Fi sleep modes. This study evaluates three critical factors for emergency response: network determinism, payload efficiency, and power autonomy.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

## I. INTRODUCTION

Indonesia is highly susceptible to various natural disasters, including floods, landslides, and earthquakes. According to the National Disaster Management Agency (BNPB), in 2024 alone, there were 1,420 flood incidents affecting over 6.3 million people and 207 landslide events resulting in 213 fatalities [1]. During the emergency response phase, conventional telecommunication infrastructure such as cellular networks is often damaged or overloaded, severely impeding evacuation and rescue coordination [2]. This critical communication gap underscores the urgent need for a standalone, resilient emergency communication system that

operates independently of vulnerable infrastructure, offers wide coverage, and consumes minimal power.

Long Range (LoRa) technology has emerged as a promising solution for IoT and wireless sensor network applications, primarily due to its exceptional long-range transmission capabilities and remarkably low power consumption characteristics [3]. Operating within the unlicensed Industrial, Scientific, and Medical (ISM) frequency bands, LoRa employs advanced Chirp Spread Spectrum (CSS) modulation techniques that enable reliable and robust communication links even in challenging environments characterized by high noise levels and signal interference.

TABLE I  
LoRa CONFIGURATION PARAM (LLCC68 E220-900T-22D)

Parameter	Value	Description
Frequency	906.125 MHz	Channel 56
Transmit Power	22 dBm	Maximum power for extended range
Air Data Rate	2.4 kbps	Balances range and payload size
Baud Rate	9600 bps	UART communication with ESP32
Parity	8N1	8 data bits, no parity, 1 stop bit
Packet Size	32 Bytes	Maximum payload per LoRa packet
Spreading Factor (SF)	12 (implied)	Highest SF for maximum range
Transmit Mode	Transparent	Direct transmission without addressing
WoR Cycle	500 ms	Wake-on-Radio interval for power saving
Address	0x1234	Network identifier

Several studies have explored LoRa-based systems for disaster management. Syed et al. [4] conducted a feasibility analysis of a LoRa mesh network using Meshtastic firmware, reporting operational durations exceeding 28 h and ranges 2-6 times greater than Wi-Fi. However, their work was limited to theoretical simulation and did not incorporate victim-centric features or a custom protocol stack optimized for emergency messaging. Hong [5] implemented a disaster-resilient mesh network combining LoRa and NerveNet, integrating Wi-Fi and LoRa links. While providing network resilience, the system focused on infrastructure deployment rather than end-user devices and suffered from LoRa's inherent low data rate of approximately 1 kbps, limiting functionality beyond simple text messages. Chen et al. [6]

proposed a UAV-based rescue system combining LoRa and NB-IoT; however, the NB-IoT component introduces a critical dependency on cellular infrastructure that is vulnerable to disaster damage. Matraccia et al. [7] provided a comprehensive survey of post-disaster communication technologies and proposed integrated space-air-ground architectures, but their work remained conceptual without hardware implementation or empirical performance validation. Schmidt et al. [8] developed BPol, a Disruption-Tolerant Networking (DTN) overlay for LoRa, demonstrating improved delivery probability in sparse networks. However, the added complexity of DTN protocols increases computational overhead on resource-constrained devices.

A notable research gap exists in the development of a fully integrated, hardware-implemented system that combines robust long-range communication (LoRa), accurate location awareness GPS with dead reckoning, and a low-power short-range fallback interface (BLE) into a single cohesive device with a well-defined operational state machine and multi-hop relay capability. Furthermore, prior works lack a detailed performance evaluation across a wide range of technical param in distinct environmental conditions.

The integration of Internet of Things (IoT) technologies has demonstrated significant potential in various monitoring applications, including environmental sensing for health and safety [9], gas leak detection using robotic platforms [10], and wearable sensor-based activity recognition for neurological disorders [11]. These studies highlight the effectiveness of IoT architectures in collecting real-time sensor data, applying machine learning for pattern recognition, and enabling remote monitoring through cloud platforms. Drawing inspiration from these approaches, this study adopts a similar IoT framework to build an emergency communication system capable of real-time environmental and situational awareness.

TABLE II  
COMPARISON OF RELATED WORKS AND PROPOSED SYSTEM

Feature	Syed et al. [4]	Hong [5]	Chen et al. [6]	Schmidt et al.	Proposed System
Hardware Implementation	Simulation Only	Partial (Raspberry Pi)	Partial (UAV Node)	Simulation + Partial	Full Prototype, Field Device, Beacon Network, and Headquarter Device
End-User Interaction	None	Infrastructure-Focused	Operator Only	None	9-State FSM + Dual Interface, custom mobile apps and physical
Payload Format	JSON (Verbose)	Standard	Standard	Protocol Buffers	Custom Packed Binary with rang size 3-13 Bytes
Network Topology	Mesh (Meshtastic)	Mesh (NerveNet)	Star (LoRaWAN)	DTN Overlay	Linear Multi-Hop Chain
Failure Detection Mechanism	None	Limited	None	DTN Custody	Heartbeat + Missing Beacon with 35 m timeout
Location Awareness	None	None	UAV GPS Only	None	GPS with Dead Reckoning + Smartphone Fallback
Automatic Motion Detection	No	No	No	No	Yes, with Haversine detect >5m within 5 s threshold
Comprehensive Parametric Evaluation	Partial (Simulation)	Partial	Partial	Simulation	16 Params; Range tested in urban & rural area, others at urban area
Power Profiling & Optimization	Estimated	Not Reported	Not Reported	Not Reported	Detailed Component-Level Analysis + Duty Cycle Proposals

This study aims to design, implement, and evaluate a prototype of an integrated emergency communication system leveraging LoRa, GPS, and BLE technologies. The main contributions of this research are:

- A fully implemented three-tier hardware architecture with custom finite-state machine logic, with detailed wiring diagrams and component specifications.
- An efficient binary communication protocol with eight messages types optimized for LoRa, including a comparative baseline analysis against standard JSON formatting.
- Performance was assessed across 16 param, testing communication range across urban and rural environments to quantify site-specific effects while conducting all other replicable tests on-campus.
- A comparative analysis against existing LoRa-based emergency communication systems, highlighting specific improvements in payload efficiency, user interaction, and failure detection.
- Detailed power consumption profiling with actionable optimization recommendations including GPS duty cycling and Wi-Fi modem sleep strategies.

## II. METHOD

This research employed an experimental methodology encompassing requirements analysis, system design, prototype implementation, and performance testing.

### A. System Architecture Overview

The proposed system comprises three distinct device types forming a linear multi-hop communication chain. The architecture is intentionally designed as a strict linear chain rather than a dynamic mesh network. This design choice prioritizes deterministic latency and predictable routing over adaptive topology, which is advantageous in emergency scenarios where reliable, low-latency communication along a known evacuation path is more critical than arbitrary peer-to-peer connectivity.

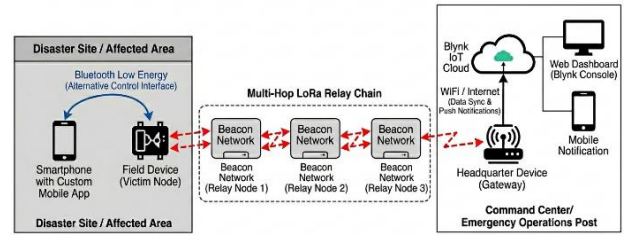


Figure 1. Overall system architecture diagram

The three device types and their primary functions are:

- **Field Device:** The end-user node deployed in disaster-prone areas. It collects situational data, e.g., disaster type and severity level, acquires GPS coordinates, and provides dual input interfaces, i.e. physical buttons and BLE mobile application. It is assigned a unique Area ID (e.g., 591).
- **Beacon Network:** Intermediate relay nodes that extend communication range and monitor network integrity. Each beacon is assigned a sequence number (e.g., Beacon 1, Beacon 2) along a specific Area ID's communication path.
- **Headquarter Device:** The command center node positioned at an emergency operations center. It aggregates incoming messages, sends acknowledgments (ACK), interfaces with the Blynk cloud platform for remote monitoring, and provides local audio alerts via a piezo buzzer.

### B. Field Device Design and Implementation

The Field Device is the primary user-facing component deployed in disaster-prone areas. It is designed to be portable, battery-powered, and operable by non-technical users under stressful conditions.

- 1) **Hardware Configuration:** The device is built around an ESP32-WROOM-32 microcontroller [12], selected for its dual-core processing capability, integrated WiFi and Bluetooth radios, and multiple hardware serial interfaces.

TABLE III  
FIELD DEVICE HARDWARE SPECIFICATIONS

Component	Model / Part Number	Key Specifications	Configuration / Notes
Microcontroller	ESP32-WROOM-32	240 MHz Dual-Core Xtensa LX6, 520 KB SRAM, 4 MB Flash	Operates at 3.3V logic level
LoRa Module	LLCC68 E220-900T-22D	906.125 MHz, 22 dBm Tx Power, -146 dBm Sensitivity	UART2 (GPIO16 RX, GPIO17 TX), M0=GPIO19, M1=GPIO23, AUX=GPIO18
GPS Module	u-blox NEO-M8L	72-channel GNSS, 3D Accelerometer & Gyroscope (Dead Reckoning)	UART1 (GPIO32 TX, GPIO33 RX), 1 Hz NMEA output
Display	OLED 0.91" SSD1306	128x32 pixels, I2C Interface	SDA=GPIO21, SCL=GPIO22, 3.3V
Input Sensors	TTP223 Capacitive Touch (x2)	2.0-5.5V Operating Voltage, 60ms Response Time	Button 1=GPIO25, Button 2=GPIO26 (Active LOW with pull-down)
Power Source	Lithium-Polymer Battery	3.7 V Nominal, 2500 mAh Capacity	Dimensions: 25 mm x 30 mm x 10 mm
Power Regulation	DC-DC Step-up Converter	Input: 0.9-5V → Output: 5V, 150 kHz Switching	Efficiency up to 85%
Pull-up Resistors	4.7 kΩ	±5% Tolerance	Connected to LoRa RXD, TXD, AUX pins for signal stability

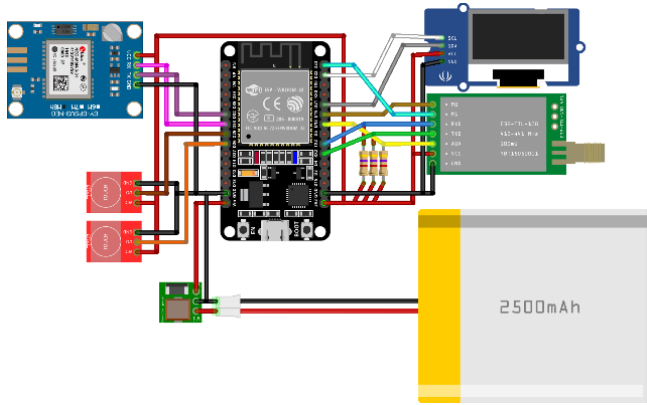


Figure 2. Wiring diagram of the Field

2) *Finite-State Machine Logic:* The firmware implements a deterministic finite-state machine (FSM) to manage user interaction and system status. This approach prevents invalid operations and provides clear visual feedback on the OLED display. The FSM consists of nine distinct states enumerated as follows:

- STATE\_IDLE: Device powered on, waiting user input.
- STATE\_SELECTING\_DISASTER: User cycles through four disaster types using Button 1.
- STATE\_SELECTING\_DESTRUCTION: User cycles through four severity levels using Button 1.
- STATE\_CONFIRM: Summary displayed; Button 2 confirms and sends, Button 1 cancels back to disaster selection.
- STATE\_WAITING\_ACK: Help message sent; listening for ACK with 60-second timeout.
- STATE\_SUCCESS: ACK received; 5-minute cancellation window active.
- STATE\_FAILED: ACK timeout / transmission failure.

- STATE\_CANCEL\_AVAILABLE: Substate within SUCCESS indicating cancellation is possible.
- STATE\_WAITING\_CANCEL\_ACK: Cancel request sent; waiting for ACK.

3) *Automatic Motion Detection:* An independent monitoring routine continuously reads GPS coordinates. Using the Haversine formula to calculate the great-circle distance between the current position and a stored reference position, the system evaluates whether displacement exceeds 5 m within any 5-second window.

$$d = 2r \cdot \arcsin \left( \sqrt{\sin^2 \left( \frac{\Delta\phi}{2} \right) + \cos(\phi_1)\cos(\phi_2)\sin^2 \left( \frac{\Delta\lambda}{2} \right)} \right)$$

Where  $d$  is distance in m,  $r$  is Earth's radius (6,371,000 m),  $\phi$  is latitude in radians, and  $\lambda$  is longitude in radians. If the displacement condition is met and the device is not already in an active emergency state, an Alarm message is autonomously generated and transmitted without user intervention. This feature addresses scenarios where the device and its user are displaced by floodwaters or landslide debris, rendering manual operation impossible.

4) *BLE Server Implementation:* The ESP32's built-in BLE radio is configured as a server with the following custom UUIDs:

- Service UUID: 4fafc201-1fb5-459e-8fcc-c5c9c331914b
- Write Characteristic UUID: beb5483e-36e1-4688-b7f5-ea07361b26a8
- Notify Characteristic UUID: 1c95d5e3-d8f7-413a-bf3d-7a2e5d7be87e

The BLE server advertises with a device name formatted as FieldDevice\_XXXX, where XXXX is the configured Area ID. A locking mechanism ensures that when the mobile application has sent a command and is awaiting acknowledgment, the physical touch sensors are temporarily disabled to prevent conflicting inputs.

TABLE IV  
FINITE-STATE MACHINE (FSM) STATE TRANSITION TABLE

Current State	Trigger / Input	Action	Next State
IDLE	Button 1 Press	Initialize disaster index = 1	SELECTING_DISASTER
SELECTING_DISASTER	Button 1 Press	Increment disaster index (wrap 1→4)	SELECTING_DISASTER
SELECTING_DISASTER	Button 2 Press	Store selected disaster code	SELECTING_DESTRUCTION
SELECTING_DESTRUCTION	Button 1 Press	Increment destruction index (wrap 1→4)	SELECTING_DESTRUCTION
SELECTING_DESTRUCTION	Button 2 Press	Store selected destruction code	CONFIRM
CONFIRM	Button 1 Press	Cancel operation	SELECTING_DISASTER
CONFIRM	Button 2 Press	Construct and send Help message; start 60s timer	WAITING_ACK
WAITING_ACK	ACK Received	Display "Send Success"; start 5-min cancel timer	SUCCESS
WAITING_ACK	Timeout (60s)	Display "Send Failed"	FAILED
SUCCESS	Timeout (5 min)	Cancel window expired; reset to idle	IDLE
SUCCESS	Buttons 1+2 (hold 6s)	Send Cancel Help message	WAITING_CANCEL_ACK
WAITING_CANCEL_ACK	ACK Received	Display "Cancelled"; reset to idle	IDLE
FAILED	Auto	Reset system	IDLE

### C. Beacon Network Design and Implementation

The Beacon Network serves as an intermediate relay node to extend communication range beyond direct line-of-sight limitations and to monitor network integrity along the communication chain.

1) *Hardware Configuration*: The beacon device uses the same core components as the Field Device but omits the GPS module, OLED display, and touch sensors to reduce cost and power consumption.

TABLE V  
BEACON NETWORK HARDWARE SPECIFICATIONS

Components	Model	Notes
Microcontroller	ESP32-WROOM-32	Same as Field Device
LoRa Module	LLCC68 E220-900T-22D	Same configuration as Field Device
Power Source	Li-Po 3.7V 2500 mAh	Same as Field Device
Power Regulation	DC-DC Step-up (5V)	Same as Field Device
Pull-up Resistors	4.7 kΩ	Same as Field Device

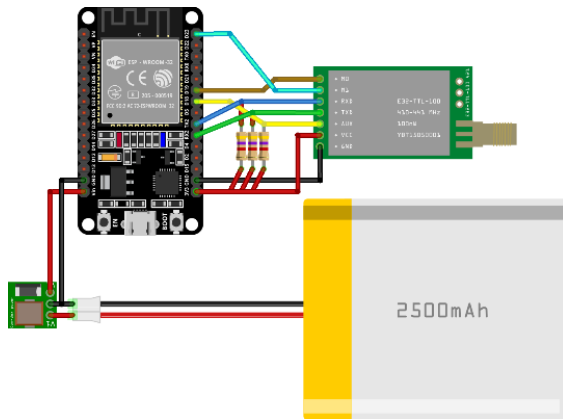


Figure 3. Wiring diagram of the Beacon Network

2) *Message Relay and Deduplication Logic*: The beacon device operates exclusively in a listening and forwarding mode. Upon receiving any LoRa packet, it inspects the type field, i.e., first byte:

- Messages types destined for the Headquarter Device, i.e., Help, Alarm, Cancel Help, and Location Response are forwarded to the next beacon in the chain, i.e., closer to HQ using `relayToNext()`.
- Messages types originating from the Headquarter Device, i.e., Acknowledgment and Location Request are forwarded toward the Field Device using `relayToPrev()`.

A critical feature is the deduplication algorithm. Each received message is hashed using the DJB2 algorithm, and the hash is stored in a history buffer with a 60-second Time-to-Live (TTL).

$$hash = 5381 + \sum_{i=0}^{n-1} (hash \times 33 + data[i])$$

If an identical hash is received within the 60-second window, the duplicate message is discarded without forwarding. This prevents message loops and redundant processing in the multi-hop network. The history buffer has a capacity of 50 entries, and a cleanup routine runs every 5 m to remove expired entries.

3) *Heartbeat and Missing Node Detection*: Each beacon periodically transmits a Heartbeat message every 30 m. The message contains the sender's position in the chain, i.e., `senderBeacon` and the intended receiver, i.e., `receiverBeacon`. Simultaneously, each beacon monitors the expected arrival of Heartbeat messages from the preceding node, i.e., closer to the Field Device. If no Heartbeat is received within 35 m (i.e., 30-minute interval plus 5-minute tolerance), the beacon assumes the preceding node has failed and generates a Missing Beacon message to the Headquarter Device, alerting command staff to the communication break in that specific segment.

4) *Power Management*: To extend battery life, the beacon implements dynamic frequency scaling (DFS) on the ESP32. When in idle listening mode, the CPU frequency is reduced from 240 MHz to 80 MHz. The main loop employs adaptive delays with 100 millis in power-save mode and 10 millis in normal operation. While the LoRa module remains in continuous receive mode as it must be ready to relay messages at any time, these CPU optimizations provide a measurable reduction in overall power consumption.

### D. Headquarter Device Design and Implementation

The Headquarter Device is positioned at the emergency operations center and serves as the data aggregation and notification hub.

1) *Hardware Configuration*: The Headquarter Device uses the same core components as the Field Device with addition piezo buzzer but omits the GPS module, OLED display, and touch sensors to reduce cost and power consumption.

TABLE VI  
HEADQUARTER DEVICE HARDWARE SPECIFICATIONS

Components	Model	Notes
Microcontroller	ESP32-WROOM-32	Same as Field Device
LoRa Module	LLCC68 E220-900T-22D	Same configuration as Field Device
Audio Alert	TMB12A03 Piezo Buzzer	Connected to GPIO27
Power Source	Li-Po 3.7V 2500 mAh	Same as Field Device
Pull-up Resistors	4.7 kΩ	Same as Field Device
Power Regulation	DC-DC Step-up (5V)	Same as Field Device

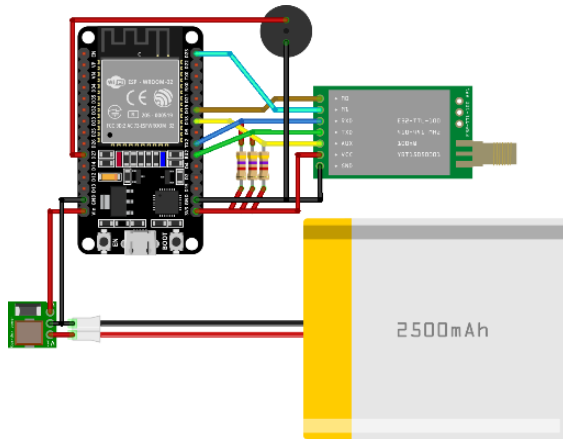


Figure 4. Wiring diagram of the Headquarter Device.

2) *Message Processing and Notification:* Upon receiving a message via LoRa, the device parses the packed structure according to the message type identifier. The payload is decoded into human-readable strings.

TABLE VII  
DISASTER AND DESTRUCTION CODE MAPPING

Code	Disaster Type	Code	Destruction Level
1	Earthquake	1	Minor (no casualties, infrastructure damage)
2	Flood	2	Moderate (<10 missing or <20 injured)
3	Landslide	3	Severe (<5 fatalities or >10 missing or >20 injured)
4	Fire	4	Massive (>5 fatalities)

The following actions are taken based on message type:

- Help or Alarm: Piezo buzzer activated for 2-second duration. Message details transmitted to Blynk platform via Wi-Fi, updating dashboard widgets. Push notification triggered as “Critical Alert” to bypass silent/DND modes.
- Cancel Help: Buzzer not activated. Blynk dashboard updated to reflect cancellation.
- Location Response: Coordinates displayed on Blynk dashboard map widget.
- Missing Beacon: Buzzer activated; alert sent to Blynk indicating specific beacon number.

3) *Time Synchronization:* The Headquarter Device synchronizes its internal clock with an NTP server with `asia.pool.ntp.org` and GMT+7 upon startup and hourly thereafter using `getLocalTime()` with a 10-second timeout. Accurate timestamps are appended to all logged events in the format DD/MM/YYYY-HH:MM: SS.

#### E. Custom Mobile Application

A cross-platform mobile application was developed using the Flutter framework [13] to serve as an alternative user interface for the Field Device. The application is designed to

operate on Android devices running version 12 or higher and provides a complete fallback control channel when the physical buttons on the Field Device are inaccessible, damaged, or when the device is mounted in a location that makes manual interaction difficult.

1) *Device Discovery and Connection Management:* Upon launch, the application requests the necessary Bluetooth and location permissions from the user. It then initiates a BLE scan, filtering for devices whose advertised name begins with the prefix “FieldDevice\_”. Each discovered device is displayed in a scrollable list showing the device name including its Area ID and the Received Signal Strength Indicator (RSSI) value in dBm. The RSSI provides a visual cue to the user regarding proximity and expected connection quality. Tapping on a device entry initiates a connection request with a 15-second timeout. Upon successful connection, the application performs service discovery to locate the custom Service UUID and its associated Write and Notify characteristics. The connection state is displayed prominently, and the application subscribes to notifications on the Notify characteristic to receive real-time status updates from the Field Device.

2) *Emergency Message Composition:* Once connected, the main interface presents the user with two dropdown selection menus. The first menu allows selection of the disaster type and the second menu allows selection of the destruction severity level. Below the selection menus, the application displays the current status of the Field Device, which is continuously updated via BLE notifications. The status display uses color-coded backgrounds: green for success states, blue for waiting or idle, and red for failure states. A prominent “Kirim Permintaan Bantuan” button initiates the emergency transmission.

3) *Integrated GPS Fallback:* The application incorporates the geolocator plugin to access the smartphone's internal GPS receiver. When the user presses the SEND HELP button, the application requests the current device location with high accuracy settings. The obtained latitude and longitude are embedded into the command payload sent to the Field Device. This mechanism provides a critical redundancy layer: if the Field Device's own GPS module is experiencing signal degradation, e.g., inside a damaged building or has failed entirely, the system can still transmit accurate location coordinates derived from the user's smartphone. If the smartphone GPS fails to acquire a fix within a 30-second timeout, the application displays a warning but still allows the message to be sent with placeholder coordinates, which the user can manually confirm.

4) *State Synchronization and Locking:* The application maintains an internal finite-state machine that mirrors the nine states of the Field Device firmware. When a state change notification is received via BLE, the application updates its user interface to disable the SEND HELP button and display

the appropriate waiting message. This prevents the user from inadvertently sending duplicate requests. The application also implements the 5-minute cancellation window. After a SUCCESS state is received, a “Batalkan Bantuan” button becomes visible along with a countdown timer. Pressing this button sends a Cancel Help command to the Field Device. A locking mechanism ensures that when the application is actively controlling the device, i.e., the application has sent a command and is awaiting acknowledgment, the physical touch sensors on the Field Device are temporarily disabled to prevent conflicting inputs.

F. Blynk Dashboard Configuration

The Blynk IoT platform serves as the cloud-based monitoring and control interface for the Headquarter Device, enabling remote situational awareness for emergency response coordinators. A dedicated template named “Disaster Alert System” was created within the Blynk Console. Communication between the ESP32 Headquarter Device and the Blynk cloud occurs via a set of ten virtual pins start from V0 through V9. Each pin is configured with a specific data type and serves a distinct function within the dashboard. The mapping is defined as follows:

TABLE VIII  
BLYNK VIRTUAL PIN MAPPING

Virtual Pin	Widget Type	Data Type
V0	Label	String
V1	Label	String
V2	Label	String
V3	Label	Double
V4	Label	Double
V5	Label / Chart	Integer
V6	Label	String
V7	Label	String
V8	Terminal	String
V9	Number Input	Integer

1) *Dashboard Layout and Widgets:* The “Latest Message” widget (V0) prominently shows the most recent event. A “Status” widget (V6) displays real-time counters for Help, Alarm, Cancel, and Missing Beacon events formatted as “H:XX, A:XX, C:XX, M:XX”. A grid section presents geospatial data including Latitude (V3), Longitude (V4), and an Area ID display with a historical chart (V5). The Disaster Type (V1), Destruction Level (V2), and Missing Beacon (V7) indicators are arranged. The “Location Request” widget (V9), entering an Area ID sends a Location Request command from the Headquarter Device to the specified Field Device. Finally, an “Alarm Log” terminal widget (V8) for time-stamped history of all received messages and system events.

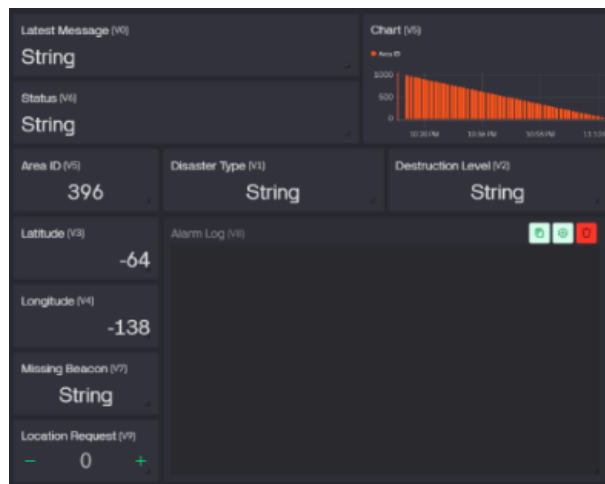


Figure 5. Blynk Dashboard Layout for Desktop

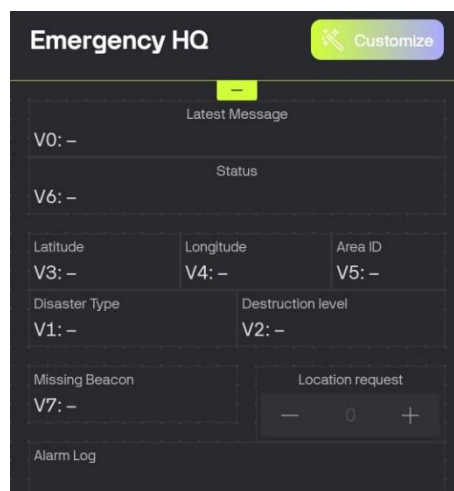


Figure 6. Blynk Dashboard Layout for Smartphone

2) *Event and Notification Configuration:* A custom event named “Disaster” was defined within the Blynk template. The event is configured with type “Critical” and a red color code for visual prominence. The notification settings are configured such that any message triggering this event, e.g., reception of a Help or Alarm message will:

- Send an email notification to the device owner's registered email address.
- Send a push notification to the Blynk IoT application on the owner's smartphone.
- Deliver the push notification as a “Critical Alert,” which bypasses the device's silent and Do Not Disturb modes, ensuring that emergency alerts are always audible and visible.

The frequency of notifications is rate-limited to one per second to prevent alert fatigue during multiple rapid transmissions. The event is displayed on the device timeline within both the Blynk Console and Blynk IoT application.

### G. Custom Communication Protocol Detailed Specification

Given the severe bandwidth constraints of LoRa technology and the need for low-latency emergency messaging, a custom binary protocol was designed from the ground up. The protocol eschews verbose human-readable formats like JSON or plain string in favor of tightly packed C structures with fixed field sizes and deterministic layout. All multi-byte numeric fields are encoded in little-endian byte order, which is the native format of the ESP32 microcontroller, eliminating the need for byte-swapping operations and reducing computational overhead.

The protocol defines eight distinct message types, each identified by a single-byte type field, i.e., first byte of every packet. To eliminate compiler-added padding bytes, all structures are declared with the `__attribute__((packed))` directive.

1) *Help Message*: This message is transmitted by the Field Device when a user manually initiates an emergency request. It contains the full situational report required by first responders.

TABLE IX  
HELP MESSAGE CONTENT

Field Name	C Data Type	Bytes	Valid Range
Type	uint8_t	1	1
areaId	UInt16_t	2	0-65535
Disaster	uint8_t	1	1-4
Destruction	uint8_t	1	1-4
Latitude	Float	4	-90.0 to +90.0
Longitude	Float	4	-180.0 to +180.0

2) *Acknowledgment Message*: This message is sent by the Headquarter Device in response to a successfully received Help, Alarm, or Cancel Help message. It serves as a positive confirmation to the originating Field Device that its transmission was received and is being processed.

TABLE X  
ACKNOWLEDGMENT MESSAGE CONTENT

Field Name	C Data Type	Bytes	Valid Range
Type	uint8_t	1	2
areaId	UInt16_t	2	0-65535

3) *Alarm Message*: This message is autonomously generated by the Field Device when its motion detection algorithm detects displacement exceeding 5 m within a 5-second window. It alerts the command center that the Field Device has been moved, possibly due to floodwaters or landslide debris.

TABLE XI  
ALARM MESSAGE CONTENT

Field Name	C Data Type	Bytes	Valid Range
Type	uint8_t	1	3
areaId	UInt16_t	2	0-65535
Latitude	Float	4	-90.0 to +90.0
Longitude	Float	4	-180.0 to +180.0

4) *Cancel Help Message*: This message is sent by the Field Device when a user manually cancels a previously sent Help request within the 5-minute cancellation window. It prevents unnecessary dispatch of emergency resources.

TABLE XII  
CANCEL HELP MESSAGE CONTENT

Field Name	C Data Type	Bytes	Valid Range
Type	uint8_t	1	4
areaId	UInt16_t	2	0-65535

5) *Location Request Message*: This message is sent by the Headquarter Device to a specific Field Device to request its current geographic coordinates. This is useful for tracking a device that may have been displaced or for routine position verification.

TABLE XIII  
LOCATION REQUEST MESSAGE CONTENT

Field Name	C Data Type	Bytes	Valid Range
Type	uint8_t	1	5
areaId	UInt16_t	2	0-65535

6) *Location Response Message*: This message is sent by the Field Device in response to a Location Request. It provides the device's current GPS coordinates.

TABLE XIV  
LOCATION RESPONSE MESSAGE CONTENT

Field Name	C Data Type	Bytes	Valid Range
Type	uint8_t	1	6
areaId	UInt16_t	2	0-65535
Latitude	Float	4	-90.0 to +90.0
Longitude	Float	4	-180.0 to +180.0

7) *Heartbeat Message*: This message is transmitted periodically by Field Device and Beacon Network nodes to monitor the health and connectivity of the multi-hop communication chain. Each node sends a Heartbeat to its immediate successor toward the Headquarter Device.

TABLE XV  
HEARTBEAT MESSAGE CONTENT

Field Name	C Data Type	Bytes	Valid Range
Type	uint8_t	1	7
areaId	UInt16_t	2	0-65535
senderBeacon	uint8_t	1	0-255
receiverBeacon	uint8_t	1	0-255

8) *Missing Beacon Message*: This message is generated by a Beacon Network node when it fails to receive the expected Heartbeat from its predecessor within the 35-minute timeout period. It alerts the Headquarter Device to a potential node failure or communication disruption at a specific point in the relay chain.

TABLE XVI  
MISSING BEACON MESSAGE CONTENT

Field Name	C Data Type	Bytes	Valid Range
Type	uint8_t	1	1
areaId	UInt16_t	2	0-65535
missingBeacon	uint8_t	1	0-255

#### H. Testing Scenarios

Performance evaluations were conducted in two distinct environments to assess the system's robustness under varying conditions.

1) *Test Environments*: The first environment was the urban campus area, Universitas Malikussaleh in Lhokseumawe characterized by multi-story buildings, moderate Wi-Fi interference, and mixed open spaces and obstructions. The second environment was the rural area, Bener Meriah Regency characterized by undulating terrain, dense vegetation, and minimal radio frequency interference.

2) *Performance Param, Targets, and Measurement Methods*: The 16 performance param evaluated in this study, their targets, and the detailed measurement methods are presented below. Unless explicitly noted otherwise, all tests were conducted at the Universitas Malikussaleh campus. Only the maximum direct communication range was measured in two environments to compare urban and rural propagation.

TABLE XVII  
COMPREHENSIVE PERFORMANCE PARAM AND MEASUREMENT METHODS

Parameter	Target	Data Measurement Method
Packet Delivery Ratio (PDR)	> 90%	Send 20 Help messages from Field Device to Headquarter Device. $PDR = (\text{Number of ACKs received by Field Device} / 20) \times 100\%$ . Tested at incremental distances 1000 m in urban campus area.
End-to-End Latency	< 60 s	Measured using stopwatch, start when message send by Field Device and end when Field Device receive ACK. Average of 10 successful trials at 1000 m distance.
BLE Range (Outdoor)	> 10 m	Smartphone moves away from Field Device until BLE disconnection. Distance measured using Google Earth measurement tool. Test repeated 3 times; maximum stable distance recorded.
BLE Range (Indoor)	> 10 m	Smartphone outdoors, Field Device indoors. Same measurement method as above.
GPS Accuracy (Outdoor)	< 10 m RMSE	Field Device and calibrated smartphone placed at same open-sky location. 60

		coordinate samples recorded at 1-second intervals. RMSE calculated using Haversine distance between corresponding samples.
GPS Accuracy (Indoor)	< 10 m RMSE	Same method as above, but Field Device placed inside concrete building.
Motion Detection Sensitivity (>5m)	100% (5/5)	Field Device physically displaced >5 m within 5 s. Test repeated 5 times. Success = Alarm message triggered and received by Headquarter Device.
Motion Detection Specificity (<5m)	0% False Alarms (0/5)	Field Device displaced <5 m within 5 s. Test repeated 5 times. Success = No Alarm triggered.
Usability (Manual Buttons)	< 120 s	5 participants unfamiliar with system. Time measured from IDLE state to WAITING_ACK state using stopwatch. Average of 5 trials.
Usability (Mobile App)	< 120 s	Same 5 participants. Time measured from app launch to message sent. Average of 5 trials.
Power Consumption (Field Device)	> 12 h	Operate Field Device with fully charged 2500 mAh battery in IDLE state with GPS active, OLED on, and LoRa listening. Record time until shutdown.
Power Consumption (Beacon Network)	> 12 h	Operate Beacon with fully charged battery in relay mode. Record time until shutdown.
Power Consumption (Headquarter Device)	> 12 h	Operate Headquarter Device with fully charged battery, Wi-Fi connected to Blynk. Record time until shutdown.
Maximum Direct Communication Range (urban campus and rural area)	> 1000 m	Incremental distance testing in urban and rural area: start at 500 m, increase by 500 m until PDR < 90%. Then refine with 100, 50, 10, 5, and 1 m. Record maximum distance with PDR $\geq$ 90% from 4 messages.
Extended Range with Beacon Network	> 1500 m	Place one Beacon at 1000 m. Test communication between Field Device (0 m) and HQ (2000 m). Measure PDR from 4 messages.

### III. RESULTS AND DISCUSSION

This section presents the evaluation results in a structured manner. It begins with the documentation of the physical prototype implementation, followed by the performance evaluation of each parameter defined in the Methodology. All tests, except for the maximum direct communication range

test, were conducted in the vicinity of the Universitas Malikussaleh campus. The maximum range test was performed in two distinct environments, urban campus and rural area to assess environmental impact on signal propagation.

#### A. Prototype Implementation

Functional prototypes of all three device types were successfully assembled in compact enclosures.

1) *Field Device Prototype*: The Field Device was constructed in a compact portable enclosure. The front panel provides access to the OLED display and two capacitive touch sensors, while the internal compartment houses the microcontroller, communication modules, and power supply.

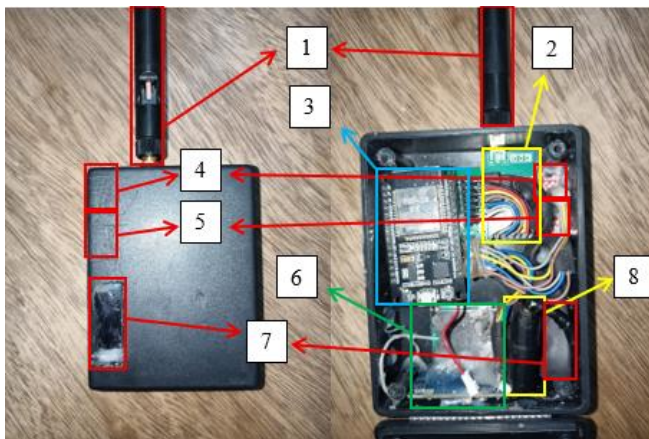


Figure 7. Field Device prototype, front view (left) and internal component layout (right)

Referring to the labelled components in the Figure 7: the 5 dB antenna (1) connects to the LoRa module LLCC68 E220-900T-22D (2) for long-range transmission. The ESP32-WROOM-32 (3) is mounted on female headers with a DC-DC step-up converter beneath it. Two TTP223 touch sensors (4, 5) serve as Button 2 and Button 1 respectively. The u-blox NEO-M8L GPS module (6) provides positioning, while the 0.91-inch OLED (7) displays the FSM state. Power is supplied by a 3.7 V 2500 mAh Li-Po battery (8).

The finite-state machine operated exactly as designed: during testing, the device correctly sequenced through the STATE\_IDLE, STATE\_SELECTING\_DISASTER, STATE\_SELECTING\_DESTRUCTION, and STATE\_CONFIRM states. Upon confirmation, it transitioned to STATE\_WAITING\_ACK and displayed the waiting message. The 60-second timeout was consistently enforced, and the device correctly transitioned to STATE\_SUCCESS upon receiving a valid ACK or to STATE\_FAILED upon timeout. The 5-minute cancellation window in STATE\_SUCCESS functioned correctly, allowing cancellation by holding both buttons for 6 s with the device transitioning to STATE\_WAITING\_CANCEL\_ACK and ultimately confirming cancellation upon receiving the corresponding ACK from the Headquarter Device.

2) *Beacon Network Prototype*: The Beacon Network omits the display, GPS, and touch sensors to reduce cost and power consumption. Its enclosure contains only the essential communication and power components.

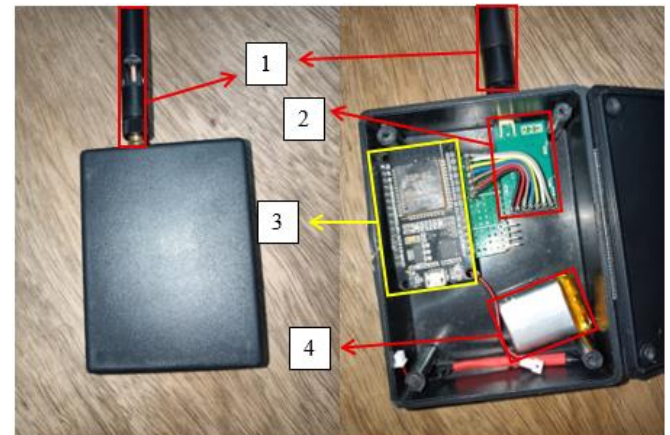


Figure 8. Beacon Network prototype, front view (left) and internal component layout (right)

Referring to the labelled components in the Figure 8: the ESP32-WROOM-32 (3) manages relay logic and heartbeat monitoring. The LoRa module LLCC68 E220-900T-22D (2) with its 5 dB antenna (1) handles reception and forwarding. A 2500 mAh Li-Po battery (4) provides autonomy. The firmware successfully implemented message relay with DJB2 hash-based deduplication and a heartbeat mechanism that monitors the preceding node. During verification, duplicate messages within 60 s were correctly discarded, and the beacon reliably forwarded only new messages.

3) *Headquarter Device Prototype*: The Headquarter Device also omits the OLED display, GPS, and touch sensors but with addition of a piezo buzzer for local audio alerts and relies on Wi-Fi for cloud connectivity.

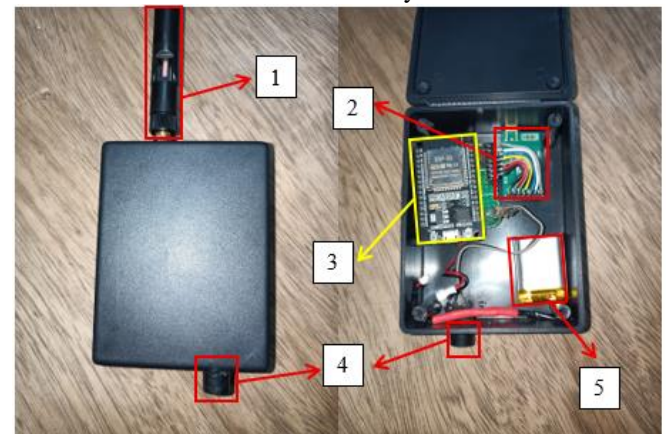


Figure 9. Headquarter Device prototype, front view (left) and internal component layout (right)

Referring to the labelled components in the Figure 9: the LoRa module LLCC68 E220-900T-22D (2) with its 5 dB antenna (1) for receiving messages and send ACK to Field

Device. The ESP32-WROOM-32 (3) decoding and parsing the incoming messages. A 2500 mAh battery (5) supplies backup power. The buzzer (4) is externally mounted for clear sound projection. Upon receiving emergency messages, the device activated the buzzer for a 2-second alert, synchronized time via NTP, and transmitted parsed data to the Blynk platform without errors.

### B. Blynk Dashboard Functionality

The Blynk IoT platform was configured as the cloud-based monitoring interface for the Headquarter Device. A dedicated template called “Disaster Alert System” was created, utilising ten virtual pins (V0–V9) to transmit emergency data. All widgets updated within approximately one second of message reception at the Headquarter Device when an active internet connection was available. The multi-channel notification system operated as configured: push notifications were delivered to the operator’s smartphone with an average latency under two s. The “Critical Alert” setting ensured that notifications bypassed silent and Do Not Disturb modes, making them always audible and visible. Email notifications

were also successfully delivered as a secondary confirmation channel.

The offline queuing capability was validated by disconnecting the smartphone from the internet, triggering an alarm, and then reconnecting the smartphone. The queued push notification was delivered immediately upon reconnection, confirming that no critical alerts are lost during intermittent connectivity. The Location Request feature operated as *intended*; entering an Area ID into the V9 input widget triggered a Location Request message over the LoRa network, and the corresponding Location Response was displayed on the dashboard upon receipt.

The web-based Blynk Console and the mobile Blynk IoT application both displayed real-time information clearly. The “Latest Message” widget (V0) showed the most recent event, the “Status” widget (V6) tracked counts of Help, Alarm, Cancel, and Missing Beacon messages, and the “Alarm Log” terminal (V8) recorded a chronological history with NTP-synchronised timestamps. The Latitude (V3) and Longitude (V4) widgets populated automatically with transmitted coordinates, enabling responders to pinpoint the incident location.

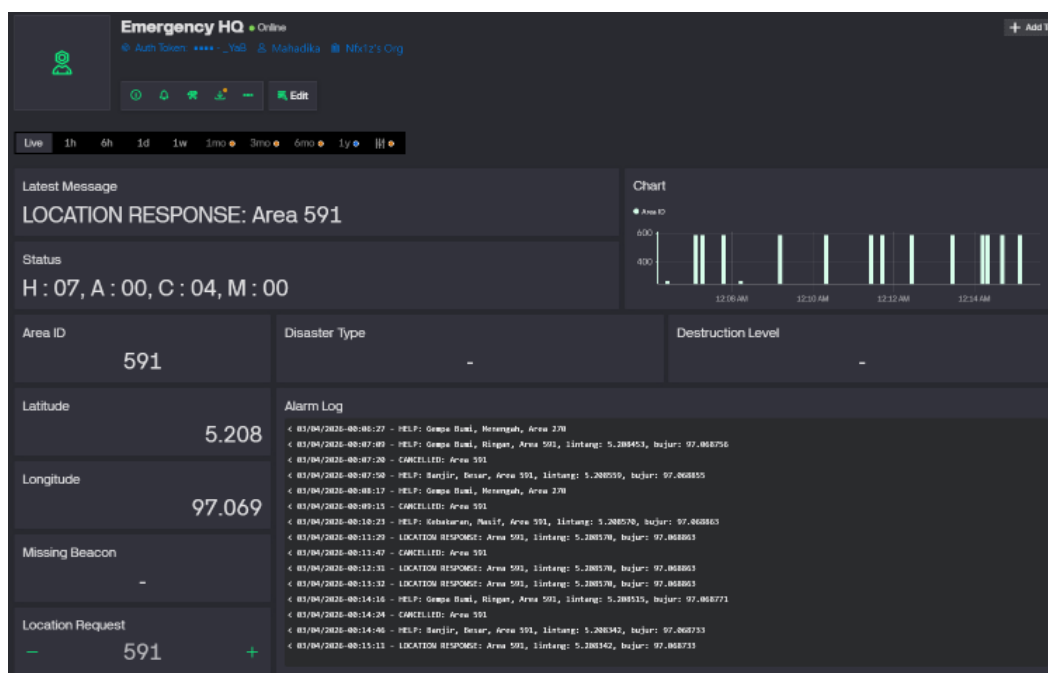


Figure 10. Blynk Web Dashboard (Console)

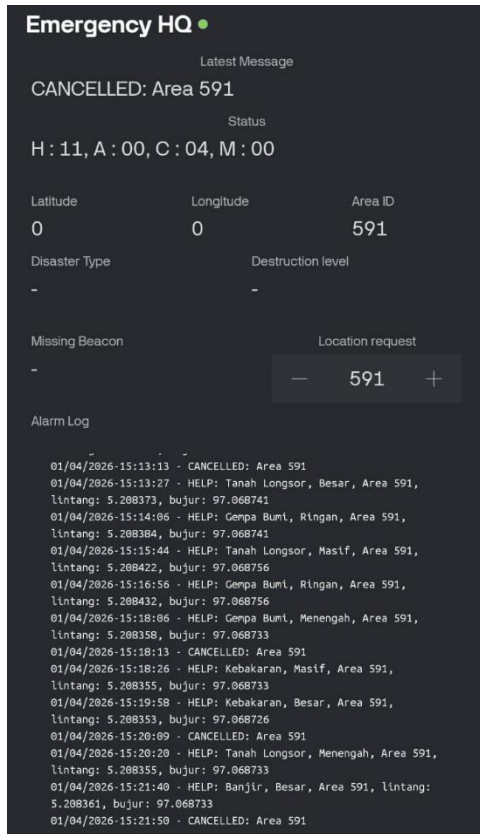


Figure 11. Blynk Mobile Dashboard (IoT App)

C. Mobile Application Functionality

The custom cross-platform mobile application, built with Flutter, was tested for BLE connectivity, user interaction, and GPS fallback capabilities. Upon launch, the application requested the necessary Bluetooth and location permissions and performed a BLE scan. Devices advertising with the prefix `FieldDevice_` were displayed in a scrollable list, showing the Area ID and Received Signal Strength Indicator (RSSI) value. Tapping a device entry initiated a connection request with a 15-second timeout. After successful connection and service discovery, the application subscribed to notifications on the Notify characteristic to receive real-time state updates from the Field Device.

The main interface presented two dropdown menus for disaster type and severity level selection. The current device state was displayed with colour-coded backgrounds—green for success, yellow for idle/waiting, red for failure—mirroring the nine-state Field Device FSM. Pressing the “Kirim Permintaan Bantuan” button transmitted the selected param, along with coordinates obtained from the smartphone’s built-in GPS via the geolocator plugin.

The integrated GPS fallback was verified by deliberately disabling the Field Device’s GPS module. The application successfully embedded the smartphone’s coordinates into the Help message payload, and the Headquarter Device received and displayed the correct location. When smartphone GPS acquisition failed within a 30-second timeout, the application

displayed a warning but still allowed transmission with placeholder coordinates. The locking mechanism preventing simultaneous physical button and application control operated correctly, avoiding command conflicts.

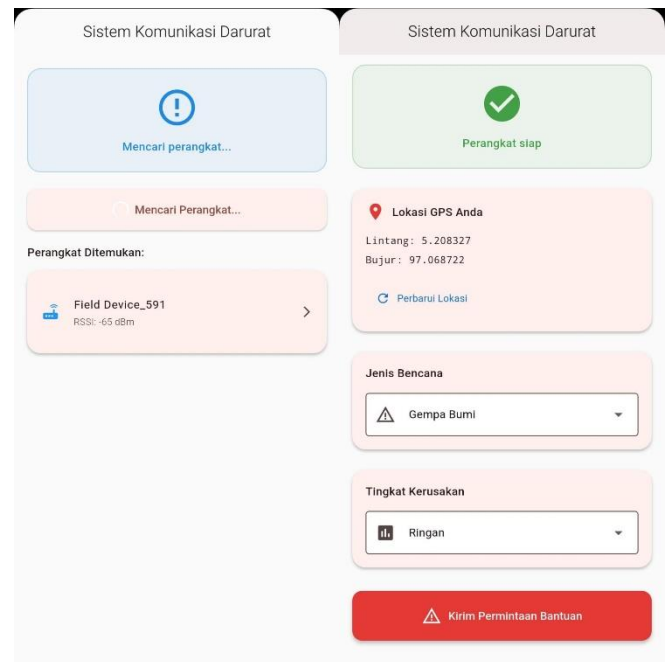


Figure 12. Custom Mobile Application UI

D. Baseline Comparison of Communication Protocol

To quantify the benefit of the custom packed binary format, an identical Help message was encoded in three representations:

- JSON, verbose human-readable format with keys and values.
- Plain String, comma-separated values as used in the initial design specification.
- Custom Packed Binary, that implemented `__attribute__((packed))` C structure.

TABLE XVIII  
PROTOCOL EFFICIENCY BASELINE

Format	Example	Bytes
JSON	{ "type":1,"areaId":591,"disaster":1,"destruction":4,"lat":40.753,"lon":-73.983 }	80
Plain String	1,591,1,4,40.753,-73.983	24
Custom Packed Binary	[0x01][0x4F][0x02][0x01][0x04][0x06F][0x12][0x83][0xC0][0xF5][0x28][0x5C][0x8F][0xC2]	13

The plain string already improves over JSON by removing field names and brackets, but it still carries ASCII digit overhead and requires run-time string parsing. The packed binary completely eliminates delimiters and represents all data in fixed-width native binary fields.

### E. Summary of Performance Testing

All key performance param were evaluated under controlled conditions. Except for maximum range that tested in both urban and rural environments, all measurements were performed at the Universitas Malikussaleh campus.

TABLE XIX  
COMPREHENSIVE PERFORMANCE EVALUATION

Parameter	Target	Result	Status
Packet Delivery Ratio (PDR) at 1000 m	> 90%	100% (20/20)	Achieved
End-to-End Latency	< 60 s	1.02 s (average)	Achieved
BLE Range – Outdoor	> 10 m	16 m	Achieved
BLE Range – Indoor	> 10 m	11 m	Achieved
GPS Accuracy – Outdoor (RMSE)	< 10 m	0.945 m	Achieved
GPS Accuracy – Indoor (RMSE)	< 10 m	28.68 m	Not Achieved
Motion Detection Sensitivity (>5 m in 5 s)	100%	100% (5/5)	Achieved
Motion Detection Specificity (<5 m)	0% false alarms	0% (0/5)	Achieved
Usability – Physical Buttons	< 120 s	14.84 s (average)	Achieved
Usability – Mobile Application	< 120 s	23.85 s (average)	Achieved
Power – Field Device	> 12 h	4.32 h	Not Achieved
Power – Beacon Network	> 12 h	9.18 h	Not Achieved
Power – Headquarter Device	> 12 h	8.03 h	Not Achieved
Direct Max Range – Urban Campus	> 1000 m	1044 m	Achieved
Direct Max Range – Rural Mountainous	> 1000 m	1288 m	Achieved
Extended Range with one Beacon	> 1500 m	2000 m (PDR 100%)	Achieved

The system met or exceeded its targets on all communication, localisation, and usability param. Outdoor GPS achieved sub-meter accuracy, motion detection operated with perfect sensitivity and no false alarms, and both user interfaces enabled rapid emergency reporting well within the two-minute threshold. The linear multi-hop relay doubled the effective range from the maximum direct distances to a full 2000 m while maintaining 100 % PDR, confirming the practicality of the linear multi-hop architecture for spanning larger disaster zones.

The only metrics that did not reach their targets were indoor GPS accuracy and power autonomy. Indoor accuracy degraded to 28.68 m RMSE due to building attenuation; this limitation is mitigated in practice by the mobile application's smartphone GPS fallback. The three devices operated for

4.32 h for the Field Device, 9.18 h for the Beacon Network, and 8.03 h for the Headquarter Device on the 2500 mAh batteries, each well short of the 12-hour goal.

The power results merit further scrutiny. The Li-Po cells used in the prototypes, labelled as 2500 mAh, measure only  $25 \times 30 \times 10$  mm. Given that the total steady-state current draw of the Field Device is well under 169.3 mA with GPS and OLED active and the Beacon and Headquarter draw even less, the observed runtimes are suspiciously low for a genuine 2500 mAh capacity. For example, a 4.32-hour lifetime would imply an average current of approximately 731.4 mA from the battery, which far exceeds the measured consumption of all components combined. This discrepancy suggests the actual capacity of these particular cells is probably far below the label rating, a common problem with small-form-factor Li-Po batteries obtained from non-specialist sources. Therefore, the runtime figures reported here likely reflect a combination of the system's real power draw and the limited capacity of the specific batteries used. Follow-up tests with verified, high-quality batteries are recommended to establish a definitive power envelope.

### IV. CONCLUSION

This research successfully designed, implemented, and rigorously evaluated an integrated emergency communication system based on LoRa, GPS, and BLE. The physical prototypes, a Field Device with a nine-state finite-state machine, a linear multi-hop Beacon relay chain with heartbeat-based failure detection, and a Headquarter Device linked to the Blynk IoT platform operated as intended. The custom packed binary protocol reduced airtime by over 83 % compared to JSON, contributing to a 100 % Packet Delivery Ratio at 1000 m and an average end-to-end latency of 1.02 s.

Twelve of the sixteen performance param met or surpassed their targets, including outdoor GPS accuracy (0.945 m RMSE), motion detection (100 % sensitivity, 0 % false alarms), usability (14.84 s physical, 23.85 s app), BLE range (16 m outdoor, 11 m indoor), and communication range extension to 2000 m via a single beacon relay. Indoor GPS accuracy (28.68 m) was the only metric that fell short, and it is already mitigated by the mobile application's smartphone GPS fallback.

The greatest challenge remains power consumption. The prototype runtimes of 4.32 h, 9.18 h, and 8.03 h fall well below the desired 12-hour mark. This is partially attributable to the likely over-rated capacity of the small form-factor Li-Po cells used, but even with genuine 2500 mAh batteries the system would still require aggressive duty cycling of the GPS, Wi-Fi, and CPU to achieve a full day of autonomous operation. Implementing the proposed sleep and Wake-on-Radio strategies is projected to boost lifetimes beyond the target, and future iterations should include these optimisations alongside the use of verified high-capacity batteries.

Beyond the power issue, the system exhibits two inherent limitations: the linear chain topology restricts scalability to roughly five hops, and the absence of encryption leaves

communications vulnerable. Both are deliberate trade-offs that favour deterministic latency and lightweight implementation for directed evacuation routes. Future work will focus on embedding lightweight authenticated encryption (e.g., ASCON), evaluating the system with more nodes in larger field trials, and transitioning to solar-assisted power to eliminate the battery bottleneck entirely.

In summary, the proposed system provides a robust, low-cost, and empirically validated communication solution for disaster-prone areas. It offers a solid baseline and a replicable methodology for the development of deterministic, energy-aware emergency networks.

#### REFERENCES

- [1] Badan Nasional Penanggulangan Bencana, *Buku Data Bencana Indonesia 2024*, 2024th ed., vol. 3, no. 1. Jakarta: Pusat Data Informasi dan Komunikasi Kebencanaan Badan Nasional Penanggulangan Bencana, 2025. [Online]. Available: [https://bnpb.go.id/storage/app/media/Buletin Info Bencana/Buku Data Bencana 2024/20250613\\_Buku Data 2024.pdf](https://bnpb.go.id/storage/app/media/Buletin%20Info%20Bencana/Buku%20Data%20Bencana%202024/20250613_Buku%20Data%202024.pdf)
- [2] E. Rizal, Y. Winoto, T. Sugito, C. Nugroho, and F. I. Septian, "Disaster communication in the digital age: a community-based case study of media, education, and local knowledge in Pangandaran, Indonesia," *Front. Commun.*, vol. 10, no. October, p. 1632436, Oct. 2025, doi: 10.3389/fcomm.2025.1632436.
- [3] D. D. Pebrian, "Alat Ukur Sinyal Lora Untuk Mengetahui Jangkauan Antara End Device Dengan Gateway," *J. Inform. dan Tek. Elektro Terap.*, vol. 13, no. 1, Jan. 2025, doi: 10.23960/jitet.v13i1.5700.
- [4] D. Shahnowaz Syed, S. B. Hossain, A. Islam, T.-A.- Islam Jim, and S. R. Zishan, "Design and Feasibility Analysis of a LoRa Based Communication System for Disaster Management," *AIUB J. Sci. Eng.*, vol. 23, no. 2, pp. 135–144, Aug. 2024, doi: 10.53799/ajse.v23i2.1055.
- [5] L. C. Hong, "Disaster Resilient Mesh Network Using Lora and Nervenet," *Universiti Tunku Abdul Rahman*, 2022.
- [6] X. Chen, S. Jin, S. Chen, and Z. Fang, "UAV Emergency Rescue System Based on Combined LoRa and NB-IoT Communication Technologies," *Proc. 2025 4th Int. Conf. Big Data, Inf. Comput. Network, BDICN 2025*, pp. 617–621, 2025, doi: 10.1145/3727353.3727452.
- [7] M. Matracia, N. Saeed, M. A. Kishk, and M. S. Alouini, "Post-Disaster Communications: Enabling Technologies, Architectures, and Open Challenges," *IEEE Open J. Commun. Soc.*, vol. 3, no. July, pp. 1177–1205, 2022, doi: 10.1109/OJCOMS.2022.3192040.
- [8] D. Schmidt, F. Kuntke, M. Bauer, and L. Baumgartner, "BPoL: A Disruption-Tolerant LoRa Network for Disaster Communication," *2023 IEEE Glob. Humanit. Technol. Conf. GHTC 2023*, pp. 440–447, 2023, doi: 10.1109/GHTC56179.2023.10354717.
- [9] M. Fikry, Bustami, and E. Suzanna, "IOT-Based Environmental Monitoring and Its Implications For Student Mental Health," *Rev. Gestão Soc. e Ambient.*, vol. 19, no. 2, p. e011183, Feb. 2025, doi: 10.24857/rsgsa.v19n2-022.
- [10] J. Khairina, N. Nurdin, and M. Fikry, "Aplikasi Teknologi Internet of Thing pada Robot Pendeteksi Kebocoran Gas Amonia (NH3)," *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 11, no. 1, pp. 25–34, Feb. 2026, doi: 10.29100/jupi.v11i1.7457.
- [11] F. Maulana *et al.*, "Comparison of Tree-Based and Deep Learning Models for Tremor Related to Parkinson ' s Disease," *Int. J. Act. Behav. Comput.*, vol. 2025, no. 3, pp. 1–26, 2025, doi: 10.60401/ijabc.128.
- [12] M. N. Nizam, Haris Yuana, and Zunita Wulansari, "Mikrokontroler ESP 32 Sebagai Alat Monitoring Pintu Berbasis Web," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 6, no. 2, pp. 767–772, Oct. 2022, doi: 10.36040/jati.v6i2.5713.
- [13] M. Muslim, R. P. Sari, and S. Rahmayuda, "Implementasi Framework Flutter pada Sistem Informasi Perpustakaan Masjid," *Coding J. Komput. dan Apl.*, vol. 10, no. 01, pp. 46–59, May 2022, doi: 10.26418/coding.v10i01.52178.
- [14] A. Lawi, A. A. Dermawan, D. E. Kurniawan, I. M. Reza, F. E. Gulo, and Z. A. Hasibuan, "Exploring the Landscape of Deep Learning Techniques for IoT Data: A Systematic Literature Review", *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 15, no. 2, pp. 578–588, Apr. 2025.
- [15] Kurniawan D E 2017 Push Notification System Pada Prototipe Kendali Listrik Rumah CESS (Journal Comput. Eng. Sci. Syst. 2 89-92 doi: 10.24114/cess.v2i2.6463
- [16] D. G.C., A. Ladas, Y. A. Sambo, H. Pervaiz, C. Politis and M. A. Imran, "An Overview of Post-Disaster Emergency Communication Systems in the Future Networks," in *IEEE Wireless Communications*, vol. 26, no. 6, pp. 132-139, December 2019, doi: 10.1109/MWC.2019.1800467.