Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

June 2026. Vol. 5. No. 3; e-ISSN: 2808-4519

Water Level Prediction Based on Internet of Things and Machine Learning

Muhammad Fauzi Ariyo[1]*, Irwansyah[2], Agus Triyono[3]

[1,2,3]*Department of Computer Engineering Technology, Politeknik Negeri Samarinda, Indonesia*
fauziariyo12@gmail.com[1], irwansyah@polnes.ac.id[2], triyono@polnes.ac.id[3]*

Abstract

This study develops an Internet of Things and machine learning based water-level prediction system for Jalan Embun Suryana, Samarinda. The device uses an ESP32, a JSN-SR04T ultrasonic sensor to measure water-surface distance, and a tipping-bucket ombrometer for rainfall. Data are synchronized to Firebase and transformed into a simple accumulated-rainfall feature to train a simple linear regression model. The research stages comprised Requirements Analysis, System Design, Sensor Data Collection, Model Training, Model Testing, Device Testing, and Integration. An Android application provides real-time visualization and early-warning notifications. Experiments indicate good sensor accuracy (JSN-SR04T; tipping-bucket 0.7 mm/tip) and model performance (MSE = 0.562 cm², RMSE = 0.749 cm, R^2 = 0.992). In sum, data acquisition, prediction, and notifications operated as designed, and the simple linear regression produced water-level forecasts. At the study site, the association between increasing rainfall and rising water level was corroborated by repeated manual field measurements.

***Keywords**: ESP32; Internet of Things; Samarinda; Simple linear regression; Water level prediction.*

1. Introduction

Flooding is one of the most destructive natural disasters with a massive impact on community activities, infrastructure functionality, and socio-economic systems. In Indonesia, Samarinda City is one of the regions highly vulnerable to this flood disaster [1]. The problem of water inundation in Samarinda generally occurs due to excessive surface runoff triggered by natural factors, such as low topography and high rainfall, as well as human activities that affect drainage performance, including clogged channels due to trash accumulation [1].

One of the main challenges in flood mitigation efforts is the delay in obtaining real-time information regarding water level increases. This delay leaves the community with insufficient time to prepare and take preventive actions [2]. Currently, available water level monitoring systems are mostly manual or depend on unreliable communication infrastructure, making them less effective in providing rapid and accurate early warnings. Therefore, a continuous, accurate, and real-time water level monitoring system is critically needed to support disaster mitigation [2].

The advancement of Internet of Things (IoT) technology and machine learning offers promising solutions for continuous environmental monitoring and early disaster prediction [3], [4]. Several previous studies have proposed the use of IoT and machine learning algorithms to predict floods [5]. For example, studies have utilized deep learning with Raspberry Pi and ultrasonic sensors, or complex algorithms such as Long Short-Term Memory (LSTM), Random Forest, and polynomial regression for weather and flood forecasting. However, the implementation of these complex algorithms often requires high computational costs. Furthermore, localized flood prediction that combines direct rainfall measurement with low-power devices is still rarely explored [6].

To address these limitations, this study proposes the design and implementation of an IoT-based water level monitoring and prediction system using a low-power ESP32 microcontroller [7]. The system integrates a waterproof JSN-SR04T ultrasonic sensor to measure the water surface distance and a tipping-bucket ombrometer to measure local rainfall intensity. The gathered field data are transmitted via Wi-Fi to Firebase Realtime Database for fast and reliable cloud storage. For the prediction mechanism, this study utilizes a simple linear regression model. This method was chosen because of its efficiency and accuracy in analyzing the pattern of rising water levels based on current rainfall data without requiring massive computational power [8].

The primary objective of this study is to design and build an IoT system to monitor and predict water levels in flood-prone locations, specifically at Jalan Embun Suryana, Samarinda City. The monitoring and prediction results are presented to users through an Android application that provides real-time visualizations, a 24-hour trend graph, and automated early warning notifications when the water level surpasses predefined safe thresholds. Through this design, the proposed system is expected to produce an effective early warning methodology to reduce the risk of material losses and support disaster mitigation steps by the community and authorities

2. Research Method

This research was conducted using a systematic system development life cycle approach, which encompasses several sequential stages: requirements analysis, system design, sensor data collection, algorithm model training, testing, and final system integration. The operational testing and implementation of the proposed system were carried out in a flood-prone area located at Jalan Embun Suryana, Samarinda City.

2.1. System Architecture

The proposed system is designed with an Internet of Things (IoT) architecture that integrates field hardware devices, a cloud-based database, and a user interface application [9]. The central hardware control unit utilizes the ESP32 Devkit V1 microcontroller, which is equipped with a built-in Wi-Fi module to facilitate wireless data communication. For the input units, the system employs a waterproof JSN-SR04T ultrasonic sensor mounted facing the water surface to measure the distance, alongside a tipping-bucket ombrometer to measure the local rainfall intensity. The comprehensive architecture and the data flow of the system are presented in Figure 1.

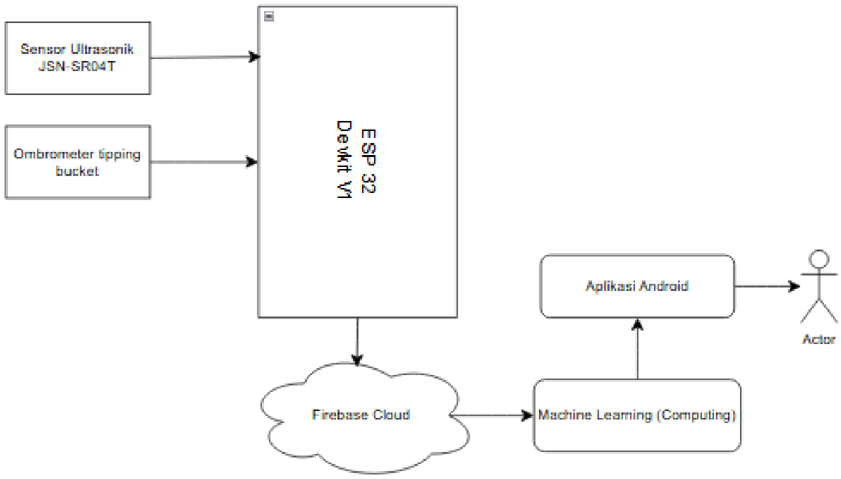

Figure 1: Architecture of the system

The hardware control center utilizes an ESP32 Devkit V1 microcontroller, which incorporates a built-in Wi-Fi module for data communication [9], [10]. As input units, the system employs a JSN-SR04T ultrasonic sensor installed above the water surface to measure distance, and a tipping-bucket ombrometer to measure local rainfall intensity [7]. Each time the tipping-bucket container fills and discharges, the system registers a single "tip," which is then converted into a rainfall measurement of 0.7 mm. The system's power supply is provided by a portable battery housed within a waterproof panel to maintain operational reliability in an outdoor environment [9].

2.2. System Workflow

The operational workflow of the software in executing the data flow from upstream to downstream is performed sequentially. The data acquisition process begins with time synchronization using the Network Time Protocol (NTP) in the Asia/Makassar (UTC+8) time zone to generate ISO-8601 formatted timestamps. During the data collection stage, rainfall is measured using a tipping-bucket ombrometer with a 150 ms debounce applied to prevent double counting caused by noise, where each tip is converted into 0.7 mm of rainfall. Meanwhile, the water surface distance data is acquired by the JSN-SR04T sensor through a multi-sampling method, which is followed by outlier removal and average calculation. The actual water level is then calculated based on the difference between the sensor installation height (35 cm) and the measured distance. All validated data are structured into a JSON payload and transmitted in real-time to the Firebase Realtime Database. Firebase acts as an intermediary that stores historical data records while providing access to the latest data for local computing devices. A local Python script retrieves this historical data and processes it using a simple linear regression algorithm to generate water level prediction values. The comprehensive workflow of the proposed system is illustrated in Figure 2.

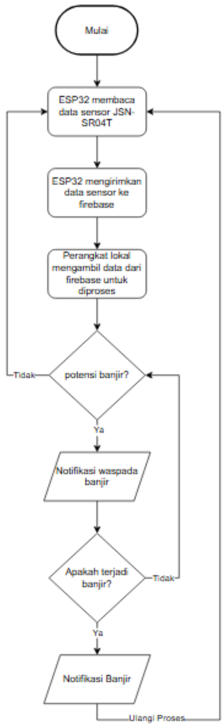

Figure 2: Workflow of the proposed system

Following the prediction process, the results are sent back to Firebase to be accessed by the user's Android application. This application serves to present real-time data visualization and automatically triggers early warning notifications if the predicted value or the actual water level exceeds the predefined safe thresholds. After completing these steps, the system returns to the initial monitoring state and continuously repeats the entire data acquisition cycle.

3. Result and Discussion

3.1. JSN-SR04T Sensor Testing

Hardware testing is focused on validating the accuracy level of the JSN-SR04T ultrasonic sensor in measuring water surface distance and calibrating the tipping-bucket ombrometer for rainfall intensity. The evaluation of the ultrasonic sensor readings was conducted by placing a reflector at several reference points varying from 20 cm to 40 cm, as presented in Table 1.

Table 1: JSN-SR04T Sensor Test Result

No	Distance (cm)	Average Reading (cm)	Error (%)
1	20	20.6	3.00
2	25	25.7	2.80
3	30	30.9	3.00
4	35	36	2.86
5	40	41	2.50

Based on Table 1, the JSN-SR04T ultrasonic sensor demonstrates highly stable measurement performance. The recorded average error rate is very low, ranging from 2.50% to 3.00%. This error rate remains well within the safe tolerance limits for open channel water level monitoring applications.

These testing results ensure that the JSN-SR04T sensor is capable of generating input data with a high level of validity and reliability. Given its consistent performance across various measurement points, the data from this sensor can be relied upon as the foundation for prediction computations by the artificial intelligence system in the subsequent data processing stage.

3.2. Prototype Implementation

The water level monitoring and prediction system was developed by integrating an ESP32 Devkit V1 microcontroller with a waterproof JSN-SR04T ultrasonic sensor and a tipping-bucket ombrometer. The system is housed within a protective electrical panel box, measuring 25 x 25 cm, which serves as a shield against rain and environmental exposure, as shown in Figure 3.

Figure 3: Hardware implementation of the monitoring system within the protective enclosure

The hardware assembly is organized on a perforated PCB located on the left side of the panel, hosting the ESP32 microcontroller and necessary connectivity ports. A portable power bank is mounted on the right side to ensure a stable power supply, allowing the device to operate independently in the field. The JSN-SR04T sensor is connected externally to measure water levels, while the ombrometer tipping-bucket is positioned atop the panel to accurately record rainfall intensity. This robust integration ensures that the system remains reliable and protected during continuous environmental monitoring.

3.3. Prediction Model Evaluation

The simple linear regression model was trained using observational data to map the relationship between accumulated rainfall and water level fluctuations. The visualization of the data distribution and the regression line resulting from the system training can be seen in Figure 4.

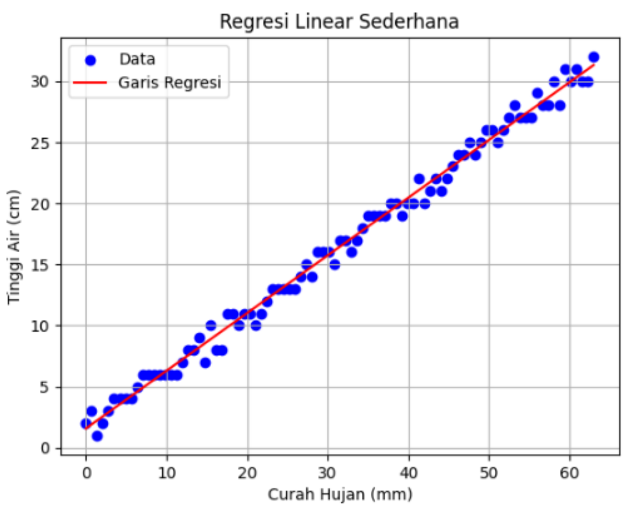

Figure 4: Simple linear regression graph of water level

Based on computational testing, the model demonstrated excellent performance with a Mean Squared Error (MSE) of 0.562 cm² and a Root Mean Squared Error (RMSE) of 0.749 cm. Furthermore, this model recorded a coefficient of determination (R^2) of 0.992. An R^2 value approaching 1 proves that the algorithm possesses a very high level of reliability in predicting water level rise trends based on the latest rainfall intensity data.

3.4. Mobile Application Interface

The mobile monitoring application acts as the primary interface for users to access real-time system status and environmental data. The main dashboard is designed with a dark theme to ensure clarity, providing an immediate overview of current conditions, including rainfall intensity, current water level, and the machine learning-based prediction value. An intuitive risk-level banner provides rapid status classification, ranging from "Level 1 – Safe" to "Level 4 – High Flood," allowing users to instantly assess potential hazards at the monitoring site, as illustrated in Figure 5.

Figure 5: Main dashboard of the mobile application

Furthermore, the application provides deeper analytical insights by displaying a 24-hour trend graph, which visualizes the correlation between water level fluctuations and rainfall patterns. This dual-axis chart allows users to observe how historical precipitation directly influences water levels over time. Additionally, the dashboard integrates a location map using OpenStreetMap (OSM) to pinpoint the precise installation site of the sensor hardware, ensuring users are well-informed about the geographical context of the data being reported, as depicted in Figure 6.

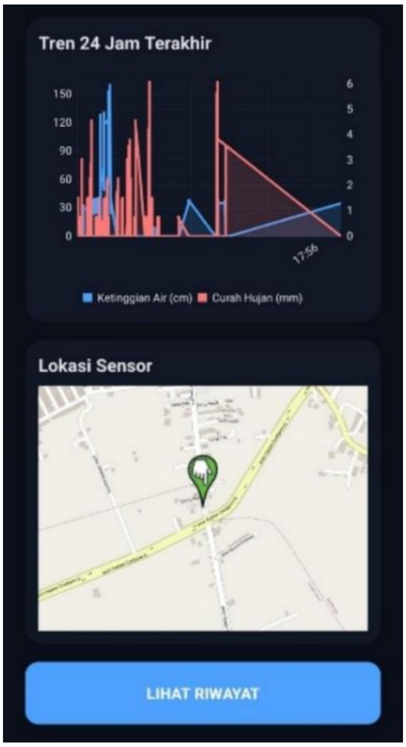

Figure 6: 24-hour trend visualization and sensor location map

To facilitate long-term data review, the application includes a "View History" feature, enabling users to navigate through archived measurements chronologically. This feature is essential for analyzing seasonal patterns or reviewing specific flood events, thereby supporting more informed decision-making and improved community preparedness for future flood mitigation.

3.5. System Integration and Interface Testing

Data communication testing demonstrates that the ESP32 successfully transmits measurement data packets every 5 seconds to the Firebase Realtime Database without any packet loss. The Android application on the user's device is capable of retrieving data from the cloud with low latency, providing real-time monitoring metrics and 24-hour trend graph visualizations. During early warning scenario testing, the system logic successfully executed the notification functions optimally. When the actual water level data or predicted values exceed the warning thresholds, the application instantly sends a push notification (e.g., a "Water Level ≥ 23 cm" alert) to the user's phone screen.

Table 2: System notification test result

Condition	Water Level/Prediction	Notification Display
If water level or prediction exceeds 15 cm	18	Ambang Jarak Terlewati Jarak kini 18,0 cm (melewati 15,0 cm)
If water level or prediction exceeds 23 cm	24	Tinggi air ≥ 23 cm Tinggi terukur 26,0 cm (melewati 23 cm)

Evaluation in Table 2 shows that the application logic successfully executes the notification functions precisely. The ESP32 consistently transmits data packet updates to the Firebase Realtime Database every 5 seconds without any issues. When the application detects a surge in actual water level data or predicted values exceeding the 15 cm or 23 cm thresholds, the system instantly displays a push notification on the user's phone screen, including the timestamp and hazard status. This alert transmission has also been optimized using a 30-second rate-limiting mechanism to prevent message congestion (*spam*) in the event of rapid water level fluctuations around the threshold values.

4. Conclusion

This research successfully designed and implemented an end-to-end water level monitoring and prediction system utilizing Internet of Things (IoT) and machine learning technology. The hardware, based on the ESP32 microcontroller, effectively acquires rainfall data from a tipping-bucket ombrometer calibrated at 0.7 mm per tip with a 150 ms debounce and water surface distance data from a JSN-SR04T sensor, which undergoes multi-sampling and outlier removal to ensure measurement precision. The system maintains high reliability by transmitting data periodically every 5 seconds to the Firebase Realtime Database, allowing for seamless integration of a simple linear regression algorithm within the system pipeline to generate accurate water level forecasts. Furthermore, the Android application functions optimally by providing near real-time data visualization and executing threshold-based early warning notifications, which are supported by a 30-second rate-limit mechanism to prevent notification spamming. Overall, the system demonstrates operational feasibility with low latency, serving as a robust prototype for field-based flood mitigation instruments. For future development, it is recommended to conduct extensive field testing using actual data, enhance the device's physical shielding against extreme weather, implement a local data buffer to prevent historical data loss during internet outages, and optimize model accuracy by applying more complex computational algorithms such as gradient boosting.

Acknowledgement

The authors would like to thank the Department of Information Technology at Politeknik Negeri Samarinda for providing the necessary facilities for this research. We acknowledge the support provided by Irwansyah during the initial stages of this study. A special and heartfelt appreciation goes to Agus Triyono for his extensive guidance, insightful feedback, and continuous support throughout the research process. Furthermore, we are deeply grateful to our colleagues and all individuals who contributed their time and effort to ensure the successful completion of this study.

References

[1] H. Setiawan *et al.*, "ANALISIS PENYEBAB BANJIR DI KOTA SAMARINDA," 2020. [Online]. Available: https://ejournal.upi.edu/index.php/gea

[2] U. Dirgantara and M. Suryadarma, "REVOLUSI INDUSTRI 4.0: INTERNET OF THINGS, IMPLEMENTASI PADA BERBAGAI SEKTOR BERBASIS TEKNOLOGI INFORMASI (BAGIAN 1) Hari Mantik."

[3] G. Al-Rawas, M. R. Nikoo, M. Al-Wardy, and T. Etri, "A Critical Review of Emerging Technologies for Flash Flood Prediction: Examining Artificial Intelligence, Machine Learning, Internet of Things, Cloud Computing, and Robotics Techniques," Jul. 01, 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/w16142069.

[4] M. Motta, M. de Castro Neto, and P. Sarmento, "A mixed approach for urban flood prediction using Machine Learning and GIS," *International Journal of Disaster Risk Reduction*, vol. 56, Apr. 2021, doi: 10.1016/j.ijdrr.2021.102154.

[5] A. Mosavi, P. Ozturk, and K. W. Chau, "Flood prediction using machine learning models: Literature review," Oct. 27, 2018, *MDPI AG*. doi: 10.3390/w10111536.

[6] A. Kurniawan, "SEJARAH, CARA KERJA DAN MANFAAT INTERNET OF THINGS."

[7] A. Andang, N. Hiron, A. Chobir, and N. Busaeri, "Investigation of ultrasonic sensor type JSN-SRT04 performance as flood elevation detection," in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Aug. 2019. doi: 10.1088/1757-899X/550/1/012018.

[8] K. Kunverji, K. Shah, and N. Banu Shah, "A Flood Prediction System Developed Using Various Machine Learning Algorithms." [Online]. Available: https://ssrn.com/abstract=3866524

[9] M. Herath, T. Jayathilaka, Y. Hoshino, and U. Rathnayake, "Deep Machine Learning-Based Water Level Prediction Model for Colombo Flood Detention Area," *Applied Sciences (Switzerland)*, vol. 13, no. 4, Feb. 2023, doi: 10.3390/app13042194.

[10] H. Jamaludin, P. Sultan, and I. Shah, "Designing ESP32 Base Shield Board for IoT Application Politeknik Designing ESP32 Base Shield Board for IoT Application," 2020.