

Implementation of Fuzzy Logic for Early Warning System for Flood Disaster in Cilacap District

Erna Alimudin¹, Arif Sumardiono², Zaenurrohman³

^{1,2,3} Electronics Engineering, Cilacap State of Polytechnic, Jl. Dr. Soetomo No. 1, Cilacap 53212, Indonesia

ARTICLE INFO

Article historys:

Received : 28/10/2024

Revised : 15/01/2025

Accepted : 12/02/2025

Keywords:

EWS; Flooding; Fuzzy Logic;
Mamdani; Membership Function

ABSTRACT

Indonesia's tropical climate and high rainfall frequently lead to unnoticed river overflows, causing significant health and economic losses. A Flood Early Warning System (EWS) using LoRa communication and an Arduino Mega2560 microcontroller was developed, incorporating ultrasonic, DHT11, rain gauge, and anemometer sensors to measure water level, temperature, humidity, rainfall, and wind speed. Unlike previous systems, it applies fuzzy logic for improved accuracy by considering rainfall and wind speed. Sensor errors ranged from 2.14% to 3.19%, with data transmission to ThingSpeak taking 11 seconds (99.94% accuracy) and to a website averaging 15.03 seconds. Fuzzy logic enhanced the system's warning accuracy.



This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/)

Corresponding Author:

Erna Alimudin

Electronics Engineering, Cilacap State of Polytechnic, Jl. Dr. Soetomo No. 1, Cilacap 53212, Indonesia

Email: ernaalimudin@pnc.ac.id

1. INTRODUCTION

Indonesia, as a tropical country, experiences two main seasons: rainy and summer. [1] Prolonged rainy seasons accompanied by extreme weather have often caused extreme weather changes in recent years [2]. Unexpected high rainfall can cause flooding, as happened in Cilacap Regency in 2021 in Jeruklegi District and Kawunganten District in 2021 [3]. BMKG Climatology Station Semarang recorded rainfall with moderate to extreme intensity in the central and southern parts of the district. Concentrations of rainfall with extreme intensity exceeding 150 mm/min in 24 hours occurred [4]. The district had extreme rainfall causing the overflowing of existing rivers, so that many houses were submerged in water [5], [6].

The flood resulted in many houses being submerged and agricultural land damaged, causing huge losses to the community [7, 8]. In addition to physical damage, the flood also had an impact on economic and social aspects [7, 9]. Many farmers experienced crop failure, and small business owners suffered financial losses due to damaged or lost products [10]. Flooding also made access to health facilities difficult, disrupting community activities and increasing the risk of diseases such as diarrhea and dengue fever [11].

Looking at the above cases to overcome this problem, the development of a more sophisticated Early Warning System (EWS) is needed [12]. An EWS that combines rainfall, water level and wind speed sensors, and using the Fuzzy Logic method, will provide more accurate early warnings. With this system, the community can be better prepared for flooding and reduce the impact caused.

2. RESEARCH METHOD

2.1. Block Diagram

The block diagram shown in Figure 1 is an overview of the Early Warning System (EWS) system for flood disasters using the Arduino Mega 2560 microcontroller. The system is designed to detect extreme weather conditions, such as high rainfall, temperature and humidity, wind speed, and water level, through connected sensors. The data collected is then sent through the LoRa communication module to be monitored and processed to provide early warning to the community.

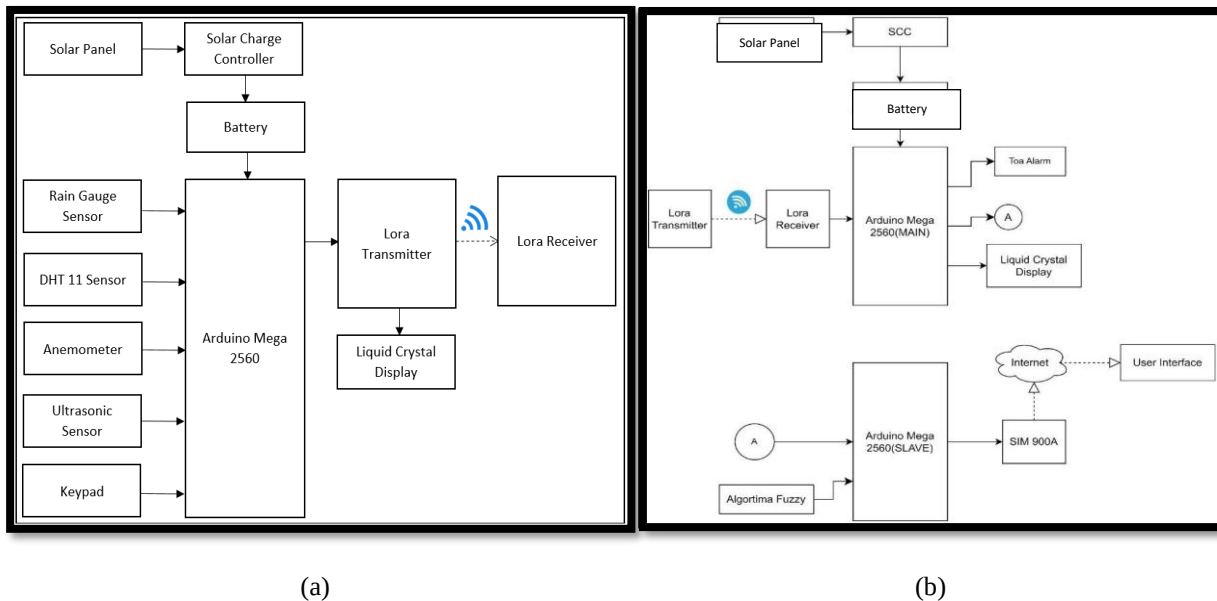


Figure 1. Block diagram of (a) transmitter and (b) receiver

Figure 1 (a) explains that the system consists of several main components that work together. The solar panel is connected to the Solar Charge Controller (SCC) to charge the battery, which serves as the system's power source. The Arduino Mega 2560 acts as the main controller that collects data from the Rain Gauge sensor (rainfall), DHT11 (temperature and humidity), Anemometer (wind speed), and Ultrasonic sensor (water level). The data is then displayed on the LCD screen and sent through the LoRa Transmitter to the LoRa Receiver for further processing in Figure 1 (b). The keypad is used as a manual input tool if needed.. The sensor data is processed by Fuzzy Logic algorithms to obtain more accurate results and transmitted to the database for further analysis. The whole system is well integrated between hardware and software, and has redundancy in data transmission, ensuring high system reliability for web-based monitoring.

The process starts with the initialization of the I/O pins, serial UART, and LCD in the system. The sensor then reads the environmental data, which is displayed on the LCD of the transmitter node before being sent to the receiver node. If the data is successfully sent, it is further processed by the Arduino Mega 2560 at the receiver node, and the data is sent to the Fuzzy Logic algorithm to get more accurate results. The processed data is displayed on the LCD and sent to the SIM900A module to be uploaded to the internet. If the internet is available, the sensor data is sent to the ThingSpeak database to be displayed on the web interface. If the data is received properly, the system will display the sensor data and output from Fuzzy Logic on the website for further monitoring.

2.2. Block Diagram of Receiver Node

The Receiver Node block diagram describes an Early Warning System (EWS) system for flood disasters that uses LoRa technology and an Arduino Mega 2560 microcontroller. This system is designed to detect and provide early warnings related to potential flooding through connected sensors and utilizes

the Fuzzy Logic method for data analysis. Data transmission is done wirelessly through the LoRa network and SIM 900A and displayed on the screen and user interface to increase community preparedness in the face of flood disasters.

Figure 1 (b) explains that this system consists of two main parts, namely the transmitter and receiver units. The solar panel charges the battery through the Solar Charge Controller (SCC), which then supplies power to the Arduino Mega 2560 (MAIN) as the main controller. Data received from the LoRa Receiver is sent to the alarm (Toa) and displayed on the Liquid Crystal Display. The signal is forwarded to the Arduino Mega 2560 (SLAVE) which processes the data using the Fuzzy Logic algorithm, then sends the results via the SIM 900A module to the internet to be displayed on the user interface.

2.3. Block Diagram of Fuzzy Logic On EWS

The following block diagram illustrates the application of the Fuzzy Logic method to an Early Warning System (EWS) system for flood detection. Utilizing various sensors that measure rainfall, water level, and wind speed, the resulting data is analyzed using fuzzy logic to produce more accurate decisions in detecting potential flooding. The system is designed to provide early warnings to the community in order to take appropriate precautions.

Data from ultrasonic sensors, anemometer, and rain gauge enter the fuzzification stage, where the data is converted into fuzzy values. Next, the predefined fuzzy rules are used in the fuzzy inference process to analyze the data. The results of this inference then go through the defuzzification stage to convert the fuzzy values into concrete values, which then produce outputs in the form of early warnings or other decisions related to flood potential.

2.4. Tool Design

1. The Overall Design



Figure 2. Overall Design

Figure 2 shows the overall design of the device used in the IoT-based monitoring and early warning system. The device is equipped with a solar panel at the top as a power source, which allows the device to operate autonomously in remote locations without relying on external power sources. There is a control box at the bottom of the panel that serves to control and display data from the sensors. It is designed with a sturdy iron frame structure to support the equipment and ensure its stability when installed in the field.

2.5. Electrical System Design

1. Electrical Design of Transmitter Node

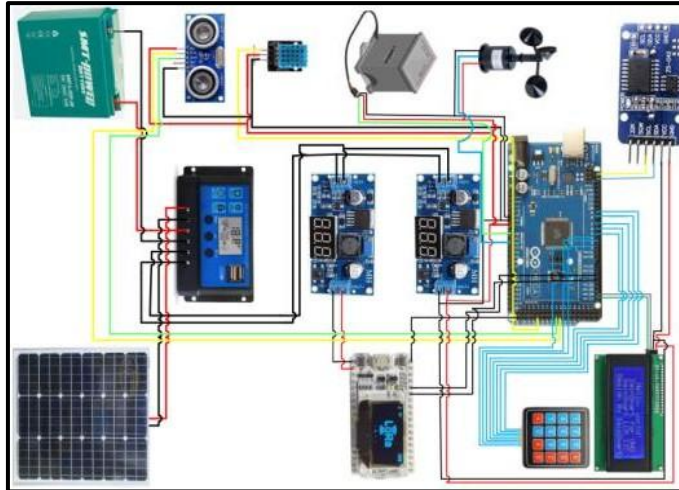


Figure 3. Transmitter Node Electrical Design

Figure 3 shows the electrical system design of the transmitter node in the Early Warning System (EWS) development project. This transmitter node includes several main components, such as a solar panel used to charge the battery as a power source, an ultrasonic sensor, a rain sensor (rain gauge), a temperature and humidity sensor (DHT11), and an anemometer to measure wind speed. All of these sensors are connected to the Arduino Mega 2560, which serves as the data processing center. The Arduino is also connected to a LoRa communication module to transmit data to the receiving node, as well as an LCD screen and keypad as the user interface. This design enables the collection and transmission of environmental data to detect potential flood disasters.

3. RESULTS AND DISCUSSION

This research aims to design and develop an Early Warning System (EWS) system for flood detection using the Fuzzy Logic method. The system integrates various sensors, such as rain gauge, ultrasonic water level sensors, and anemometer, to monitor environmental conditions that could potentially cause flooding. The data obtained from these sensors is then processed using a Fuzzy Logic algorithm to provide more accurate predictions of potential flooding.

3.1 Equation Result of Fuzzy Logic EWS Implementation with MATLAB

The Mamdani method in the Early Warning System (EWS) for flood detection is used to improve accuracy in disaster detection, by utilizing ultrasonic sensors, wind speed sensors, and rainfall at the sending node (transmitter). Data from these sensors is sent to the receiver node and processed using the Mamdani fuzzy logic algorithm. The defuzzification process is carried out using the centroid (center of area) method, which aims to find the center point of the area under the fuzzy output curve. This technique calculates the ratio between the integral of the membership value multiplied by the output variable, and the integral of the membership value itself. The centroid method was chosen because it provides accurate and representative results. The stages start from fuzzification, which is the process of converting crisp input data into fuzzy input, with several variables used in this study.

1. Fuzzification

Fuzzification is a process where the input data of definite values (crisp input) is entered into fuzzy input. In this study, several variables were used.

Table 1. Fuzzification

Variables	Sets	Domains	Membership Function
Water Level	Low	[0 0 60 108]	Trapezoid
	Medium	[80 150 220]	Triangle
	High	[180 240 300 300]	Trapezoid
Wind Speed	Low	[0 0 6 12]	Trapezoid
	Medium	[10 12 14]	Triangle
	Fast	[12 22 50 50]	Trapezoid
Rainfall	Low	[0 0 12 25]	Trapezoid
	Medium	[15 35 55]	Triangle
	Heavy	[45 102 160]	Triangle
	Extreme	[150 175 180 180]	Trapezoid

Table 1 shows the fuzzification stage in the fuzzy system for flood detection, which includes three input variables: water level, wind speed, and rainfall. Each variable has several fuzzy sets, such as “low,” “medium,” and “high” for water level, “low” and “fast” for wind speed, and “low,” “medium,” “heavy,” and “extreme” for rainfall. Each set has a different domain with a trapezium or triangle-shaped membership function, which is used to convert the exact input data into fuzzy values. This process is essential for interpreting sensor data in a fuzzy logic-based flood detection system.

2. Implication Function Application

Tests that have been carried out, the Mamdani fuzzy logic system in the flood disaster EWS is able to integrate water level, wind speed, and rainfall data well, resulting in accurate predictions of safe, alert, or dangerous conditions. Using the centroid defuzzification method, this system can provide a more representative early warning according to the input from the sensor.

Table 2. Rule Fuzzy

No	Water Level	Wind Speed	Rainfall	Output
1	Low	Toned	Heavy	Safe
2	Low	Toned	Extreme	Safe
3	Medium	Low	Low	Safe
4	Medium	Low	Medium	Alert
5	Medium	Low	Heavy	Alert
6	Medium	Toned	Medium	Safe
7	Medium	Toned	Heavy	Alert
8	Medium	Toned	Extreme	Alert
9	High	Medium	Medium	Alert
10	High	Medium	Heavy	Danger
11	High	Medium	Extreme	Danger
12	High	Toned	Low	Alert

Based on the data in Table 2, the Mamdani fuzzy logic system successfully classifies flood conditions into three main categories: safe, alert, and danger, taking into account the variables of water level, wind speed, and rainfall. In conditions of low to moderate water levels with wind speed and rainfall that are also low or moderate, the system outputs “safe”. However, when the different variables show higher values, such as moderate or high water levels and heavy to extreme rainfall, the output changes to “alert” or “danger”. The system shows good capability in providing early warning based on the environmental conditions measured by the sensors.

3. Defuzzification

Defuzzification is converting the output fuzzy set to the form of firm numbers (crisp). In the Mamdani method using the Centroid (Center of area) calculation method in Equation 1, 2, and 3.

$$Output = \frac{Momentum}{Area} \quad (1)$$

$$Momentum = \int_a^b x \cdot \mu(z) \cdot dx \quad (2)$$

$$Area = \int_a^b \mu \cdot dx \quad (3)$$

Before performing defuzzification, the fuzzification process must be carried out first. This is because fuzzification changes clear (crisp) input data into fuzzy values that can be processed with fuzzy logic rules. After the fuzzy rules are applied and produce fuzzy values, then defuzzification is carried out to change the results back to crisp values that can be understood or applied in practical decisions. Thus, fuzzification is the initial step to prepare data in a form that can be processed using fuzzy logic, and defuzzification is the final step to change the results back into a useful form. Fuzzification of water level, wind speed, dan rainfall shown at Figure 7-9.

Case example: The implementation case in this case takes an example, When the water level is (150) cm, wind speed is (25) m/s, and rainfall is (60) ml/min what is the output result? The water level can calculated by equation 4,5, or 6. The wind speed can calculated by equation 7, 8, or 9. And the rainfall level can calculated by equation 10, 11, 12, or 13.

a. Fuzzification

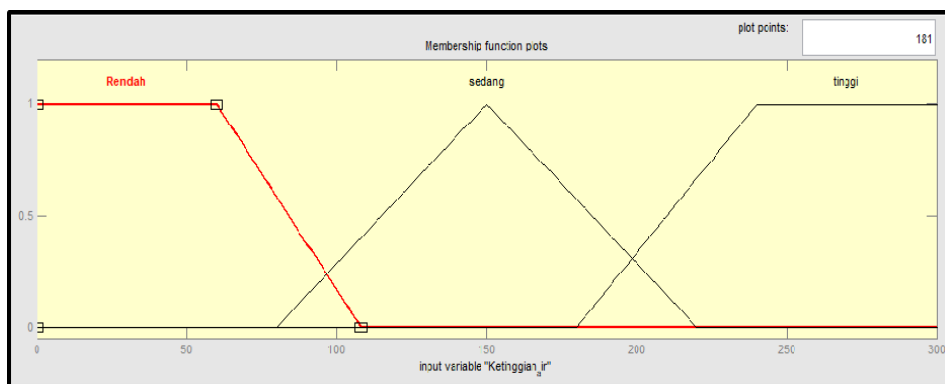


Figure 7. Water Level Fuzzification

Water Level

$$\text{Low} = 0 \quad (4)$$

$$\text{Medium} = \frac{x-a}{b-a} = \frac{150-80}{150-80} \quad (5)$$

$$\text{High} = 0 \quad (6)$$

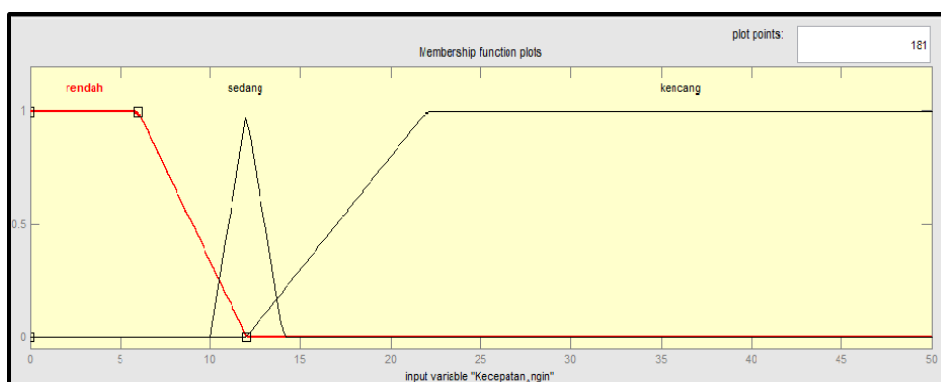


Figure 8. Wind Speed Fuzzification

Wind Speed

Low = 0

(7)

Medium = 0

(8)

Fast = 1

(9)

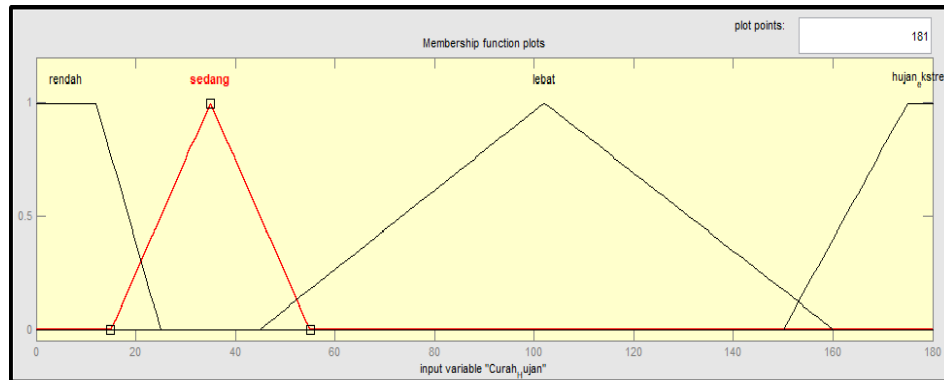


Figure 9. Rainfall Fuzzification

Rainfall

Low = 0

(10)

Medium = 0

(11)

Heavy = $\frac{x-a}{b-a} = \frac{60-45}{102-45} = 0.263$

(12)

Extreme = 0

(13)

b. Inference

After knowing the membership degree of each membership function, the next step is inference, which is taking the MIN output value of each membership function:

$$\begin{aligned} [R23] &= \text{water}(\text{medium}) \& \& \text{wind}(\text{fast}) \& \& \text{rainfall}(\text{heavy}) \text{ Then Caution.} \\ &= \text{MIN water}(1) \& \& \text{wind}(1) \& \& \text{rainfall}(0.263) \\ &= \text{MIN}(1, 1, 0.263) = 0.263 \end{aligned}$$

c. Defuzzification

In the Mamdani method, affirmation uses the centroid (center of area) method. Calculated by Equation 14-18. Supervised output range = 4 – 6.

$$\text{Centroid} = \frac{\text{Momentum}}{\text{Area}} \quad (14)$$

$$\text{Momentum} = \int_a^b x \cdot 0.263 \, dx \quad (15)$$

$$= 0.263 \left(\frac{1}{2} \right)_4^6 = 0.263 \left(\frac{1}{2} 6^2 - \frac{1}{2} 4^2 \right) = 0.263 \left(\frac{36}{2} - \frac{16}{2} \right) = 2.63$$

$$\text{Area} = \int_4^6 0.263 \, dx = 0.263 [x]_4^6 = 0.263 (6 - 4) = 0.526 \quad (16)$$

$$\text{Output} = \frac{\text{Momentum}}{\text{Area}} \quad (17)$$

$$= \frac{2.63}{0.526} \quad (18)$$

= 5 (Manual calculation result)

= 5 (Arduino program result)

= 5 (Matlab result)

From this comparison, it can be concluded that manual calculations cannot be used as a reference because the difference between rule 1 and the others has almost the same difference. In the Arduino program results, a program is made that references calculations based on matlab. Thus, the Arduino generated program can be classified to distinguish the output shown in Figure 10-12.

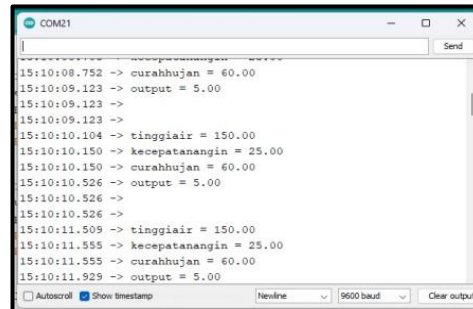


Figure 10. Results on Arduino IDE serial monitor

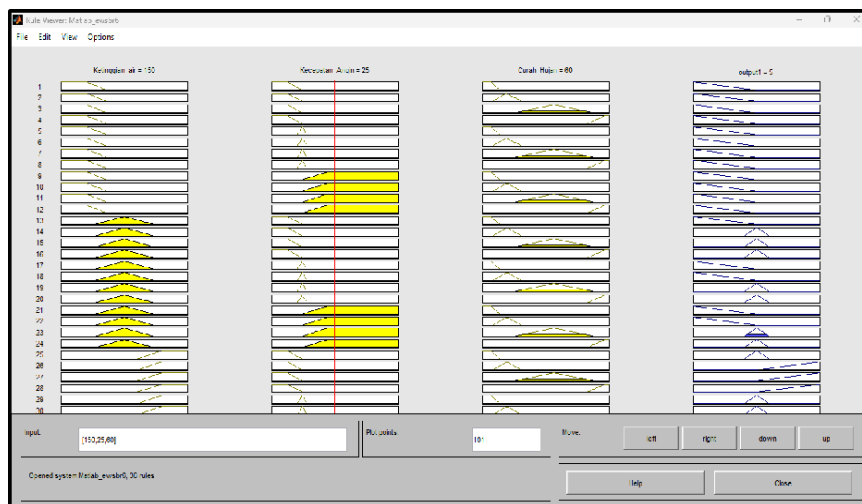


Figure 11. Results in MATLAB

From Table 2, there are 36 rows indicating there are 36 rules and there are 4 columns consisting of 3 fuzzy inputs and 1 fuzzy output from each rule made. In the input column there is a red line indicating the value of the input rule, and there is the oldest color among the outputs that is the appropriate rule from the input that has been determined in the lower input column, the mark on each output rule indicates the membership of the resulting output.

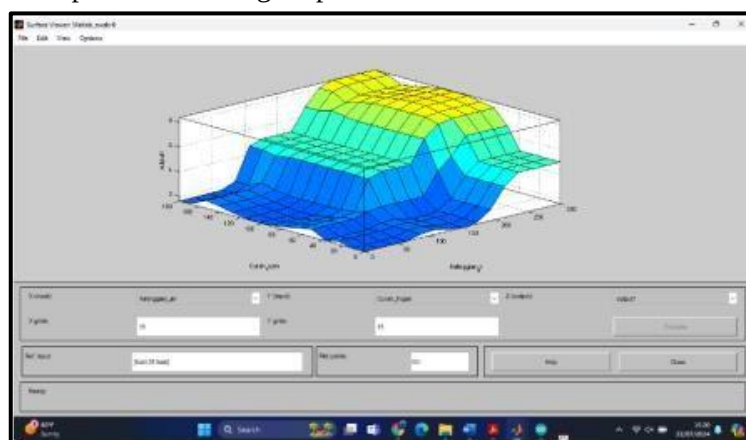


Figure 12. Surface graph in MATLAB

From Figure 12, the surface graph in MATLAB shows that rainfall and water level are very influential in flood disasters, because the graph shows that when the water level increases and rainfall increases, the fuzzy logic output is also higher, while wind speed is not very influential in the occurrence of flood disasters.

3.2 Fuzzy Logic Algorithm Testing with MATLAB

The EWS that has been designed and has been completed is then tested and compared to the results of the Fuzzy Logic Algorithm with simulations in MATLAB. The results of the Fuzzy Algorithm Test can be seen in the test images between EWS and MATLAB in attachments 2 and 3. From these 11 tests it can be concluded that there is a difference of 0%, so the Fuzzy Logic Algorithm on EWS is very good.

4. CONCLUSION

Rainfall and water level greatly affect the results of flood detection, while wind speed is less important while wind speed is not very influential in flood detection influence in flood disaster detection. This can be known from the rules and results of the Fuzzy Algorithm. The Fuzzy Logic algorithm that has been carried out and the results are very accurate with 11 data that has been tested with MATLAB software, while on the website when the water level is 157, the wind speed is 0, and the rainfall is 54.40 has the same fuzzification value as the MATLAB software with a difference of 0 by 0.

REFERENCES

- [1] I. Suhardjo, "Operasi Pintu Air Bendung Gerak Serayu Dalam Usaha Pengendalian Banjir Dan Irigasi," *Teodolita (Media Komun. Ilm. di Bid. ...)*, pp. 1–11, 2011, [Online]. Available: <http://ejournal.unwiku.ac.id/teknik/index.php/JT/article/download/73/59>
- [2] I. N. W. Satiawan, I. B. F. Citarsa, and others, "Desain Buck Converter Untuk Charging Baterai Pada Beban Bervariasi: Buck Converter Design For Battery Charging On Various Loads," *DIELEKTRIKA*, vol. 5, no. 1, pp. 30–35, 2018.
- [3] ASEAN Coordinating Centre for Humanitarian Assistance, "Indonesia, Flooding in Cilacap Regency, Central Java (21 Jul 2021)," 2021. <https://reliefweb.int/report/indonesia/indonesia-flooding-cilacap-regency-central-java-21-jul-2021>
- [4] I. W. Harmoko and B. Ruslana, Zauyik Nana (Stasiun Klimatologi Klas I Semarang, "Analisis Singkat Kejadian Banjir Di Kabupaten Cilacap (Tanggal 21 Juli 2021)," 2021.
- [5] Asian Disaster Information Network, "Indonesia, Flooding in Cilacap Regency, Central Java." [Online]. Available: <https://adinet.ahacentre.org/report/indonesia-flooding-in-cilacap-regency-central-java-20210721>
- [6] I. Nozomi, "Penerapan Data Mining Untuk Peringatan Dini Banjir Menggunakan Metode Klastering K-Means (Studi Kasus Kota Padang)," *J. Sains Inform. Terap.*, vol. 2, no. 2, pp. 39–44, 2023, doi: 10.62357/jsit.v2i2.165.
- [7] H. P. Adi, I. Wahyudi, and F. (Universitas I. S. A. Ni'am, "Model Mitigasi Dan Penanganan Banjir Air Pasang Laut Untuk Ketahanan Pangan Dan Permukiman (Studi Kasus Di Kecamatan Kawunganten, Kabupaten Cilacap)," 2019.
- [8] E. Alimudin, A. Sumardiono, and Zaenurrohman, "Sistem Peringatan Dini Bencana Banjir Menggunakan Multi Sensor," S00202212254, 2022
- [9] F. R. Usman, W. Ridwan, and I. Z. Nasibu, "Sistem Peringatan Dini Bencana Banjir Berbasis Mikrokontroler Arduino," *Jambura J. Electr. Electron. Eng.*, vol. 1, no. 1, pp. 1–6, 2019, doi: 10.37905/jjee.v1i1.2721.
- [10] R. A. Priatim, M. Asri, and S. Abdussamad, "Rancang Bangun Prototipe Peringatan Dini Banjir Menggunakan Raspberry Pi Berbasis IoT," *Jambura J. Electr. Electron. Eng.*, vol. 5, no. 2, pp.

216–221, 2023, doi: 10.37905/jjee.v5i2.19696.

- [11] L. Soulibouth, H. S. Hwang, and D. H. Shin, “The Impact of Flood Damage on Farmers, Agricultural Sector and Food Security in Laos: A Regional Case Study of Champhone District, Savannaket Province,” *J. Int. Dev. Coop.*, vol. 16, no. 2, pp. 151–170, 2021, doi: 10.34225/jidc.2021.16.2.151.
- [12] A. Nurdianto, D. Notosudjono, and H. Soebagia, “Rancang bangun sistem peringatan dini banjir (early warning system) terintegrasi internet of things,” *J. Online Mhs. Bid. Tek. Elektro*, vol. 01, pp. 1–10, 2018.