ILKOM Jurnal Ilmiah Vol. 16, No. 2, August 2024, pp. 151-160
Accredited 2nd by RISTEKBRIN No. 200/M/KPT/2020; E-ISSN 2548-7779 | P-ISSN 2087-1716

Research Article **Open Access (CC–BY-SA)**

Telegram bot-based Flood Early Warning System with WSN Integration

Abdul Wahid[a,1,*]; Jumadi M. Parenreng[a,2]; Welly Chandra Kusumah Kusnandar[a,3]; Puput Dani Prasetyo Adi[b,4]; Dendy Mahabror[b,5]; Ros Saniningrum[b,6]
[a] *Universitas Negeri Makassar, Jl. A.P. Pettarani, Makassar, Indonesia*
[b] *National Research and Innovation Agency (BRIN-RI), Jakarta, Indonesia*
[1] *wahid@unm.ac.id;* [2] *jparenreng@unm.ac.id;* [3] *wellychandrakk@gmail.com;* [4]*pupu008@brin.go.id;* [5]*mahabror_dee@yahoo.com,*
[6]*ross001@brin.go.id*
* *Corresponding author*

Article history: Received May 20, 2023; Revised May 25, 2023; Accepted July 02, 2024; Available online August 20, 2024.

Abstract

Indonesia experiences frequent flooding, with data from the National Disaster Management Agency (BNPB) revealing that floods account for 41% of all natural disasters (1,441 incidents) recorded in 2021. These floods cause significant property damage and casualties. To address this challenge, we have developed a prototype flood early warning system. This system utilizes ultrasonic sensors for real-time water level detection. Sensor data is transmitted to designated personnel through a website interface. Additionally, the system leverages a Telegram bot to deliver flood early warnings directly to the community residing in flood-prone areas. The sensor data comparison test yielded an error rate of only 0.6175% with an average difference of 1 cm, demonstrating the system's accuracy and functionality. Furthermore, a notification test conducted ten times achieved 100% accuracy. The Telegram bot successfully sent text message alerts (alert 1, alert 2, alert 3) with an average delivery time of 4.07 seconds. This prototype offers a promising solution for flood mitigation. By providing real-time water level data and issuing timely alerts via a user-friendly Telegram bot, the system empowers communities to prepare for potential flooding and minimize associated risks.

Keywords: Alert; Early Warning System; Flood; Water Level Detection; Wireless Sensor Network.

Introduction

Disaster is an incident or a series of incidents caused by natural and/or non-natural factors and humans that threaten and disrupt human life and livelihood, such as causing casualties, environmental damage, property damage, and psychological impacts. Disasters are natural phenomena that have a significant impact on humanity. These events can include floods, earthquakes, volcanic eruptions, landslides, tsunamis, and others [1]-[3]. A flood is a natural disaster that occurs when excess water inundates a land area. This happens due to high rainfall over a long period or in succession, causing rivers, dams, and irrigation canals to overflow, often due to littering by the community, which clogs the water channels [4]-[6].

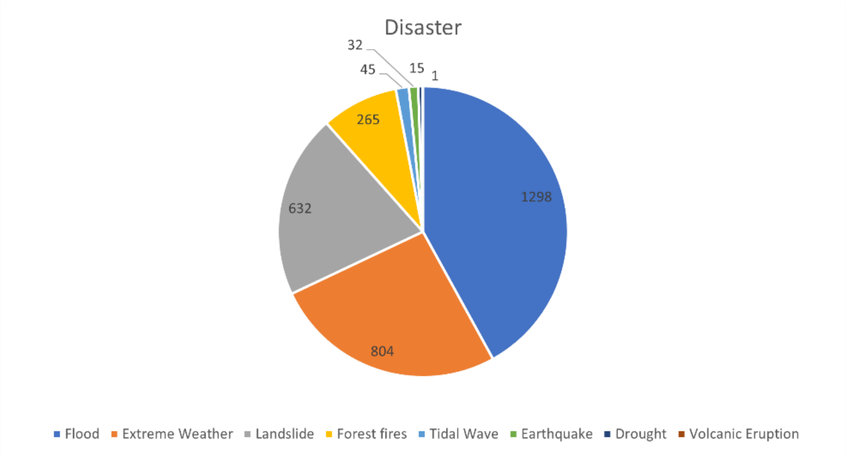

Figure 1. Indonesia Disaster Statistics

http://dx.doi.org/10.33096/ilkom.v16i2.1699.151-160

Indonesia is one of the countries that is frequently hit by floods. According to data from the National Disaster Management Agency (BNPB), from January 1 to June 18, 2021, there were a total of 1,441 natural disasters that hit Indonesia. Based on Figure 1., the most frequent disaster is flooding with a high intensity of 41%. This number is certainly not small because it causes many impacts in various aspects, including social, economic, and environmental aspects. This results in both material losses and casualties for the community. Various efforts have been made to minimize the losses caused by floods. One way to do this is to use technology, such as utilizing sensor-based flood early warning systems. With this tool, people in flood-prone areas can be warned early in the event of a flood [7].

Flooding poses a significant threat to communities worldwide, causing loss of life, property damage, and economic disruption. Effective flood early warning systems (EWS) are crucial for mitigating these impacts by providing timely alerts to affected areas [8]. However, existing EWS often face limitations, such as restricted notification range, reliance on outdated technologies, and a lack of real-time monitoring capabilities [9].

Previous research on flood EWS has explored various approaches, including buzzer-based systems, SMS gateways, and Internet of Things (IoT)-based monitoring. While these systems address some limitations, they also introduce new challenges. Buzzer-based systems by [10] have a limited notification range due to their reliance on sound alerts. Additionally, local monitoring restricts remote access. SMS gateway systems by [11] overcome the range limitation but suffer from the obsolescence of SMS technology. Moreover, local LCD-based monitoring limits remote access. IoT-based monitoring systems by [12] enable real-time water level monitoring via a web interface. However, they lack a notification system and have limited sensor coverage.

To address the limitations of existing EWS, we propose a Telegram bot-based system integrated with a Wireless Sensor Network (WSN). This system aims to provide real-time monitoring and alerts to communities while expanding sensor coverage to capture flood-prone areas effectively. Key Features of the Proposed System are Real-time monitoring and alerts: The Telegram bot provides real-time updates on water levels and issue flood alerts to subscribed users; Comprehensive sensor coverage: The WSN enables the deployment of sensors in multiple flood-prone locations, expanding the system's coverage area; Scalability and adaptability: The system can be easily scaled to accommodate additional sensors and users, making it adaptable to varying flood risk scenarios; User-friendly interface: The Telegram bot provides a user-friendly interface for accessing real-time data and receiving alerts.

The proposed Telegram bot-based EWS with WSN integration offers a promising solution to address the limitations of existing systems and enhance flood preparedness. By combining real-time monitoring, comprehensive sensor coverage, and a user-friendly interface, this system can effectively alert communities and mitigate the impacts of flooding.

Method

This research employs a Research & Development (R&D) methodology. R&D is a process or series of steps aimed at creating a new product or improving an existing one. The product may not necessarily be entirely new or non-existent but can involve refining existing research products to enhance their effectiveness in various aspects [13]-[15].

In this study, the R&D methodology is employed to develop and refine a flood early warning system. The research process involves:

- Defining objectives: Clearly outlining the specific goals and objectives of the flood early warning system.
- Conducting research: Gathering and analyzing data on flood patterns, sensor technologies, and communication protocols.
- Designing and developing the system: Creating the hardware and software components of the flood early warning system, integrating sensor networks, communication infrastructure, and user interfaces.
- Testing and evaluation: Rigorously testing the system in simulated and real-world environments to assess its performance and identify areas for improvement.
- Refining the system: Based on testing and evaluation results, refining the system's components and functionalities to enhance its effectiveness and reliability.

The R&D methodology provides a structured and systematic approach to developing and refining the flood early warning system. By following this methodology, the study aims to create a robust and effective system that can provide timely and accurate flood warnings to communities, minimizing the impacts of flooding.

A. System Architecture

The proposed flood early warning system utilizes a combination of technologies to provide real-time monitoring and alerts to communities and government agencies. The system architecture is depicted in **Figure 2** and can be summarized as follows:

- WSN: A network of sensors, including NodeMCU ESP8266 and ultrasonic sensors, collects water level data every two seconds.
- Sink Node: The Sink Node receives data from the WSN nodes and transmits it to a database for processing.
- Data Processing: The data is analyzed to determine the flood warning level (*Siaga 1* to *Siaga 3*).
- Telegram Bot Notifications: Flood alerts are automatically sent to registered users' smartphones via a Telegram bot.
- Real-time Monitoring Website: A real-time monitoring website allows government agencies and administrators to view water level data remotely.

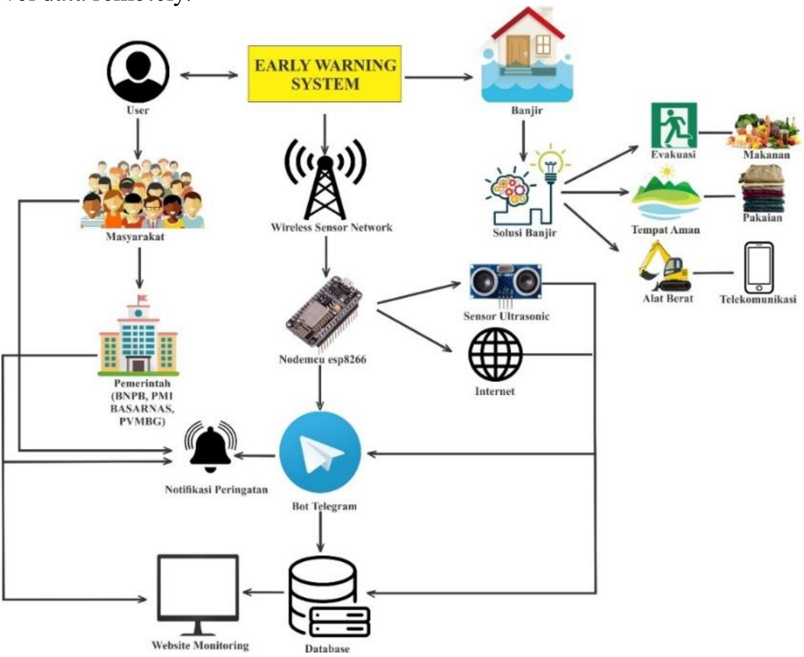

Figure 2. System Architecture

In the event of a flood, government agencies promptly deploy to the affected area to evacuate residents to safe locations. Heavy machinery is crucial for efficient and safe evacuation. Additionally, telecommunication infrastructure, food, and clothing are essential for sustaining affected communities and enabling communication with family members. By implementing a comprehensive flood early warning system and coordinating effective disaster response measures, communities and government agencies can significantly reduce the impact of floods and safeguard lives and livelihoods.

B. WSN Node System Design

The WSN Node hardware design, as depicted in **Figure 3**, consists of two primary components: the NodeMCU ESP8266 microcontroller board and the ultrasonic sensor [16]. The ultrasonic sensor has four pins that need to be connected to the NodeMCU ESP8266 board as follows [17]:

- VCC Pin: Connect the VCC pin of the ultrasonic sensor to the 3.3V pin of the NodeMCU ESP8266.
- GND Pin: Connect the GND pin of the ultrasonic sensor to the GND pin of the NodeMCU ESP8266.
- Trigger Pin: Connect the Trigger pin of the ultrasonic sensor to the D5 pin of the NodeMCU ESP8266.
- Echo Pin: Connect the Echo pin of the ultrasonic sensor to the D6 pin of the NodeMCU ESP8266.

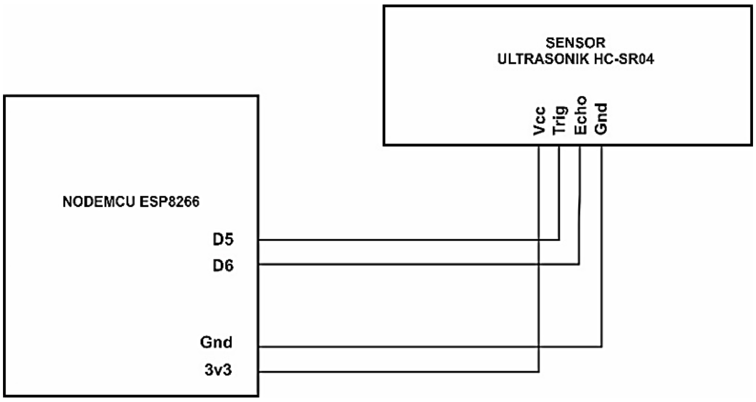

Figure 3. WSN Node System Design

These connections enable the NodeMCU ESP8266 to communicate with the ultrasonic sensor and gather water level data, which is crucial for the flood early warning system [18].

C. Flowchart Design of Flood Early Warning System

Figure 4 illustrates the operational workflow of the flood early warning system. The WSN Nodes continuously collect water level data every two seconds. This data is aggregated at the Sink Node, which is connected to a database for processing. The processed data is then used to generate a real-time monitoring website and trigger flood warning notifications via a Telegram bot.

Data Processing and Notification Mechanism:

1. Data Collection and Aggregation: WSN Nodes collect water level data every two seconds, and the Sink Node aggregates this data.
2. Data Processing: The collected data is processed to determine the corresponding flood warning level.
3. Website Monitoring: The processed data is used to update the real-time monitoring website, providing users with up-to-date water level information.
4. Telegram Bot Notifications: If the water level exceeds the predefined thresholds, the system automatically sends corresponding flood warning messages to registered users via the Telegram bot. Flood Warning Levels and Notifications,
   - *Siaga 1* (Alert Level 1): Water level between 100 cm and 199 cm triggers an Alert Level 1 notification.
   - *Siaga 2* (Alert Level 2): Water level between 200 cm and 299 cm triggers an Alert Level 2 notification.
   - *Siaga 3* (Alert Level 3): Water level above 300 cm triggers an Alert Level 3 notification, indicating a severe flood situation.
5. Continuous Data Updates: After sending a notification, the system updates the relevant data in the database and on the website to ensure real-time information dissemination.

This comprehensive data processing and notification mechanism ensures that communities are promptly alerted to potential flooding hazards, enabling them to take timely protective measures and minimize the impact of floods.

D. Decision Algorithm Flowchart

Figure 5 explains the steps of the decision algorithm. WSN node collects data and the WSN Node is given 3 conditions, namely *siaga 1*, *siaga 2* and *siaga 3*. Each of these conditions is given a decision algorithm that counters or calculate 5 times in each condition so that the notification sent is truly valid according to the conditions that occur in the field if the counter or calculation is not fulfilled, it counters or calculate from the beginning again.

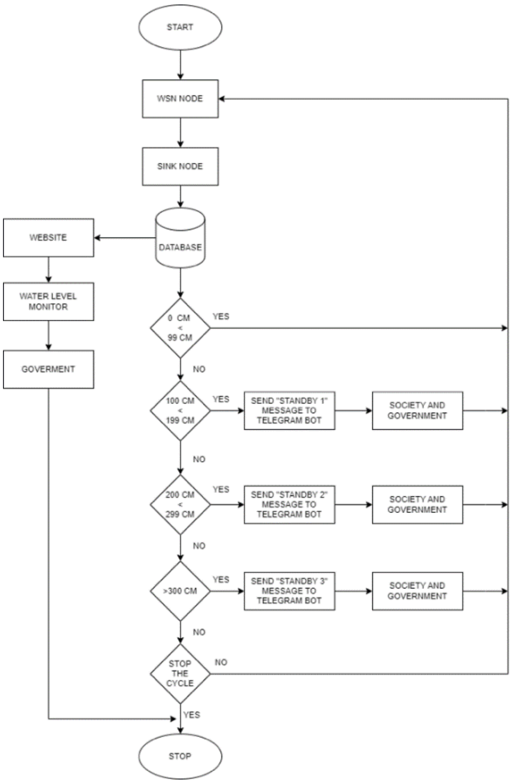

Figure 4. Flowchart Design of Flood Early Warning System

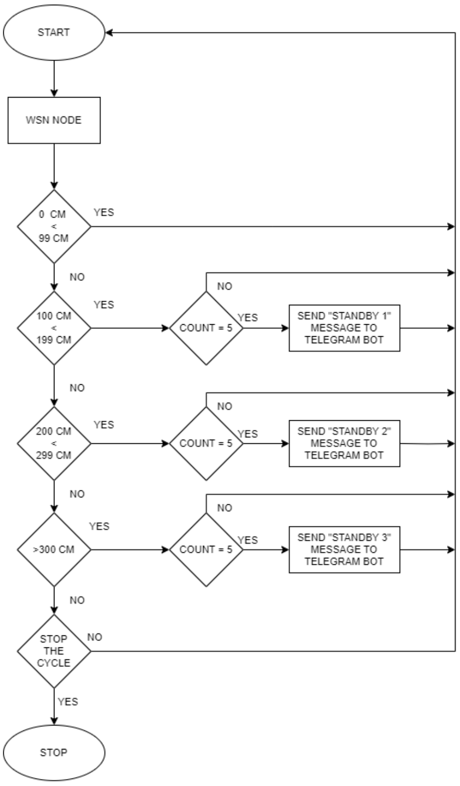

Figure 5. Decision Algorithm Flowchart

E. Mathematic Formula for Ultrasonic Signal

Ultrasonic signals experience attenuation or signal interference due to interference or objects in front of them, the type used in this research is HC-SR04, this ultrasonic sensor uses ultrasonic waves that have a very high frequency of 20 kHz or 20,000 Hz [19]. From the working principle of ultrasonic sensors, it can be used to measure the distance of objects and determine actuators, for example to monitor water levels in reservoirs or rivers as used in this research. Next is how if the ultrasonic sensor or sound waves read the types of objects, if you can see the resulting signal is different depending on the object of reflection. This equation is called Frequency response $(M(\omega))$ as the Equation 1.

$$M(\omega) = e^{-\alpha\,(\omega)x} e^{-j\frac{x(\omega)}{V_p(\omega)}} \tag{1}$$

While when hitting the object signal $y(t)$ has an equation as in Equation 2. Where this equation is used to provide a value signal that hits a multi-layer object. It is essential to analyze the type or shape of the signal generated from the ultrasonic sensor which is used to emit a signal to the object which can then provide signal feedback based on the object [20].

$$y(t) = \sum_{n=0}^{N} A_n h(t - \tau_n) \tag{2}$$

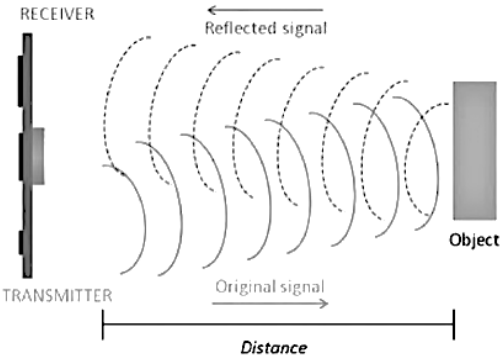

Figure 6. Working Principle of Ultrasonic Sensor

Results and Discussion

Figure 7 illustrates the assembly process of the Flood Early Warning System with Telegram Bot Based on WSN. The system utilizes two primary components: the NodeMCU ESP8266 microcontroller board and the HC-SR04 ultrasonic sensor. The HC-SR04 ultrasonic sensor is a device that measures the distance of an object in front of it. It employs two digital pins for communication: the Trigger pin for emitting a signal and the Echo pin for receiving the reflected signal from the object.

WSN Node Assembly Steps

1) Connect the Sensor to the NodeMCU [16]:
    a. Connect the VCC pin of the HC-SR04 sensor to the 3.3V pin of the NodeMCU ESP8266.
    b. Connect the GND pin of the HC-SR04 sensor to the GND pin of the NodeMCU ESP8266.
    c. Connect the Trigger pin of the HC-SR04 sensor to the D5 pin of the NodeMCU ESP8266.
    d. Connect the Echo pin of the HC-SR04 sensor to the D6 pin of the NodeMCU ESP8266.

2) Power Supply Connection:
    a. Connect the 5V pin of the NodeMCU ESP8266 to the 5V pin of the USB power supply.
    b. Connect the GND pin of the NodeMCU ESP8266 to the GND pin of the USB power supply.

3) WSN Node Deployment:
    a. Securely mount the assembled WSN Node at the desired location.
    b. Ensure the ultrasonic sensor is facing the direction where water level measurements are required.

By following these assembly steps, the WSN Nodes can be effectively integrated into the flood early warning system, enabling them to collect water level data and contribute to the overall flood monitoring and notification mechanism.

Figure 7. The IoT Node Use HC-SR04

A. Sensor Testing

The purpose of this testing is to determine the accuracy of the ultrasonic sensors used in the prototype device by comparing the water level readings displayed on the monitoring website with measurements taken using a measuring tape [21]. This comparison assesses the discrepancy between the sensor readings and the actual water level. Testing methodology as the following:

- Sensor Readings: Record the water level readings displayed on the monitoring website for various water levels.
- Meter Tape Measurements: Simultaneously measure the actual water level using a measuring tape at the same locations where the sensor readings were obtained.
- Comparison and Analysis: Compare the sensor readings from the monitoring website with the corresponding meter tape measurements. Calculate the difference between the two sets of measurements for each water level.
- Accuracy Evaluation: Evaluate the accuracy of the ultrasonic sensors by analyzing the calculated differences. Determine the extent to which the sensor readings deviate from the actual water level measurements.
- Error Analysis: Identify any systematic or random errors that may be contributing to the discrepancies between the sensor readings and the actual water level measurements.

By conducting this testing and analysis, the accuracy of the ultrasonic sensors can be thoroughly assessed, providing valuable insights into the performance of the prototype device and identifying potential areas for improvement.

Table 1. Comparison of Measurement

No	Use a Ruler (Cm)	Ultrasonic sensor (Cm)	W.E.B. Monitoring	Different (Cm)	Relative Error (100 %)
1.	5 cm	5cm	5cm	0 cm	0 %
2.	15 cm	15 cm	15 cm	0 cm	0 %
3.	30 cm	30 cm	30 cm	0 cm	0 %
4.	45 cm	45 cm	45 cm	0 cm	0 %
5.	60 cm	59 cm	59 cm	1 cm	1,694 %
6.	75 cm	74 cm	74 cm	1 cm	1,351 %
7.	90 cm	89 cm	89 cm	1 cm	1,123 %
8.	105cm	104cm	104cm	1 cm	0,961 %
9.	120 cm	119 cm	119 cm	1 cm	0,840 %
10.	135 cm	134 cm	134 cm	1 cm	0,746 %
Average				1 cm	0,6715 %

Description:

- Using a ruler (cm): Actual distance
- Ultrasonic sensor (cm): Measurement Result
- Difference (cm): The result of subtracting the actual distance - measurement result
- Relative error (100%): Obtained by using the Equation 3.

Based on **Table 1**. the results of several trials of comparison between the distance displayed by web monitoring and the actual distance, the error is calculated using the Equation 3.

$$\text{Relative error \%} = \left(\frac{JS-JP}{JP}\right).100\% \qquad (3)$$

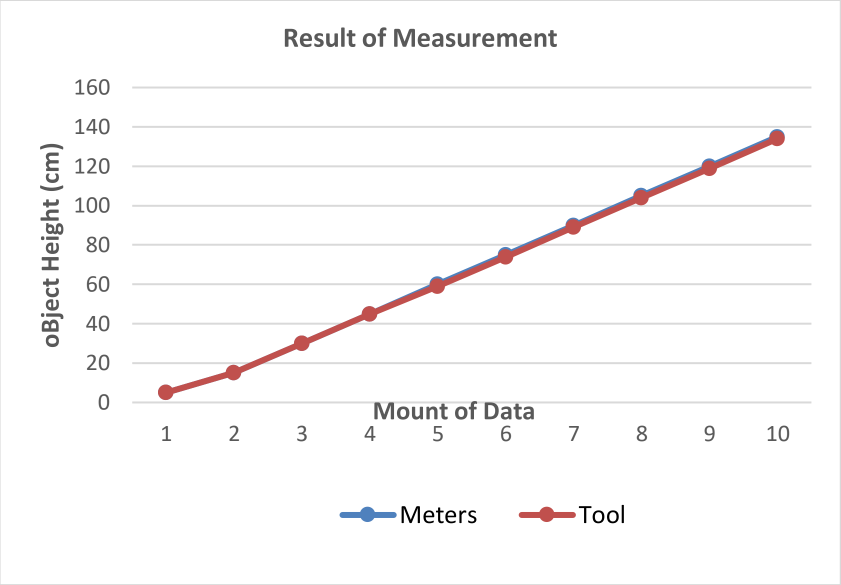

Figure 8. Sensor Testing Chart

Figure 8, the Sensor Data Comparison Chart, illustrates the differences between the water level measurements obtained using a measuring tape and the readings displayed on the monitoring website. The Comparative Analysis result:

- Measurements 1 to 4: For the first four measurements, there is no discrepancy between the two measurement methods. Both the measuring tape and the monitoring website consistently display the same water level values.
- Measurements 5 to 10: For measurements 5 to 10, a slight discrepancy of 1 cm is observed between the two methods. The measuring tape consistently indicates water levels 1 cm higher than the values displayed on the monitoring website.

Despite the minor discrepancies in measurements 5 to 10, the overall accuracy of the sensor data comparison is exceptionally high. The relative error value is calculated to be 0.6175%, which falls within the category of "very acceptable." This indicates that the ultrasonic sensors used in the prototype device provide highly reliable and accurate water level measurements.

The sensor data comparison chart demonstrates the effectiveness of the ultrasonic sensors in measuring water levels. The overall accuracy of the system is commendable, with a relative error of less than 1%. This confirms the suitability of the prototype device for flood monitoring applications.

B. Notification Testing Via Telegram

The objective of this testing phase is to verify the successful delivery of notification messages to the Telegram app during flood warnings [22], [23]. The steps of this testing :

- Create Simulation Scenarios: Create three simulated flood warning scenarios representing *Siaga 1* (Alert Level 1), *Siaga 2* (Alert Level 2), and *Siaga 3* (Alert Level 3).
- Trigger Notifications: Trigger the corresponding notification messages for each simulated flood warning scenario.
- Monitor Telegram App: Observe the Telegram app to confirm the receipt of the notification messages.
- Record Results: Record the results of each test case, indicating whether the notification message was successfully delivered and displayed in the Telegram app.

Evaluation Criteria

- Successful Delivery: A successful delivery is determined if the corresponding notification message is received and displayed in the Telegram app within a reasonable timeframe.
- Accuracy of Notification Content: The notification message content should accurately reflect the simulated flood warning scenario, including the correct alert level (*Siaga 1*, *Siaga 2*, or *Siaga 3*).

Based on the testing results, the system successfully delivers notification messages to the Telegram app during simulated flood warnings. The notification content accurately reflects the corresponding alert level, ensuring that users are promptly informed about the severity of the flood situation. This confirms the effectiveness of the notification mechanism in disseminating critical information during flood events.

Table 2. Telegram Notification Testing

Testing	**Notification Sending**														
	Alert 1 (50 – 100 cm)					***Alert 2 (101 – 150 cm)***					***Alert 3 (151 – 200 cm)***				
	1	***2***	***3***	***4***	***5***	***1***	***2***	***3***	***4***	***5***	***1***	***2***	***3***	***4***	***5***
1	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
2	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
3	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
4	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
5	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
6	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
7	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
8	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
9	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
10	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

The results of several trials of sending notifications to the telegram application, the accuracy is calculated using the Equation 4. Result of analyze from 150 of total data and 0 of error data obtain the result of accuracy equal 100%.

$$Accuracy = \frac{Total\ Data - Error}{Total\ Data} \times 100\% \tag{4}$$

C. Testing the Time to Send Chat to Telegram

Based on **Table 3**. test results from several trials testing the time to send chat to telegram, the average time to send chat is calculated using the Equation 5.

$$Average = \frac{Number\ of\ Values}{Lots\ of\ Data} \tag{5}$$

From the Equation 3, number of values is 122,19 and number of data is 30, So as to produce an average result is 4.07 seconds. At this stage, testing is carried out on how long it takes to send messages from sensor readings to the telegram chat application. This test aims to find out how long it takes for the system to deliver messages to the public through the telegram application when a flood warning occurs. In this test, 10 experiments were carried out and 3 sizes of alert status conditions were made, namely, 10 cm - 19 cm alert 1, 20 cm - 29 cm alert 2 and 30 cm and above alert 3. Time is measured using a cellphone stopwatch when the text message notification has entered the telegram application. In this test, the average time to send a message is 4.07 seconds. The following is a test of the time to send chat to telegram, the data obtained is as shown in **Table 3**.

Table 3. Testing the Time to Send Chat to Telegram

No	**Alert 1 (10-19 cm)**	**Alert 2 (20–29 cm)**	**Alert 3 (>30 cm)**
1	4,05 second	4,09 second	3,43 second
2	2,95 second	4,13 second	5,73 second
3	2,79 second	3,30 second	5,96 second
4	2,70 second	5,63 second	6,09 second
5	4,30 second	3,24 second	3,88 second
6	4,96 second	4,81 second	5,52 second
7	4,43 second	3,24 second	5,18 second
8	3,68 second	4,30 second	3,61 second
9	2,67 second	3,66 second	2,11 second
10	5,12 second	3,14 second	3,49 second
Average			4,07 second

Conclusion

Based on the research findings, the following conclusions can be drawn regarding the Flood Early Warning System with Telegram Bot Based on WSN:

1. Effective Data Acquisition and Visualization: The designed system effectively acquires and displays water level data from multiple WSN Nodes. The monitoring website successfully presents the collected data in the form of five graphs, updating every five seconds. This real-time data visualization enables users to monitor water levels and identify potential flood risks promptly.
2. Reliable Notification Delivery: The system consistently delivers flood warning notifications to Telegram users. During testing, all ten notification attempts, encompassing *Siaga 1* (Alert Level 1), *Siaga 2* (Alert Level 2), and *Siaga 3* (Alert Level 3) messages, were successfully sent to the Telegram app. The average notification delivery time was 4.07 seconds, demonstrating timely dissemination of critical information during flood events.
3. Leveraging IoT and WSN Technologies: The system effectively utilizes IoT and WSN technologies to collect water level data from multiple locations. This approach enables comprehensive monitoring of flood conditions across a wider area, enhancing the overall effectiveness of the flood early warning system.

In summary, the Flood Early Warning System with Telegram Bot Based on WSN has proven to be a reliable and effective tool for flood monitoring and notification. Its ability to accurately collect and display water level data, coupled with its timely and reliable notification delivery mechanism, makes it a valuable asset for flood risk mitigation efforts.

Acknowledgement

Thanks to Universitas Negeri Makassar (UNM) and all colleagues at Universitas Negeri Makassar, hopefully this article can be a reference for researchers, students, lecturers, and the public, and can continue to be developed along with technological developments, especially the Internet of Things and Artificial Intelligence.

References

[1] B. Merz *et al.*, “Impact Forecasting to Support Emergency Management of Natural Hazards,” *Reviews of Geophysics*, vol. 58, no. 4, p. e2020RG000704, 2020, doi: 10.1029/2020RG000704.

[2] M. Krichen, M. S. Abdalzaher, M. Elwekeil, and M. M. Fouda, “Managing natural disasters: An analysis of technological advancements, opportunities, and challenges,” *Internet of Things and Cyber-Physical Systems*, vol. 4, pp. 99–109, 2024, doi: 10.1016/j.iotcps.2023.09.002.

[3] L. Velotti, “Natural Hazards: Tornadoes,” in *Encyclopedia of Security and Emergency Management*, L. R. Shapiro and M.-H. Maras, Eds., Cham: Springer International Publishing, 2019, pp. 1–4. doi: 10.1007/978-3-319-69891-5_209-1.

[4] F. Ahmad, “A study of Flood on Learning Attitude at Secondary level,” 2018.

[5] T. Osayomi *et al.*, “‘I lost all I had to the flood…’: A Post-Disaster Assessment of the 2018 Kogi State Flood in Nigeria,” 2022.

[6] K. Y. Lim, N. A. Zakaria, and K. Y. Foo, “Water quality assessment of urban catchment after the large-scale flood event: The worst natural tragedy at Pahang River, Malaysia,” *DWT*, vol. 175, pp. 32–42, 2020, doi: 10.5004/dwt.2020.24790.

[7] Q. Sholihah, W. Kuncoro, S. Wahyuni, S. Puni Suwandi, and E. Dwi Feditasari, “The analysis of the causes of flood disasters and their impacts in the perspective of environmental law,” *IOP Conf. Ser.: Earth Environ. Sci.*, vol. 437, no. 1, p. 012056, Feb. 2020, doi: 10.1088/1755-1315/437/1/012056.

[8] M. Yahya, J. Mabe Parenreng, Suhartono, M. Syahid Nur Wahid, A. Saputra, and A. Wahid, “Early Warning Urgent Logistic Disaster in Rural Area based on Classification Algorithm: Indonesian Case,” in *2021 IEEE 5th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, Purwokerto, Indonesia: IEEE, Nov. 2021, pp. 277–282. doi: 10.1109/ICITISEE53823.2021.9655850.

[9] F. N. I. Sari, D. N. Sugianto, and K. Kunarso, “Early Warning System Based on Historical Coastal Flood Events in Semarang City, Indonesia,” *GT*, vol. 17, no. 2/2022, pp. 14–25, Jul. 2022, doi: 10.21163/GT_2022.172.02.

[10] C. Umari, E. Anggraini, and R. Z. Muttaqin, “Rancang Bangun Sistem Peringatan Dini Banjir Berbasis Sensor Ultrasonik dan Mikrokontroler Sebagai Upaya Penanggulangan Banjir,” vol. 4, no. 2, 2017.

[11] A. Hernowo, Setiyono, and S. S. Prayogo, “Prototipe Sistem Peringatan Dini Banjir Berbasis PLC Omron dan Arduino Uno Sebagai SMS Gateway,” *tekno*, vol. 23, no. 2, pp. 122–132, 2018, doi: 10.35760/tr.2018.v23i2.2462.

[12] D. Satria, S. Yana, R. Munadi, and S. Syahreza, "Sistem Peringatan Dini Banjir Secara Real-Time Berbasis Web Menggunakan Arduino dan Ethernet," *jtik*, vol. 1, no. 1, p. 1, Oct. 2017, doi: 10.35870/jtik.v1i1.27.

[13] K. B. Akhilesh, *R&D Management*. New Delhi: Springer India, 2014. doi: 10.1007/978-81-322-1946-0.

[14] S. Magistretti, C. Dell'Era, R. Verganti, and M. Bianchi, "The contribution of Design Thinking to the R of R&D in technological innovation," *R&D Management*, vol. 52, no. 1, pp. 108–125, 2022, doi: 10.1111/radm.12478.

[15] A. Wahid, S. Luhriani, and R. Jefri, "Implementasi Papan Pengumuman Digital Terintegrasi Berbasis Multimedia Website di Jurusan Bahasa Inggris Fakultas Bahasa dan Sastra UNM," in *Prosiding Seminar Ilmiah Sistem Informasi dan Teknologi Informasi*, Makassar, Indonesia: Stmik Dipanegara, Makassar, 2015, pp. 199–207.

[16] Y. S. Parihar, "Internet of Things and Nodemcu A review of use of Nodemcu ESP8266 in IoT products," vol. 6, p. 1085, Jun. 2019.

[17] P. W. Rusimamto, E. Endryansyah, L. Anifah, R. Harimurti, and Y. Anistyasari, "Implementation of arduino pro mini and ESP32 cam for temperature monitoring on automatic thermogun IoT-based," *IJEECS*, vol. 23, no. 3, p. 1366, Sep. 2021, doi: 10.11591/ijeecs.v23.i3.pp1366-1375.

[18] J. Ko, C. Lu, M. B. Srivastava, J. A. Stankovic, A. Terzis, and M. Welsh, "Wireless Sensor Networks for Healthcare," *Proc. IEEE*, vol. 98, no. 11, pp. 1947–1960, Nov. 2010, doi: 10.1109/JPROC.2010.2065210.

[19] W. Xu, C. Yan, W. Jia, X. Ji, and J. Liu, "Analyzing and Enhancing the Security of Ultrasonic Sensors for Autonomous Vehicles," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5015–5029, Dec. 2018, doi: 10.1109/JIOT.2018.2867917.

[20] A. Djalilov, E. Sobirov, O. Nazarov, S. Urolov, and I. Gayipov, "Study on automatic water level detection process using ultrasonic sensor," *IOP Conf. Ser.: Earth Environ. Sci.*, vol. 1142, no. 1, p. 012020, Mar. 2023, doi: 10.1088/1755-1315/1142/1/012020.

[21] K. Janani, A. Palanivelu, and R. Sandhya, "Diagnostic accuracy of dental pulse oximeter with customized sensor holder, thermal test and electric pulp test for the evaluation of pulp vitality: an in vivo study," *BDS*, vol. 23, no. 1, Jan. 2020, doi: 10.14295/bds.2020.v23i1.1805.

[22] E. Ramadhani, A. Syarifurrahman, and G. P. Mahardhika, "Design System: Networks Status Notification using Telegram Messenger," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 852, no. 1, p. 012131, Jul. 2020, doi: 10.1088/1757-899X/852/1/012131.

[23] H. Husain, I. P. Hariyadi, K. A. Latif, and G. T. Aditya, "Implementation of Port Knocking with Telegram Notifications to Protect Against Scanner Vulnerabilities," *matrik*, vol. 23, no. 1, pp. 215–228, Nov. 2023, doi: 10.30812/matrik.v23i1.3459.