

# IoT-Based Real-Time River Monitoring and Early Flood Warning Using ESP32 and HC-SR04

Asri Mulyani\*, Dede Kurniadi, Rizki Esa Saputra

**Abstract**—Floods remain one of the most frequent and destructive natural disasters in Indonesia, primarily caused by rising river water levels. This study presents a prototype of an IoT-based river water level monitoring system using an HC-SR04 ultrasonic sensor, an ESP32 microcontroller, and the Blynk platform for real-time monitoring and alerting. The system classifies water levels into three categories: safe, alert, and danger, triggering local alarms (buzzers) and smartphone notifications accordingly. Evaluation results show a high sensor accuracy of up to 99.7%, with a notification delay averaging 1.2 seconds and a buzzer response time of less than 0.5 seconds. The system has been tested through black box testing and field simulations and has shown reliable performance for early flood warnings. These findings demonstrate the system's potential in enhancing flood preparedness, particularly in vulnerable communities. Future enhancements include deploying waterproof sensors and integrating additional alert platforms such as SMS or sirens.

**Index Terms**— Blynk, Early Flood Warning, ESP32, HC-SR04, IoT, River Monitoring.

## I. INTRODUCTION

Floods are among the most frequent and destructive natural disasters in Indonesia [1], often triggered by unmonitored increases in river water levels, particularly in regions with high rainfall intensity. Floods account for the highest proportion of recorded disaster events due to weather, climate, and hydrometeorological hazards (44%), rank second in terms of the largest economic loss (31%), and cause the third highest number of fatalities (16%) [2]. Floods result in infrastructure damage, economic losses, disruption of daily activities, and threats to public safety. Contributing factors include inadequate drainage systems, rapid urbanization, and reduced water infiltration due to impervious surfaces such as asphalt and concrete.

In many flood-prone areas, residents continue to rely on manual river inspections, especially during the rainy season, which delays emergency responses and reduces preparedness. An automated river water level monitoring system capable of real-time data transmission and early warning alerts is crucial for enhancing disaster mitigation strategies. Effective flood management requires making the right decisions before, during, and after a disaster occurs [3]. This can be achieved through the implementation of an integrated early warning system, which enables a rapid response to minimize the negative impacts of flooding. Recent studies emphasize that Internet of Things (IoT)-based flood monitoring systems significantly reduce flood risks by providing early alerts

and real-time data access to communities and authorities [4].

The advancement of IoT technology has enabled the development of environmental monitoring systems that combine sensors, microcontrollers, and mobile applications. IoT allows users to connect devices to the internet to gather data and remotely manage processes or machinery, facilitating continuous data acquisition and the transmission of real-time alerts through wireless networks [5]. IoT-based technology has also played an important role in strengthening public safety and optimizing emergency response systems by facilitating real-time monitoring and early warning capabilities [6]. Although IoT technology cannot prevent disasters from occurring, it is very useful in supporting preparedness and providing mitigation data, such as early warning information on disasters [7]. Recent studies such as those by Syamsi et al. [8] and Abdulle et al. [9] further reinforce the role of IoT in disaster mitigation. Their findings validate the accuracy of ultrasonic-based monitoring systems and emphasize the importance of multi-platform alerts for increasing community preparedness against flooding risks.

Research by Sharma et al. [10] demonstrates that the implementation of an IoT can significantly enhance disaster management systems by enabling early detection and rapid response. Their framework outlines how various disaster scenarios, such as floods, earthquakes, and wildfires, can be effectively managed by deploying interconnected IoT devices that collect and transmit environmental data in real time. This layered architecture allows local data processing at the edge level for immediate decision-making. The study further emphasizes that timely data exchange, automated alert systems, and real-time monitoring made possible by IoT not only reduce the impact of disasters but also support smarter, more resilient urban infrastructure. By integrating IoT devices to provide real-time water monitoring with high sensor accuracy and stable peer-to-peer communication, Rohan et.al. [11] research shows a significant improvement in disaster response thanks to ease of use and data that is directly received by emergency responders.

Numerous previous studies have delved into this area; however, most exhibit limitations in terms of system integration, alert diversification, or sensor reliability. For instance, Priatim et al. [12] used HC-SR04 sensors with a Raspberry Pi and ThingSpeak-Blynk integration, but relied on limited application functionality and had delayed data visualization. Pratita et al. [13] employed ESP32 and a Proportional Integral Derivative (PID) controller with HC-SRF05 sensors, yet reported a 1.97% average error rate due to ripple and lighting interference. Wandu and Ashari [14] designed a system using ESP32 with HC-SR04 and MH-RD sensors, where alerts were sent only through Telegram

Asri Mulyani (Corresponding Author), Dede Kurniadi, and Rizki Esa Saputra are with Institut Teknologi Garut, Jawa Barat, Indonesia. Correspondence: [asrimulyani@itg.ac.id](mailto:asrimulyani@itg.ac.id).

bots without on-site alarms or broader platform support. Hanggara and Eka Putra [15] developed a system using Arduino UNO, ESP-01, and HC-SR04, with data sent to ThingSpeak, triggering sirens locally but lacking smartphone notifications, and Wicaksono et al. [16] employed various notification methods—Telegram bots, SMS gateways, and websites—yet often lacked multi-platform integration and on-site alerts. Subsequent studies by Pratama et al. [17] created a system using HC-SR04 and NodeMCU ESP8266 to store water level data in a MySQL database and display it via a website, but the system lacked physical alerts and had a 5–6.8 second data delay, Herdiana and Triatna [18], and Hendryady et al. [19] incorporated HC-SR04 ultrasonic sensors with the Blynk app, yet still presented limitations, such as the absence of multi-level warning indicators or physical alert mechanisms. A similar approach was successfully implemented using Long Range Wide Area Network (LoRaWAN) and HC-SR04 sensors, demonstrating effective flood level categorization and mobile app notification integration. Furthermore, Syamsi et al [8] validated the HC-SR04 reliability by reporting a measurement precision of 99.94% and an average error rate of only 0.08%, confirming its suitability for real-time, non-contact water level monitoring.

This study addresses the limitations of previous research by integrating real-time water level monitoring with a warning system designed in three levels based on water surface height, physical alarms (buzzers), and notifications via smartphones. This combination strengthens the preparedness of officers and communities in responding to potential flooding, thereby supporting more effective disaster mitigation efforts.

Compared to prior studies, this research enhances flood monitoring by integrating an HC-SR04 ultrasonic sensor and ESP32 microcontroller with the Blynk app. Unlike systems that rely solely on Telegram or web dashboards, this prototype features both remote smartphone notifications and on-site alerts via a buzzer. Although it does not use additional display hardware such as Light Emitting Diode (LED) or DF Mini Player, the system ensures local alerts through audio signals and real-time updates to Blynk. The system was created to improve both accessibility and responsiveness for monitoring officers in the field.

This research contributes by integrating multi-level alerts (safe, warning, danger) that simultaneously provides local (buzzer) and cloud-based (Blynk app) notifications, combined with real-time performance analysis such as notification delay and buzzer response. Furthermore, the system verifies local alert independence during temporary internet disconnection, ensuring continued operation even under connectivity loss. These combined features have not been quantitatively evaluated together in previous ESP32–HC-SR04–Blynk–based flood monitoring systems.

Although the hardware configuration (ESP32 and HC-SR04) is similar to several existing works, the contribution of this study lies in the integration and quantitative evaluation of system components that are rarely analyzed together in previous IoT-based flood warning prototypes. First, the system verifies local alert independence,

ensuring the buzzer continues to operate during temporary cloud or Wi-Fi disconnection—an essential requirement for early-warning resilience but seldom tested in earlier ESP32–Blynk implementations. Second, this study provides measured real-time performance, including notification latency (1.2 s), buzzer activation time ( $\leq 0.5$  s), and cloud refresh intervals, whereas most previous works only report functional demonstrations without numerical evaluation. Third, a multi-level dual-alert mechanism (safe–warning–danger) is implemented on both the physical device and cloud platform, and its reliability is tested under repeated dynamic water-level scenarios. Finally, this system achieves high accuracy (99.7%) at very low cost (USD 10.8), offering a practical community-level solution that addresses affordability and deployability—gaps that are often overlooked in studies focusing primarily on technical feasibility.

A review of existing studies shows that most IoT-based flood monitoring systems still face several limitations. Many implementations rely solely on cloud alerts (e.g., Telegram, SMS, or dashboards) without local fail-safe mechanisms such as independent buzzer activation. Several studies do not evaluate real-time performance quantitatively, leaving notification latency, buzzer response time, or data refresh intervals unmeasured. Others provide only single-level or binary alerts, lack multi-platform communication, or present no assessment of system behavior during temporary network loss. Furthermore, prior works typically report functional demonstrations rather than controlled calibration with accuracy analysis.

Based on these gaps, this study contributes by integrating and evaluating multi-level dual alerts (buzzer + cloud), measuring end-to-end latency and response times, analyzing local alert independence during network disconnection, and validating sensor accuracy under controlled calibration. These combined contributions address gaps not simultaneously covered in previous ESP32–HC-SR04–based flood warning prototypes.

## II. METHOD

This study employed the prototyping development methodology, which is particularly suitable for system development involving continuous user feedback and iterative refinement [20].

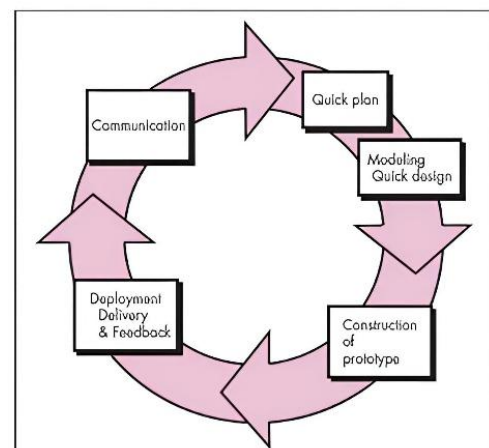


Fig. 1. Flow of the prototyping method

### A. Hardware Architecture

The water level monitoring system comprises an HC-SR04 ultrasonic sensor, an ESP32 microcontroller, a 16×2 Liquid Crystal Display (LCD), and a buzzer for local alerts. The sensor measures the distance from the river surface and sends the data to the ESP32, which processes it to determine the water level status. A smartphone power bank is used to power the system. The processed data is displayed locally and also sent to the Blynk app via Wi-Fi. This setup allows remote monitoring and on-site alarm activation.

### B. Decision Logic and Classification Thresholds

The system uses a threshold-based classification to define three water level conditions:

TABLE I. WATER LEVEL CLASSIFICATION

| Water level | Distance   | System action                                                                                                   |
|-------------|------------|-----------------------------------------------------------------------------------------------------------------|
| Safe        | > 25 cm    | Display status and distance on LCD and Blynk app; buzzer OFF                                                    |
| Warning     | 10 – 25 cm | Display status on LCD and Blynk; yellow LED ON; buzzer sounds briefly; smartphone notification sent             |
| Danger      | ≤ 10 cm    | Display status "Danger!" on LCD and Blynk; red LED ON; buzzer sounds continuously; smartphone notification sent |

The classification thresholds used in this study were determined based on interviews with officers from the Cimanuk River monitoring post and direct observations of typical water level variations during the dry and rainy seasons. The “warning” level was set at a 10 cm rise from the normal level, representing early signs of rapid water increase, while the “danger” level at 25 cm indicates a critical condition where overflow potential becomes significant for nearby residential areas. Although these thresholds were locally informed, future calibration will align them with official hydrological standards and flood stage criteria to enhance system accuracy and regional applicability.

### C. Blynk Integration and Notification Mechanism

The ESP32 connects to the Blynk Cloud using Wi-Fi to send real-time water level data. The Blynk app visualizes data using a gauge and a time-series chart, and issues push notifications for "Warning" and "Danger" conditions. Notification frequency is throttled using the Blynk Notification Limit setting to prevent spam alerts.



Fig. 2. Blynk working principle

## III. RESULTS AND DISCUSSION

The results of this research are presented by explaining the stages in developing a prototype of an IoT-based river water level monitoring system using the HC-SR04 ultrasonic sensor and the ESP32 microcontroller. This system was built by following the prototyping methodology, which consists of several iterative phases to gradually shape the final solution.

### A. Communications

In the communication phase, initial observations and interviews were conducted with water monitoring officers and flood-affected communities living along the Cimanuk River in Garut. The purpose of this phase was to identify the real challenges faced during the flood, such as the absence of a real-time water level monitoring system and an early warning system. The input obtained formed the basis for determining the system requirements.

### B. Quick Plan

In the second stage, the system flow was designed based on data obtained from interviews, which was then visualized in the form of a flowchart. This design aims to provide an overview of the water level monitoring system to be developed. The flowchart showing the workflow of the water level sensor is presented in Fig 3.

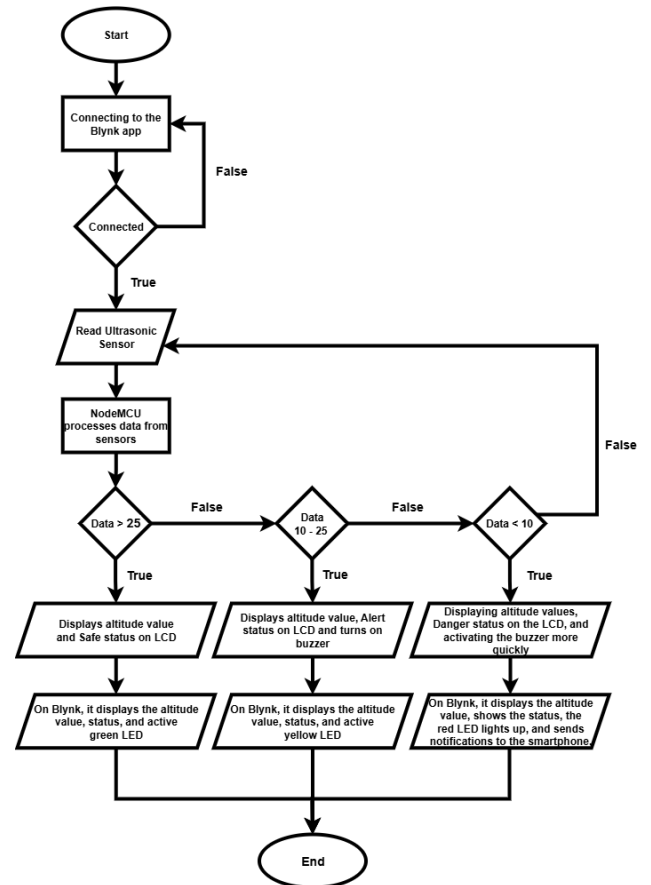


Fig 3. System flowchart

The process begins by connecting the device to the Blynk app. If the connection is successful, the ultrasonic sensor will read the water level in real time. The data obtained from the sensor is then processed by NodeMCU to be classified into three levels of conditions: safe, alert, and danger. In the safe condition, when the distance between the water surface and the sensor is more than 25 cm, the system only displays the water level value and status on the physical LCD and virtual LCD in the Blynk app without activating the buzzer. Meanwhile, in the caution condition, when the distance is between 10 cm and 25 cm, the system will display the data on the LCD, turn on the indicator LED, activate the buzzer, and send a notification to the user's smartphone. In a “danger” condition, where the water level is less than 10 cm, the system will display the status on the LCD, turn on the red

LED, and activate the buzzer as an emergency warning signal. Additionally, the system is equipped with a notification delay setting via the Notification Limit feature in Blynk Cloud, ensuring notifications to the smartphone are sent only once every minimum 10-second interval to prevent repeated notifications in quick succession that could disturb the user.

### C. Modeling Quick Design

In the third stage, a block diagram was designed as a detailed representation of the system workflow based on the previously developed flowchart. This block diagram served as a guide in constructing the prototype and visualizing the interaction between components such as the ultrasonic sensor, NodeMCU ESP32 microcontroller, buzzer, and the Blynk app. In addition, a schematic circuit was also created to depict the wiring and electronic configuration required for assembling the hardware. Both the block diagram and the schematic circuit played a crucial role in ensuring accuracy and consistency during the development process. The block diagram and schematic are presented in Fig. 4.

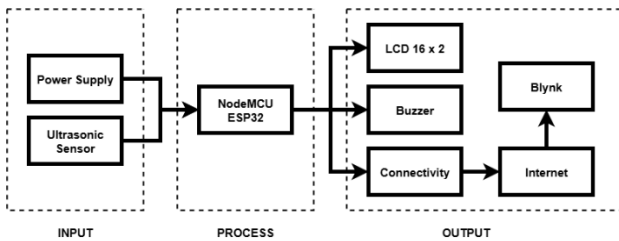


Fig 4. Block diagram

Fig. 4 shows a Block diagram of an IoT-based water level monitoring system consisting of three main parts, namely Input, Process, and Output. In the Input block, there are two important components: Power Supply, which provides power for the entire system, and the HC-SR04 Ultrasonic Sensor, which detects the distance between the water surface and the sensor.

The data from the sensor is then transmitted to the Process block, which is the NodeMCU ESP32, acting as a microcontroller to process the distance data and determine the water condition status. The processing results are then transmitted to various components in the Output block, such as a 16×2 LCD to display water level information and status locally, a buzzer as an audio indicator when the water reaches a warning or danger level, and the Blynk app, which receives and displays data in real-time via an internet connection.

The connection between the NodeMCU and the internet is established via Wi-Fi connectivity, which in this implementation uses a smartphone hotspot or Wi-Fi network. Through this connection, NodeMCU can send data to the Blynk Cloud, enabling users to monitor water conditions remotely via the smartphone app. This connectivity also allows the system to send automatic notifications to the smartphone. Fig 5. shows a diagram of the system circuit designed based on the procedures that have been carried out.

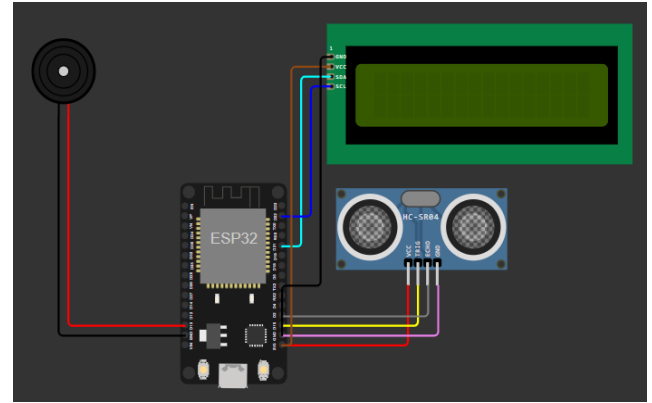


Fig 5. Schematic circuit

TABLE II. COMPONENT PRICES

| Component | Amount | Price  |
|-----------|--------|--------|
| Esp32     | 1      | \$3.3  |
| HC-SR04   | 1      | \$0.7  |
| LCD 16x2  | 1      | \$1.5  |
| Buzzer    | 1      | \$0.3  |
| Battery   | 1      | \$3    |
| Box       | 1      | \$2    |
| Total     |        | \$10.8 |

### D. Construction Of Prototype

The construction of prototype phase involved assembling the hardware and writing the firmware using Arduino IDE with C++.

TABLE III. CODE DECISION LOGIC

```

if (distance > 25) {
    updateBuzzer(false);
    status = "Status : Safe";
    updateBlynkLEDs(true, false, false);
} else if (distance > 10) {
    status = "Status : Warning";
    updateBlynkLEDs(false, true, false);
    if (alarmEnabled) {
        updateBuzzer(true);
        buzzerDelay = 300;
    } else {
        updateBuzzer(false);
    }
} else {
    status = "Status : Danger!";
    updateBlynkLEDs(false, false, true);
    if (alarmEnabled) {
        updateBuzzer(true);
        Blynk.logEvent("banjir_alert",
"Status: Danger!");
    } else {
        updateBuzzer(false);
    }
}

```

The prototype system applies threshold logic using distance data from the HC-SR04 sensor, categorizing water levels into "Safe" (>25 cm), "Warning" (10–25 cm), and "Danger" (≤10 cm). Each status activates specific LED indicators and buzzer responses, with a flood alert notification sent via the Blynk app in the "Danger" condition.

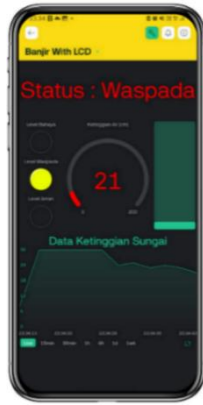


Fig 6. Real-time water level monitoring on Blynk app

Fig. 6 shows the Blynk app interface, which functions as a real-time monitoring platform displaying water level data obtained from ultrasonic sensors. The interface includes a gauge that displays the water level in real time in centimeters, status indicators for “Safe,” “Warning,” and “Danger” levels, and a graph that records river water level trends over time. When the water level reaches the “Warning” threshold, an alert is sent to the user. This enables users to monitor river conditions remotely via their smartphones and take preventive actions if necessary.

#### E. Deployment Delivery & Feedback

In the fifth stage, the application is tested using the Blackbox testing method, which aims to ensure that every function in the system runs as expected based on user requirements. Testing focuses on the output of each input without regard to the internal structure of the program code.



Fig 7. Results of hardware assembly

The use of a controlled water container in this study is not intended to replicate full river dynamics, but rather to allow repeatable, stable, and measurable distance variations needed to evaluate sensor accuracy, threshold classification stability, latency performance, and the independence of local alerts. Controlled laboratory testing is a common approach in IoT prototyping studies because it minimizes environmental noise and enables systematic validation of core functionalities before any potential future field deployment. While this study focuses solely on laboratory-scale evaluation, field testing is identified as future work to further assess system robustness under real environmental conditions once the prototype is fully finalized.

This testing included the system's response to changes in water level, as well as the reaction of devices such as LCD screens, buzzers, LEDs, and notifications via the Blynk app. If any discrepancies were found between the generated output and the expected output, evaluation and correction were performed on the problematic components. Table II displays the results of the testing that has been conducted.

TABLE IV. BLACKBOX TESTING

| Test name                          | Input                                   | Expected process                              | Expected output                                                                                                | Status |
|------------------------------------|-----------------------------------------|-----------------------------------------------|----------------------------------------------------------------------------------------------------------------|--------|
| Safe level measurement             | Water distance > 25 cm                  | Distance reading, no alarm trigger            | LCDs Status: Clear, Buzzer off, Blynk Green LED on, other LEDs off, distance value sent to Blynk               | Pass   |
| Level measurement Alert            | Water distance > 10 cm and $\leq$ 25 cm | Reads distance, buzzer turns on               | LCDs Status: Alert, Yellow LED on, other LEDs off, buzzer sounds momentarily, distance displayed on Blynk app. | Pass   |
| Hazard level measurement           | Water distance $\leq$ 10 cm             | Read distance, buzzer on, event sent to Blynk | The LCD's Status: Danger! Flood!, Red LED on, buzzer sounds continuously, event log sent to Blynk app          | Pass   |
| Data transmission to the Blynk app | Random sensor spacing                   | Data is sent in real-time to Blynk            | Distance and status values appear on Blynk app                                                                 | Pass   |
| Status display on LCD              | All levels (Safe-Hazard)                | LCD's status and distance                     | Line 1: Status, Line 2: Water distance in cm                                                                   | Pass   |
| Alert notification                 | Distance $\leq$ 10 cm                   | System sends event logs to Blynk app          | Turn on the notification in the app, the buzzer sounds, and the LCD status reads "Danger! Flood!"              | Pass   |

The test results show that all main functions operate as expected. The system successfully displays water level data, transmits data to the Blynk app in real time, and sends early warning notifications when the water reaches a critical level. These successful results were achieved thanks to a careful review and evaluation process during the design and development phase.

To evaluate the accuracy of the HC-SR04 sensor, tests were conducted at multiple distances ranging from 1 cm to 50 cm. Manual measurements were taken using a ruler with a precision of 1 millimeter, and the corresponding sensor readings were recorded to calculate the error and

accuracy. The results of the accuracy test are summarized in Table V.

TABLE V. ACCURACY TEST RESULTS OF THE HC-SR04 SENSOR

| Test Scenario | Manual Measurement (cm) | Sensor (cm) | Error (cm) | Accuracy (%) |
|---------------|-------------------------|-------------|------------|--------------|
| Safe          | 50                      | 49.6        | 0.4        | 99.20        |
| Safe          | 40                      | 39.8        | 0.2        | 99.50        |
| Safe          | 30                      | 29.8        | 0.2        | 99.33        |
| Warning       | 20                      | 19.9        | 0.1        | 99.50        |
| Warning       | 15                      | 14.9        | 0.1        | 99.50        |
| Warning       | 12                      | 12          | 0          | 100          |
| Danger        | 10                      | 10          | 0          | 100          |

| Test Scenario | Manual Measurement (cm) | Sensor (cm) | Error (cm) | Accuracy (%) |
|---------------|-------------------------|-------------|------------|--------------|
| Danger        | 5                       | 5           | 0          | 100          |
| Danger        | 2                       | 2           | 0          | 100          |
| Danger        | 1                       | 1           | 0          | 100          |

By comparing sensor readings with visual ruler measurements across ten test scenarios, the study found that the HC-SR04 ultrasonic sensor provided consistent and stable measurements under controlled laboratory conditions.

While the HC-SR04 sensor achieved 99.7% accuracy under controlled laboratory conditions, this value reflects the ideal performance of the sensor in stable, low-noise environments. Real river environments typically contain additional factors such as water turbulence, flow velocity changes, floating debris, rainfall, wind-induced ripples, and temperature fluctuations, all of which may introduce measurement noise and reduce effective accuracy. Because this study focuses on controlled prototyping and quantitative calibration, validation under real environmental conditions was not conducted and remains an important direction for future work. Therefore, the reported accuracy should be interpreted as laboratory-calibrated performance rather than field-level reliability.

The accuracy values shown in Table V were obtained by comparing ten sets of ultrasonic sensor readings against manual measurements using a precision ruler with a 1 mm resolution. Each measurement was repeated three times at varying water levels ranging from 1 cm to 50 cm, and the average difference was used to compute the accuracy percentage. The resulting mean accuracy of 99.7% demonstrates that the HC-SR04 sensor performs consistently under controlled conditions.

Although the accuracy calculation demonstrates consistent sensor readings under controlled conditions, the present study did not perform comprehensive statistical analyses such as standard deviation, repeatability testing, regression fitting, or RMSE evaluation due to the limited size of the calibration dataset. These metrics are important for assessing sensor stability, noise characteristics, and long-term performance. Future studies will incorporate a larger number of measurements under varying environmental conditions to enable statistical validation and reliability assessment using RMSE, standard deviation, correlation coefficients, and multi-iteration repeatability analysis. This approach will allow a more robust evaluation of the sensor's performance beyond the preliminary calibration results presented in this prototype stage.

In terms of scalability, the proposed system can be extended to support multiple sensor nodes deployed along different points of a river. Each node can operate independently using local alert mechanisms while transmitting data to a centralized cloud platform. However, a Wi-Fi-based architecture may not be ideal for long river segments due to range limitations and increased bandwidth usage when multiple devices send data simultaneously. For large-scale deployment, long-range communication technologies such as LoRa/LoRaWAN or GSM/MQTT are more suitable because they support low-power operation, wide coverage, and more stable multi-node communication. Network latency may increase slightly as more nodes publish data to the cloud, but this can be

mitigated through optimized sampling intervals, MQTT-based lightweight messaging, and gateway buffering. Therefore, while the current prototype demonstrates single-node performance, the architecture can be scaled into a distributed multi-node system with appropriate communication technology and network design.

In addition to the Wi-Fi connection used in this prototype, alternative communication technologies such as GSM, MQTT, and LoRa/LoRaWAN are widely used in flood early-warning systems and offer different trade-offs in terms of cost, coverage, latency, and reliability. GSM-based systems (SMS/Call) typically provide wide-area coverage and higher reliability in remote locations, but incur higher operational costs due to SIM card and data subscription fees, and generally exhibit 2–5 seconds latency depending on network load. LoRa provides ultra-long-range connectivity (up to 5–10 km) with extremely low power consumption and no subscription fees, making it suitable for multi-node river monitoring; however, it offers low bandwidth and limits message size. MQTT is a lightweight communication protocol optimized for IoT, providing low-latency message delivery (<1 second) and efficient multi-node data synchronization, but requires a stable gateway or cloud broker. Compared to these alternatives, Wi-Fi offers the lowest cost and fastest local latency but is limited by coverage and network stability, making it suitable for prototype-level testing and community-scale deployments.

By presenting Wi-Fi performance alongside the characteristics of GSM, LoRa, and MQTT, this study positions the proposed system within the broader landscape of IoT communication technologies and clarifies the practical considerations for selecting communication methods in large-scale flood monitoring deployments.

In terms of power management, the current prototype was operated using a portable power bank for testing convenience. For long-term deployment, this setup is impractical; future implementations will incorporate solar-powered charging systems or lithium-ion batteries, along with low-power operation modes such as ESP32 deep sleep to extend operational lifespan.

A similar validation was also carried out by previous researchers under more dynamic and turbulent conditions. A study concluded by Pereira et al [21] evaluated the accuracy of ultrasonic sensors for water level measurements in open channels, considering both permanent and transient flow regimes.

TABLE VI. REAL-TIME SYSTEM RESPONSE

| Evaluation Parameters                   | Measured Value     | Description                                                       |
|-----------------------------------------|--------------------|-------------------------------------------------------------------|
| Notification delay (Blynk app)          | 1.2 seconds        | Time from threshold detection to smartphone notification          |
| Buzzer response time                    | $\leq 0.5$ seconds | Time from threshold detection to buzzer activation                |
| Sensor data update interval (Blynk app) | $\leq 1$ second    | Real-time data refresh interval on the Blynk cloud platform       |
| Local alert independence                | Yes                | The Buzzer works even if the network is temporarily disconnected. |

To evaluate the real-time reliability of the system, several performances were measured during testing. The average time delay between water level threshold detection

and notification delivery to the smartphone via the Blynk app was approximately 1.2 seconds. The buzzer response time after threshold detection was nearly instantaneous ( $\leq 0.5$  seconds). The response time analysis was conducted using a 2.4 GHz Wi-Fi network under stable connectivity conditions. Each measurement was repeated ten times to ensure consistency. The system achieved an average notification delay of 1.2 seconds with a standard deviation of 0.16 seconds, demonstrating stable network performance and reliable alert transmission during the tests. These results indicate that the system responds promptly to rising water levels, providing both remote and on-site alerts with minimal latency, which is essential for effective early flood warnings.

To evaluate the performance and novelty of the proposed system, a comparative analysis was conducted with several previous studies related to IoT-based flood early warning systems. Table V presents a summary of this comparative analysis to highlight the strengths and limitations of each approach, including the current study.

TABLE VII. COMPARATIVE ANALYSIS OF IoT-BASED FLOOD WARNING SYSTEMS

| Study               | Technology          | Communication method                      | Result                                                                                                    |
|---------------------|---------------------|-------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| This study          | IoT (Wi-Fi + Blynk) | ESP32, HC-SR04, Buzzer, Blynk             | 99.7% accuracy; 1.2 s notification delay; low cost (~USD 10)                                              |
| Ragnoli et al [22]  | LoRa / LoRaWAN      | LoRa Node, Ultrasonic Sensor, Solar Panel | long-range transmission (>5 km); validated in real flood environment; excellent energy efficiency         |
| Nurzaid et al. [23] | GSM                 | GSM (SMS / Call)                          | Provides three water-level stages (Normal–Warning–Danger); sends SMS and call alerts to users; ~3 s delay |
| Nur [24]            | Arduino + GSM       | GSM                                       | Real-time FEWS with Blynk App and solar power                                                             |
| Priatim et al. [12] | Raspberry Pi, Blynk | HC-SR04                                   | There is a 5-second time gap between each level change                                                    |

Although the prototype has demonstrated stable performance under controlled laboratory conditions, further validation in real river environments is necessary to assess its robustness against environmental disturbances such as wind, rainfall, debris movement, and flow turbulence. Future experiments will focus on deploying the system near an actual river to evaluate sensor stability, signal noise, and overall system reliability under dynamic field conditions.

#### IV. CONCLUSION

This research successfully developed and evaluated a real-time river water level monitoring system based on IoT technology. The prototype integrates an HC-SR04 ultrasonic sensor with an ESP32 microcontroller and the Blynk application to measure and transmit water level data. Through threshold-based classification, the system provides dual alerts via both a buzzer and smartphone notifications. Test results indicate that the system operates

reliably, with a measurement accuracy of up to 99.7%, a notification delay of 1.2 seconds, and real-time responsiveness.

While the system performed well under laboratory-scale simulations, field validation remains an important future step. The next stage of research will involve deploying the prototype in real river environments to examine its resistance to environmental factors such as temperature variation, humidity, debris, and water turbulence, thereby ensuring its readiness for practical flood monitoring applications. To increase robustness, future work will also focus on replacing the ultrasonic sensor with waterproof models such as JSN-SR04T and expanding alert mechanisms to include public sirens, SMS, or integration with community-based warning infrastructure.

In addition to the laboratory validation, the scalability and sustainability of the proposed system have also been considered. The architecture can be expanded into a multi-node network where several sensors are deployed along a river segment to provide distributed monitoring and real-time data synchronization to the cloud. System maintenance is simplified by using modular components that can be easily replaced or recalibrated when needed, ensuring long-term operability at a low cost. Furthermore, the system can be integrated with national disaster management platforms such as Indonesia's Badan Nasional Penanggulangan Bencana (BNPB) through standardized data APIs or cloud-to-cloud communication, enabling early warning coordination and faster response across regional and national levels.

Despite its promising performance, the system still has several limitations that must be addressed before real-world deployment. The current prototype was tested only under controlled laboratory conditions and has not yet been evaluated in environments with varying temperature, rainfall, debris movement, or turbulent flow, which may affect long-term accuracy and signal stability. From a cost perspective, the low hardware cost of USD 10.8 makes the system highly suitable for community-level adoption, but large-scale deployment would require additional investment in long-range communication modules, protective enclosures, and sustainable power sources such as solar panels.

In the future, the system can be integrated with regional or national flood warning infrastructures, including BPBD and BNPB platforms, through standardized APIs, MQTT brokers, or cloud-to-cloud communication. Such integration would enable automated data sharing, wider alert dissemination, and coordinated responses during flood events. These enhancements, along with field validation and multi-node scalability, represent important directions for further development to ensure the system's practical feasibility and long-term reliability.

#### REFERENCES

- [1] F. Arifin, A. Muhibuddin, and H. Saleh, "Mitigasi Kawasan Rawan Banjir Pada Daerah Aliran Sungai Jeneberang Dengan Menggunakan Sistem Informasi Geografis," *Urban Reg. Stud. J.*, vol. 6, no. 1, pp. 71–82, 2023, doi: 10.35965/ursj.v6i1.3813.
- [2] N. Rindsfuser, A. P. Zischg, and M. Keiler, "Monitoring flood risk evolution: A systematic review," *iScience*, vol. 27, no. 9, p. 110653, 2024, doi: <https://doi.org/10.1016/j.isci.2024.110653>.
- [3] L. Wang *et al.*, "A review of the flood management: from flood control to flood resilience," *Heliyon*, vol. 8, no. 11, p. e11763, 2022, doi: 10.1016/j.heliyon.2022.e11763.

- [4] A. M. Abdulle, A. S. Ali, A. S. Ali, M. A. Ali, S. A. Kahie, and N. A. Mohamed, *IoT-based river monitoring and alerting system to mitigate flood damage*. INC, 2024. doi: 10.1016/B978-0-443-13268-1.00022-4.
- [5] R. Setiawan and M. Sujatmika, "Prototype Deteksi Banjir Menggunakan Sensor Ultrasonik, Dan Water Level Sensor Dengan Notifikasi Blynk," *J. Teknol. Dan Sist. Inf. Bisnis-JTEKSIS*, vol. 4, no. 2, p. 462, 2022, [Online]. Available: <https://doi.org/10.47233/jteksis.v4i2.573>
- [6] W. Leong, "Internet of Things for Enhancing Public Safety, Disaster Response, and Emergency Management," p. 61, 2025, doi: 10.3390/engproc2025092061.
- [7] M. Anbarasan *et al.*, "Detection of flood disaster system based on IoT, big data and convolutional deep neural network," *Comput. Commun.*, vol. 150, no. August 2019, pp. 150–157, 2020, doi: 10.1016/j.comcom.2019.11.022.
- [8] N. Syamsi, M. Ulfah, and D. Lesmidayarti, "Water Level Monitoring for Flood Early Mitigation Based on Internet of Things (IoT)," *Tepian*, vol. 5, no. 2, pp. 58–64, 2024, doi: 10.51967/tepian.v5i2.2504.
- [9] A. M. Abdulle, A. S. Ali, A. S. Ali, M. A. Ali, S. A. Kahie, and N. A. Mohamed, "22 - IoT-based river monitoring and alerting system to mitigate flood damage," in *Advanced Studies in Complex Systems*, F. B. T.-C. I. and B. in C. S. Al-Turjman, Ed., Morgan Kaufmann, 2024, pp. 233–243. doi: <https://doi.org/10.1016/B978-0-443-13268-1.00022-4>.
- [10] K. Sharma, D. Anand, M. Sabharwal, P. K. Tiwari, O. Cheikhrouhou, and T. Frikha, "A Disaster Management Framework Using Internet of Things-Based Interconnected Devices," *Math. Probl. Eng.*, vol. 2021, no. 1, p. 9916440, Jan. 2021, doi: <https://doi.org/10.1155/2021/9916440>.
- [11] M. Rohan, M. Khatun, M. Rahman, and A. Pathak, "Enhancing Flood Disaster Response Through Real-Time Monitoring and IoT: The Case of SENTRYLeaf," *Internet Things Cloud Comput.*, vol. 13, no. 1, pp. 1–14, 2025, doi: 10.11648/j.iotcc.20251301.11.
- [12] R. A. Priatim, M. Asri, and S. Abdussamad, "Rancang Bangun Prototipe Peringatan Dini Banjir Menggunakan Raspberry Pi Berbasis IoT," *Jambura J. Electr. Electron. Eng.*, vol. 5, no. 2, pp. 216–221, 2023, doi: 10.37905/jjee.v5i2.19696.
- [13] L. A. Pratita, S. Nurcahyo, and H. K. Safitri, "Sistem Monitoring dan Kontrol Level Air pada Pengelolaan Sumber Daya Air Menggunakan Kontrol PID," vol. 3, no. 3, pp. 132–141, 2024.
- [14] I. A. Wandu and A. Ashari, "Monitoring Ketinggian Air dan Curah Hujan Dalam Early Warning System Bencana Banjir Berbasis IoT," *IJEIS (Indonesian J. Electron. Instrum. Syst.)*, vol. 13, no. 1, 2023, doi: 10.22146/ijeis.83569.
- [15] F. D. Hanggara and R. D. Eka Putra, "Purwarupa Perangkat Deteksi Dini Banjir Berbasis Internet Of Things," *J. Inform. dan Rekayasa Elektron.*, vol. 4, no. 1, pp. 87–94, 2021, doi: 10.36595/jire.v4i1.349.
- [16] A. M. Wicaksono, Y. Hasan, A. Rahman, I. O. Things, and S. Ultrasonik, "Rancang Bangun Sistem Pendeteksi Banjir pada Waduk Menggunakan Water Level Sensor Berbasis IOT (Internet of Thing)," vol. 15, no. x, pp. 173–177, 2021.
- [17] N. Pratama, U. Darusalam, and N. D. Nathasia, "Perancangan Sistem Monitoring Ketinggian Air Sebagai Pendeteksi Banjir Berbasis IoT Menggunakan Sensor Ultrasonik," *J. Media Inform. Budidarma*, vol. 4, no. 1, p. 117, 2020, doi: 10.30865/mib.v4i1.1905.
- [18] Y. Herdiana and A. Triatna, "Prototype Monitoring Ketinggian Air Berbasis Internet of Things Menggunakan Blynk Dan Nodemcu Esp8266 Pada Tangki," *J. Inform.*, vol. 07, pp. 1–11, 2020.
- [19] D. Hendryady, N. Syam, S. Komputer, and I. Bina Adinata, "Prototype Monitoring Pendeteksi Banjir Menggunakan Aplikasi Blynk Berbasis Esp32," *Amm. J. Syst. Inf. Comput. Inst. Teknol. dan Bisnis Bina Adinata*, vol. 1, no. 1, pp. 20–34, 2023, [Online]. Available: <https://journal.bina-adinata.ac.id/index.php/ammatoa/article/view/4>
- [20] R. S. Pressman, *Software Quality Engineering: A Practitioner's Approach*, vol. 9781118592. 2014. doi: 10.1002/9781118830208.
- [21] T. S. R. Pereira, T. P. de Carvalho, T. A. Mendes, and K. T. M. Formiga, "Evaluation of Water Level in Flowing Channels Using Ultrasonic Sensors," *Sustain.*, vol. 14, no. 9, pp. 1–20, 2022, doi: 10.3390/su14095512.
- [22] M. Ragnoli, G. Barile, A. Leoni, and G. Ferri, "An Autonomous Low-Power LoRa-Based Flood-Monitoring System," 2020.
- [23] N. M. Zain, L. S. Elias, Z. Paidi, and M. Othman, "Flood Warning and Monitoring System ( FWMS ) using GSM Technology," vol. 5, no. 1, pp. 8–19, 2020.
- [24] H. Nur and P. Wisudawan, "Design and Implementation of Real-Time Flood Early Warning System ( FEWS ) Based on IoT Blynk Application," vol. 13, no. 2, pp. 113–121, 2021.



**Asri Mulyani** is an Associate Professor in the Department of Computer Science at Institut Teknologi Garut and a certified lecturer by the Ministry of Research, Technology, and Higher Education of the Republic of Indonesia. She is a Doctoral candidate in Computer Science at Dian Nuswantoro University. Her research interests include computer vision, deep learning, intelligent systems, software engineering, and machine learning. (email: [asrimulyani@itg.ac.id](mailto:asrimulyani@itg.ac.id)).



**Dede Kurniadi** is an Associate Professor in the Department of Computer Science at Institut Teknologi Garut and a certified lecturer by the Ministry of Research, Technology, and Higher Education of the Republic of Indonesia. His Doctoral degree was in Computer Science from Binus University. His research interests include intelligent systems, deep learning, machine learning, computer vision, and software engineering. (email: [dede.kurniadi@itg.ac.id](mailto:dede.kurniadi@itg.ac.id)).



**Rizki Esa Saputra** received his bachelor's degree in Informatics Engineering from Institut Teknologi Garut, Indonesia. His research interests focus on software engineering, the Internet of Things, and software development. (email: [rizkiesasaputra@gmail.com](mailto:rizkiesasaputra@gmail.com))