The International Journal of Embedded Computer Engineering (IJEmCE), Vol. 2, No. 2, October 2026.
https://ejurnal.ung.ac.id/index.php/IJEmCE

IoT-Based Flood Early Detection System Study Using ESP32, HC-SR04 Sensor, ThingSpeak, and Telegram

Iskandar Zulkarnain Nasibu[1*], Bambang Panji Asmara[2], Moh. Ali Algifari Usman[3], Ikhsan Hidayat[4], Nabilah Auliya Fahmi[5], Muh. Ibrahim Al Ghiffari[6], Reja Aswad Podungge[7], Alde Reza Purnama Daliuwa[8]

Computer Engineering, Universitas Negeri Gorontalo[1-8]

mohalialgifariusman@gmail.com

Abstract **–** Floods are one of the most frequent natural disasters in Indonesia and cause significant human and material losses. Effective flood disaster management requires an early warning system capable of providing real-time information to the public and relevant agencies. This study designs and implements a prototype of an Internet of Things (IoT)-based flood early detection system using an ESP32 microcontroller as the main processing unit. An HC-SR04 ultrasonic sensor is used to continuously measure water level. Water level data is sent to the ThingSpeak cloud platform for monitoring and data visualization purposes, as well as forwarded as warning notifications via Telegram Bot to registered users. In addition, a buzzer is installed as a local alarm to provide immediate warnings at the monitoring location. Tests conducted show that the system is able to detect changes in water level with an average measurement error of 1.05% and an average notification latency of 2.7 seconds. This system successfully distinguishes four status levels: safe, alert I, alert II, and emergency, which can be utilized as an affordable and easy-to-implement IoT-based flood monitoring solution.

Keywords:
Early Detection;
ESP32;
Flood;
IoT;
Thingspeak

Article History:
Received: Sept 1, 2026
Revised: Sept 12, 2026
Accepted: Octo 1, 2026
Published: Octo 31, 2026

Copyright © 2026 IJEmCE. All rights reserved.

I. INTRODUCTION

Floods are a natural disaster that repeatedly strikes various regions in Indonesia, particularly during the rainy season. According to a study [1], global flood management is now shifting from conventional control approaches to resilience based on real-time monitoring systems. Meanwhile, reference [2] emphasizes that continuous flood risk monitoring is key to reducing human and material losses due to flooding.

Flood mitigation efforts require a reliable early warning system capable of delivering information quickly to stakeholders. Conventional early warning systems are limited by high installation costs and minimal integration with digital platforms that are easily accessible to the public [3]. Reference [4] shows that ultrasonic sensors are an accurate and affordable solution for measuring water levels in open channels.

The development of Internet of Things (IoT) technology opens up great opportunities to develop more affordable and easy-to-implement environmental monitoring systems. In the journa [5] it is proven that the ESP32 combined with the HC-SR04 sensor is able to provide real-time river level monitoring with sufficient accuracy. Reference [6] also confirms that the HC-SR04 sensor on the NodeMCU-based IoT platform provides consistent and reliable distance readings.

Several previous studies have explored the use of IoT technology for flood monitoring systems. [7] a flood detection system was designed using IoT-based ESP32 with a prototype method. [8] a flood water level measuring device was developed using ESP32 integrated with IoT-based Telegram. However, these studies still lack the simultaneous integration of long-term cloud data storage, historical visualization, and a user-friendly notification system.

This study proposes a prototype of an early flood detection system that integrates an HC-SR04 sensor for water level measurement, an ESP32 microcontroller as the processing unit, the ThingSpeak platform for cloud data storage and visualization, and a Telegram Bot for notifications. The literature in [9] demonstrates that combining a Telegram Bot with an IoT cloud platform significantly enhances monitoring accessibility; therefore, a similar approach is adopted in this study. The integration of these four components

is expected to provide a comprehensive and efficient solution for IoT-based flood monitoring.

II. RESEARCH METHODS

This study employs an experimental research approach using a prototype-based system design method. The research stages encompass system requirements identification, hardware design, software design, system integration, as well as performance testing and evaluation.

A. System Architecture

The developed system consists of three main layers: the hardware layer, the communication layer, and the application layer. This three-layer architectural approach aligns with the design in [10] for developing an ESP32-based river water level monitoring system for flood mitigation.

At the hardware layer, the HC-SR04 sensor periodically measures the water surface distance every 2 seconds. This data is then processed by the ESP32 to determine the water level status. At the communication layer, the ESP32 transmits data to ThingSpeak via the HTTP protocol every 30 seconds and sends notifications to the Telegram Bot via HTTPS whenever a status change occurs or a hazardous condition is detected. A review in [11] emphasizes that the selection of the appropriate communication protocol strictly determines data transmission reliability in IoT-based flood monitoring systems.

B. Hardware Components

Table I presents the hardware components used, along with their respective functions and specifications. Figure 1 shows the overall physical prototype of the assembled system. The selection of the ESP32 as the primary microcontroller is based on the findings in [12], which confirm the capability of the ESP32 to simultaneously manage sensors and network connections with efficient power consumption.

TABLE I.
Hardware Components of the Flood Early Detection System

No	Component	Function	Spesification
1	ESP32 Devkit v1	Primary microcontroller	240 MHz, Wi-Fi & BT
2	HC-SR04	Ultrasonic sensor	2-400 cm, 3 mm resolution
3	Buzzer	Local warning alarm	5V, 2,3 kHz frequency
4	Jumper Wires	Electrical current and data signal connector	15 cm lenght, male to male dan male to female types
5	BreadBoard	Circuit prototyping board	400 tie-points, dimensions 8.2 x 5.5 cm
6	Resistor	Voltage divider	1 kΩ
7	ThingSpeak	Cloud IoT platform	15-second update interval
8	Telegram Bot	User notification	API HTTP/HTTPS

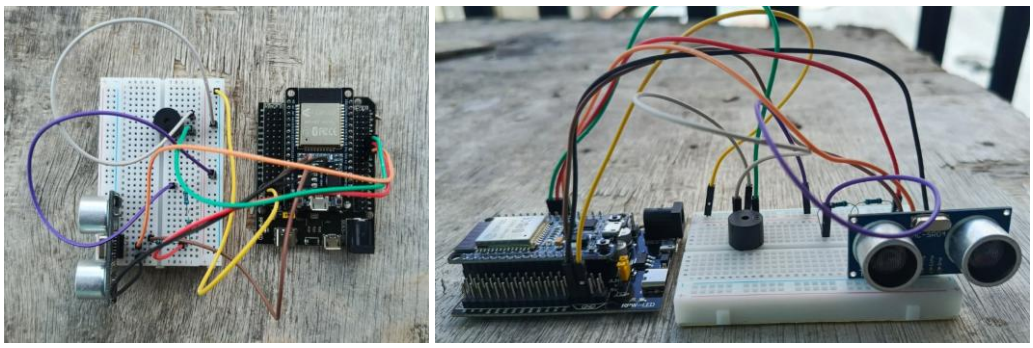


FIGURE 1.
Prototype of the Early Flood Detection System (ESP32, HC-SR04, Buzzer, and Testing Container

C. Software Design

The system software is developed using the Arduino IDE with the C++ programming language. The main program running on the ESP32 consists of four primary functions: system initialization, sensor reading, data transmission to ThingSpeak, and Telegram notification delivery. The overall system workflow is illustrated in the flowchart in Figure 2.

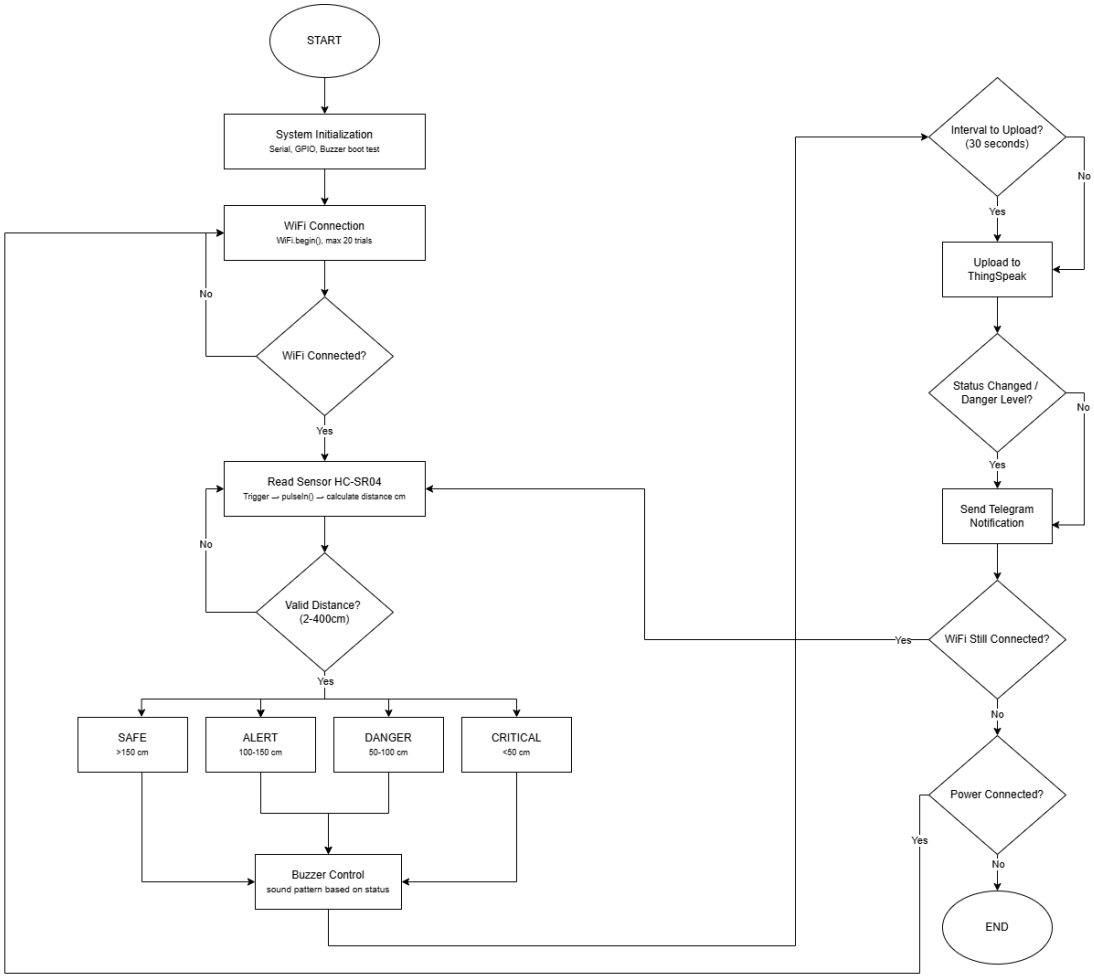


FIGURE 2.
Workflow Flowchart of the Early Flood Detection System

D. Determination of Status Thresholds

The system implements four levels of warning status that refer to the standards used by the Regional Disaster Management Agency (RDMA), namely: (1) Safe Status, if the water level is above 150 cm from the sensor; (2) Alert I Status, if the water level is within the range of 101–150 cm; (3) Alert II Status, if the water level is within the range of 50–100 cm; and (4) Emergency/Flood Status, if the water level is below 50 cm from the sensor, indicating that the water has nearly reached the critical point. Each status change will trigger an automatic notification via the Telegram Bot to registered users, while simultaneously activating a buzzer with a distinct sound pattern according to the danger level. The water level data is sent to ThingSpeak every 15 seconds to adhere to the minimum data update limit of the free ThingSpeak account. ThingSpeak's capability to receive, store, and visualize sensor data in real-time makes it an appropriate platform for IoT-based environmental monitoring applications [13].

E. Testing Procedure

System testing is conducted in two stages. The first stage is sensor accuracy testing, which compares the readings of the HC-SR04 sensor with manual measurements using a ruler as a reference across four different water level scenarios. The second stage is overall system response testing, which measures the notification latency from the moment a water level change occurs until the notification is received by the user via Telegram.

III. RESULT AND DISCUSSION

A. Sensor Accuracy Testing Result

The accuracy testing of the HC-SR04 sensor was conducted by comparing the distance measured by the system with the actual distance measured using a ruler across four water level condition scenarios. Table II

presents the complete results of this testing.

TABLE II.
Sensor Accuracy and System Response Testing Results

Condition	Measured Distance (cm)	Actual Distance (cm)	Error (%)	System Status
Safe	155,6	156	0,4	Normal
Alert I	149,2	150	0,8	Warning
Alert II	79,2	80	0,8	Danger
Flood	7,8	10	2,2	Emergency

Based on the data in Table II, the average measurement error percentage of the HC-SR04 sensor is 1.05%. The highest error value occurs under the emergency condition (2.2%), which is presumably caused by the imperfect reflection of ultrasonic waves on the turbulent water surface due to simulated flood conditions. Overall, the accuracy level of the sensor remains within a highly acceptable tolerance limit for a prototype-scale flood monitoring application.

B. System Response Testing Results

The latency testing for Telegram notifications was conducted over ten trials for each status change condition. The test results showed an average notification latency of 2.7 seconds, with a minimum value of 2.1 seconds and a maximum value of 4.3 seconds. This latency variation was influenced by the internet network conditions during the testing.

The ThingSpeak system successfully received and displayed water level data consistently at the configured 15-second intervals. The historical graphs generated by ThingSpeak can be seen in Figure 7, which facilitates the trend analysis of water level changes over time, thereby supporting better decision-making for disaster management agencies.

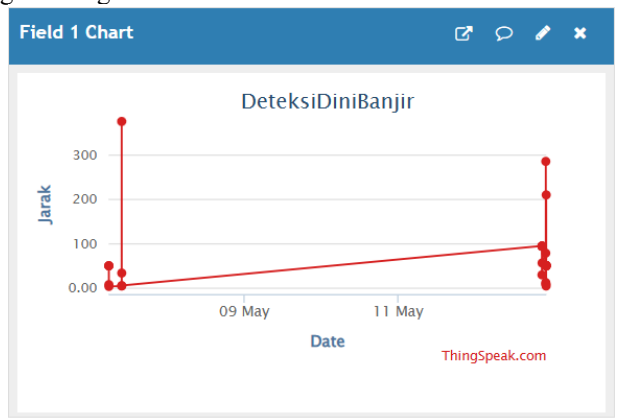


FIGURE 3.
Dashboard View ThingSpeak: Water Level Graph

Warning notifications are automatically sent via the Telegram Bot to registered users whenever a change in the water level status occurs. Figure 8 shows the display of notifications received by users on the Telegram application for each status condition.

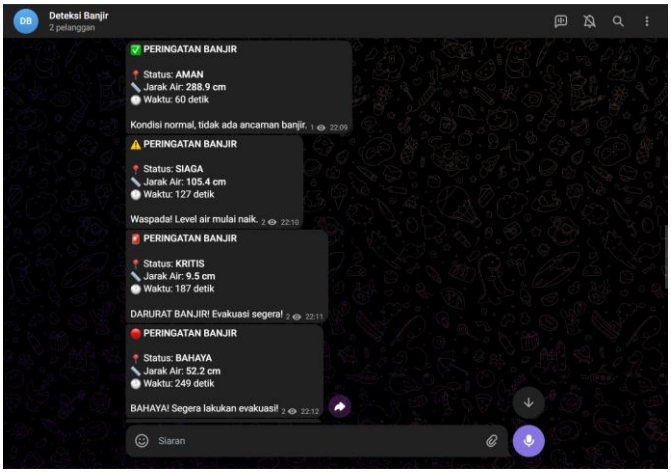


FIGURE 4.
Telegram Bot Notification Display for Each Warning Status Level

C. Discussion

The test results indicate that the developed flood early warning system prototype functions according to its design. The integration of the HC-SR04 sensor, ESP32, ThingSpeak, and Telegram Bot produces a comprehensive system, as it encompasses detection, data storage, visualization, and notification within an integrated workflow.

The use of the ESP32 proved effective as a processing unit due to its capability to handle Wi-Fi communication simultaneously with sensor reading processes and program logic execution. Compared to research [5], which used NodeMCU and only provided data access via a local web server, the proposed system offers a distinct advantage in terms of data accessibility by utilizing a public cloud platform.

Compared to research [7], which used an ESP32 with a monitoring platform lacking public cloud integration, the proposed system is superior regarding data accessibility and historical data storage through ThingSpeak. The use of a Telegram Bot as a notification medium aligns with the findings of [14], who implemented Telegram in their flood early warning system, and it has proven to deliver low notification latency and easy access for users at no additional cost.

Nevertheless, this system still has several limitations. The dependency on an internet connection makes the system vulnerable to network service disruptions. Furthermore, the use of a free ThingSpeak account limits the data update frequency to a minimum of once every 15 seconds, which could pose a constraint during rapid changes in water levels. For future development, it is recommended to add local data storage as a backup and to utilize a more reliable network connection, such as a cellular network with a GSM/4G module, as recommended in the comprehensive review of IoT-based early warning systems [15] [16].

IV. CONCLUSION

This research successfully designed and implemented an IoT-based flood early warning system prototype using the ESP32, HC-SR04 sensor, ThingSpeak, and Telegram Bot. Based on the test results, the system is capable of measuring water levels with an average error of 1.05% and sending warning notifications to users with an average latency of 2.7 seconds. The system also successfully differentiates between four danger status levels automatically and delivers appropriate notifications according to the detected conditions.

The integration of ThingSpeak as a cloud data storage and visualization platform with the Telegram Bot as a notification medium is proven to produce a comprehensive, easily accessible system that can be implemented at a relatively low cost. This prototype has the potential for further development by adding supporting sensors, a backup GSM-based communication module, and water level prediction algorithms using machine learning techniques to enhance the system's early warning capabilities.

ACKNOWLEDGMENT

The author would like to express gratitude to the lecturer of the Internet of Things course for providing guidance and direction throughout the research and writing process of this article. Gratitude is also extended to all group members who contributed to the design, implementation, and testing of the system.

REFERENCES

[1] *L. Wang et al., "A review of the flood management: from flood control to flood resilience," Nov. 01, 2022, Elsevier Ltd. doi: 10.1016/j.heliyon.2022.e11763.*

[2] *N. Rindsfüser, A. P. Zischg, and M. Keiler, "Monitoring flood risk evolution: A systematic review," Sep. 20, 2024, Elsevier Inc. doi: 10.1016/j.isci.2024.110653.*

[3] *M. Esposito, L. Palma, A. Belli, L. Sabbatini, and P. Pierleoni, "Recent Advances in Internet of Things Solutions for Early Warning Systems: A Review," Mar. 01, 2022, MDPI. doi: 10.3390/s22062124.*

[4] *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," Sustainability (Switzerland), vol. 14, no. 9, May 2022, doi: 10.3390/su14095512.*

[5] *A. Mulyani, D. Kurniadi, and R. E. Saputra, "IoT-Based Real-Time River Monitoring and Early Flood Warning Using ESP32 and HC-SR04," 2026, doi: 10.56741/jnest.v5i01.1252.*

[6] *V. N. Sulistyawan, N. A. Salim, F. G. Abas, and N. Aulia, "Parking Tracking System Using Ultrasonic Sensor HC-SR04 and NODEMCU ESP8266 Based IoT," in IOP Conference Series: Earth and Environmental Science, Institute of Physics, 2023. doi: 10.1088/1755-1315/1203/1/012028.*

[7] *Y. Basir, M. Rizki, A. Pratama, and M. W. Aminullah, "Perancangan Sistem Pendeteksi Dan Penanggulangan Banjir Menggunakan ESP32 Berbasis IoT," Jurnal Ilmiah GIGA, vol. 26, no. 1, doi: 10.47313/jig.v%vi%i.2127.*

[8] *P. T. Wikantama and R. Puspitasari, "Perancangan Perangkat Pengukur Ketinggian Banjir dengan ESP32 dan Telegram Berbasis IoT," Elektriese: Jurnal Sains dan Teknologi Elektro, vol. 13, no. 02, pp. 107–114, Nov. 2023, doi: 10.47709/elektriese.v13i02.3108.*

[9] *D. N. Bestari and A. Wibowo, "An IoT-Based Real-Time Weather Monitoring System Using Telegram Bot and Thingsboard Platform," International Journal of Interactive Mobile Technologies, vol. 17, no. 6, pp. 4–19, 2023, doi: 10.3991/ijim.v17i06.34129.*

[10] *N. A. Pramono, B. A. Purwandani, O. Ghaisyani, F. P. P. Mallisa, and F. I. Sofyan, "Development a prototype of river water level monitoring system using ESP32 based on internet of things for flood mitigation," in Journal of Physics: Conference Series, Institute of Physics, 2023. doi: 10.1088/1742-6596/2498/1/012039.*

[11] *S. Rosadi and P. Studi Sistem Komputer, "PROTOTYPE SISTEM PENDETEKSI BANJIR MENGGUNAKAN NODEMCU DAN PROTOKOL MQTT BERBASIS INTERNET OF THINGS," Jurnal Sistem Informasi dan Informatika (Simika) P-ISSN, vol. 5, pp. 2622–6901, 2022.*

[12] *W. Musa, I. Hidayat, S. Alam, B. Asmara, and A. Tolago, "PENGONTROLAN ENERGI PANAS DAN KELEMBABAN MENGGUNAKAN SENSOR DHT22 DAN ESP32 PADA PROSES FERMENTASI TEMPE GEMBUS," Journal Of Renewable Energy Engineering, vol. 2, pp. 50–55, Apr. 2024, doi: 10.56190/jree.v2i1.33.*

[13] *A. H. Miry and G. A. Aramice, "Water monitoring and analytic based ThingSpeak," International Journal of Electrical and Computer Engineering, vol. 10, no. 4, pp. 3588–3595, 2020, doi: 10.11591/ijece.v10i4.pp3588-3595.*

[14] *A. D. Rakhman, I. Komang Somawirata, M. Ardita, and I. K. Somawirata, "Flood Early Warning System with Notification Using Telegram Bot Based on IoT System," Journal homepage: Journal of Electrical Engineering and Computer (JEECOM), vol. 7, no. 1, 2025, doi: 10.33650/jeecom.*

[15] *M. Siddique, T. Ahmed, and M. S. Husain, "Flood Monitoring and Early Warning Systems – An IoT Based Perspective," Jul. 31, 2023, European Alliance for Innovation. doi: 10.4108/eetiot.v9i2.2968.*

[16] *Tolago, A. I., Nadifa, U., Asri, A. A. O., & Hidayat, I. (2026). Flood Susceptibility Mapping of Gorontalo Province Based on Climate and Spatial Data Using Machine Learning. Journal of Electrical Engineering and Computer (JEECOM), 8(3), 204-212.*