Journal of Electrical Engineering and Computer (JEECOM)
Vol. 7, No. 1 (2025), DOI: 10.33650/jeecom.
p-ISSN: 2715-0410; e-ISSN: 2715-6427

Flood Early Warning System with Notification Using Telegram Bot Based on IoT System

Aditya Djulqodri Rakhman [1], I Komang Somawirata [1], Michael Ardita [1]
[1] Electrical Engineering, National Institute of Technology Malang, Malang, Indonesia

Article Info

Article history:

Received February 28, 2025
Revised March 5, 2025
Accepted April 22, 2025

Keywords:

Early Warning System
Variable Message Sign
Internet of Things
Telegram Bot
NodeMCU ESP8266

ABSTRACT

Flooding is one of the disasters that often occurs in Indonesia, especially in areas with high rainfall, such as Malang City. To improve the effectiveness of early warning, this research develops an Early Warning System (EWS) information dissemination system based on the Internet of Things (IoT). This system combines Variable Message Sign (VMS) as a visual display media and a Telegram bot to provide real-time notifications to the public. NodeMCU ESP8266 is the main microcontroller connected to a WiFi network to access data on the server. Tests were conducted to assess the accuracy of VMS synchronization, the speed of notification response, and the effectiveness of the buzzer in providing warnings. The test results show that synchronizing two VMS with different internet networks has an average time difference of 1.16 seconds, while with the same network, it is only 0.03 seconds. In addition, the VMS and Telegram bot can deliver information quickly and accurately, while the buzzer functions according to the set warning level. With this system, the dissemination of flood early warning information becomes more effective and easily accessible to the public, especially those in vulnerable areas or on the move.

Corresponding Author:
I Komang Somawirata,
National Institute of Technology Malang City and 65141, Indonesia
Email: kmgsomawirata@lecturer.itn.ac.id

1. INTRODUCTION

Flooding is one of the disasters that often occurs in various parts of Indonesia[1]. One factor that contributes significantly to flood events is climate change, which causes increased rainfall and more frequent sea level rise[2],[3]. Malang City is one of the areas in Indonesia that has high rainfall. Over the past five years, Malang's flooding intensity has increased significantly, with a record of more than 700 flood events in 2019[4]. The Malang City Government has sought flood mitigation by installing 11 Flood Early Warning System (EWS) devices in flood-prone locations. These devices can detect floods early to reduce their impact on infrastructure, mobility, and the community's economy[5]. A flood early warning system is a series of mechanisms to detect signs of potential flooding[6]. This system integrates data communication with sensors that detect events and make decisions automatically. All stages work synergistically to identify threats that can cause negative impacts[7].

Currently, science and technology are experiencing a very rapid increase, so an innovative flood early warning system based on the Internet of Things (IoT) has emerged as a creative solution[8], [9]. In early flood detection, IoT plays a role in developing a system capable of measuring and sending user information and notifications [10]. This system utilizes sensors to monitor river water levels in real time and send data to web platforms or social media to provide early warnings. Although efficient, people must actively access the website or social media bot to get information. This limitation is a challenge, especially for traveling people who cannot monitor the system directly. Previous research has shown the effectiveness of IoT in flood monitoring, such as research conducted by Muhammad Jafar Siddik using NodeMCU ESP8266 and ultrasonic sensors to send water level data to the ThinkSpeak database with a transmission delay of about 5 seconds[11]. Meanwhile, another study conducted by Panji Tegar Wikantama developed an ESP32-based device that provides Telegram notifications within 2-3 seconds with "Alert" and "Danger" warning categories[12]. Although fast, these two studies have weaknesses in the efficiency of information dissemination, especially for mobilizing people.

This research aims to design an IoT-based flood Early Warning System (EWS) information dissemination device to provide early warnings effectively and in real-time. This system integrates a variable message sign (VMS) as a visual information display medium in specific locations and notifications via the Telegram bot to provide direct warnings to the public. This device uses NodeMCU ESP8266 as the main microcontroller, which can connect to a WiFi network[13], enabling fast and accurate data communication. Telegram Bot is an open-source technology provided by Telegram Messenger LLP, allowing developers to build Telegram bot applications [14]. Meanwhile, a Variable Message Sign (VMS) is an electronic board that can display messages that can be changed as needed, such as text, warnings, or road instructions in real time[15]. VMS will be placed in strategic locations such as crowd centers and flood-prone areas so that the public can access information without relying on smartphones or the internet. The system still relies on Telegram to provide real-time notifications, but it uses a more comprehensive approach to cover all levels of society. This device is expected to provide early warning information for floods efficiently, easily accessible, and valuable for the community, especially those mobilizing with the collaboration between VMS and Telegram notifications.

2. METHOD

2.1. Block Diagram

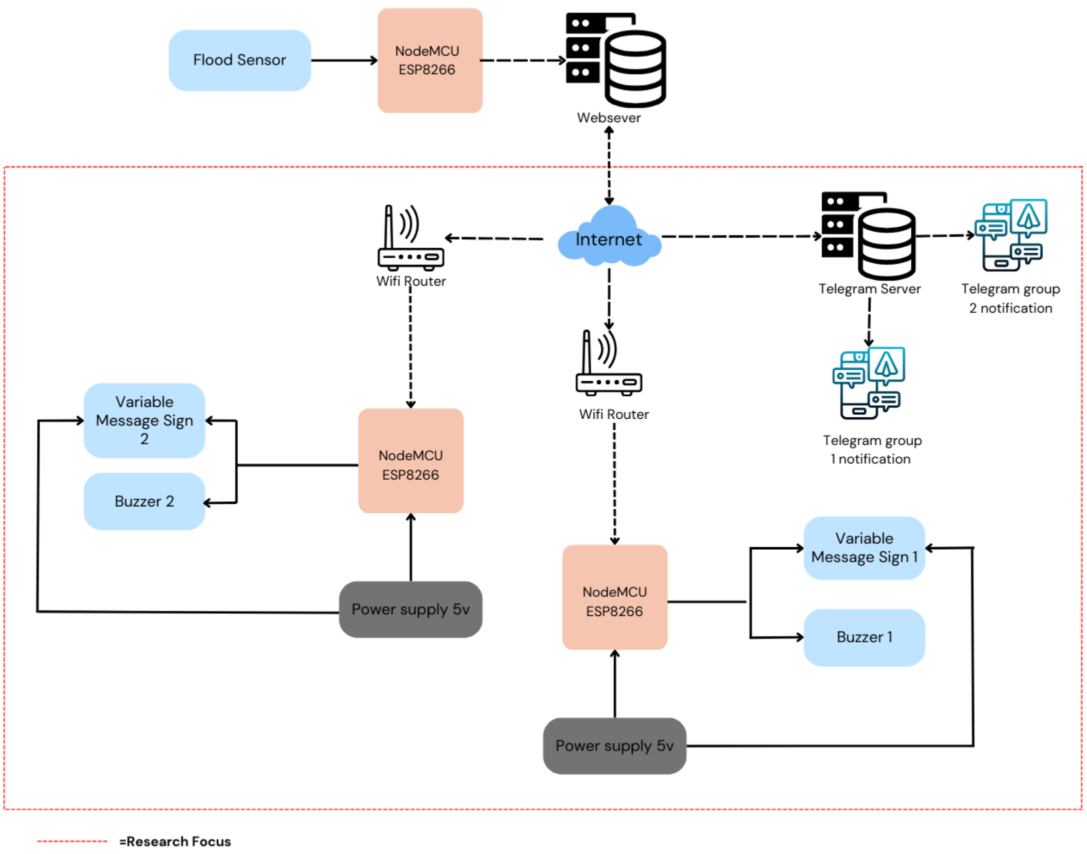

Figure 1. System block diagram

This research focuses on designing a device to disseminate information from the existing flood Early Warning System device to the public. The Block Diagram in Figure 3.1 illustrates the NodeMCU ESP8266-based flood monitoring system with internet support to provide early warning through local notifications and alarms. This system starts with a flood sensor that detects water levels or flood conditions. Data from the sensor is then sent to the first NodeMCU ESP8266. This NodeMCU connects the sensor with a web server, where the data is stored and can be monitored online.

Through an internet connection and Wi-Fi router, the web server connects to the Telegram server to send automatic notifications to the user's Telegram app when an emergency condition is detected. This allows users to receive early warnings on smartphones in real-time. On the other hand, there is a second NodeMCU ESP8266 that controls alarm devices such as buzzers and Variable Message Signs (VMS). The buzzer acts as an audible alarm to warn the neighborhood, while the VMS displays messages or information related to flood conditions.

This system uses a 5V power supply to power the NodeMCU and VMS. The system workflow starts when the sensor detects the water level and sends data to the first NodeMCU, which then forwards it to the web server and triggers the sending of Telegram notifications. The second NodeMCU will activate the buzzer and VMS to provide local alerts.

2.2. Application Flowchart on Server

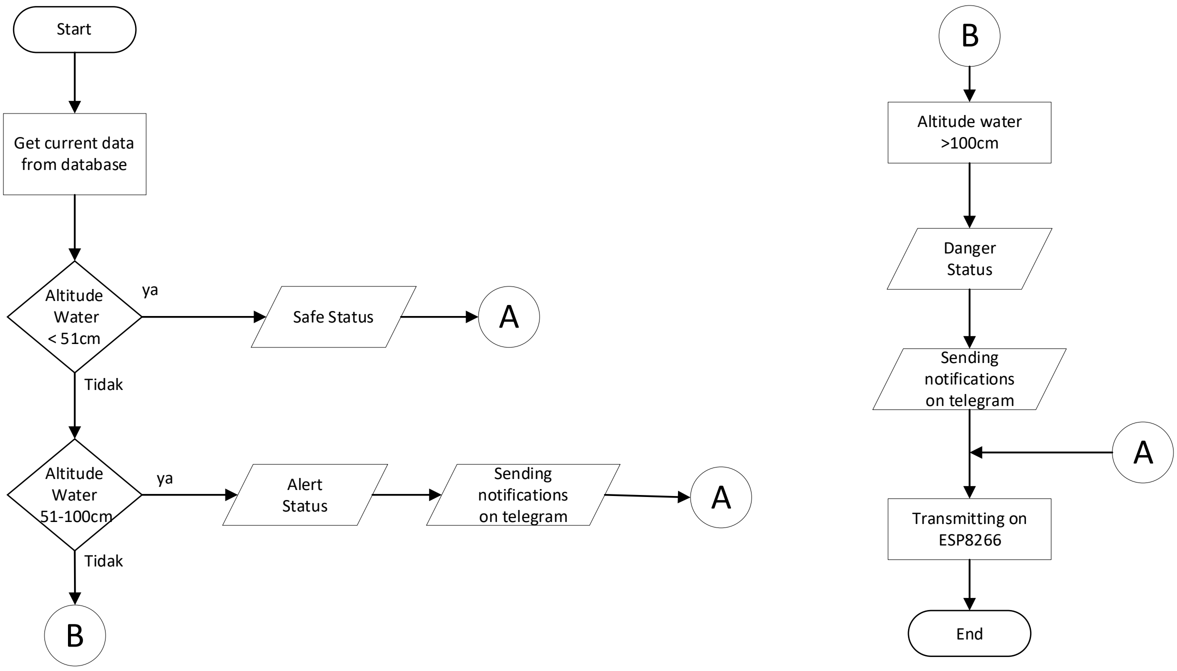

Figure 2. Flowchart of the application on the server

Figure3. Shows the flow of the flood warning level. The process starts when the system is active, and the flood sensor sends data related to the water level. If the water level is below 51 cm, the status is declared Safe, and no notification is sent, but the data is directly forwarded to the ESP8266 for further processing. If the water level is within the range of 51-100 cm, the system changes the status to Alert and sends a warning notification to Telegram as a sign that water conditions must be considered. Furthermore, suppose the water level exceeds 100 cm. In that case, the status changes to Danger, and the system immediately sends an emergency notification via Telegram to alert the user that a critical condition has been reached. Once the status is set and the notification is sent, the data and status are forwarded to the ESP8266 to control related devices, such as the buzzer or Variable Message Sign (VMS). The process ends once all steps have been executed, and the system is again ready to read new data from the sensors. With this flow, the system quickly responds to potential flooding through online alerts or local alarm activation.

2.3. VMS Client Side Flowchart

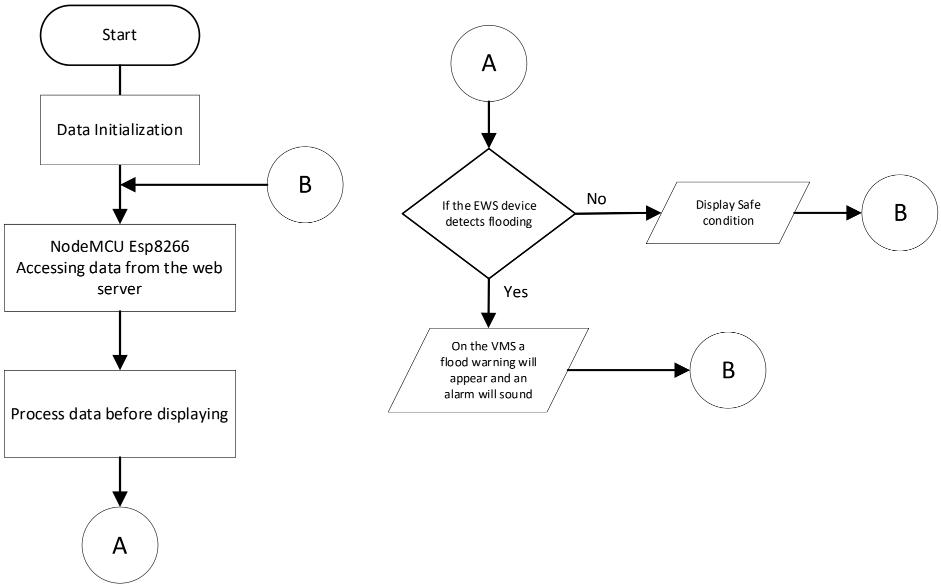

Figure 3. System flowchart

Figure 3. is a Flowchart that describes the workflow of the Flood *Early Warning System* information system. The data initialization stage is the initial stage of data preparation, then NodeMCU ESP8266 accesses the required data from the *web server*. The data will be processed first before being displayed. If the EWS device detects a potential flood, the VMS will display a flood warning, and an alarm will sound. If the EWS device does not detect a possible flood, the alarm will not sound, and the VMS will display safe conditions.

2.4. Tool Design

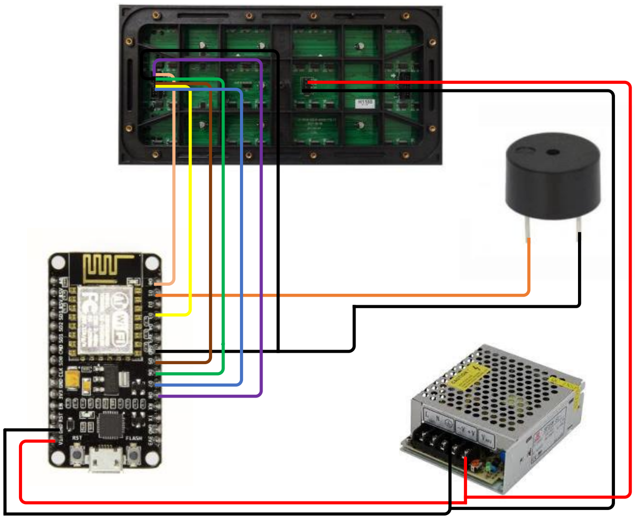

Figure 4. Tool design

Figure 4. is the wiring of the *Early Warning System* flood information dissemination device, using a 5V power supply as the power supply of the P10 LED module and the ESP8266 NodeMCU. After that, the ESP8266 will process the data accessed on the EWS flood webserver. The data processing results will be displayed on the P10 LED module, which functions as a water-level VMS. If the data received identifies flooding, the VMS will display a warning status, and the buzzer will sound.

2.5. Telegram Bot Design

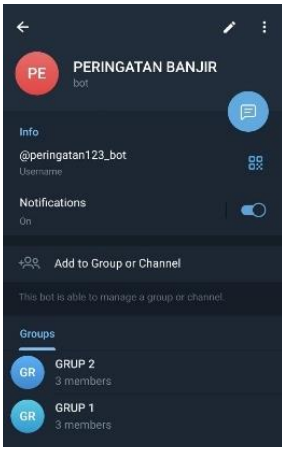

Figure 5. Telegram bot

Figure 5 shows the design of the Telegram bot to send notifications related to the IoT-based flood EWS. The Telegram bot is named "BANJIR WARNING." The Bot is then divided into groups, namely Group 1 and Group 2, which aim to simulate sending notifications to two different groups/regions. To send messages to both groups, adding both chat IDs from the group to the program is enough.

3. RESULTS AND DISCUSSION

3.1. Testing Data Retrieval from Server

The first testing stage is to test the data retrieval process from the server using the NodeMCU ESP8266. Here, the data used is simulation data inputted into the Database created in Figure 6, which contains the ID, sensor value, status, and date. The latest data from the Database will then be displayed on the web server, as shown in Figure 7.

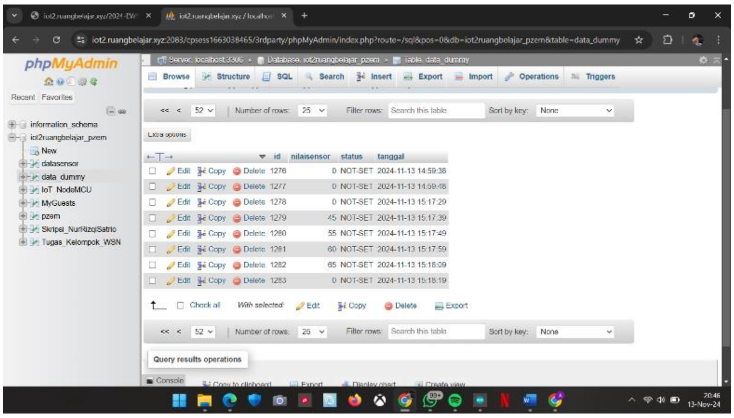

Figure 6. Database

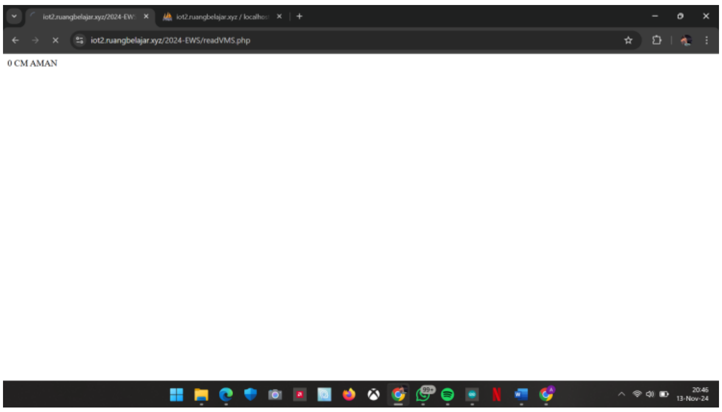

Figure 7. Web server

This test aims to determine the time of data retrieval from two different VMSs and then determine the difference in value from the request time of the two VMSs as reference material to analyze the synchronization of two different VMSs.

a. Testing Using different Providers

Table 1. testing the request time of 2 VMS different providers

No.	VMS Request Time 1	VMS Request Time 2	Time Difference (seconds)
1	15:02:32	15:02:32	0
2	15:02:42	15:02:42	0
3	15:02:52	15:02:52	0
4	15:03:02	15:03:02	0
5	15:03:12	15:03:12	0
6	15:03:22	15:02:22	0
7	15:03:32	15:02:33	1
8	15:03:42	15:03:43	1
9	15:03:53	15:03:53	0
10	15:04:03	15:04:03	0
11	15:04:13	15:04:13	0
12	15:04:23	15:02:24	1
13	15:04:33	15:02:34	1
14	15:04:43	15:04:44	1
15	15:04:53	15:04:54	1
16	15:05:03	15:05:04	1
17	15:05:13	15:05:14	1
18	15:05:23	15:05:25	2
19	15:05:34	15:05:35	1
20	15:05:44	15:05:45	1
21	15:05:54	15:05:56	2
22	15:06:04	15:06:16	2
23	15:06:14	15:06:26	2
24	15:06:24	15:06:36	2
25	15:06:34	15:06:46	2
26	15:06:44	15:06:56	2
27	15:06:54	15:07:07	3
28	15:07:04	15:07:17	3
29	15:07:14	15:07:27	3
30	15:07:25	15:07:27	2
	Average time difference		1,16

Table 1. compares request times between the two VMSs and then determines the time difference of each request to the server from the two VMSs. The two VMS are connected to different internet networks; VMS 1 is connected to a cellphone hotspot with an internet speed of 62 Mbps, while VMS 2 is connected to the Electro Lab lecture room wifi with an internet speed of 32 Mbps. This difference in internet speed also affects the request time made by the two VMS. Thus, based on 30 experimental samples, the average value of the difference in request time between the two VMS is 1.16 seconds.

b. Testing Using the same Provider

Table 2. testing request time of 2 same VMS providers

No.	VMS Request Time 1	VMS Request Time 2	Time Difference (seconds)
1	11:41:31	11:41:31	0
2	11:41:41	11:41:42	1
3	11:41:52	11:41:52	0

4	11:42:02	11:42:02	0
5	11:42:12	11:42:12	0
6	11:42:23	11:42:23	0
7	11:42:33	11:42:33	0
8	11:42:43	11:42:43	0
9	11:42:53	11:42:53	0
10	11:43:04	11:43:04	0
11	11:43:14	11:43:14	0
12	11:43:24	11:43:24	0
13	11:43:35	11:43:35	0
14	11:43:45	11:43:45	0
15	11:43:55	11:43:55	0
16	11:44:05	11:44:05	0
17	11:45:06	11:45:06	0
18	11:45:16	11:45:16	0
19	11:45:26	11:45:26	0
20	11:45:36	11:45:36	0
21	11:45:46	11:45:46	0
22	11:45:57	11:45:57	0
23	11:46:07	11:46:07	0
24	11:46:17	11:46:17	0
25	11:46:28	11:46:28	0
26	11:46:38	11:46:38	0
27	11:46:48	11:46:48	0
28	11:46:49	11:46:49	0
29	11:46:59	11:46:59	0
30	11:47:09	11:47:09	0
Average time difference			0.03

Table 2. shows the comparison of request time between two VMSs. VMS are connected to the same internet network; VMS 1 and VMS 2 are connected to a cellphone hotspot with an internet speed of 62 Mbps. The results show that one out of 30 experimental samples had a different request time between the two VMS, so the average difference was 0.033 seconds. This can be caused by several factors, one of which is that the network is experiencing congestion, and the time of sending data from one of the devices to the server can be affected.

The two tests that have been carried out have found that the accuracy or synchronization of the two VMS in making requests to the server, both when using the same provider and different providers, is as expected because the average difference is relatively low, under 2 seconds.
All symbols used in the equations should be defined in the following text.

3.2. Testing Data Appearance on VMS

In this test, test whether the data previously accessed by ESP8266 from the server can be displayed on the VMS. Furthermore, the test results can be used to prove whether the data on the server is the same as the data displayed on the VMS, as shown in the table below.

Figure 8. 1VMS testing

Table 3. VMS Testing

NO	Data on Server (cm)	Display on VMS 1	Display on VMS 2
1	10	10 CM SAFE	10 CM SAFE
2	20	20 CM SAFE	20 CM SAFE
3	30	30 CM SAFE	30 CM SAFE
4	40	40 CM SAFE	40 CM SAFE
5	50	50 CM SAFE	50 CM SAFE
6	60	60 CM ALERT	60 CM ALERT
7	70	70 CM ALERT	70 CM ALERT
8	80	80 CM ALERT	80 CM ALERT
9	90	90 CM ALERT	90 CM ALERT
10	100	100 CM ALERT	100 CM ALERT
11	110	110 CM DANGER	110 CM DANGER
12	120	120 CM DANGER	120 CM DANGER
13	130	130 CM DANGER	130 CM DANGER
14	140	140 CM DANGER	140 CM DANGER
15	150	150 CM DANGER	150 CM DANGER

Table 3 shows the results obtained from testing the appearance of data on the VMS. All simulation data entered on the server can be displayed correctly on the VMS, accompanied by a predetermined warning level status. Every 10 seconds, the VMS updates the display with the latest data from the server. If the data on the server does not change, the VMS continues to display the previous data.

3.3. Alert Alarm Testing

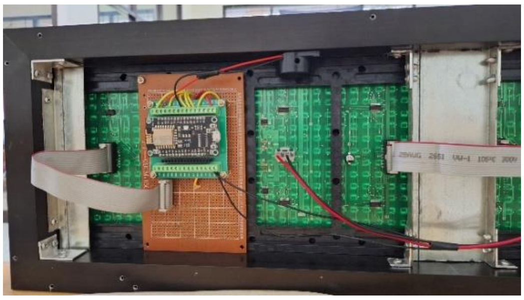

Figure 9. 2Warning alarm test

In the warning alarm test, a piezoelectric buzzer has been assembled, as shown above. If the warning level is detected as "ALERT," then the buzzer will sound intermittently for 1 second for a duration of 6 seconds. Meanwhile, when the warning level is detected as "DANGER," the buzzer will sound intermittently for 2 seconds for a duration of 6 seconds. The test results can be seen in the table below.

Table 4. Buzzer Testing

NO	Data on Server (cm)	Buzzer Condition 1	Buzzer 2 Condition
1	15	Die	Die
2	30	Die	Die
3	45	Die	Die
4	60	Beep (active for 6 seconds at 1-second intervals)	Beep (active for 6 seconds at 1-second intervals)
5	75	Beep (active for 6 seconds at 1-second intervals)	Beep (active for 6 seconds at 1-second intervals))
6	90	Beep	Beep

		(active for 6 seconds at 1-second intervals)	(active for 6 seconds at 1-second intervals)
7	105	Beep (active for 6 seconds at 2 second intervals)	Beep (active for 6 seconds at 2 second intervals)
8	120	Beep (active for 6 seconds at 2 second intervals)	Beep (active for 6 seconds at 2 second intervals)
9	135	Beep (active for 6 seconds at 2 second intervals)	Beep (active for 6 seconds at 2 second intervals)
10	150	Beep (active for 6 seconds at 2 second intervals)	Beep (active for 6 seconds at 2 second intervals)

The warning alarm test results in the table above show that the buzzer functions properly in this system. When the data on the server is 15, 30, and 45, the buzzer will turn off because the condition is identified as safe. When the data on the server is 60, 75, and 90, the buzzer will sound for 6 seconds with an interval of 1 second because the condition is identified as alert. When the data on the server is 105, 120, 135, and 150, the buzzer will sound for 6 seconds with an interval of 2 because the condition is identified as dangerous.

3.4. Telegram Notification Testing

Telegram bot notification testing is conducted on the two groups created to prove whether the Telegram bot can send notifications correctly and simultaneously to both groups. When conditions are safe or the data on the server is detected as secure, the Telegram bot will not send any notifications.

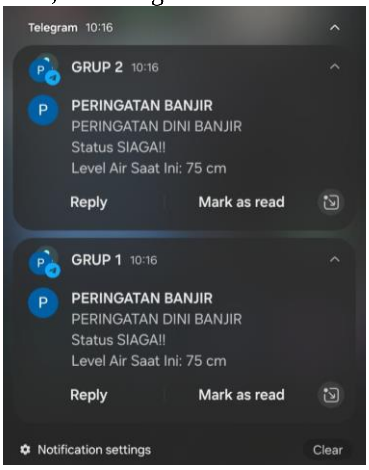

Figure 10. Alert level notification testing

Figure 4.10 is a test result showing a Telegram bot notification. The notification displays flood early warning information that is automatically sent when the data on the server indicates the alert level. The notification states, "Alert Status!!!, Current Water Level: 75 cm," which means that the current condition indicates a STANDBY status with a water level of 75 cm. It can also be seen in the picture above that the Telegram bot managed to send notifications simultaneously to two different groups.

Figure 11. Hazard-level notification testing

Figure 4.11 shows the test results in the form of notifications sent by the Telegram bot as part of the flood early warning system. This notification is automatically sent when data received from the server shows that conditions have reached the danger level. In the notification, there is a message that reads, "DANGER STATUS!!!, Current Water Level: 100 cm." This message shows that the current water level has reached 100 cm, indicating the dangerous condition. In addition, it can be seen in the image that the Telegram bot successfully sent notifications to two different groups simultaneously.

3.5. Functional Analysis

The *Early Warning System* flood information dissemination device that has been designed has system reliability that can be seen from the consistency of the tests that have been carried out. Based on the test results, the overall system performed well. This can be proven by the similarity of all data values on the server with those accessed via ESP8266 and displayed on the VMS. In addition, buzzers and Telegram bots can correctly send alerts if there is a change in status on the server that identifies a flood warning.

The main factor affecting this system's reliability is the internet network quality used. This system accesses data from the server using the internet network. Hence, the stability of the internet network dramatically affects the success of accessing data from the server and sending Telegram bot notifications.

4. CONCLUSION

This research successfully designed an IoT-based flood early warning information dissemination system, integrating a Variable Message Sign (VMS), buzzer, and notification via Telegram bot. The test results show that the synchronization of two VMS in making requests to the server is quite good, with an average time difference of 1.16 seconds on different providers and 0.03 seconds on the same provider, indicating adequate synchronization performance. The VMS also showed an accurate response to data changes on the server, where the SAFE, WARNING, and DANGER statuses were displayed according to the water level range. Buzzers responded to data changes with sound patterns corresponding to safe, alert, and dangerous conditions, while Telegram bots successfully sent real-time notifications when changes in status levels occurred. The reliability of the system is affected by the stability of the internet network, as data transmission and notifications depend on the available connection. Nonetheless, the device effectively delivers information quickly and widely to the public, especially in flood-prone areas or areas that become community mobilization points. With these results, the designed flood early warning system has functioned well and met expectations in providing accurate and fast information.

ACKNOWLEDGMENTS

On behalf of the authors, we would like to thank our supervisors, Dr Eng Komang Somawirata ST MT. and Dr. Michael Ardita, ST., MT. for their guidance during this research, providing advice, opinions, and motivating us throughout this study process.

REFERENCES

[1] M. F. Kurniati, P. Periyadi, A. Ariansyah, E. Febriana, and H. Rahmadina, "Dissemination of Flood Early Warning Information at Kemuning Banjarbaru Village Office," vol. 2, no. 2, 2024.

[2] W. Widodo *et al.*, "Implementation of Flood Early Warning System (Flews) as a Flood Mitigation Effort in Sungai Nangka Village, South Balikpapan," *Abdimas Univers.*, vol. 5, no. 1, pp. 159-167, Jul 2023, doi: 10.36277/abdimasuniversal.v5i1.298.

[3] A. Sumardiono, E. Alimudin, Z. Zaenurrohman, and H. Susanti, "Design of Monitoring Early Warning System for Flood Disaster Based on River Flow Height Using SIM900 Modem and Internet of Things," *Infotekmesin*, vol. 13, no. 1, pp. 112-117, Jan 2022, doi: 10.35970/infotekmesin.v13i1.1019.

[4] G. Vijanarki, "Malang is Flood Prone: Malang City Spatial Planning Problems," *ITN Malang*, 2024.

[5] K. kurniawan, "BPBD Malang City Adds 11 EWS Devices Installed at Flood Prone Points: Proposal from the Community," *TribunJatim.com*, 2024.

[6] M. N. Ruhyat, R. Rahmadewi, and Y. Saragih, "IMPLEMENTATION OF TRANSRECEIVER MODULE NRF24L01 AS A WIRELESS DATA SENDER AND RECEIVER ON A BANJIR EARLY WARNING MONITORING SYSTEM," vol. 6, no. 1.

[7] V. H. C. Putra, G. Kanugrahan, and A. P. Wahyu, "Flash Flood Early Warning System in VULcanic SAND MINING AREA USING INTERNET OF THINGS," 2024.

[8] Yogyakarta University of Technology, A. F. Waluyo, T. R. Putra, and Yogyakarta University of Technology, "Flood Early Warning Based on Internet Of Things (IOT) and Telegram," *Infotek J. Inform. And Technol.*, vol. 7, no. 1, pp. 142-150, Jan 2024, doi: 10.29408/jit.v7i1.24109.

[9] P. N. Sari and F. W. Christanto, "Early Warning System Based on Internet of Things (IoT) in Flood Prone Areas in Semarang City," *J. Appl. Multimed. Netw.*, vol. 7, no. 2, pp. 62-70, Dec 2023, doi: 10.30871/jamn.v7i2.6503.

[10] I. A. Wandi and A. Ashari, "Monitoring Water Levels and Rainfall in an IoT-Based Flood Disaster Early Warning System," *IJEIS Indones. J. Electron. Instrum. Syst.*, vol. 13, no. 1, Apr 2023, doi: 10.22146/ijeis.83569.

[11] M. J. S. Simanjuntak, S. Sundari, and Y. D. Lestari, "Design and Implementation of a Web-Based Flood Level Monitoring System and IoT (Internet Of Things) Using Ultrasonic Sensors," 2022.

[12] P. T. Wikantama and R. Puspitasari, "Design of Flood Level Measuring Device with ESP32 and Telegram Based on IoT," *Elektriese J. Science and Technol. Electro*, vol. 13, no. 02, pp. 107-114, Nov 2023, doi: 10.47709/elektriese.v13i02.3108.

[13] M. A. Hudhoifah and D. I. Mulyana, "Implementation of Temperature and Humidity Monitoring in Mushroom House for Oyster Mushroom Cultivation Using NodeMCU - ESP8266 in Wirasana Village Purbalingga: Implementation of Temperature and Humidity Monitoring in Mushroom House for Oyster Mushroom Cultivation Using NodeMCU - ESP8266 in Wirasana Village Purbalingga," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 2, pp. 472-480, Feb 2024, doi: 10.57152/malcom.v4i2.1222.

[14] G. C. Lenardo, Herianto, and Y. Irawan, "Utilization of Telegram Bot as Academic Information Media at STMIK Hang Tuah Pekanbaru," *JTIM J. Technol. Inf. and Multimed.*, vol. 1, no. 4, pp. 351-357, Feb 2020, doi: 10.35746/jtim.v1i4.59.

[15] F. T. Julfia, E. Satryawati, D. Setiadi, W. Ariani, and R. Rijadi, "Analysis of VMS (Variable Message Sign) Message Delivery Process on Five Jabodetabek Toll Road Sections," *J. Educ.*, vol. 5, no. 2, pp. 1685-1695, Jan 2023, doi: 10.31004/joe.v5i2.805.