THE IoT-BASED EARLY WARNING SYSTEM FOR DETECTING HIGH TIDE FLOODS (ROB-EWS) IN TAMBAK LOROK, SEMARANG INDONESIA

ABSTRACT

Background and Objectives: The Tambaklorok fishing village area, located on the northern coast of Semarang City, Indonesia, is susceptible to sea level fluctuations, which often lead to tidal flooding and seawater encroaching into residential areas. Consequently, a seawater flood detector (rob) is needed for effective disaster mitigation for affected communities. This study aims to implement a disaster mitigation system by installing an IoT-based Seawater Flood (Rob) Early Warning System to reduce disaster risks and their impacts on the community. **Methods**: The IoT-based Early Warning System is designed to detect floods by monitoring sea level height parameters. **Results:** IoT-based devices measure sea level using ultrasonic sensors, with the readings processed by a microprocessor. The data is then periodically analyzed and transmitted to an application in real-time. This technology allows for real-time information dissemination through web applications and alerts in the form of sirens. These warnings enable residents to evacuate and move to safer areas. The siren alerts are categorized into two levels: Alert 1 indicates that seawater is approaching the land boundary, while Alert 2 signals that water has begun to enter residential areas. Additionally, designated evacuation zones will guide affected communities to safer areas. **Conclusion:**The installation of an IoT-based Early Warning System delivers real-time information to help mitigate the impact of rising sea levels (rob) in the Tambak Lorok fishing village community

Keywords: Tambak Lorok, Detection, Internet of things, Mitigation, Rob early warning system

1. INTRODUCTION

The Tambak Lorok fishing area is the largest fishing village in Semarang City, located along the coast of the Java Sea. Situated in Tanjung Mas Village, in the northern part of Semarang City, this area spans approximately 84.48 hectares and directly borders the Java Sea.[1]. The Tambak Lorok area serves multiple purposes, including as a settlement for fishermen, a center for trade, a fish landing and auction site, as well as a venue for socio-cultural activities and other economic endeavors[2]. Residents of Tambak Lorok village utilize marine resources like shrimp and small fish to produce shrimp paste. Additionally, the community processes mangroves into snacks or chips, representing another way they leverage natural resources for economic gain [3]

Tambak Lorok has the characteristic of lowlands, with a height of only 0.5 meters above sea level. The land in this area consists of weathered and sedimentary structures, making it vulnerable to land subsidence at a rate of around 10 centimeters per year, exacerbating tidal flooding. With a population of around 9,000 people, Tambak Lorok residents face a significant risk if tidal flooding occurs [4]Tidal flooding presents serious risks, including potential loss of

life and damage to residential buildings. To address these future hazards, coastal areas require an early warning system. Such technology would allow local residents to prepare in advance, mitigating risks and minimizing the impacts of significant flooding. These impacts include infrastructure damage, loss of life, economic losses to fisheries and tourism businesses, and damage to private property [5]Based on the tidal flood incidents in the Tambak Lorok area, there is a critical need for an effective and timely evacuation route and early warning system [6]

Current advancements in technology enable the connection of water level monitoring with various devices. This technology is commonly referred to as IoT, which conceptually involves connecting surrounding objects through internet connections. The growing internet infrastructure allows for connectivity with computer networks and various other parameters. MySWL is an IoT service platform designed to facilitate the management and integration of IoT systems across various applications. Some key aspects of the development and features typically offered by platforms like MySWL[7] Automatic Alerts: MySWL provides a customizable alert system that notifies users about abnormal conditions or significant events [8] Notifications involve sending alerts through various channels such as email, whatsapp, or mobile applications. The services used in this study include Real-Time Database and WebApp. The WebApp service allows data to be accessed through a web browser, making it easier for the community to monitor river water levels (ESP32)[9]In addition to being accessible via a web browser, the ESP32 also allows access to the Real-Time Database from SQL over the internet. It includes a warning module that triggers alerts when the water level reaches a certain dangerous threshold.The Rob-Early Warning technology system, which combines ESP32 with SQL, can enhance IoT projects by offering an affordable, scalable, and flexible data management solution. This integration leverages real-time processing and advanced query capabilities to improve overall system performance [10]Some of the advantages of SQL technology are using a relational data model that allows storing and managing data in interconnected tables, very useful for applications that require complex relationships between data and referential integrity, using a flexible query language to perform complex operations, such as combining, grouping, and filtering data. This allows for in-depth data analysis and complex reports. [11] SQL databases support ACID (Atomicity, Consistency, Isolation, Durability) transactions, which ensure data integrity and consistency even in situations with many users or simultaneous transactions[12]. SQL is often easier to scale vertically by adding more resources to an existing server. This can be a good solution for applications with very intensive workloads.[13]. In the application of an IoT-based early warning system, MySWL is a tool used within the MySQL Workbench environment that directly interacts with SQL. SQL automation and MySWL facilitate the automatic execution of SQL commands. The features of MySWL extend SQL functionality by providing a means to automate and manage complex tasks in MySQL. When SQL queries are executed, MySQL interprets the SQL commands and performs the appropriate operations on the database. MySQL is responsible for processing the queries, managing the data, and returning the results[14].

IoT-Based Early Warning and Mitigation System (MySWL-SQL) for Tidal Flood Detection in Tambak Lorok Fishing Village, Semarang is a technology-driven approach designed

to tackle the challenges of tidal flooding, which can be caused by tides, rising sea levels, or other coastal factors. This system aims to enhance community resilience by providing timely information, enabling proactive measures, and ultimately safeguarding lives and property.

2. RESEARCH METHODS

This research was conducted using an experimental method to test the author's hypothesis, where ultrasonic sensors were employed for water level measurement and a MySQL platform was used to develop the Early Warning System.[15] The research process was carried out systematically, involving activities such as formulating ideas, reviewing literature, developing prototype design concepts, implementing the design in several phases, and then integrating all components into a complete system[16]

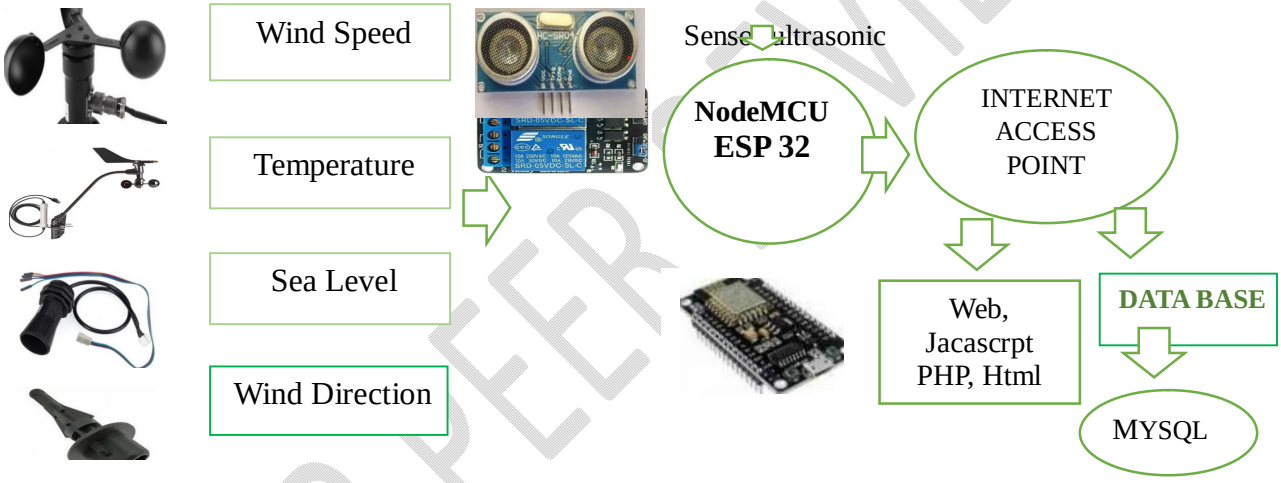

Fig 1. Diagram Rancangan Sistem

General System Flow Design is the process of outlining how the entire system operates, from input to output, and how each component interacts with one another. Figure 1 illustrates the system flow design and demonstrates that the NodeMCU ESP32 microcontroller serves as an IoT device for processing data using a specialized program.[17] Data is collected from an ultrasonic sensor, which measures the distance to the water surface. Once the data is gathered, the microcontroller processes it based on a pre-programmed distance threshold. If the water level data meets the predetermined conditions, it is displayed on the website. Additionally, if the data indicates an Alert or Danger status, an alarm is triggered to signal an emergency.

3. RESULTS AND DISCUSSION

1) Database Design[18]

Database design is crucial for creating systems that require data processing and management. For an early flood detection system using MySQL [19] database, the database design must support efficient data storage, retrieval, and processing to ensure accurate and timely flood monitoring

Table 1. Table Design

tb_sensor	Nama Field	Id	Sensor	Date
	Type of Data	Int	int	Timestamp
	Length of size	Auto increment	11	Auto

User Table:
Create Table tb_users (Id IntAuto_Increment Primary Key, Username Varchar(50) Not Null, email Varchar(100), Created_At Timestamp Default Current_Timestamp);
Product Table
Create Table Tb_Products (Id Int Auto_Increment Primary Key, Product_NameVarchar(100) Not Null, Price Decimal(10, 2) Not Null, Stock_Quantity Int Default 0);
Usage In Queries
Adding data: Insert Into Tb_Users (Username, Email) Values
Retrieving Data By Id:
Select * From Tb_Users Where Id = 1;
Updating Data By Id:
Update Tb_Users Set Email = hmkl @sinus.ac.id = 1; Delete From Tb_Users Where Id = 1;

The tb_sensor table is used to store data collected from sensor devices. This data may include various types of information such as temperature, wind speed, wind direction, and sea water level as generated by the sensors. The tb_sensor table can be modified and customized based on the specific needs of the system or application being used. The table name follows a common naming convention, where tb is an abbreviation for "table," and sensor indicates that the table is related to sensor data.

2) Water Surface Level Reading Hardware[20]

Microprocessor integrated into an integrated board (Wemos D1 Mini). The water surface level is measured using an ultrasonic sensor, specifically a waterproof type such as the HC-SR04[21]This ultrasonic sensor module emits ultrasonic waves after receiving a trigger pulse of at least 100 microseconds (μs). The echo pin switches to High once the ultrasonic module has completed sending the waves and switches to Low after receiving the feedback or after 38 milliseconds (ms) if the object is at a certain distance. For ultrasonic sensors, a travel time of 1

ms corresponds to a distance of 0.0343 cm (round trip), so a travel time of 38 ms translates to approximately 650 cm (or 6.5 meters).

The sensor, such as the HC-SR04, requires a trigger signal to measure distance. Microcontrollers or development boards, including the ESP32 or Raspberry Pi, can be used to interface with the ultrasonic sensor and process the measurements.[22]JavaScript plays a crucial role in controlling and processing the signals, particularly when using a server environment such as Node.js or a web server running on the ESP32. In these setups, JavaScript can handle the signal processing from the ultrasonic sensor, manage data flow, and interface with the microcontroller to interpret and act upon the sensor data. This setup allows for real-time monitoring and interaction with the ultrasonic sensor data through a web interface or server-side application. [23]. IoT-Based Early Warning and Mitigation System for Rob Flood Detection in Sea Waters Using JavaScript and MySQL. In this IoT-based early warning and mitigation system for detecting rob floods, various technologies are integrated to monitor, analyze, and respond to sea conditions that may lead to tidal flooding. The system employs a combination of hardware and software components to provide real-time alerts and facilitate proactive measures.

3) **System Components**

a) **IoT Hardware**[24]

Water Level Sensor: Measures sea water height in real-time to monitor and detect changes in sea conditions. Ultrasonic Sensors: These sensors measure the distance to the water surface by emitting ultrasonic waves and calculating the time it takes for the waves to return. They are effective for detecting water level changes from above the surface. Pressure Sensors: These sensors are placed underwater and measure the pressure exerted by the water column above them. The pressure data is then used to calculate the water depth. Pressure sensors are suitable for direct measurements of water depth.

b) Communication Module[25]

Communication Module: Facilitates the transmission of data from the sensor to the server, enabling remote monitoring and data management. ESP32 with Wi-Fi: This module integrates a microcontroller with built-in Wi-Fi capabilities, allowing it to connect to wireless networks and send sensor data to a server or cloud-based platform. The ESP32 is commonly used for its versatility, low power consumption, and ease of integration with various sensors and system. A module for sending data from the sensor to the server. Examples include ESP32 with Wi-Fi.

c) Power Supply

Power Supply: Provides the necessary electrical power to the IoT devices, ensuring their proper operation and functionality. The power supply component is crucial for maintaining continuous operation of the system, including sensors, communication modules, and processing units. It can include: Battery Packs: Portable and rechargeable batteries that provide power to IoT devices in remote or off-grid locations. AC-DC Adapters: Converts alternating current (AC) from a wall outlet to the direct current (DC) required by the IoT devices. Solar Panels: Renewable energy sources that can be used to charge batteries or provide direct power, particularly in outdoor or isolated environments.

d) Backend Software[26]

Server: Manages the data received from sensors, processes the data, and executes the necessary logic. The server is responsible for handling requests, performing data analysis, and orchestrating the overall system functionality. Node.js: A runtime environment that enables the server to handle asynchronous operations and manage real-time data effectively. Node.js is well-suited for applications that require high performance and scalability, making it an excellent choice for handling data from IoT sensors. Database: Stores sensor data and related information for retrieval, analysis, and historical record-keeping. The database is essential for managing and querying large volumes of data efficiently. MySQL: A relational database management system that organizes data into structured tables and supports SQL queries. MySQL is widely used for its reliability, ease of use, and robust performance in handling large datasets.

e) Frontend Software[27]

User Interface: A web application designed for monitoring data and receiving alerts. This interface allows users to interact with the system, view real-time data, and receive notifications regarding critical events. Key features include: **Real-Time Monitoring**: Displays live sensor data and visualizes trends and alerts through charts, graphs, or dashboards. **Alert Management**: Provides notifications and alerts based on predefined thresholds or conditions, ensuring users are informed of any potential issues or emergencies. **JavaScript**: Utilized for creating dynamic and interactive web applications. JavaScript enables the implementation of client-side logic, real-time updates, and seamless user interactions. It is often used alongside frameworks like React, Angular, or Vue.js to enhance the functionality and user experience of the web application.

4) System Design

Sensor and Development Board Connection [28]

Waer Sensor: **Connection to Development Board**: Connect the water level sensor to a development board such as the **ESP32**. This involves wiring the sensor's output pins to the appropriate input pins on the ESP32, ensuring that the connections are secure and correctly configured. **Data Measurement and Transmission**: The sensor measures the water height in real-time. The collected data is transmitted to the ESP32 for further processing. The sensor's readings are continuously sent to the ESP32 to monitor changes in water level accurately.[28]

ESP32. Data Collection: The ESP32 collects data from the water level sensor. It processes the incoming sensor data, which includes interpreting the readings and applying any necessary calculations or logic based on the sensor's output. **Data Transmission**: After processing, the ESP32 sends the collected data to the server via Wi-Fi. The ESP32 connects to a wireless network and transmits the data to a backend server or cloud service, where it can be stored, analyzed, and accessed through the web application.[29]

5) Server Setup

Node.js Server:

a) **Provides API Endpoints**: The Node.js server exposes API endpoints that allow it to receive data from the water level sensor. These endpoints handle incoming HTTP requests, process the sensor data, and ensure it is correctly formatted and received by the server.[30]

b) **Processes and Stores Data**: Upon receiving the sensor data, the server processes it according to the defined logic (e.g., calculating water levels, checking thresholds). The processed data is then stored in a **MySQL** database, ensuring it is available for historical analysis and retrieval.[31]

c) **Provides APIs for Frontend**: The server also exposes APIs for the frontend application. These APIs allow the web application to retrieve and display data and alerts. The server responds to requests from the frontend, providing real-time data updates and alert information as needed.[26]

d) **MySQL Database**:

Data Storage: The MySQL database stores sensor data, configuration settings, and related alert information. This ensures that all relevant data is persistently available for analysis, monitoring, and historical reference.

e) **Possible Table Structures**:

```
CREATE TABLE sensor_data (
id INT AUTO_INCREMENT PRIMARY KEY,
sensor_idVARCHAR(50),
timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
water_level FLOAT
);
CREATE TABLE flood_alerts (
id INT AUTO_INCREMENT PRIMARY KEY,
alert_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
water_level FLOAT,
alert_message TEXT
);
```

6) **Early Warning System Setup**

a) **Flood Detection Logic: Define a Water Level Threshold**: Establish a specific water level threshold that indicates a risk of rob flooding. This threshold should be based on historical data, expert recommendations, and safety considerations. It represents the critical level at which intervention or alerting is necessary. **Implement Detection Logic on the Server**: Develop and deploy logic on the Node.js server to continuously monitor incoming data from the sensors. The server should compare the measured water levels against the defined threshold. If the water level exceeds this threshold, the server generates an alert to signal potential flooding. This involves: [32]

b) **Data Comparison**: Comparing real-time sensor data with the predefined threshold value and **Alert Generation**: Triggering an alert when the threshold is exceeded, which may include logging the event, creating an alert record, and preparing notification content.

c) **Alert Delivery: Store Alerts in the Database**: Once an alert is generated, it should be stored in the `flood_alerts` table in the MySQL database. This allows for tracking, historical analysis, and retrieval of alert information. Each alert entry should include details such as the time of the alert, water level at the time, and any associated message. **Send Notifications**: Depending on the urgency and requirements, notifications can be sent via email or SMS. This can be achieved using various services and APIs

7) Implementation

a) Code node.js to collect and process data

```
javascript
Copy code
const express = require('express');
const mysql = require('mysql');
const app = express();
const port = 3000;
app.use(express.json());
// MySQL connection
const connection = mysql.createConnection({
host: 'localhost',
user: 'root',
password: '',
database: 'flood_detection'
});
connection.connect();
// Endpoint to receive sensor data
app.post('/api/sensor', (req, res) => {
const { sensor_id, water_level } = req.body;
const timestamp = new Date().toISOString();
 // Save data to MySQL
const query = 'INSERT INTO sensor_data (sensor_id, water_level, timestamp) VALUES (?, ?, ?)';
connection.query(query, [sensor_id, water_level, timestamp], (err, results) => {
if (err) throw err;
// Check for flood alert
```

```
if (water_level> 100) { // Example threshold
const alertMessage = 'Flood warning: water level is too high!';
const alertQuery = 'INSERT INTO flood_alerts (water_level, alert_message) VALUES (?, ?)';
connection.query(alertQuery, [water_level, alertMessage], (err, results) => {
if (err) throw err;
console.log('Flood alert saved to database.');
});
}
res.send('Data received');
});
});
app.listen(port, () => {
console.log(`Server running on http://localhost:${port}`);
});
```

JavaScript Code for Frontend

Use a framework like React or Vue.js to build the user interface:

```
html
Salin kode
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Flood Detection Dashboard</title>
</head>
<body>
<h1>Flood Detection Dashboard</h1>
<div id="alerts"></div>
<script>
async function fetchAlerts() {
const response = await fetch('http://localhost:3000/api/alerts');
const data = await response.json();
const alertsDiv = document.getElementById('alerts');
alertsDiv.innerHTML = '';
data.forEach(alert => {
```

```
const alertElement = document.createElement('div');
alertElement.textContent = `Alert at ${alert.alert_time}: ${alert.alert_message}`;
alertsDiv.appendChild(alertElement);
});
}
fetchAlerts();
setInterval(fetchAlerts, 10000); // Refresh alerts every 10 seconds
</script>
</body>
</html>
```

b) Sensor Reading Interval and Data Handling

Sensor Reading Interval:The system is configured to read water level data at one-minute intervals. This means that the data reflects changes in water level every minute, providing periodic updates on the water level.[33]

8) **Data Transmission to Realtime Database**[34]:

a) **ESP32 Internet Access**: The ESP32 microcontroller requires an internet connection to send data to the Realtime Database. This necessitates a reliable internet access point for continuous data transmission.

b) **Database Configuration**: The Realtime Database is configured to store four types of data:**Water Level**: Measured in centimeters, **Temperature**: Ambient temperature readings.**Wind Speed**: Measurement of wind speed.**Wind Direction**: Direction from which the wind is blowing.

c) **Rate of Change**: The system calculates the rate of change in water level in centimeters per minute. This metric helps in predicting potential flooding events based on the rate at which the water level is rising.**Forecasting Flood Likelihood**:Using the rate of change and real-time data, the online monitoring system and warning devices can predict the likelihood and timing of floods. By analyzing trends and rate changes, predictions can be made for possible flood events.**Communication Configuration**:A JavaScript-based configuration is used to manage communication with the Realtime Database. This configuration ensures that data is properly sent to and retrieved from the database.

9) Warning Hardware Setup

Microprocessor and Connectivity[35]

a) **ESP32**: The warning hardware uses the ESP32 microprocessor, which connects to the Realtime Database over the internet. The ESP32 reads data from the database to monitor water levels and other environmental factors.[36]

b) **Warning Mechanism**:

Siren: A siren is used to provide varying warning signals. The system is programmed to activate the siren under two critical conditions based on water level and rate of change:**Approximately 15 Minutes Before.Flooding**: When water levels are between 210 cm and 250 cm.**Approximately 5 Minutes Before Flooding**: When water levels exceed 250 cm.The siren issues early warnings to alert individuals about impending flooding, giving them time to take preventive measures.[37]

10)Testing Results[38]

a) **Web Monitoring Prototype**: The developed web monitoring system allows public access through a web browser.

b) **Real-Time Updates**: The data displayed on the web interface is updated in real-time, reflecting the current information stored in the Realtime Database. This ensures that users have access to the most recent data for effective monitoring and decision-making.

The display, accessible through a web browser, is as follows:

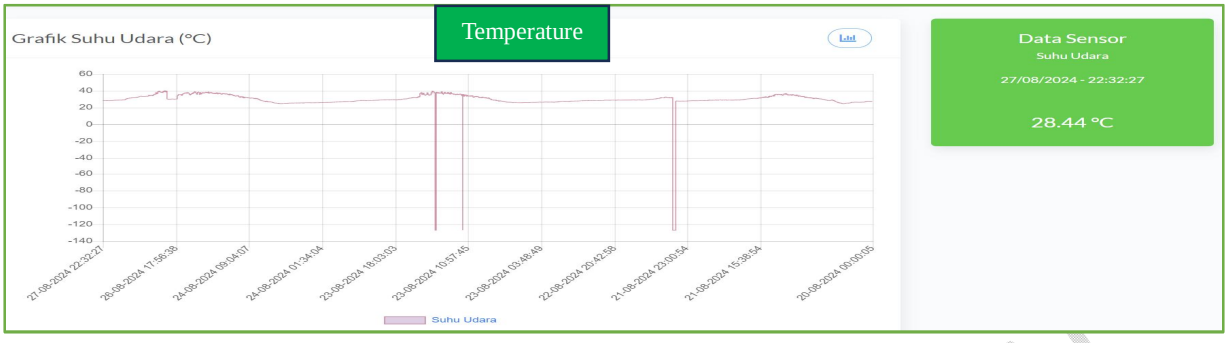

Figure 2. Parameter display on Web SensorKu

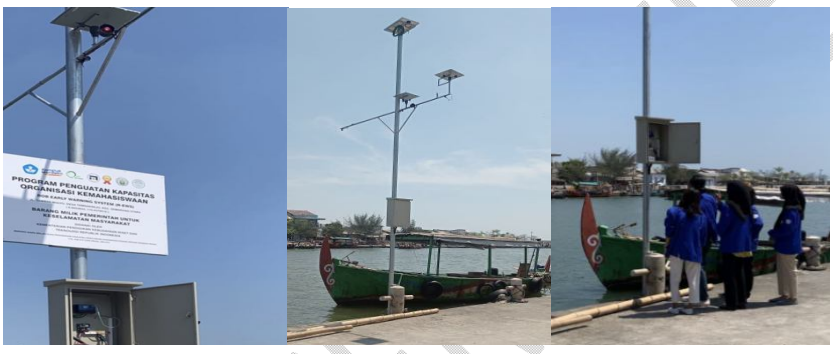

Figure 3. Condition of R-EWS installation location

CONCLUSION

Rob-EWS has become an effective tool in reducing the risk of tidal flood disasters in Tambak Lorok Village in Semarang City. Early detection capabilities, combined with robust data analysis and timely notification, have made a significant contribution to reducing the impact of tidal floods. Continued improvement and expansion will further enhance the effectiveness and resilience to future flood events.

DATA AVAILABILITY

All necessary data has been assembled along with accompanying documentation. This study will help researchers identify crucial areas of the Web-Based ROB Disaster Mitigation Information System in Tambak Lorok Village, Semarang, Central Java.

CONSENT

The author(s) have gathered and archived respondents' written consent in accordance with international or university standards.

DATA AVAILABILITY

The manuscript includes all the datasets created or analyzed during this study

DISCLAIMER (ARTIFICIAL INTELLIGENCE)

The authors declare that no generative AI tools, including Large Language Models or text-to-image generators, were used in the creation or editing of this manuscript. This confirms that AI tools were not involved in the development or editing process of the manuscript.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCE

1. I. Akbar, H. W. Poerbo, and W. K. Soedarsono. Adaptive urban design principles for land subsidence and sea level rise in coastal area of Tambak Lorok, Semarang. *IOP Conference Series: Earth and Environmental Science*, IOP Publishing, 2019, p. 12005.
2. A. S. Nurhayati. Cultural Values' Dimensions of Tambak Lorok Fisheries Community. *CL-LAMAS 2019: Proceedings of the First International Conference on Culture, Literature, Language Maintenance and Shift, CL-LAMAS 2019, 13 August 2019, Semarang, Central Java, Indonesia*, European Alliance for Innovation, 2019, p. 33.
3. A. Luthfi, F. Husain, K. B. Prasetyo, M. S. Mustofa, and A. B. Santoso. Resilience of Small Fishermen in the Development of Tambak Lorok Marine Tourism Village in Semarang City. *Proceedings of the 3rd International Conference on Social and Political Development (ICOSOP 3)*, 2019, pp. 126–131.
4. C. Amin, S. Sukamdi, and R. Rijanta. Exploring Typology of Residents Staying in Disaster-Prone Areas: A Case Study in Tambak Lorok, Semarang, Indonesia. *Forum Geografi*, 2018, pp. 24–37.
5. A. R. Okvitasari, A. R. Fatoni, A. Bahtiar, N. Faridatussafura, A. Hermanto, and M. F. N. Aulady. The impact of climate change on potential rob floods and its effect on regional spatial planning on the Surabaya coast. *Calam. A J. Disaster Technol. Eng.*, vol. 1, no. 2, pp. 114–126, 2024.
6. P. S. Herbanu, A. Nurmaya, R. M. Nisaa, and R. A. Wardana. The zoning of flood disasters by combining tidal flood and urban flood in Semarang City, Indonesia. *IOP Conference Series: Earth and Environmental Science*, IOP Publishing, 2024, p. 12028.
7. C. Bell. *MySQL for the Internet of Things*. Springer, 2016.
8. O. Ozdilek and D. Z. Seker. A web-based application for real-time GIS. *XXth ISPRS Congress*, 2004, pp. 1–5.
9. N. Chatterjee, S. Chakraborty, A. Decosta, and A. Nath. Real-time communication application based on android using Google firebase. *Int. J. Adv. Res. Comput. Sci. Manag. Stud*, vol. 6, no. 4, 2018.
10. A. de Jesús Plasencia-Salgueiro. IoT-based Drip Irrigation Monitoring and Controlling

System using NB-IOT, ESP32+ Raspberry Pi. *Rev. Cuba. Ciencias Informáticas*, 2023.
11. A. Meier and M. Kaufmann, *SQL & NoSQL databases*. Springer, 2019.
12. N. Banothu, S. Bhukya, and K. V. Sharma. Big-data: Acid versus base for database transactions. *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, IEEE, 2016, pp. 3704–3709.
13. K. Schweinsberg and L. Wegner. Advantages of complex SQL types in storing XML documents. *Futur. Gener. Comput. Syst.*, vol. 68, pp. 500–507, 2017.
14. M. Darwis, H. A. Al Banna, S. R. Aji, D. Khoirunnisa, and N. Natassa. IoT Based Early Flood Detection System with Arduino and Ultrasonic Sensors in Flood-Prone Areas. *J. Tek. Inform.*, vol. 16, no. 2, pp. 133–140, 2023.
15. M. D. Prasad, M. Vejendla, N. R. Sai, K. G. Guptha, and P. D. K. Reddy. *Emerging Technologies Transforming the Future.* GCS PUBLISHERS, 2023.
16. J. C. Patni, H. K. Sharma, R. Tomar, and A. Katal.*Database Management System: An Evolutionary Approach*. Chapman and Hall/CRC, 2022.
17. D. Hercog, T. Lerher, M. Truntič, and O. Težak. Design and implementation of ESP32-based IoT devices. *Sensors*, vol. 23, no. 15, p. 6739, 2023.
18. K. Chaduvula, B. R. Markapudi, and C. R. Jyothi. Design and Implementation of IoT based flood alert monitoring system using microcontroller 8051. *Mater. Today Proc.*, vol. 80, pp. 2840–2844, 2023.
19. R. A. Lakshmi, M. M. Lakshmi, P. Swetha, and T. D. Prakash. Flood Detection and Early Warning System .2020.
20. S. Konde and D. S. Deosarkar. OT based water quality monitoring system. *2nd International Conference on Communication & Information Processing (ICCIP)*, 2020.
21. D. Adebayo Adeniyi and N. Ifeagwu Emmanuel. Flood Detection and Monitoring System in Otuoke Community using IOT. 2024.
22. M. M. Gabriel and K. P. Kuria. Arduino uno, ultrasonic sensor HC-SR04 motion detector with display of distance in the LCD. 2020.
23. S. D. K. Moddable, P. Hoddie, and L. Prader. IoT Development for ESP32 and ESP8266 with JavaScript, .
24. R. Fadilah, R. Ruslan, and A. Imran. Development Of Internet Of Things (Iot) Based Flood Early Warning Tools. *J. Electr. Eng. Informatics*, vol. 1, no. 2, pp. 45–50, 2024.
25. A. I. Paganelli *et al.*A conceptual IoT-based early-warning architecture for remote monitoring of COVID-19 patients in wards and at home. *Internet of Things*, vol. 18, p. 100399, 2022.
26. B. Secker. Development of an IoT System for Environmental Monitoring: Software. 2021.
27. A. Albeshri, A. AlreshidI, and A. A. Alanazi. Software Engineering for IoT-Driven Data Analytics Applications.
28. N. Kamal, A. Hammad, T. Salem, and M. Omar. Early Warning and Water Quality, Low-Cost IoT Based Monitoring System. *JES. J. Eng. Sci.*, vol. 47, no. 6, pp. 795–806, 2019.
29. V. Menon, R. Arjun Rathya, A. Prasad, A. Gopinath, N. B. Sai Shibu, and G. Gayathri, Exploring iot-enabled multi-hazard warning system for disaster-prone areas. *Advances in Computing and Network Communications: Proceedings of CoCoNet 2020, Volume 1*, Springer, 2021, pp. 405–422.
30. L. Mänty. Comparison Of Communication Protocols In The Alarm Management For Remote Systems. 2023.
31. G. Suciu, M. Anwar, A. Ganaside, and A. Scheianu. IoT time critical applications for

environmental early warning. *9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, IEEE, 2017, pp. 1–4.
32. M. Kaur, P. D. Kaur, and S. K. Sood. ANFIS-based flood detection and vulnerability assessment framework. *Hydrol. Sci. J.*, vol. 67, no. 15, pp. 2310–2326, 2022.
33. Y. Permatasari, M. R. Firdaus, H. Zuhdi, H. Fakhrurroja, and A. Musnansyah. Development of IoT Control System Prototype for Flood Prevention in Bandung Area. *JOIV Int. J. Informatics Vis.*, vol. 7, no. 3, pp. 1016–1021, 2023.
34. N. Byaruhanga, D. Kibirige, S. Gokool, and G. Mkhonta. Evolution of Flood Prediction and Forecasting Models for Flood Early Warning Systems: A Scoping Review. *Water*, vol. 16, no. 13, p. 1763, 2024.
35. V. V Krzhizhanovskaya *et al*. Flood early warning system: design, implementation and computational modules. *Procedia Comput. Sci.*, vol. 4, pp. 106–115, 2011.
36. A. F. S. Susanti *et al.*Instrumentation for Monitoring and Early Warning of Tidal Flood Using ESP32S2 Microcontroller and A01NYUB Ultrasonic Sensor in Tambakrejo, Semarang. *IOP Conference Series: Earth and Environmental Science*, IOP Publishing, 2024, p. 12046.
37. R. Mardiati and M. R. Effendi. Early warning system of flood disaster using jsn-sr04 and rainfall sensor based on internet of things. *8th International Conference on Wireless and Telematics (ICWT)*, IEEE, 2022, pp. 1–7.
38. I. Suwarno, A. Ma'arif, N. M. Raharja, A. Nurjanah, J. Ikhsan, and D. Mutiarin. IoT-based lava flood early warning system with rainfall intensity monitoring and disaster communication technology,. *Emerg. Sci. J.*, vol. 4, no. 0, pp. 154–166, 2021.