

Early Warning System for Flood Disasters Using the Internet of Things

Yuyun^{1,*}, Hamka Zulfaesa¹, Latief Arda Abdul¹

¹ Department of Computer System, STMIK Handayani Makassar, Makassar, Indonesia

*Corresponding author. Email: yuyunwabula@handayani.ac.id

ABSTRACT

Floods are one of the most frequently occurring natural disasters in Indonesia. It is therefore of special concern to reduce the risk of flood fatalities and other damage. The purpose of this study is to design a flood early warning system based on the Internet of Things (IoT). In this work, we use an ultrasonic HC-SR04 sensor to collect information about water levels. Further, we use an MCU8266 node as a microcontroller to analyze the sensor data and test it using a fuzzy inference system (FIS). To monitor the server activities, a Blink application is used as an iCloud IoT to hook up to smartphone devices. The test results show that all the devices function properly and that the system can classify water levels into normal, standby, alert, and danger categories in real time. Thus, the server can send information to the output device before a flood disaster occurs.

Keywords: Flood, Early Warning System, MCU8266, Fuzzy Inference System, HC-SR04

1. INTRODUCTION

Geographically, Indonesia is an archipelago country located between two continents, Asia and Australia. As a result, the season changes every six months, from the rainy season to the dry season and vice versa. In Indonesia, rainfall in each region depends on the shape of the terrain, slope direction, and wind direction. Based on the above constellation, Indonesian rainfall in each region can be divided into three rainfall patterns: monsoon rain patterns, equatorial rainfall patterns, and local rainfall patterns [1]. Makassar City, the capital of South Sulawesi Province, is a city that is vulnerable to flooding. Makassar City is categorized as an area with gentle slopes, with heights of 1-22 m above sea level (masl). In general, the flood area in Makassar City is a low area with an elevation of 1-4 masl; it is located along the Tello River watershed, the Jenne'berang River watershed, and the Pampang River [2].

The Perumnas Antang Block 10 area, situated in the Manggala subdistrict, is a location that is often hit by floods when the rainy season arrives. An early warning system for floods is therefore necessary to reduce the number of casualties due to flooding. Such a system, however, is currently absent. Many researchers have studied early warning systems, including design systems for monitoring water conditions, such as water level, water flow, and precipitation level [3]. These systems

used wireless sensor network (WSN) technology, the mobile general packet radio service (GPRS), and VirtualCOM to monitor water conditions. Then, [4] proposed a tailored sensor network to forecast, map, and monitor urban flash floods. [5] proposed a PIC microcontroller device to detect current water levels and a GSM modem to provide notifications. Additionally, [6] offered a design for rainstorm calculations based on three criteria: point rainstorms, surface rainstorms, and the distribution of rainstorms. [7] described a GridStix-based flood monitoring system to support the adaptation of WSN technology to changing environmental conditions. Further, [8] created a web-based system that utilized hydrological observation stations to monitor flood data, while [9] proposed WSN technology with an Arduino microcontroller and an XBee transceiver as information devices of the alert center. All these existing systems, however, cannot properly deal with real-time data.

To solve the problem of real-time data, some researchers proposed Internet of Things (IoT) technology to monitor water conditions. [10] utilized an ESP-8266 microcontroller, which was connected to a water level sensor and a water velocity sensor. Then, [11] deployed IoT technology with two sensors, namely an ultrasonic and a water sensor, by applying an IEEE802.11 communication format that measures water levels in real time. Further, [12] applied an MCU node in a flood warning and monitoring system, but this study used a

manual scale for determining flood level categories. The purpose of this study is to design an early warning system for flood disasters using the fuzzy inference system (FIS) method for dividing the scale of floodwater height levels. In this research, we used an ultrasonic HC-SR04 sensor to obtain water level information and utilized an MCU8266 node as a microcontroller to collect the sensor data. We used a Blink application to save the water height data. Furthermore, the application used iCloud IoT technology to connect to smartphone devices.

2. METHOD

2.1. MCU Node

To monitor flood status in real-time conditions, we used wireless sensor network WSN technology to inform people about current water conditions. In this case, the WSN for the flood disaster early warning system contains an ESP8266 MCU node. This node operates as a microcontroller or as a data processing unit equipped with Wi-Fi to get Internet access. This node was developed from ESP8266, which uses e-Lua firmware. The MCU node uses the Lua programming language, which is a package from ESP8266. Besides supporting the Lua language, the MCU node also supports Arduino IDE software. Before it can be utilized, this board must first be flashed to support the tools used. If the Arduino IDE device is used, then Ai-Thinker firmware should be applied to support the AT command set [13]. Data collected by the MCU node is then sent to and subsequently saved on the online server by utilizing Blink's iCloud. The data forwarded to the user's smartphone to get the water level information via the smartphone device. Figure. 1 shows the flow diagram of the flood disaster early warning system.

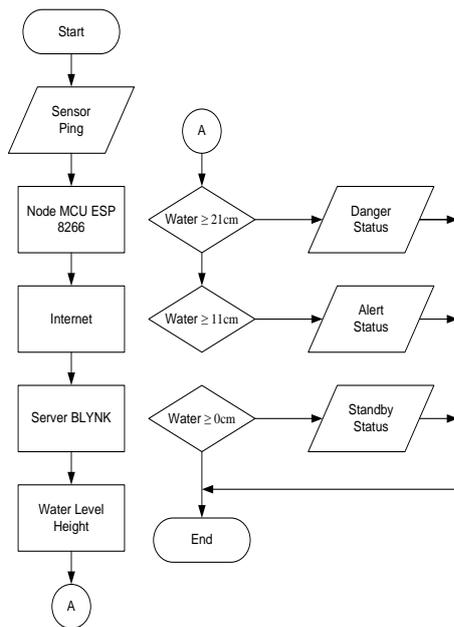


Figure 1 Flow Diagram For The Water Level Status

The system is designed in the form of a prototype and consists of a vessel (a place that contains water), a PING sensor, an ESP8266 MCU node, a Blynk server, and a smartphone (see Figure. 3).

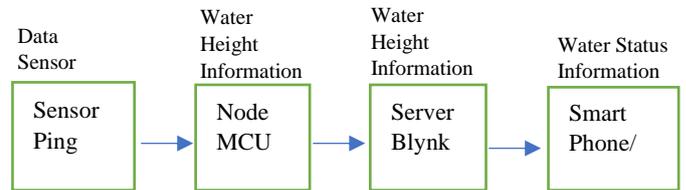


Figure 2 System Workflow

The system's workflow is as follows:

- The PING sensor retrieves water level data and sends it to the ESP8266 MCU node. The sensor data contains water level information, expressed in centimeters (see Figure. 2).
- The ESP8266 MCU node receives data about water levels and processes it using the FIS method. The aim is to normalize the fuzzy membership functions to obtain normal, standby, alert, and danger status.
- The data processed by the MCU node is then stored on the Blynk server to be given to the user. The smartphone displays the water height level information with the status (normal, standby, alert, and danger).

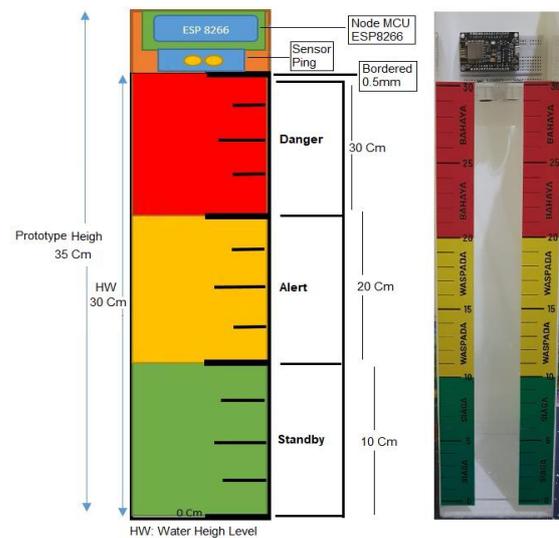


Figure 3 Hardware Prototype Design For The Simulation Process

Figure. 3 shows the proposed prototype. The prototype's height is 35 cm, and the vessel's height is 30 cm. In this simulation, the PING sensor and the MCU node are placed on the container. When water is poured into the container, the PING sensor will read the movement of the water. The data from the PING sensor is then forwarded to the MCU node to be processed and converted into information. If the water level is 0-10 cm, the PING sensor will read the standby status, while 11-20 cm is the alert status and 21-30 cm is the danger status.

The data from the MCU node is subsequently saved on the Blink server. The data stored on the Blink server is then submitted to the user's smartphone to get the latest information on the status of the water level.

2.2. Fuzzification

Determining the degree of membership in a fuzzy system is known as fuzzification. This process changes numeric variables into linguistic variables. The following is the fuzzification process for the water level.

$$\mu_{standby}(x) = \begin{cases} 1; & x \leq 9 \\ \frac{10-x}{10-9}; & 9 < x < 10 \\ 0; & x \geq 10 \end{cases} \quad (1)$$

The alert status membership function:

$$\mu_{alert}(x) = \begin{cases} 0; & x \leq 9 \text{ atau } x \geq 20 \\ \frac{x-9}{10-9}, & 9 < x < 10 \\ \frac{20-x}{20-19}, & 19 \leq x < 20 \\ 1; & 10 \leq x \leq 19 \end{cases} \quad (2)$$

The alert status membership function:

$$\mu_{danger}(x) = \begin{cases} 1; & x \geq 20 \\ \frac{x-19}{20-19}; & 19 < x < 20 \\ 0; & x \leq 19 \end{cases} \quad (3)$$

3. RESULT AND DISCUSSION

3.1. Hardware Testing Analysis

The testing of the hardware prototype begins by providing a 3.5-volt supply to the microcontroller. The ESP8266 MCU microcontroller node is active when the indicator lamp is red. To activate the PING sensor, a 5-volt signal is inserted into the power supply. The device status is active when the light indicator is on (see Figure 4). In this study, the PING sensor reads the water movement, which rises and falls, and then sends the reading to the ESP8266 MCU node to get the water height information.

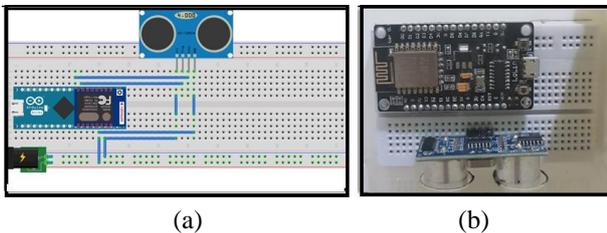


Figure 4 Water level measurement circuit, (a) the circuit board designed using an application and (b) direct installation onto the circuit board

3.2. System Testing

The testing of the device starts by making sure that the smartphone has installed the IoT water level

application. After the application is installed, it is checked whether the ESP8266 MCU node is connected. The test was carried out in two places by using two ESP8266 MCU nodes, which were named according to the location of the area to be monitored. The status of the water level was divided into the categories normal, standby, alert, and danger by the indicator lights.

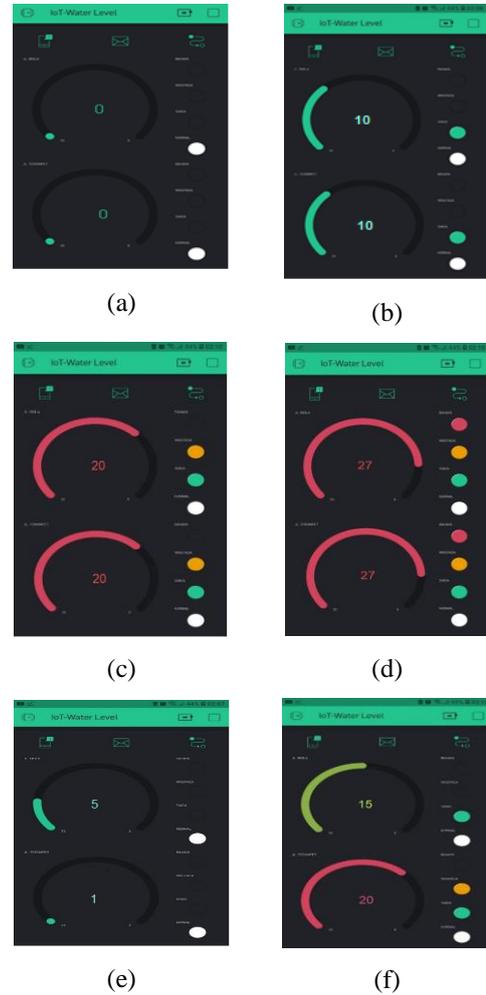


Figure 5 Application Test Result: (a) Normal, (b) Standby, (c) Danger, (d) Danger, (e) Standby, and (f) Alert and Danger Status

Figure 5 shows the IoT water level application. The application is first tested by activating the PING sensor. The result is an indicator light, as shown in Table I

Table 1. Table of testing Result

Water Level	Indicator
Normal	White
Standby	Green
Alert	Yellow
Danger	Red

From Table I and Figure 5, it can be concluded that when the water level is in the normal category, the

indicator light in the application is white. When the water level is in the standby category, the indicator light is green; when the water level is in the alert category, the indicator light is yellow; and when the water level is in the danger category, the indicator light is red. The test result shows that all devices are functioning properly, as the indicator lights are on or off in accordance with the instructions.

The next step is testing the water level by conducting several experiments to measure the height of the surface water, the water level, and the application indicators. The test in Table 2 starts without water in the vessel, and the result was a white indicator light. Then the water height was increased to 1 cm and further from 2 to 10 cm, and the result was a green indicator light. Then we proceeded with a water level of 11-19 cm, and the result was a yellow indicator light. The last experiment was to put water in the vessel with a height of 20-30 cm, and the result was that the red indicator light was active. Thus, all the devices worked properly. An alternative description of Table 2 is that if the water level is in the standby category, the height of the water surface is $\pm 1-10$ cm. The alert water level occurs if the depth of the water surface is $\pm 11-19$ cm, and if the water level is in the danger category, the depth of the surface water is $\pm 20-30$ cm. The test results can be seen in Table 2.

Table 2. Testing of Water Depth, water heigh and application light indicator

No	Water Depth (cm)	Water Status	Indicator light
1	± 0	Normal	White
2	± 1 to ± 10	Standby	Green
3	± 11 to 19	Alert	Yellow
4	± 20 to 30	Danger	Red

The next tests were carried out based on water surface depth, road location, and application indicators. The trial was located in JL. Trompet and JL. Biola in Perumahan Antang Blok 10. The reason for choosing these locations is that here the land has a slope so that when it is raining for a long time, the surrounding areas will be flooded. The test results are shown in Figure 5.

3.3. System Testing

Table 2 is a comparison of direct observations of the prototype (manual testing) and system testing involving 33 trials. The water level heights of 9.1 and 19.1 cm produce different results in the manual and system testing. In the manual testing, a height of 9.1 cm has the standby status, whereas in the system testing, it produces the alert status. Further, a height of 19.1 cm is the alert status in the manual testing and the danger status in the system testing. The formula for the testing accuracy is

Table 3. Comparison of Manual testing and System Testing

Manual Data				Fuzzy System		
Rule	Water High (cm)	Rainfall	Status	Water High (cm)	Rainfall	Status
1	0	No Rain	Standby	0	No Rain	Standby
2	1, 2, 3	Drizzle, No Rain, Heavy	Standby	1, 2, 3	Drizzle, No Rain, Heavy	Standby
3	4, 5, 6	Drizzle, No Rain, Heavy	Standby	4, 5, 6	Drizzle, No Rain, Heavy	Standby
4	7, 8, 9	Drizzle, No Rain, Heavy	Standby	7, 8, 9	Drizzle, No Rain, Heavy	Standby
5	9,1	Heavy	Standby	9,1	Heavy	Alert
6	10, 11, 12	Drizzle, No Rain, Heavy	Standby	10, 11, 12	Drizzle, No Rain, Heavy	Standby
7	13, 14, 15	Drizzle, No Rain, Heavy	Alert	13, 14, 15	Drizzle, No Rain, Heavy	Alert
8	16, 17, 18	Drizzle, No Rain, Heavy	Alert	16, 17, 18	Drizzle, No Rain, Heavy	Alert
9	19	Drizzle	Alert	19	Drizzle	Alert
10	19,1	Drizzle	Alert	19,1	Drizzle	Danger
11	20, 21, 22	Drizzle, No Rain, Heavy	Alert	20, 21, 22	Drizzle, No Rain, Heavy	Alert
12	23, 24, 25	Drizzle, No Rain, Heavy	Danger	23, 24, 25	Drizzle, No Rain, Heavy	Danger
13	26, 27, 28	Drizzle, No Rain, Heavy	Danger	26, 27, 28	Drizzle, No Rain, Heavy	Danger
14	29	Drizzle	Danger	29	Drizzle	Danger
15	30	No Rain	Danger	30	No Rain	Danger

$$\begin{aligned}
 \text{Accuracy} &= \frac{\text{Accurate data}}{\text{Whole data}} \times 100\% \\
 &= \frac{31}{33} \times 100\% = 93,93\%
 \end{aligned}
 \tag{4}$$

4. CONCLUSION

The conclusion of this research is that the created prototype can detect water levels and send information to a smartphone by giving a notification message or sending a message via email. The tests with different water levels show that the application indicators operate properly, obtaining an accuracy rate of 93.93% for the simulation data of 33 trials. Thus, this prototype can provide correct information about water levels. We expect that the prototype and the proposed method can contribute to IoT applications in the field of disaster response systems. In future research, image and video output may be added to the prototype.

ACKNOWLEDGMENTS

This study was supported by Ministry of Research and Technology/National Research and Innovation and STMIK Handayani Makassar.

REFERENCES

- [1] "Gambar Umum Kota Makassar" Rencana Pembangunan Jangka Menengah Daerah Kota Makassar Tahun 2014-2019
- [2] "Geografi Regional Indonesia" Jurusan Pendidikan Geografi Fakultas Ilmu Pengetahuan Sosial Universitas Pendidikan Indonesia 2008.
- [3] J. Sunkpho and C. Ootamakorn "Real-time flood monitoring and warning system." *Songklanakarin Journal Science and Technology*. 33 (2), 227-235, 2011
- [4] N. Bin Chang and D. H. Guo, "Urban flash flood monitoring, mapping, and forecasting via a tailored sensor network system," *Proc. 2006 IEEE Int. Conf. Networking, Sens. Control. ICNSC'06*, pp. 757–761, 2006.
- [5] "Flood Detection using Sensor Network and Notification via SMS and Public Network 5.pdf".
- [6] Z. Li, H. Zhang, V. P. Singh, R. Yu, and S. Zhang, "A simple early warning system for flash floods in an ungauged catchment and application in the Loess Plateau, China," *Water (Switzerland)*, vol. 11, no. 3, 2019, doi: 10.3390/w11030426.
- [7] D. Hughes et al., "An Intelligent and Adaptable Grid-based Flood Monitoring and Warning System," *Proc. UK E-Science All Hands Meet.*, pp. 53–60, 2006.
- [8] E. Tate and K. Cauwenberghs, "An innovative flood forecasting system for the demer basin: A case study," *Int. J. River Basin Manag.*, vol. 3, no. 3, pp. 163–167, 2005, doi: 10.1080/15715124.2005.9635255.
- [9] S. N. Sakib, T. Ane, N. Matin, and M. S. Kaiser, "An intelligent flood monitoring system for Bangladesh using wireless sensor network," *2016 5th Int. Conf. Informatics, Electron. Vision, ICIEV 2016*, pp. 979–984, 2016, doi: 10.1109/ICIEV.2016.7760145.
- [10] H. Yuliandoko, Subono, V. A. Wardhany, S. H. Pramono, and P. Siwindarto, "Design of flood warning system based IoT and water characteristics," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 16, no. 5, pp. 2101–2110, 2018, doi: 10.12928/TELKOMNIKA.v16i5.7636.
- [11] T. Perumal, M. N. Sulaiman, and C. Y. Leong, "Internet of Things (IoT) enabled water monitoring system," *2015 IEEE 4th Glob. Conf. Consum. Electron. GCCE 2015*, pp. 86–87, 2016, doi: 10.1109/GCCE.2015.7398710.
- [12] M. S. Mohd Sabre, S. S. Abdullah, and A. Faruq, "Flood Warning and Monitoring System Utilizing Internet of Things Technology," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, no. 4, pp. 287–296, 2019, doi: 10.22219/kinetik.v4i4.898.
- [13] Aan Tohir Efendi "Sistem Pengendali Pintu Berbasis WEB Menggunakan Node MCU ESP8266" Jurusan Teknik Komputer, Sekolah Tinggi Manajemen Informatika dan Komputer AKAKOM Yogyakarta, 2017."Gambar Umum Kota Makass Pembangunan Jangka Menengah Daerah Kota Makassar Tahun 2014-2019 E.M.