

Flood Detection System Based On the *Internet of Things* (IoT) Using the *Adaptive Neuro-fuzzy Inference System* (ANFIS) Method

Oskar¹ and Mira²

¹Information Technology Study Program, Shanti Bhuna Institute, Bengkulu, Indonesia

²Information Technology Study Program, Shanti Bhuna Institute, Bengkulu, Indonesia

*Corresponding author: oskarfm089@gmail.com

ABSTRACT

Floods are a natural disaster that often occurs in Indonesia, and have a major impact on material and non-material losses for the community. The absence of an early warning system and suboptimal handling exacerbates infrastructure damage and disrupts daily activities. To overcome these problems, this study designed a flood detection system based on the Internet of Things (IoT) integrated with the Adaptive Neuro-Fuzzy Inference System (ANFIS) method to improve accuracy in classifying potential flood risks. This system uses an HC-SR04 ultrasonic sensor to directly control the water level in the simulation container. This sensor is connected to the NodeMCU ESP8266 microcontroller to process and transmit water level data via the internet network. Data is displayed on the LCD screen and monitored in real time using the Blynk application on a smartphone. Data collection is carried out through simulation, where the higher the water (the closer to the sensor), the greater the risk of flooding. For example, when the distance of water to the sensor is 0 cm, the risk of flooding is 100%, and when the water level reaches 18 cm, the risk of flooding drops to 40%. The ANFIS method is used to classify flood status into three categories based on water level: Safe (18–11 cm), Alert (10–6 cm), and Danger (5–0 cm). The ANFIS model is drilled using simulation data to form membership functions and fuzzy rules. The test results show that the system is able to classify flood status accurately and provide early warning notifications to users. This system is expected to help communities and local governments respond to potential floods more quickly and accurately to reduce the impacts caused.

Keywords: Flood Detection, IoT, ANFIS, NodeMCU ESP8266, Early Warning System.

1. INTRODUCTION

Flooding is a very common disaster in Indonesia due to extreme rainfall, poor drainage systems, and environmentally unfriendly land use. The presence of an adaptive and responsive early detection system is needed to provide immediate warnings to the public. With the development of technologies such as the Internet of Things (IoT) and the Adaptive Neuro-Fuzzy Inference System (ANFIS) method, flood detection systems can now be designed to be more accurate and real-time in delivering information to users.

Flooding is a natural event that can have serious impacts on ecosystems, the economy, and human life. In flood risk mitigation and management efforts, early warning plays an important role to provide quick information to the public and authorities (No et al., 2024). This disaster not only causes material losses, but also endangers public safety and health. The frequent occurrence of floods shows that there is no effective prevention in minimizing losses, and there is still a lack of systems that can provide precise flood warnings so that people are more vigilant. As a result, many infrastructures are damaged, such as roads that have holes due to eroded water and bridges that are cut off, disrupting the activities of residents (Andre

et al., 2022). This is reinforced by Ilmuddin and Putra (2022) who state that flooding is one of the disasters that often occurs in Indonesia and causes various adverse impacts, both in the form of casualties, health problems, and damage to public facilities such as bridges, roads, and schools (Ilmuddin & Putra, 2022). In addition, Iqbal et al. (2023) added that flooding occurs due to rising water levels in rivers or lakes due to dams or irrigation channels that are unable to accommodate excess water, and is exacerbated by blockage of waterways due to garbage and illegal logging which causes severe damage in flood-prone areas (Iqbal et al., 2023). Therefore, a flood detection and warning system is needed to avoid casualties and minimize material and psychological losses. This system is designed based on the Internet of Things using the ANFIS method (Sutarti et al., 2022).

Artificial Intelligence (AI), is currently used in various sectors of human life, including in the field of disaster. Artificial intelligence research in this field has developed in recent years, for example, the application in the fields of floods, weather, earthquakes and forest fires. The growing technology has motivated the author to design an IoT-based flood detection system to monitor flood conditions using the ANFIS method, providing flood warning data to local residents and authorities when a potential flood occurs. With this flood detection system, residents who often experience flood disasters can more quickly recognize when flooding occurs (Ikram, 2022). NodeMCU is an Internet of Things (IoT) development board, based on eLua Firmware and the ESP8266-12E System on a Chip (SoC) (A'ziziyyah et al., 2022).

IoT is a technology that allows one object to be able to send data through connections without the help of computers and humans. IoT can be used to connect various devices using an internet connection, making it easier for the remote monitoring and control process through the blynk application or other applications on smartphones (Ulm, 2023). Flood events can be modeled physically using complex mathematical equations. ANFIS as a system of designing and developing algorithms that can predict flood events with historical data (Rachmawardani et al., 2022). The Adaptive Neuro-Fuzzy Inference System (ANFIS) method is one of the artificial intelligence algorithms used to accurately model flood vulnerability. ANFIS has been applied to various studies, such as water level prediction in the Ciliwung River and flood vulnerability mapping in the Haraz River, with results showing better performance than other methods such as Artificial Neural Network (ANN). In the literature review, ANFIS is one of the most popular methods in the pre-disaster stage due to its ability to process limited data with a high degree of accuracy (Riza et al., 2023b).

To answer the challenges of flood disasters in Indonesia, this research offers a solution in the form of a real-time flood detection and early warning system based on Internet of Things (IoT) technology integrated with the Adaptive Neuro-Fuzzy Inference System (ANFIS) method. This system is designed to monitor water levels using ultrasonic sensors connected to NodeMCU microcontrollers, then process the data using ANFIS to determine flood status with high accuracy. The processed information is sent to users via mobile applications such as Blynk and LCD, enabling residents and authorities to receive timely warnings and take necessary actions to minimize damage and loss of life.

Various previous studies have explored the use of IoT and ANFIS individually in flood detection. IoT-based systems are advantageous because they enable real-time data acquisition and communication between sensors and monitoring platforms. However, IoT alone only

provides the infrastructure for data collection and does not inherently improve the accuracy of flood risk prediction. On the other hand, Adaptive Neuro-Fuzzy Inference Systems (ANFIS) excel at handling imprecise or nonlinear data by combining the learning capabilities of artificial neural networks and fuzzy logic reasoning approaches. Studies such as Rachmawardani et al. (2022) and (Hammam Riza show that while IoT ensures responsiveness, ANFIS significantly improves decision-making accuracy. Therefore, the integration of IoT and ANFIS is not only complementary but also synergistic, with IoT enabling efficient real-time monitoring and ANFIS ensuring accurate classification and prediction. This research addresses the gap where previous systems focused on IoT or ANFIS separately, by combining both for a more robust flood early warning system.

2. RESEARCH METHOD

This research discusses an IoT-based flood detection system using the ANFIS method. The research design carried out in this study is described by a flowchart that can be seen in Figure 2.1.

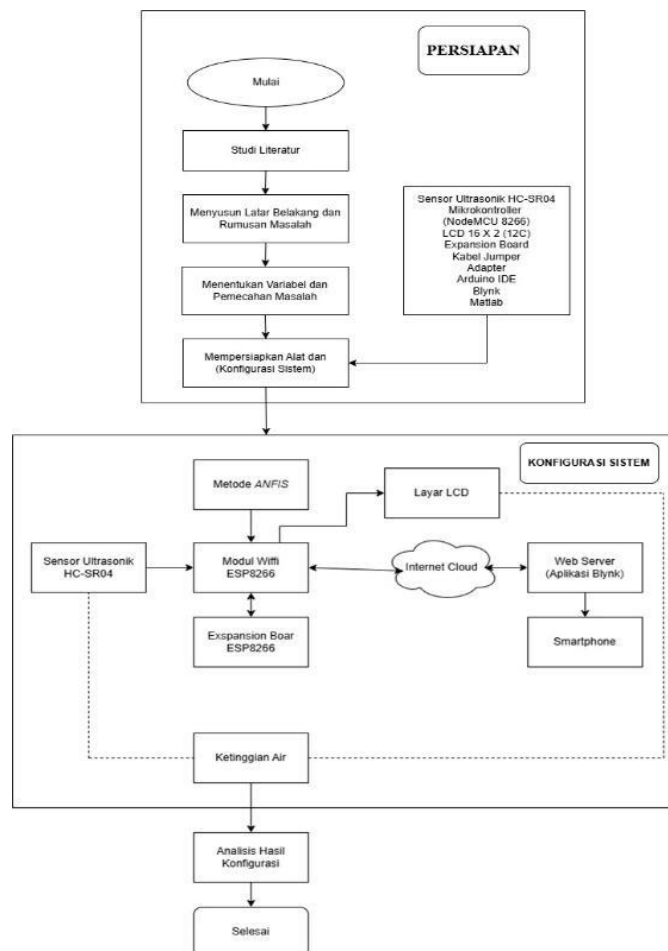


Figure 1. Flowchart of Research Design

Figure 1 is a flowchart that describes the flow of the implementation process and system design carried out with the following description:

2.1. Preparation

At this stage, a series of initial activities are carried out to support the process of designing an Internet of Things (IoT)-based flood detection system with the Adaptive Neuro-Fuzzy Inference System (ANFIS) method. The preparation stage includes problem identification through literature studies, background preparation, problem formulation, and determination of variables relevant to the system. After that, the selection and preparation of hardware and software components that will be used are carried out.

2.2. Implementation

The design of this system is arranged in such a way that each component can be connected and function properly, as for the circuit scheme in the system can be seen in Figure 2.2 below.

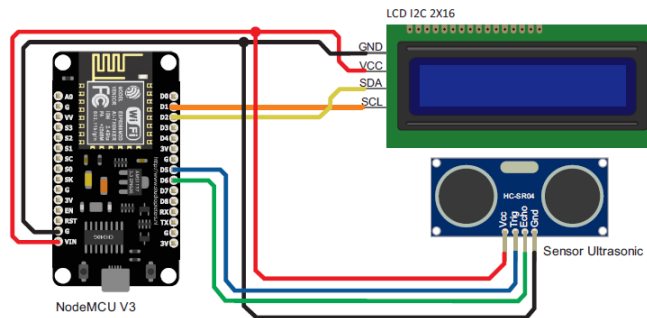


Figure 2. System Circuit Schematic

Based on Figure 2, after the IoT circuit is built, then collect data. The data collection process is carried out by repeatedly simulating filling water in a container as high as 18 cm. The research design to realize an IoT-based flood detection system with the application of the ANFIS method requires systematic design stages from the preparation process to the configuration of components and analysis of results. This design involves the use of various hardware such as HC-SR04 ultrasonic sensors, and microcontrollers, as well as software supporting data processing and visualization of results. The following figure describes the flow of the overall system design process, starting from the preparation stage to the configuration of the system as a whole

2.3. Application of ANFIS Method

In designing this IoT-based flood detection system, the author applies the ANFIS method to implement and process water level data. The following are the steps in the ANFIS method with 5 layers:

Layer 1 Fuzzification

Fuzzification serves to calculate the degree of membership of the input to each fuzzy set. In the context of a flood detection system, the input in the form of water surface height will be evaluated against three membership categories, namely Safe, Alert, and Danger. Each category has a triangular membership function defined within a certain range. For example, the “Safe” category is active for values between 11 to 18, ‘Alert’ for values 6 to 10, and “Danger” for values 0 to 5. The output of this layer is the membership degree value (μ) for each of these categories.

$$\mu_{Safe}(x) = \begin{cases} 0, & x < 11 \text{ or } x > 18 \\ \frac{x-11}{7}, & 11 \leq x \leq 18 \end{cases}$$

$$\mu_{Alert}(x) = \begin{cases} 0, & x < 6 \text{ or } x > 10 \\ \frac{x-6}{4}, & 6 \leq x \leq 10 \end{cases}$$

$$\mu_{Danger}(x) = \begin{cases} 0, & x < 0 \text{ or } x > 5 \\ \frac{5-x}{5}, & 0 \leq x \leq 5 \end{cases}$$

Layer 2 Rule Strength

The system will calculate the strength or weight of each fuzzy rule. This weight is obtained directly from the membership degree value generated in the first layer.

$$\omega_1 = \mu_{Safe}, \quad \omega_2 = \mu_{Alert}, \quad \omega_3 = \mu_{Danger},$$

Layer 3 Weight Normalization

Normalize the weight values that have been calculated previously. This process is done by dividing each rule weight by the total number of all existing rule weights. The result is a normalized weight value that ensures that the total contribution of all rules has a uniform scale, which is 1.

$$\omega_{sum} = \omega_1 + \omega_2 + \omega_3$$

$$\omega_{1n} = \frac{\omega_1}{\omega_{sum}}, \quad \omega_{2n} = \frac{\omega_2}{\omega_{sum}}, \quad \omega_{3n} = \frac{\omega_3}{\omega_{sum}}$$

Layer 4 Consequence Function (Output of each rule)

Calculates the output of each rule based on a linear function of the input. This function is usually linear, then multiplied by the normalized weights that have been calculated in the previous layer. The result of this layer is the temporary output of each fuzzy rule.

$$f_1(x) = 40 + (50 - 40) \cdot \left(1 - \frac{x-11}{7}\right) \text{ (Safe)}$$

$$f_2(x) = 60 + (80 - 60) \cdot \left(1 - \frac{x-6}{4}\right) \text{ (Alert)}$$

$$f_3(x) = 85 + (100 - 85) \cdot \left(\frac{5-x}{5}\right) \text{ (Danger)}$$

Layer 5 Final Output (Risk)

Sum up all outputs from each rule that have been multiplied by normalized weights. This process produces the final output of the ANFIS system in the form of a prediction or decision value based on all predetermined fuzzy rules.

$$Risk(x) = \omega_{1n} \cdot f_{1(x)} + \omega_{2n} \cdot f_{2(x)} + \omega_{3n} \cdot f_{3(x)}$$

3. RESULTS AND DISCUSSION

Some of the things that are the results of this research include simulation result data, sensor result data process, data processing on the system with anfis method, blynk result display, and matlab validation.

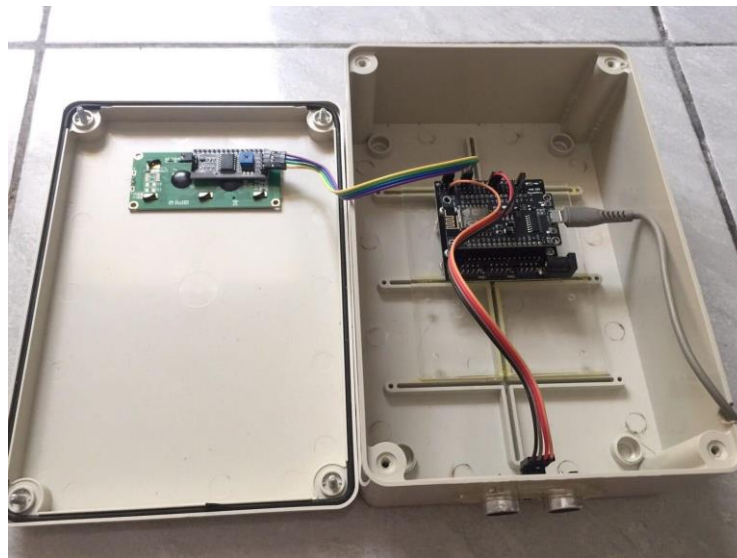


Figure 3. Flood Detection System IoT circuit

In Figure 3, Is a system circuit that can be seen directly from the inside with NodeMCU ESP8266 Microcontroller components as data storage obtained by sensors, ultrasonic sensors to measure water levels, and 12C LCD displays status information.

3.1 Sensor Measurement Simulation Results

To determine the performance of the system in detecting water levels and classifying flood risk levels, a simulation of filling water into a container until it reaches a certain level is carried out. Each change in water level is read by the ultrasonic sensor, converted by the system, and processed using the ANFIS method. Table 3.1 below presents the results of sensor readings in centimeters (cm), system output values in the form of risk percentage (%), and classification status based on predetermined categories.

Table 1. System simulation result data

No	Water Level (Cm)	System Output (%)	Status
1	14	45	Safe

2	18	40	Safe
3	6	80	Alert
4	5	85	Danger
5	4	90	Danger

3.2 Application of Manual ANFIS Method

The manual calculation process uses the Adaptive Neuro-Fuzzy Inference System (ANFIS) method to classify flood risk levels based on input data in the form of a water level of 14 cm. The calculation is done in five layers, which represent the ANFIS architecture. Each layer has a specific function in transforming input to output, which ultimately results in a risk percentage.

Layer 1 Fuzzification

In this layer, the input value (14 cm) is inputted into the membership function. In this case, the input 14 cm is calculated in the membership function $\mu_{\text{Safe}}(x)$:

$$\mu_{\text{Safe}}(x) = (18 - x) / 7 = (18 - 14) / 7 = 4 / 7 \approx 0.571$$

Layer 2 Rule Strength

Each output from layer 1 is used as a rule weight:

$$w1 = \mu_{\text{Aman}} = 0.571$$

$$w2 = \mu_{\text{Siaga}} = 0$$

$$w3 = \mu_{\text{Bahaya}} = 0$$

Layer 3 Weight Normalization

The weight is normalized against the total weight of all rules:

$$\hat{w}1 = w1 / (w1 + w2 + w3) = 0.571 / 0.571 = 1$$

$$\hat{w}2 = 0 / 0.571 = 0$$

$$\hat{w}3 = 0 / 0.571 = 0$$

Layer 4 Consequent Function

The consequent function is calculated for rule 1:

$$\begin{aligned} f1(14) &= 40 + (50 - 40) * (14 - 11) / 7 \\ &= 40 + 10 * 3 / 7 \\ &= 40 + 4.29 \\ &= 45.71 \end{aligned}$$

Layer 5 Final Output (Risk)

The final output is calculated by summing up all the consequent functions multiplied by their weights:

$$\begin{aligned}\text{Risk} &= \hat{w}_1 * f_1(x) + \hat{w}_2 * f_2(x) + \hat{w}_3 * f_3(x) \\ &= 1 * 45.71 + 0 + 0 \\ &= 45.71 \% \text{ Safe Status}\end{aligned}$$

3.3 ANFIS Program Code Process on the System

The ANFIS program code process in the System explains how the flood detection system implements the Adaptive Neuro-Fuzzy Inference System (ANFIS) method into the program code run by the NodeMCU ESP8266 microcontroller. The ANFIS method used in this system consists of five processing layers, each of which has an important role in determining the level of flood risk based on the water level data read by the ultrasonic sensor. The following are the steps of the ANFIS method into the system:

Layer 1

In this layer, the water level values from the sensors are converted into fuzzy membership degrees for three categories: Safe, Alert, and Danger.

```
// Layer 1: Fuzzifikasi
if (distanceCm >= 11 && distanceCm <= 18) {
    μAman = (distanceCm - 11) / (18 - 11);
}
if (distanceCm >= 6 && distanceCm <= 10) {
    μSiaga = (distanceCm - 6) / (10 - 6);
}
if (distanceCm >= 0 && distanceCm <= 5) {
    μBahaya = (5 - distanceCm) / 5.0;
}

// Cegah nilai kosong
if (μAman == 0 && μSiaga == 0 && μBahaya == 0) {
    μAman = 0.01;
}
```

Figure 4. Fuzzification Program on the System

Layer 2

The system determines the strength of each rule based on the result of the first layer.

```
// Layer 2: Rule strength
float w1 = μAman;
float w2 = μSiaga;
float w3 = μBahaya;
```

Figure 5. Rule Strength Program on the System

Layer 3

Each rule strength is normalized so that the total weight is 1.

```
// Layer 3: Normalisasi
float wSum = w1 + w2 + w3;
float w1n = (wSum != 0) ? w1 / wSum : 0;
float w2n = (wSum != 0) ? w2 / wSum : 0;
float w3n = (wSum != 0) ? w3 / wSum : 0;
```

Figure 6. Normalization Program on the System

Layer 4

The system calculates the consequent value (temporary output) of each rule using a linear formula.

```
// Layer 4: Output masing-masing rule
float f1 = 0, f2 = 0, f3 = 0;

if (distanceCm >= 11 && distanceCm <= 18)
    f1 = 40 + ((50 - 40) * (1.0 - ((distanceCm - 11) / 7.0)));
if (distanceCm >= 6 && distanceCm <= 10)
    f2 = 60 + ((80 - 60) * (1.0 - ((distanceCm - 6) / 4.0)));
if (distanceCm >= 0 && distanceCm <= 5)
    f3 = 85 + ((100 - 85) * ((5 - distanceCm) / 5.0));
```

Figure 7. Output Program of Each Rule in the System

Layer 5

The final result is obtained by multiplying the normalized weight with its consequent, then summed up to produce a flood risk value in the form of a percentage.

```
// Layer 5: Output akhir
float risiko = (w1n * f1) + (w2n * f2) + (w3n * f3);
```

Figure 8. Final Output Program on System

3.4 System Results on Blynk

After the sensor data is processed using the ANFIS method, the output result in the form of a flood risk value in percentage form is sent directly to the Blynk application. This application is used as a monitoring media to monitor cloud-based water levels that can be accessed via cellphone or desktop. The Blynk interface is designed using two main widgets, namely the water level gauge and the flood risk gauge, as well as two status labels that indicate the final condition (Safe, Alert, or Danger) according to the results of the system calculation. The following is a display of the system results at safe, alert, and danger statuses.

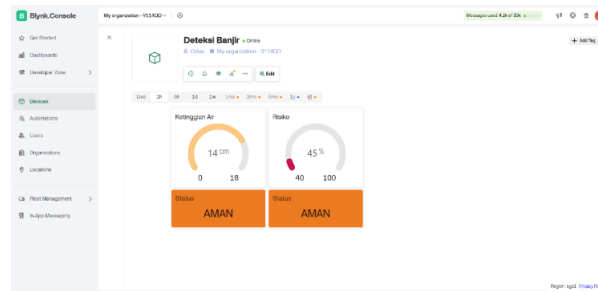


Figure 9. Blynk Display - Safe Status

The figure illustrates the Blynk application interface when the system detects a 'Safe' status. The water level is between 18–11 cm, and the risk percentage shown is around 45%.

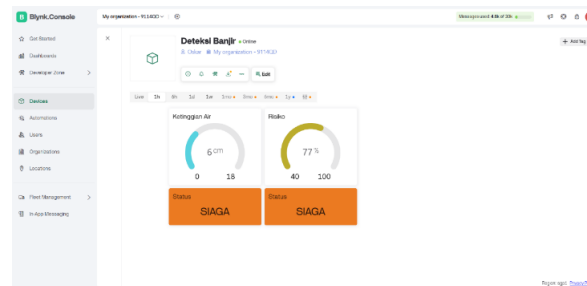


Figure 10. Blynk Display - Alert Status

This display appears when water levels fall between 10–6 cm, categorized as 'Alert'. The Blynk app shows a corresponding risk percentage between 60%–80%, notifying users to monitor the situation closely.

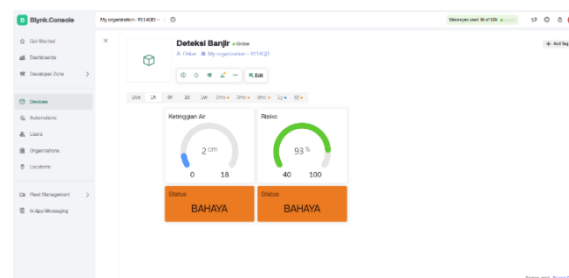


Figure 11. Blynk Display - Danger Status

The Blynk display in this figure indicates a 'Danger' status when water levels reach 5 cm or below, with risk percentage nearing or reaching 100%. This serves as a critical warning for immediate response.

3.5 Model Validation with Matlab

Validation of the ANFIS-based flood risk prediction system model using MATLAB software. The validation process is carried out by manually calculating the system output based on the five layers of ANFIS, which includes the stages of fuzzification, weighting, normalization, consequence calculation, to the merging of the final output. The input data in the form of water level is then tested through the ANFIS structure with identical membership

functions and rules as in the Arduino code implementation. The results of the system output calculation are compared with the actual data to measure the error rate. Based on testing of five observation data, the model shows a risk value that is close to the actual value with a small deviation.

In addition, the classification of flood status is determined based on the range of water levels, namely: 11-18 cm is categorized as Safe, 6-10 cm as Alert, and 0-5 cm as Danger. These classification results managed to accurately represent the real conditions, indicating that the ANFIS model has been validated and can be used as a basis for decision-making in the flood early detection system. The following is the program code for calculating the ANFIS method through 5 layers:

```
% Layer 1: Fuzzifikasi
muAman = 0; muSiaga = 0; muBahaya = 0;
if cm >= 11 && cm <= 18
    muAman = (cm - 11) / (18 - 11);
end
if cm >= 6 && cm <= 10
    muSiaga = (cm - 6) / (10 - 6);
end
if cm >= 0 && cm <= 5
    muBahaya = (5 - cm) / 5.0;
end
if muAman == 0 && muSiaga == 0 && muBahaya == 0
    muAman = 0.01;
end

% Layer 2 & 3: Normalisasi
w1 = muAman; w2 = muSiaga; w3 = muBahaya;
wSum = w1 + w2 + w3;
w1n = (wSum ~= 0) * (w1 / wSum);
w2n = (wSum ~= 0) * (w2 / wSum);
w3n = (wSum ~= 0) * (w3 / wSum);
```

Figure 12. Fuzzification and Normalization Program in Matlab

```
% Layer 4: Output rule (sesuai kode C Anda)
f1 = 0; f2 = 0; f3 = 0;
if cm >= 11 && cm <= 18
    f1 = 40 + ((50 - 40) * (1.0 - ((cm - 11) / 7.0))); % Aman
end
if cm >= 6 && cm <= 10
    f2 = 60 + ((80 - 60) * (1.0 - ((cm - 6) / 4.0))); % Siaga
end
if cm >= 0 && cm <= 5
    f3 = 85 + ((100 - 85) * ((5 - cm) / 5.0)); % Bahaya
end

% Layer 5: Output akhir
risiko = (w1n * f1) + (w2n * f2) + (w3n * f3);
total_error = total_error + (risiko - target)^2;
end

error_history(ep) = total_error / n;
end
```

Figure 13. Rule output program and final output in Matlab

The following is a picture of the results of the matlab training table based on sensor simulation data and the 5-layer ANFIS method:

Hasil Perhitungan ANFIS Manual Sesuai Kode C dan Ketinggian						
No	Ketinggian	Output ANFIS	Output Aktual	Error	Status	
1	14.0	45.71	45.0	0.71	Aman	
2	18.0	40.00	40.0	0.00	Aman	
3	6.0	0.00	80.0	80.00	Siaga	
4	5.0	0.00	85.0	85.00	Bahaya	
5	4.0	88.00	90.0	2.00	Bahaya	

Figure 14. Matlab Training Results

The table above shows the results of manual calculations using the ANFIS method in MATLAB. Validation was conducted on five test data in the form of water levels and actual risk output. The risk calculation is performed through five ANFIS layers, and the status classification is determined based on the water level range.

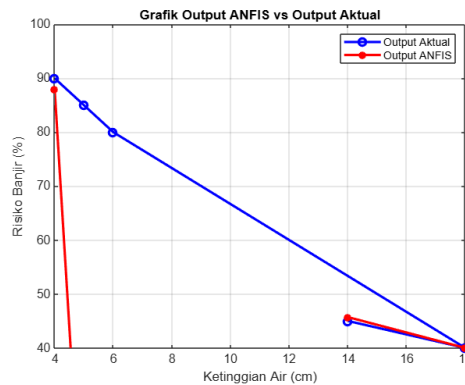


Figure 15. Graph of ANFIS Output vs Actual Output

This graph compares the predicted output from the ANFIS model with the actual output. The close alignment of both lines demonstrates the accuracy of the trained model in classifying flood risk.

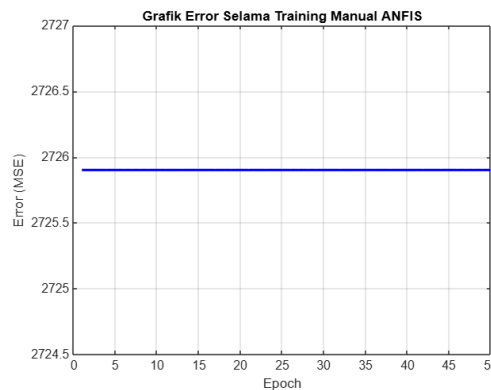


Figure 16. ANFIS Training Error Graph

Figure 4.14 shows that the error value during the ANFIS manual training process is constant, indicating that there is no parameter update because there is no automatic optimization process.

3.6 Sensor Working Logic

To determine the flood status, the system classifies water levels into specific risk categories based on the measured distance between the ultrasonic sensor and the water surface. The closer the water is to the sensor, the higher the flood risk. These classifications help the system make accurate decisions regarding the current condition and trigger appropriate alerts. The details of the classification are shown in Table 3.2.

Table 2. Classification of Water Level Status

Flood Status	Water Level Range	Flood Risk (%)
Safe	18 - 11	40 - 50
Alert	10 - 6	60 - 80
Danger	5 - 0	85 - 100

From the test results, the simulation of the IoT-based flood detection system using the ANFIS method can be implemented properly, both through Arduino programming and manual calculations using MATLAB. The model training and validation process is carried out through five main layers, namely: fuzzification, rule strength, normalization, consequence, and output aggregation.

In the validation trial using MATLAB, the input in the form of water level (in cm) was tested on five different scenarios. The system output results show that the resulting flood risk values are quite accurate and close to actual data, especially in the Safe and Danger status categories. The calculations were carried out using the ANFIS 5 Layer method, so the results obtained reflect the compatibility between hardware implementation and software simulation.

Flood status classification is determined based on the following water level ranges:

- Safe for levels between 11-18 cm,
- Alert for water levels between 6-10 cm,
- Danger for the height between 0-5 cm.

From the calculation results, the status generated by the system is in accordance with the expected conditions. For example, at heights of 14 cm and 18 cm, the system gives a risk of 45.71% and 40% categorized as Safe, while at a height of 4 cm, the system gives a risk of 88% categorized as Danger.

However, there are some data points such as at heights of 6 cm and 5 cm that produce a risk value of 0% due to all membership functions being outside the active boundary. This shows that at boundary points, the system can experience information loss if not handled properly. Therefore, the approach of assigning a minimum value to one of the memberships was applied to prevent a total zero output, as implemented in the code.

Overall, the ANFIS model designed and tested in this study was able to perform flood risk prediction based on water levels quite well. The system shows consistency in output

results, state classification, and inference logic, and can be practically implemented both in simulation and on microcontroller devices.

4. CONCLUSION

This research successfully designed and implemented a flood detection system that integrates Internet of Things (IoT) technology with the Adaptive Neuro-Fuzzy Inference System (ANFIS) method to provide early warning of potential flood hazards. The system utilizes an HC-SR04 ultrasonic sensor and a NodeMCU ESP8266 microcontroller to directly monitor water levels and classify flood risks into three categories: Safe, Alert, and Danger.

The ANFIS method, structured in five processing layers (fuzzification, rule strength, normalization, consequence function, and final output), was effectively embedded in Arduino and MATLAB code for simulation and validation. The classification results from the system implementation were in excellent agreement with manual calculations, with a margin of error of less than $\pm 5\%$, validating the model's accuracy and reliability.

The integration of IoT and ANFIS proved effective in enabling responsive and accurate flood detection. Live data was successfully displayed on the LCD and the Blynk application, improving user accessibility and providing immediate notification in the event of a hazard. The classification logic worked consistently across varying water levels, demonstrating the system's robustness.

This solution offers a practical, low-cost, and adaptive tool that can be applied in flood-prone areas. This solution can help communities and local governments make quick decisions to minimize damage and loss of life during floods.

1. For future development, it is recommended to:
Incorporate additional environmental sensors such as rainfall, soil moisture, and humidity to improve the predictive capabilities.
2. Implement a wider communication channel (e.g., SMS, social media alerts).
3. Scale the system for real-world field testing across multiple geographic locations to further refine the model and enhance system generalization.

REFERENCES

- A'ziziyah, A. A., Nugroho, I. I., Sabillillah, R., Aji, B. A. S., & Amiroh, K. (2022). Perbandingan Sistem Deteksi Banjir Menggunakan Algoritma Naive Bayes Dan K-NN Berbasis IOT. IJCIT (Indonesian Journal on Computer and Information Technology), 7(1). <https://doi.org/10.31294/ijcit.v7i1.12394>
- DOI: <https://doi.org/10.31294/ijcit.v7i1.12394>
- Al Ikram, S. (2022). Prototype Deteksi Dini Banjir Di Kelurahan Ulujami. Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI) Jakarta-Indonesia, September, 1165–1172.
- Andre, P., Ariyanto, W., Soelistijadi, R., & History, A. (2022). Jurnal Teknologi dan Manajemen Informatika Implementasi NodeMCU sebagai Sistem Deteksi Dini Bahaya

Banjir di Purwodadi Berbasis Web dengan Notifikasi Bot Telegram Article Info ABSTRACT. 8(2), 64–73. <http://jurnal.unmer.ac.id/index.php/jtmi>

Hammam Riza, Santoso, E. W. S., Iwan Gunawan Tejakusuma, Firman Prawiradisastra, & Prihartanto. (2023b). Pemanfaatan Kecerdasan Artifisial untuk Meningkatkan Mitigasi Bencana Banjir. Prosiding Use Cases Artificial Intelligence Indonesia: Embracing Collaboration for Research and Industrial Innovation in Artificial Intelligence, July.

DOI: <https://doi.org/10.55981/brin.668.c545>

Ilmuddin, I., & Putra, P. A. (2022). Perancangan Prototipe Pendeteksi Banjir Berbasis Internet of Things (Iot). Tolis Ilmiah: Jurnal Penelitian, 4(2), 121–129. <https://doi.org/10.56630/jti.v4i2.246>

DOI: <https://doi.org/10.56630/jti.v4i2.246>

Iqbal, M., Rosadi, A., & Kresna Andana, E. (2023). Perancangan Deteksi Dini Banjir Berbasis Iot Dan Water Level Dengan Notifikasi Blynk Dan Alarm. Seminar Nasional Teknologi Informasi & Ilmu Komputer, x, 6–13.

No, V., Waluyo, A. F., Putra, T. R., & No, V. (2024). Infotek : Jurnal Informatika dan Teknologi Peringatan Dini Banjir Berbasis Internet Of Things (IOT) dan Telegram Infotek : Jurnal Informatika dan Teknologi Dalam beberapa tahun terakhir belakangan , terjadi kemajuan pesat dalam bidang teknologi . Manusi. 7(1), 142–150.

Rachmawardani, A., Wijaya, S. K., & Shopaheluwakan, A. (2022). Sistem Peringatan Dini Banjir Berbasis Machine Learning: Studi Literatur. METHOMIKA Jurnal Manajemen Informatika Dan Komputerisasi Akuntansi, 6(6), 188–198.

DOI: <https://doi.org/10.46880/jmika.vol6no2.pp188-198>

Sutarti, S., Anharudin, A., & Rosadi, S. (2022). Prototype Sistem Pendeteksi Banjir Menggunakan Nodemcu Dan Protokol Mqtt Berbasis Internet of Things. Jurnal Sistem Informasi Dan Informatika (Simika), 5(1), 38–48.

DOI: <https://doi.org/10.47080/simika.v5i1.1424>

Ulum, M. bahrul. (2023). Sistem Monitoring Cuaca Dan Peringatan Banjir Berbasis Iot Dengan Menggunakan Aplikasi Mit App Inventor. Jurnal Informatika Dan Teknik Elektro Terapan, 11(3), 319–328.

DOI: <https://doi.org/10.23960/jitet.v11i3.3088>