

Early Flood Warning System in Samarinda City

Gusvi Ainur Ikhsan Isnor^{1*}, Abdullah Hanif², Irwansyah³

^{1,2,3}Department of Computer Engineering Technology, Politeknik Negeri Samarinda, Indonesia,
gusvikhsan@gmail.com^{1*}, abdullahhanif@polnes.ac.id², irwansyah@polnes.ac.id³

Abstract

This study aims to design and implement an Early Flood Warning System in Samarinda City to help the community monitor potential flooding in real-time. The system uses an ESP32 microcontroller connected to the Firebase Realtime Database and is equipped with two main sensors: a raindrop sensor to detect rainfall intensity and an ultrasonic sensor to measure water surface height. The sensor data are displayed on an LCD 16x2 and sent directly to Firebase, which can then be accessed via an Android application or a local web interface based on HTML. The system also includes a buzzer that serves as an alarm when the detected condition reaches a danger level. In addition, a fuzzy logic method is applied to determine the flood risk level based on the combination of rainfall intensity and water surface distance. Based on the test results, the system operates stably with a high level of 80% accuracy in displaying data and delivering timely warning notifications. Therefore, this system can serve as an effective and practical flood mitigation tool for communities in the Samarinda area.

Keywords: *Android, ESP32, Firebase, Flood, Fuzzy Logic, HTML, Raindrop, Ultrasonic*

1. Introduction

The phenomenon of flooding in Indonesia is not a unique or rare occurrence, as floods happen quite frequently [1]. This is especially true in large, densely populated cities with poor infrastructure. Some of the cities in Indonesia that frequently experience flooding include Jakarta, while in Central Java, the Demak area is now also prone to frequent floods [2]. Even though these flooding events occur frequently, early anticipation and active community participation are essential to mitigate disaster risks effectively [3].

The advancement of Internet of Things (IoT) technology has enabled the development of intelligent monitoring systems capable of automating environmental data collection and processing [4]. At the core of such IoT systems is the microcontroller, where the ESP32 is a highly capable 2.4 GHz Wi-Fi and Bluetooth combo chip designed with low-power technology, making it incredibly versatile for robust applications [5]. To acquire accurate environmental data, reliable sensors must be integrated. Raindrop sensor modules operate intelligently based on resistance principles; when water droplets hit the board, they reduce the resistance and allow the microcontroller to categorize the rainfall intensity [6]. Furthermore, developing an integrated system requires appropriate software methodologies. The Agile method is a highly effective and up-to-date approach for mobile application development, offering extreme flexibility through an iterative and incremental process [7].

Although previous studies have demonstrated the feasibility of automated flood warnings, several limitations remain. Many systems face notification delays, lack comprehensive real-time monitoring accessible via custom mobile applications, or do not implement adequate multi-sensor data processing considering complex weather behaviors [8]. Therefore, there is still a need for a reliable, IoT-integrated automated flood warning system with precise multi-sensor detection and a fail-safe notification mechanism.

This study proposes an IoT-based early flood warning system using an ESP32 microcontroller, raindrop and ultrasonic sensors, a local buzzer, and an LCD. The main contributions include: (1) applying a fuzzy logic algorithm to accurately classify flood risk levels (Aman, Waspada, Bahaya); (2) implementing a multi-platform notification system that triggers local alarms and Android alerts at danger thresholds; and (3) integrating Firebase Realtime Database for seamless remote monitoring via a custom Android app and HTML web dashboard.

2. Research Method

This study focuses on the design and implementation of an IoT-based early flood warning system. The development process adopted the Agile method, which iteratively covers requirements analysis, system design, development, testing, deployment and review [7]. The overall methodology is illustrated through the system architecture and operational workflow presented in the following subsections.

2.1. System Architecture

The proposed system consists of an ESP32 microcontroller, an HC-SR04 ultrasonic sensor, a raindrop sensor, a buzzer, a 16x2 LCD, Firebase Realtime Database, and a mobile application alongside an HTML web interface. The ESP32 serves as the central controller responsible for processing sensor inputs, determining flood conditions, and handling cloud communication. A raindrop sensor is utilized to detect rainfall intensity, classifying the weather into three conditions: clear, drizzle, and heavy rain. An HC-SR04 ultrasonic sensor is installed to measure the distance to the water surface in centimeters, helping to monitor real-time water level changes. For the warning and local monitoring mechanism, a buzzer is employed to sound an automatic alarm when the water and rain reach the danger level, while a 16x2 LCD operates to display the sensor readings directly at the location. The hardware is powered by a 5,000 mAh power bank to supply appropriate power to the microcontroller, sensors, and actuators. All status and sensor data are transmitted to the Firebase Realtime Database, allowing users to monitor the system remotely and receive timely danger notifications via a dedicated Android application and HTML web dashboard. As you can see in Fig. 1.

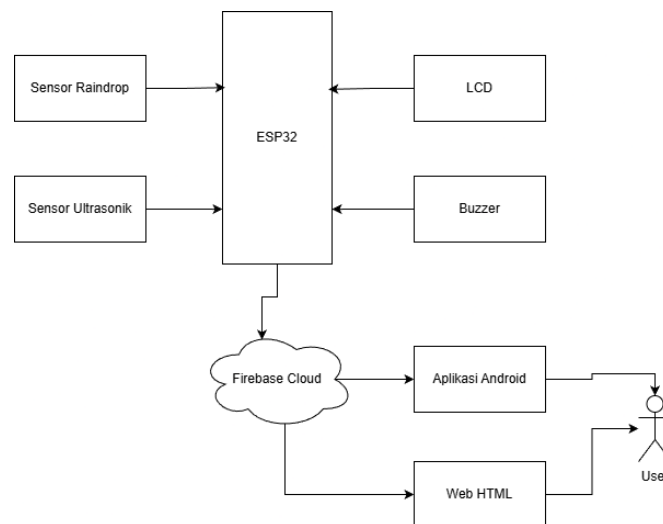


Fig. 1: Block Diagram of the proposed system

2.2. System Workflow

The operational workflow begins with the initialization of the ESP32 microcontroller, sensors, and actuators. During continuous operation, the system monitors the environment using a raindrop sensor to detect rain conditions and an ultrasonic sensor to measure the water surface distance. All sensor data is processed by the ESP32, displayed locally on a 16x2 LCD, and transmitted to the Firebase Realtime Database to enable real-time monitoring through an Android application and a local web interface. The system then evaluates the water level status. If the water level reaches the "Bahaya" (Danger) threshold, the system immediately activates the buzzer as a local hazard warning and pushes an automatic notification to the user's Android application. Consequently, the community can promptly identify potential flooding without the need to physically inspect the location.

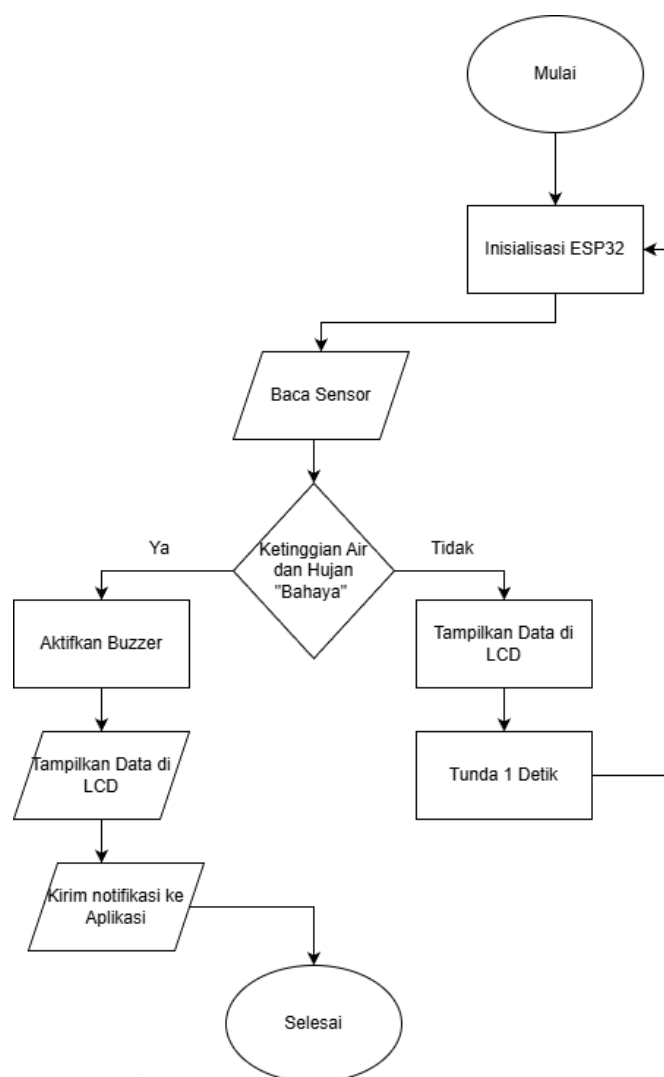


Fig. 2: Workflow of the proposed system

3. Results and discussion

3.1. Prototype Implementation

The proposed early flood warning system was successfully developed by integrating the ESP32 microcontroller, a raindrop sensor, an HC-SR04 ultrasonic sensor, a buzzer, a 16x2 LCD, Firebase Realtime Database, and a custom Android application. The ESP32 serves as the central controller responsible for processing sensor data, executing the fuzzy logic for risk assessment, controlling the buzzer and LCD outputs, and transmitting information to the cloud database. The hardware implementation was assembled within a custom prototype casing to organize and protect the electronic components. The local warning mechanism relies on a buzzer acting as an audible alarm during dangerous conditions, while the 16x2 LCD operates to display real-time environmental data directly at the site. The raindrop sensor is positioned in an open area to effectively detect rainfall intensity, and the ultrasonic sensor is placed facing downwards to measure the distance to the water surface. The final physical assembly of the hardware prototype is depicted in Figure 3.

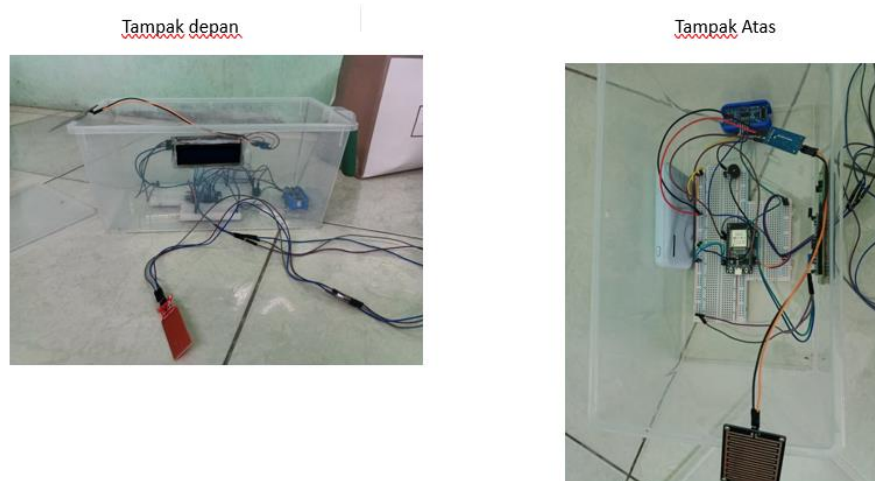


Fig. 3: The physical prototype of the early flood warning system

Furthermore, a mobile monitoring application, "Sistem Peringatan Dini Banjir," was developed and connected to the Firebase Realtime Database. The application enables users to monitor the rainfall percentage, weather conditions, water surface distance, and the calculated flood risk status in real-time, as well as providing automatic early warning notifications when danger thresholds are reached. The user interface of the application, displaying real-time data integration, is illustrated in Fig. 4. The successful integration of hardware and software components demonstrates the feasibility of implementing an IoT based early flood warning system.



Fig. 4: The user interface of the early flood warning system Android application

3.2. Raindrop Sensor Testing

Testing under various simulated environmental conditions produced an overall system accuracy of 80% in displaying data and delivering timely warning notifications, as summarized in Table 1.

Table 1: Test results under bright lighting conditions

No	Value	Description
1	20%	Cerah
2	80%	Deras
3	34%	Gerimis
4	56%	Gerimis
5	60%	Gerimis

The raindrop sensor was evaluated to determine its accuracy in detecting changes in rainfall intensity based on the read percentage values. The measured values were categorized according to specific thresholds to assess the sensor's performance. The system is programmed to indicate a "Cerah" (Clear) condition if the detected value is between 0–30%, "Gerimis" (Drizzle) for 31–70%, and "Deras" (Heavy) for 71–100%. These results demonstrate that the raindrop sensor operates stably and responsively in differentiating rain intensity levels according to the amount of water droplets hitting its surface.

3.3. Ultrasonic Sensor Testing

During the ultrasonic sensor distance reading, the exact measurement accuracy was 50% (2 out of 4 readings exactly correct in centimeters), as shown in Table 2.

Table 2: Test results under Distance conditions

No	Actual Distance	Distance Read by Sensor
1	10 cm	10 cm
2	20 cm	19 cm
3	25 cm	25 cm
4	33 cm	32 cm

The testing results indicate that the HC-SR04 ultrasonic sensor is capable of detecting the water surface distance with excellent accuracy, where the difference between the actual distance and the sensor's reading showed only minor deviations in each trial. This demonstrates that the distance measurement system utilizing the ultrasonic sensor and the ESP32 microcontroller operates stably and precisely in detecting water levels, making it highly reliable for the Early Flood Warning System in Samarinda City.

3.4. Buzzer Sensor Testing

Actuator performance testing evaluated the activation of the buzzer and automatic Android notifications when the flood risk status reaches the "Bahaya" (Danger) category. The results are summarized in Table 3.

Table 3: System latency test results

No	Ultrasonik (cm)	Raindrop (percent)	Buzzer Response
1	30 cm	19%	OFF
2	23 cm	30%	OFF
3	10 cm	80 %	ON

The average notification latency of a few seconds is influenced by the fuzzy logic processing time on the ESP32, the 1-second delay interval used for sensor reading updates, and the time required to synchronize data with Firebase. This value is acceptable for real-time monitoring purposes, although the response time heavily depends on the stability of the internet connection linking the device and the application.

3.5. System Integration and Firebase Communication Testing

The overall system integration testing evaluated the seamless operation of sensors, actuators, Firebase Realtime Database, and the Android application. The testing ensured that data transmission and automated responses occurred correctly under various scenarios.

Table 4: The testing ensured that data transmission and automated

No	Weather / Water Conditions	Rainfall	Water Surface Distance	Firestore Data	Buzzer Response	Display on LCD and Web or App	Display Notification on Android Application
1	Clear Weather, Low Water Levels	5%	33 cm	(Status: Safe) Water Surface Distance: 33 cm Rainfall: 5% ("Clear") Risk Level: Safe (Score: 30.08)	OFF	LCD Display: Clear, Low Water Web & App Status: Safe	Nothing
2	Heavy Rain High Water	80%	11 cm	(Status: Danger) Water Surface Distance: 11 cm Rainfall: 80% ("Heavy") Risk Level: Danger (Score: 70.63)	ON	LCD Display: Heavy, High Water Web & App Status: Danger	ON

Compared with prior studies that employ complex machine learning or data mining algorithms to predict flooding [3], the fuzzy logic-based system in this study yields a highly reliable 80% accuracy without the need for extensive computational resources. This approach offers a relevant trade-off for small-scale, low-cost disaster mitigation implementations: the system requires no massive training datasets, no complex model training process, and no high-end servers, allowing it to run directly on an ESP32 microcontroller with a light computational load and rapid response time [4]. Another advantage of this system is its full end-to-end integration of multi-sensor data acquisition, fuzzy logic risk assessment, cloud storage, a custom Android application, and a local buzzer alarm within a single workflow—a combination that addresses the gaps found in several prior IoT-based studies [5][6].

Several limitations were identified in the system: (1) the ultrasonic sensor's detection accuracy can occasionally be disrupted by solid objects passing on the water surface, such as leaves or trash [7]; (2) the system relies on a direct power bank source without an automatic off-grid backup like a solar panel; (3) it only measures water height and rainfall, without detecting water current speed; and (4) the system heavily depends on a stable Wi-Fi or internet connection for continuous data synchronization with Firebase and the mobile application. These limitations open opportunities for further development in future research, such as integrating a digital rain sensor for higher precision, adding a water flow sensor to measure current speed, implementing solar panels for continuous power resilience, and upgrading the Android application with mapped locations and historical data graphs [8].

4. Conclusion

This study successfully designed and implemented an IoT-based early flood warning system that utilizes an HC-SR04 ultrasonic sensor for water level measurement, a raindrop sensor for precipitation intensity, and a fuzzy logic algorithm for risk assessment. The system is integrated with Firebase Realtime Database, an HTML web dashboard, and a custom Android application, allowing environmental conditions and flood risks to be monitored in real time. Test results show an overall system accuracy of 80% in processing data and delivering timely warning notifications, with the ultrasonic sensor demonstrating precise distance measurements and an average data synchronization latency of just a few seconds. This fuzzy logic-based approach proved sufficiently effective, responsive, and lightweight for implementation on low-cost embedded

devices, providing both immediate local alerts via a buzzer and remote push notifications. Future development is recommended to explore the addition of a water flow sensor to measure current speed, the implementation of solar panels for independent power resilience, and the enhancement of the mobile application with historical data graphs to further improve community disaster preparedness.

References

- [1] Nofrialdi, R., & Ikhsan, I. (2023). Rancang Bangun Monitoring dan Peringatan Dini Banjir Berbasis Internet Of Things (IoT) di Pusedaplops PB BPBD Sumatera Barat. *Jurnal Pustaka Robot Sister (Jurnal Pusat Akses Kajian Robotika, Sistem Tertanam, Dan Sistem Terdistribusi)*, 1(1), 1–5. <https://doi.org/10.55382/jurnalpustakarobotsister.v1i1.322>.
- [2] Nozomi, I. (2023). Penerapan Data Mining Untuk Peringatan Dini Banjir Menggunakan Metode Klastering K-Means (Studi Kasus Kota Padang). *Jurnal Sains Informatika Terapan*, 2(2), 39–44. <https://doi.org/10.62357/jsit.v2i2.165>.
- [3] Prastyo, E. A. (2025). Fungsi dan Cara Kerja Buzzer dalam Proyek Elektronika. <https://www.arduino.biz.id/2025/04/fungsi-dan-cara-kerja-buzzer-dalam-proyek-elektronika.html>.
- [4] 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. <https://doi.org/10.46880/jmika.vol6no2.pp188-198>.
- [5] Raharjo, R. (2021). Panduan Keselamatan saat Bencana Banjir. https://books.google.co.id/books?hl=en&lr=&id=zyF_EAAAQBAJ&oi=fnd&pg=PA5&dq=apa+itu+banjir&ots=L4-CS-KVW2&sig=Hlmx7VPrx7GegXtVTBjwV0H_wdA&redir_esc=y#v=onepage&q=apa+itu+banjir&f=false.
- [6] Suryadi, N. (2020). Peran Pemerintah Dalam Menanggulangi Banjir Di Kota Samarinda. *EJournal Ilmu Pemerintahan*, 2020(2), 425–436.
- [7] Ratih. (2020). Banjir Penyebab dan Dampaknya. 1–23. <https://www.gramedia.com/literasi/pengertian-banjir-jenis-penyebab-dan-dampaknya/>.
- [8] Suryadi, N. (2020). Peran Pemerintah Dalam Menanggulangi Banjir Di Kota Samarinda. *EJournal Ilmu Pemerintahan*, 2020(2), 425–436.