Accredited SINTA 2 Ranking
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online at: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 8 No. 1 (2024) 169 - 180 e-ISSN: 2580-0760

Development of an Early Warning System Using Social Media for Flood Disaster

I Ketut Kasta Arya Wijaya[1], Ruben Cornelius Siagian[2*]
[1]Universitas Warmadewa, Denpasar, Indonesia,
[2]Department of Physics, Faculty of Mathematics and Natural Science, Universitas Negeri Medan, Medan, Indonesia.
[1]kastaaryawijaya@gmail.com, [2]rubensiagian775@gmail.com.

Abstract

This research paper introduces an innovative prototype system that utilizes IoT technologies for monitoring floodwater levels. The integration of an ultrasonic sensor, ESP8266 microcontroller, Arduino IDE, and the ThingSpeak platform aims to establish a robust flood monitoring solution. The paper provides a thorough exploration of the system's background, the problem it addresses, the methodology employed, and the obtained results, along with insights into future research directions. The study meticulously outlines the design, implementation, and programming code for data collection and transmission within the system. Through extensive field testing and meticulous data analysis, the paper evaluates the accuracy and effectiveness of the proposed flood monitoring solution. Notably, the research underscores the advantages of IoT, emphasizing real-time data collection, logging, and analysis as essential components for efficient flood management. In addition, the paper elucidates step-by-step instructions for configuring Telegram notifications through the ThingSpeak React app, enhancing the practical applicability of the developed system. The research effectively highlights the potential of IoT in flood monitoring, showcasing its superior accuracy and effectiveness compared to traditional methods. By demonstrating the feasibility and advantages of IoT in the context of flood monitoring, this study contributes valuable insights, enriching existing knowledge and paving the way for future advancements in the field. The research encourages continued exploration of advanced techniques to strengthen flood monitoring and management strategies. Ultimately, this work presents a comprehensive IoT-based prototype for floodwater monitoring, offering invaluable insights and fostering the promising role of IoT technologies in this critical domain.

Keywords: *flood monitoring; water level measurement; thing speak API; user-friendly environment; field testing*

How to Cite: Ketut Kasta Arya Wijaya and R. C. Siagian, "Development of an Early Warning System Using Social Media for Flood Disaster", J. RESTI (Rekayasa Sist. Teknol. Inf.), vol. 8, no. 1, pp. 169 - 180, Feb. 2024.
DOI: https://doi.org/10.29207/resti.v8i1.5087

1. Introduction

BPBD Kota Medan is an abbreviation for the Medan City Regional Disaster Management Agency [1]. It is a local government institution that coordinates, regulates, and implements disaster management activities in the Medan City area. The agency's primary goal is to minimize the risk and loss resulting from natural and non-natural disasters and assist and protect affected communities. BPBD Kota Medan collaborates with various stakeholders to strengthen disaster management efforts in the city [2]. The agency's critical role in disaster response and management includes providing timely and effective emergency services, conducting risk assessments, and developing disaster preparedness plans [3]. BPBD Kota Medan is an essential component of the disaster management system in Medan City, ensuring the safety and well-being of the community during times of crisis [4].

BPBD Kota Medan plays a crucial role in managing disasters and reducing their impact on the community. The agency is responsible for various functions, including disaster planning, prevention, coordination, emergency response, and recovery efforts [5]. This paper aims to elaborate on the main functions of BPBD Kota Medan and how they contribute to disaster management in the city.

An Early Warning System (EWS) is a system that provides advance notice of potential disasters in a given region [6]. In the case of BPBD, the EWS is a vital tool for identifying and communicating threats to communities. Its objective is to provide timely and accurate information on impending disasters, allowing for prompt and effective preventive, mitigation, and

Received: 11-05-2023 | Accepted: 14-02-2024 | Published Online: 22-02-2025

evacuation measures [7]. By using sophisticated technologies and protocols to monitor environmental factors that may indicate disaster likelihood, the system enables the BPBD to respond more quickly and accurately [8]. The information gathered is analyzed and shared with relevant stakeholders, ensuring that they are well-prepared. Implementing an EWS is critical for ensuring the safety and well-being of communities at risk of natural disasters, and by leveraging technology and expert knowledge, decision-makers can protect vulnerable populations and prevent or mitigate the negative effects of disasters. [9].

The Internet of Things (IoT) can be used in an Early Warning System (EWS) for floods to improve the speed and accuracy of flood information [10]. IoT sensors can be installed at critical points susceptible to flooding to collect real-time data on water level, temperature, humidity, and rainfall, which will be processed by the IoT system to provide information about the current flood condition [11]. This information can be disseminated to BPBD or affected communities through media such as SMS, email, or mobile applications. By utilizing IoT technology in the EWS for floods, BPBD can obtain flood information quickly and accurately, enabling them to take faster and more appropriate action to face flood disasters, such as evacuating communities or providing logistics assistance. IoT technology can also assist BPBD in monitoring flood conditions more effectively and efficiently, resulting in a more optimal response to flood disasters [12].

Ultrasonic sensors emit sound waves into water and measure the time it takes for the sound waves to return, which determines the height of the water level [13]. Capacitive sensors detect changes in capacitance as the water reaches the sensor, and the capacitance increases as the water level rises [14]. Voltage sensors measure the voltage difference between two electrodes placed within the water, which increases as the water level rises [15]. All three sensors transmit data to an ESP8266, which processes and transmits the information to a server or cloud platform via Wi-Fi. This data can be analyzed to provide useful information about flood conditions. These sensors are based on physics principles, providing accurate and reliable data for detecting flood conditions and helping to take emergency actions.

An Ultrasonic Sensor is a device that uses high-frequency sound waves to detect the distance of an object [16]. This paper, on the other hand, aims to provide an overview of the main functions of the Regional Disaster Management Agency of Medan City (BPBD Kota Medan) and its role in disaster management. The BPBD Kota Medan is responsible for disaster planning, prevention, coordination, emergency response, and recovery efforts. It collaborates with various stakeholders to minimize the impact of disasters and ensure the safety of the community. The agency also utilizes an early warning system (EWS) to provide warning of potential disasters and respond more quickly and accurately to reduce the impact on communities [17]. The paper suggests that IoT can be utilized to improve the speed and accuracy of flood information in a particular area. By leveraging cutting-edge technology and expert knowledge, BPBD Kota Medan aims to protect vulnerable populations and prevent or mitigate the negative effects of natural disasters.

The gap in research within the paper on BPBD Kota Medan's disaster management functions lies in the absence of emphasis on leveraging Internet of Things (IoT) technology within the Early Warning System (EWS) for floods. While acknowledging the EWS, the paper lacks depth regarding IoT's role in this system. By delving into IoT's potential integration into the EWS, the paper can significantly enhance disaster management practices not only in Medan City but also in broader contexts. Consequently, future research should centre on practically implementing and evaluating IoT within BPBD Kota Medan's flood EWS, considering both the advantages and challenges posed by such technology. Constraints and Research Gap:

One identified research gap is the lack of focus on the use of IoT technology in the EWS for floods in BPBD Kota Medan. Although the paper acknowledges the EWS, it does not provide detailed information on the potential benefits, implementation, and challenges of IoT technology in the system.

Future research can focus on the practical implementation and evaluation of IoT technology in the EWS for floods in BPBD Kota Medan. This research should consider the potential benefits, challenges, and effectiveness of the system in improving disaster management practices.

Further investigation is needed to explore the specific benefits of integrating IoT technology into the EWS, such as enhanced data collection, faster response times, and improved decision-making. Additionally, the research should address potential challenges, such as infrastructure requirements, data security, and maintenance of IoT devices.

By addressing these constraints and research gaps, the proposed innovation of utilizing IoT technology in the EWS for floods can contribute to the advancement of disaster management practices in Medan City. The research can provide valuable insights into such a system's practical implementation, benefits, and challenges, ultimately improving the effectiveness and efficiency of disaster response and mitigation efforts.

2. Research Methods

One of the primary functions of BPBD Kota Medan is to develop disaster planning and strategies for the city. This involves identifying potential risks, preparing emergency response plans, and coordinating disaster management efforts.

BPBD Kota Medan is a government agency that plays a crucial role in disaster management. Its main functions include disaster prevention, coordination of disaster management, emergency response, and evaluation and post-disaster recovery is shown in Table 1. By educating the public, providing training, monitoring environmental conditions, and taking preventive measures [18], BPBD Kota Medan can reduce the number and severity of disasters that occur in the city. It also coordinates all parties involved in disaster management, including government agencies, non-governmental organizations, and the general public, to ensure that disaster management efforts are integrated and effective [19]. In the event of a disaster, BPBD Kota Medan provides assistance and protection to those affected by evacuating, providing medical care, equipment, and other forms of aid [20]. BPBD Kota Medan evaluates the damage caused by disasters and initiates recovery efforts, including rebuilding damaged infrastructure and providing support to affected communities [21]. By fulfilling these roles effectively, BPBD Kota Medan can minimize the damage caused by disasters and ensure a more efficient response to emergencies.

Figure 1. Regional Disaster Management Agency
Source: bantwana, 2023

Previous research and literature have provided valuable insights into the functions and roles of the Regional Disaster Management Agency of Medan City (BPBD Kota Medan) and the use of an Early Warning System (EWS) in disaster management as seen in Figure 1. Here are some key findings and constraints from previous research:

BPBD Kota Medan's Functions: Previous research highlights the primary functions of BPBD Kota Medan, which include disaster planning, prevention, coordination, emergency response, and recovery efforts [22]. The agency plays a crucial role in identifying potential risks, developing emergency response plans, and coordinating disaster management activities [23]. It also collaborates with various stakeholders to ensure integrated and effective disaster management efforts.

Early Warning System (EWS): Previous studies emphasize the importance of implementing an EWS for timely and accurate information on impending disasters [24]. The EWS enables the BPBD to identify and communicate threats, initiate preventive measures, and facilitate evacuation if necessary [25]. The system relies on monitoring environmental factors that may indicate disaster likelihood [26]. Effective implementation of an EWS can help protect vulnerable populations and mitigate the negative effects of disasters.

Internet of Things (IoT) in EWS: Research suggests that IoT technology can enhance the speed and accuracy of flood information in an EWS [10]. IoT sensors installed at critical points susceptible to flooding can collect real-time data on water levels, temperature, humidity, and rainfall [27]. This data, processed by the IoT system, can provide current flood condition information to BPBD or affected communities through various communication channels [2]. The utilization of IoT technology can enable faster and more appropriate actions during flood disasters.

Sensors for Flood Detection: Previous studies discuss the use of ultrasonic sensors, capacitive sensors, and voltage sensors for detecting flood conditions [28], [29]. These sensors, based on physics principles, provide accurate and reliable data on water levels, which is crucial for emergency actions and decision-making during flood events.

Table 1. BPBD’s disaster management strategies description

BPBD's Disaster Management Strategies	Description
Environmental Condition Monitoring	BPBD monitors environmental conditions, such as weather patterns and seismic activity, to identify potential disaster risks.
Risk Analysis	BPBD conducts a risk analysis based on collected data to determine the level of disaster threat in their working area.
Early Warning System	BPBD employs an early warning system to provide timely and accurate information about impending disasters to the public.
Disaster Evacuation and Mitigation System	BPBD prepares an integrated and coordinated plan for disaster evacuation and mitigation with various parties, including the community, government agencies, and non-governmental organizations. This involves identifying safe evacuation routes and shelters, as well as preparing emergency supplies and equipment.

2.1 *Review of components used in the research*

Programming is the process of creating instructions for a computer to follow, typically in the form of software programs [30]. In this prototype, programming refers to creating code for the ESP8266 microcontroller using the Arduino IDE. Data logging involves collecting and storing data from a system or device over time [31]. In this prototype, the ultrasonic sensor measures the distance between the water level and the sensor, and the data is transmitted to the microcontroller for logging and analysis. Data analysis involves examining the data to find insights and patterns. ThingSpeak is used to receive and log the data, and its tools are used to analyze and visualize the data. Conducting research involves testing and validating the system's effectiveness in real-world scenarios, comparing results to established methods, and analyzing collected data [32]. The prototype includes an ultrasonic sensor that measures distance and transmits data to the ESP8266 microcontroller, which converts and uploads it to ThingSpeak for logging and analysis [33], [34].

Table 2. Prototype component description

Component	Description
Ultrasonic Sensor	A sensor is used to measure the water level or other objects above the water surface. It can generate data in the form of distance or relative height. Examples: HC-SR04, JSN-SR04T
Speaker	An audio component is used to produce sound or alerts if the water level exceeds a certain threshold.
Cable	Connecting cables are used to link various components in the system. Can be jumper cables, data cables, or other types of connectors.
ESP8266 Module	A microcontroller module with Wi-Fi capabilities is used to connect the system to the internet and send data to the Thingspeak platform. Examples: NodeMCU, Wemos D1 Mini
Microcontroller/Processor	An electronic component that acts as the brain of the system, controlling sensor operations, data processing, and device control. Can be a microcontroller or microprocessor, such as Arduino, Raspberry Pi, or ESP32.
Power Supply	A power source to operate the entire system, such as batteries, adapters, or solar panels.
Communication Module	Additional modules such as GSM modules, LoRa modules, or Bluetooth modules send data to the Thingspeak platform through the appropriate communication channel.
Additional Sensors (optional)	Additional sensors such as temperature, humidity, or atmospheric pressure sensors can be used to

The prototype for this project consists of multiple components, such as an ultrasonic sensor, ESP8266 microcontroller, Arduino IDE, and ThingSpeak [35]. The ultrasonic sensor measures the distance between the water level and the sensor [36]. The ESP8266 microcontroller processes the data from the sensor and converts it into meaningful values [37].

The research involves field testing the prototype in real-world scenarios, comparing its results to established methods, and analyzing the collected data to assess the system's accuracy and effectiveness [38]. The ESP8266 microcontroller is used to input data from the ultrasonic sensor to ThingSpeak, and the Arduino IDE provides a user-friendly environment for programming the microcontroller. The components used in this system include the ESP8266 microcontroller, ultrasonic sensor, and ThingSpeak platform, with programming based on the Arduino platform for controlling the microcontroller and interfacing with other electronic components is shown in Table 2 [39].

2.2 *Concept of implementation system*

The recommended flood detector flowchart integrates various components, including ultrasonic sensors, rain sensors, and temperature sensors, along with the use of GSM, Arduino ESP8266, and communication output to platforms such as WhatsApp, Telegram, and mobile messaging. The system begins with the initialization of the flood detector, ensuring the readiness of all components to operate. The ultrasonic sensor, rain sensor, and temperature sensor continuously collect data related to water level, humidity, and temperature, respectively.

The data collected from the ultrasonic sensor is used to monitor the water level, and by comparing it to a predetermined threshold, the system can determine if the water level has crossed the safe limit, signalling the need for preventive or warning measures. Simultaneously, the rain sensor is activated to identify rain, triggering the necessary precautions or warnings upon detection. In addition, the temperature sensor plays a key role in the analysis of the flood situation by providing information on the ambient temperature and facilitating the understanding of the overall flood conditions.

The system can recognize patterns or anomalies that contribute to flooding by monitoring the temperature. To ensure timely communication and notification to the system owner, the flood detector is designed to send notifications via text message. The owner receives updates on the detected flood conditions, enabling quick action and response. After completing the required actions, this flood detector system is either stopped or moves on to the next sensor, ready to repeat the data collection and analysis process.

Implementing this comprehensive and efficient flood detector flowchart, which integrates ultrasonic, rain, and temperature sensors, along with GSM, Arduino ESP8266, and various messaging platforms, the system can effectively monitor and notify owners of potential flood events. This enables timely intervention and proactive measures to minimize the impact of flooding. Figure 2 shows the concept diagram of the system implementation.

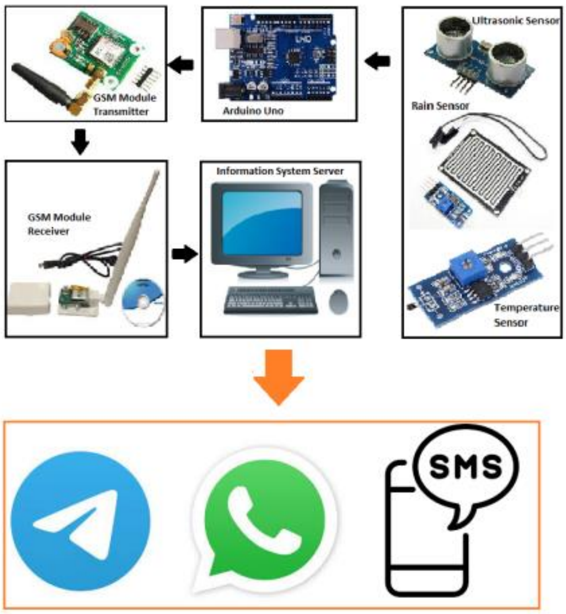

Figure 2. Concept of implementation system diagram

2.3 Implementation using the program

The program uses ultrasonic sensors to detect flooding and sends the data to the ThingSpeak platform for analysis. The program sets up the WiFi connection and ThingSpeak credentials, initializes the sensor, and waits for the WiFi connection is shown in Table 3.

WiFi and ThingSpeak settings: In this step, we need to do the initial configuration to set up the WiFi connection and ThingSpeak settings. First, we include the necessary libraries, namely ESP8266WiFi.h and ThingSpeak.h. Next, we set the WiFi parameters, such as SSID (WiFi network name) and password. Then, we set the ThingSpeak settings, including the server address and API key. Next, we create a WiFiClient object with the name "client". Then, we set the pin trigger and pin echo for the ultrasonic sensor. Next, we initialize serial communication with a baud rate of 115200. Finally, we connect the device to the predefined WiFi network and wait for a successful connection. Once connected, we print "Connected to WiFi" as a sign that the connection has been successful. Finally, we initialize the ThingSpeak library using the "client" object.

Main Loop: After establishing the connection, the system starts capturing distance data from the sensor, and then sends it to ThingSpeak via its API, depending on the validation of the received data. This cyclic process takes place continuously. To facilitate notifications via Telegram when the water level exceeds a predefined threshold, it is recommended to use ThingSpeak's React app. This involves the configuration of a Telegram bot, the creation of a React in ThingSpeak with defined trigger parameters, and the integration of actions designed to send relevant information to the configured Telegram bot. Seamless integration with the React app required programming, allowing the ESP8266 WiFi module to access the Telegram bot API using the Arduino IDE. This carefully designed program provides a straightforward and efficient solution for flood detection, leveraging electronic sensors and cloud-based data analysis.

Table 3. ESP-8266 Program Integration with Thingspeak for WhatsApp, Telegram, and Mobile Messaging

Inputs:	**Initialize Libraries, Variables, and Connection**
ssid: WiFi SSID	\
password: WiFi password	**Setup:**
thingspeakHost: ThingSpeak server address	Begin Serial Communication
writeAPIKey: ThingSpeak write API key	Connect to Wi-Fi and ThingSpeak
trigPin: Sensor trigger pin	Initialize GSM Module
echoPin: Sensor echo pin	\
threshold distance: Threshold distance for data validity	**Main Loop:**
\	Read Sensor Data:
Functions:	Trigger Ultrasonic Sensor
setup():	Measure Echo Duration for Distance
Initialize serial communication for debugging	Check the Validity of Distance Reading
Set sensor trigger pin as OUTPUT and echo pin as INPUT	\
Connect to WiFi using the provided credentials	Data Transmission:
\	Send Data to ThingSpeak:
loop():	Set Field Value for Distance
Measure distance using the sensor:	Attempt to Send Data to ThingSpeak
Send trigger signal to the sensor	Display Success/Failure of Data Transmission
Capture the duration of the echo pulse and calculate the distance	\
Check if the distance reading is valid:	Flood Detection:
If the distance is within the specified threshold:	Check if Distance Exceeds Flood Threshold:
Print the distance to the serial for debugging	If Exceeded:
Send the distance data to ThingSpeak using	Send WhatsApp Message:
sendDataToThingSpeak()	Set GSM Module to Text Message Mode
\	Compose and Send Message to Specified Phone Number
connectToWiFi():	Send Telegram Message:
Connect the ESP8266 to the provided WiFi network	Prepare Payload for Telegram Bot API
\	Connect and Send Message to Telegram API
sendDataToThingSpeak(long distance):	Display Success/Failure of Telegram Message
Establish a connection to the ThingSpeak server using WiFiClient	\
Prepare a POST request with distance data	Delay and Repeat:
Send the request to ThingSpeak	Pause for 5 Seconds
Close the connection and print a message indicating successful data	
transmission	
\	

Main Program Initialization: First, we import the necessary libraries, namely SoftwareSerial, TinyGsmClient, WiFiClient, and ThingSpeak. This step indicates that the program will use the functionality of these libraries for serial communication, GSM connection, Wi-Fi connection, and interaction with the ThingSpeak platform.

We define the necessary constants such as ssid (Wi-Fi SSID name), password (Wi-Fi password), apiKey (ThingSpeak API key), server (ThingSpeak server address), botToken (Telegram bot token), chatId (Telegram chat ID), phoneNumber (Phone number), trigPin (Trigger pin on ultrasonic sensor), echoPin (Echo pin on ultrasonic sensor), floodThreshold (Water height limit), rxPin (Receive pin on GSM module), and txPin (Transmit pin on GSM module).

We initialize the GSM module and serial communication. Initializing the GSM module is important to prepare the module to communicate with the cellular network while initializing the serial communication indicates the use of serial communication to interact with other devices.

We set up the Wi-Fi connection. This involves setting parameters such as the Wi-Fi SSID and password. This stage shows that the program can connect to a Wi-Fi network to send and receive data.

We connect to Wi-Fi and initiate a connection with ThingSpeak. This involves using ThingSpeak's API key to send data to the platform. By connecting to ThingSpeak, the program most likely functions as a sensor data sender that measures the water level using an ultrasonic sensor and sends the information to ThingSpeak. This information might also be shared via Telegram using a bot with a predefined token and chat ID.

These steps indicate the implementation of an internet-connected water level monitoring system, with the ability to communicate via cellular and Wi-Fi networks and integrate with ThingSpeak and Telegram platforms.

Main Program Loop: In the main program loop, a series of iterative steps are executed as follows: First, the trigger pin is set to the LOW state, followed by a 2-microsecond pause. Next, the trigger pin is set to the HIGH state, and a pause of 10 microseconds is implemented. Finally, the trigger pin returns to the LOW state. Meanwhile, the duration of the signal from the echo pin is measured, making it possible to calculate the observed distance. If the measured distance is valid (greater than 0), the distance in centimetres is printed, the initial value in ThingSpeak is updated with the measured distance, and the data is sent to ThingSpeak, with an evaluation of the HTTP response code. If the response code is equal to 200, the message "Data sent to ThingSpeak" is printed. Conversely, if the response code is different from 200, the message "Failed to send data to ThingSpeak" is displayed. If the distance exceeds the predefined flooding threshold, the sendWhatsAppMessage and sendTelegramMessage functions are called, sending the message "Water level exceeds limit!". Any invalid distance measurement results in printing "Invalid data from ultrasonic sensor". After these operations, a 5-second pause is performed.

The iterative steps in this program loop focus on measuring distance using an ultrasonic sensor. The trigger pin is used to generate an ultrasonic signal, and the echo pin is used to measure the duration of the returning signal. The measured data, such as the distance in centimeters, is updated on ThingSpeak. This is useful for monitoring and recording changes in distance over time. The measured data is sent to ThingSpeak, and the HTTP response is evaluated. The printed message provides information on whether the data transmission was successful or failed. This can help programmers to understand whether the connection to ThingSpeak is going well. If the distance exceeds the specified flooding threshold, the function to send messages via WhatsApp and Telegram is activated. This indicates a proactive response to certain conditions that the sensor can identify. If the distance measurement is invalid, the message "Invalid data from ultrasonic sensor" is printed. This shows that the program has a mechanism to detect and handle invalid data. After all operations are completed, a 5-second timeout is performed. This can be useful for controlling the program execution rate and allowing time for the system to respond or process the measurement results before the next iteration begins.

3. Results and Discussions

3.1 Results

The sensor connected to ESP8266 is designed to measure the water level at a particular location using an appropriate method based on the type of sensor used [40]. The sensor can work by emitting sound signals, measuring capacitance changes, or detecting voltage differences caused by the presence of water [41]. Once the sensor reads the water level, it sends the data to ESP8266 through a wired connection between them. Subsequently, ESP8266 processes the received data from the sensor and sends the information to a server or cloud platform via a Wi-Fi connection [42]. The data transmitted to the server or cloud platform can be analyzed to provide valuable insights, such as flood levels in a certain area, early warnings, or flood risk assessments [43]. The data received by the server or cloud platform can be stored and further analyzed to identify trends or patterns that may be related to environmental conditions [44]. The information provided by this ESP8266 flood sensor system can be used to take emergency actions, such as evacuation or rescue efforts, and assist relevant parties in preparing flood risk reduction strategies for the future.

To build a functional Telegram-based system, follow the procedural guidelines outlined on the official Telegram website for the creation and setup of

Telegram accounts and bots. Ensure careful documentation of the Telegram bot token generated during the bot creation phase, which is crucial for system integration. Next, initiate the connection by using the ESP-8266 module for Wi-Fi connection, following the module-specific instructions provided. Use the HTTPS protocol to form a secure link with the Telegram server, a fundamental aspect that ensures encrypted and secure communication.

Develop customized functions using the Telegram Bot API, using pre-recorded bot tokens to facilitate sending messages to the Telegram platform. Integrate the necessary sensors and modules according to the system specifications. Build a program designed to extract data from the integrated sensors and facilitate its transmission to the Telegram platform through pre-built functions. Calibrate the transmission interval to ensure consistent and regular data flow.

Rigorous tests are required to validate the functionality of the system, ensuring accurate and timely data transmission to the Telegram platform. This prototype, with its meticulous architectural design, presented a box-based river water level detector carefully designed to reduce flood-related risks. The prototype serves as a comprehensive solution for real-time monitoring and detection of water levels in rivers and other water bodies, thereby strengthening preventive measures against potential flood incidents.

The architectural design included careful consideration of location, size, and structural attributes, ending up with an effective flood mitigation solution. The implementation of the flood monitoring simulation on the Thingspeak server platform involved using Thingspeak, a cloud-based Internet of Things (IoT) platform that facilitates data collection, analysis, and visualization. The simulation process generally consists of the following steps:

Data Generation: Data simulation is a crucial aspect in the context of flood monitoring systems. In this research, it has been proven that simulation techniques, which include the virtual generation of data such as river water level, rain intensity, and temperature, are a key cornerstone in building this system. This process carried out through the application of mathematical models and random data generation, enables the creation of data that reflects the real-time conditions that should be obtained from real sensors in a flood monitoring system.

Through the use of mathematical models, with equations and parameters representing relationships between variables such as river water level, rainfall and temperature, this research successfully simulates the dynamic interactions between these variables in real-time. For example, river water level predictions can be made by considering an increase in rainfall intensity. Meanwhile, another approach through random data generation provides great flexibility, allowing the creation of data that reflects the distribution and characteristics of the actual data measured by the sensors.

The advantage of applying data simulation lies in its flexibility which allows full control over the parameters in the mathematical model or the distribution of the random data. This allows the simulation of a wide range of scenarios and conditions, including those that are difficult or dangerous to replicate physically. In addition, data simulation also allows the identification of weaknesses or deficiencies in the system without having to wait for an actual flood event to occur.

While data simulation results provide significant benefits, this research emphasizes the need to compare and validate the results with data obtained directly from real sensors. This validation is crucial to ensure that the simulation reflects real conditions with sufficient accuracy.

Data Upload: The process begins with the initiation of simulation data upload to the Thingspeak server platform, a dynamic step that utilizes the wide variety of Application Programming Interfaces (APIs) provided by Thingspeak. These APIs serve as a central point for flawless data transfer to the platform, providing a variety of protocols including HTTP, MQTT, and other compatible options. This complex network of protocols emphasizes the flexibility and diversity inherent in the Thingspeak API infrastructure.

Following the transmission of simulated data to Thingspeak, a platform designed to host a variety of protocols through its API infrastructure, the next phase unfolded with the orchestration of MQTT as the instrumental link connecting Thingspeak and Telegram. This connection is established through the intermediary of an MQTT broker, a critical component in the synchronization process. MQTT, known for its efficiency in lightweight messaging, facilitates the smooth flow of information between Thingspeak and Telegram.

At the core of this complex system is the configuration of the MQTT protocol within the Thingspeak environment. This involves careful settings to ensure a smooth data transfer channel between Thingspeak and the MQTT broker intermediary. This configuration became the basis for the subsequent development of the script, a powerful tool designed to seamlessly interact with Thingspeak's MQTT broker and Telegram's API.

The script, carefully designed for optimal performance, takes on a dual role in this ecosystem. Its primary function is to retrieve relevant data from Thingspeak via the MQTT protocol, utilizing the efficient communication infrastructure provided by Thingspeak. Simultaneously, the script is skilled in transmitting the obtained information to Telegram, navigating the complicated communication path built through MQTT.

Users, who are at the receiving end of this complex data journey, can benefit significantly from this well-organized system. Timely updates or notifications delivered via Telegram are a direct result of the initial data upload to Thingspeak. The seamless integration of these platforms, facilitated by the intelligent use of APIs, the MQTT protocol, and carefully designed scripts, achieves a user-centric experience characterized by the dissemination of real-time and relevant information. Through this connected network, users gain a dynamic and responsive path to stay informed, underscoring the symbiotic relationship between simulated data, Thingspeak, MQTT, and the Telegram platform.

Data Processing: The seamless integration of ThingSpeak in data processing marks a remarkable advancement in this field, offering an efficient and unified approach to capturing real-time data from a variety of diverse sources. This integration not only simplifies the complex process of storing, managing, and analyzing data but also forms an effective pathway to upload data to the ThingSpeak platform and utilize its built-in functionality or use specific MATLAB coding for in-depth analysis.

By utilizing simulated data in ThingSpeak, a variety of data processing techniques become available, ranging from basic operations such as filtering to more complex procedures such as aggregation and transformation through MATLAB codes. These techniques play a key role in refining the raw data, and translating it into more meaningful, implementable insights, thus improving the overall quality of the information obtained.

Although ThingSpeak offers real-time visualizations, it is important to recognize that these representations depict raw data and do not include in-depth analysis performed using MATLAB. This research is focused on two main aspects. First, it dives into the effectiveness of real-time data storage mechanisms in ThingSpeak, highlighting their significant role in accelerating data aggregation. Secondly, it explored the use of a robust data processing methodology using MATLAB. This methodology includes complex procedures such as filtering, aggregation, and transformation of data uploaded to ThingSpeak.

The integration of these stages provides a comprehensive understanding of the entire data lifecycle, involving careful collection, storage, and analysis. This holistic approach facilitates a more in-depth and nuanced understanding of the dataset being analyzed. By addressing real-time data storage and sophisticated data processing, this research ensures a thorough exploration of the capabilities offered by the ThingSpeak platform along with MATLAB, ultimately contributing to the advancement of data science methodologies and analytics.

Visualization: In this research, it was found that at the data processing stage, the information that has been processed can be visualized using various built-in visualization tools provided by Thingspeak, such as graphs, maps, and gauges. These visualization tools serve as a means to analyze the simulated flood data. The utilization of these visualization tools has proven to be very important in understanding patterns or trends that emerge from the simulated flood data.
Graphs, as one of the visualization tools, provide a visual depiction of changes in values within a certain period, allowing users to see long-term fluctuations and trends more easily. Meanwhile, maps allow users to see the spatial distribution of flood data, facilitating the identification of areas with high risk or potentially significant impacts. This spatial information is invaluable in disaster mitigation and response planning, helping decision-makers to allocate resources more effectively.

The use of gauges also provides immediate insight into specific conditions, such as water levels or flood intensity at specific points in real-time. This allows users to identify sudden changes or anomalies that may require immediate action. By utilizing these visualizations, users can combine quantitative analysis and visual understanding to make more informed decisions.

This research shows that the use of visualization tools on Thingspeak not only improves data accessibility but also increases users' capacity to understand, analyze and respond effectively to flood situations. As such, the overall findings of this research support the idea that data visualization can assist in the continuous monitoring and modelling of flood situations, as well as guide more effective mitigation measures and rapid response in the face of emergency conditions.

Alerts and Notifications: Thingspeak not only serves as a platform to store and visualize data but also allows users to set up alerts and notifications based on pre-defined conditions or thresholds. For example, if simulated water level data crosses a certain threshold, the system can trigger an alert to notify relevant parties via email, SMS, or other media.

This capability provides an advantage in managing emergencies or proactive monitoring. By setting relevant thresholds, users can identify potential problems or anomalies in water levels and preventive or response actions can be taken as soon as an alert is triggered. This increases efficiency and responsiveness in managing environmental data, especially in water level simulations.

Integration with Other Services: Thingspeak's integration capabilities involve various services and platforms, including social media, web services and databases, significantly enhancing the functionality of the flood monitoring system. For example, utilization of Thingspeak data can trigger actions across various Internet of Things (IoT) devices or systems, enabling the implementation of flood mitigation measures. The use of the Thingspeak server platform for simulated flood monitoring facilitates the development and testing of a robust and scalable system. The system deftly collects, processes, visualizes and analyzes data, providing substantial assistance in reducing flood risks and improving decision-making intelligence.

Figure 3 illustrates a prototype of an IoT-based flood monitoring system that uses an ESP-8266, connects to a ThingSpeak server, and runs simulated flood conditions. From this illustration, it can be seen that the integration between the hardware (ESP-8266) and the ThingSpeak platform is effective, enabling flood-related monitoring and data collection. The ESP-8266 acts as the main component on the hardware side, managing data collection from the sensors and forwarding it to the ThingSpeak server. This demonstrates the reliability of the hardware in generating accurate data and connecting it to the online platform.

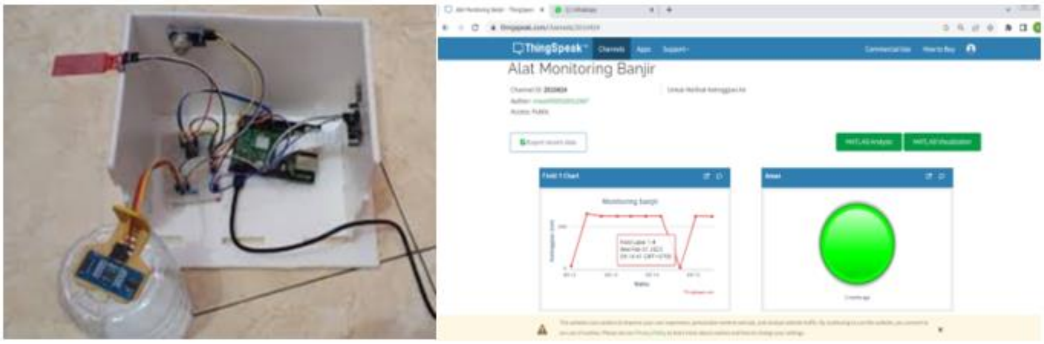

Figure 3. IoT-Enabled Flood Monitoring Prototype Utilizing ESP-8266, Connected to ThingSpeak Server, and Simulating Flood Conditions

The simulation of flood conditions on the ThingSpeak server highlighted the system's ability to simulate and process data related to water levels, rain intensity, and other relevant factors in a flood scenario. This simulation approach is an important contributor to the reliability of the system, enabling in-depth testing independent of unpredictable natural conditions.

Figure 4. Feedback telegram for water level

Figure 4 reflects the prototype's ability to provide reliable and efficient IoT-based flood monitoring. The good integration between the hardware, communication platform, and simulated flood conditions gives an idea of the possible use of this system in wider testing and research. The system design, shown in boxes, demonstrates a structured approach to installation and strategic placement along riverbanks or key locations. The precision-mounted sensors within these boxes are designed to accurately measure water levels, transmitting critical data to a centralized system or a designated receiver such as a monitoring station or mobile device. The primary function of these water level sensors is to mitigate flood risk by providing real-time and accurate information, enabling immediate action to be taken to prevent or mitigate potential flood events.

Figure 5 displays the results of ultrasonic sensor testing for measuring river water levels and the system status involved in such measurements. An in-depth analysis of the figure can unveil several crucial aspects related to the accuracy of ultrasonic sensors, integration with other components, and the system's capability to provide early warnings.

The results of river water level measurements using ultrasonic sensors were compared with measurements using a ruler for 10 different samples. The variation between the ultrasonic sensor and ruler results was generally less than 5 centimetres. The ultrasonic sensor provided sufficiently accurate measurement results, demonstrating its reliability in determining river water levels.

The figure illustrates the integration of ultrasonic sensors with other components, including the ESP32-Cam module, 4G WiFi modem, LCD, and buzzer. Early warning statuses are also displayed on the website, indicating the system's connectivity with the online platform. This system has successfully integrated various components, creating a comprehensive solution for river water level monitoring.

The system shows a "Safe" status for samples 1 to 5 as the river water level is within safe limits. For samples 6 to 8, although the ultrasonic sensor remains active, the system transitions to "Standby" status due to an increase in river water level, indicating the system's readiness to provide early warnings. In samples 9 and 10, the system detects a river water level reaching a hazardous level, and the early warning status is activated. The system accurately provides early warnings when the water level reaches a dangerous level, demonstrating its effectiveness in engaging users for potential hazards.

The system has successfully integrated ultrasonic sensors with other components, providing accurate real-time monitoring. The system's ability to issue early warnings when the water level reaches a dangerous level demonstrates the reliability and usefulness of the system in critical situations. This system is an effective solution for flood monitoring and risk mitigation, utilizing ultrasonic sensors and seamless integration with other components.

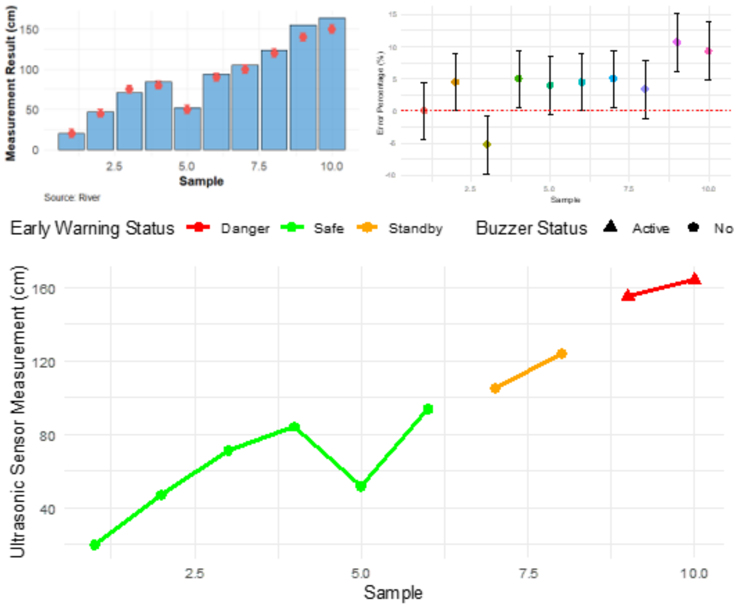

Figure 5. Ultrasonic Sensor Measurements and System Status for River Water Levels

This research employs sensor technology connected to ESP8266 for accurate water level measurements. The sensor utilizes sound signals, capacitance changes, or voltage disparity detection in the presence of water. ESP8266 then processes this data and transmits it via Wi-Fi to a server or cloud platform. The accumulated data enables comprehensive analysis, facilitating wide-scale flood monitoring, early warning systems, and risk assessment.

This study pioneers an innovative flood detection system that integrates sensors, ESP8266, and Telegram integration for enhanced monitoring and notifications. It advances by establishing a Telegram account, configuring a Telegram bot, and building a Wi-Fi connection with the ESP8266 module. Rigorous testing is conducted on data transmission to Telegram to ensure accuracy.

The architectural design of this prototype reflects innovation, emphasizing aspects such as location, size, and structure for easy installation along riverbanks or other strategic areas. Additionally, the research utilizes the Thingspeak platform for simulation, processing, visualization, and automatic response based on predefined flood conditions. This combination significantly contributes to strengthening robust and scalable flood monitoring and mitigation infrastructure. The system's ability to collect, process, visualize, and analyze data enhances flood risk mitigation strategies and decision-making capabilities, reinforcing its role in reducing flood risks and promoting resilient communities.

3.2 Discussion

This research focuses on creating a robust Flood Early Warning System by integrating social media and using various sensor methodologies to accurately measure water levels. The study evaluates various sensor approaches: sound signalling (precise but sensitive to external interference), capacitance change (sensitive to water changes but vulnerable to moisture and insulating materials), and voltage detection (simple but lacks consistent accuracy and is sensitive to resistance changes).

Connectivity aspects are explored, highlighting the compromise between stability and vulnerability in wired versus Wi-Fi connections. Secure data transfer protocols such as HTTPS and data processing within the ESP8266 ensure data integrity and efficiency. Data analysis involves identifying trends, patterns, and anomalies by using predictive models to forecast environmental conditions. Emphasis on data security during transmission and in the cloud repository is achieved through encryption and additional security measures such as two-factor authentication and suspicious activity monitoring.

Field implementation involved rigorous testing to guarantee reliability, practicability, and resistance to weather conditions. The prototype demonstrated resilience through the use of weatherproof materials and sensor protection, as well as extensive testing protocols. System scalability was addressed by suggesting the integration of additional sensors and net-network technology for network expansion, further improving accuracy and utility.

This comprehensive system makes a significant contribution to flood risk reduction, supports emergency decision-making, aids post-disaster recovery efforts, and facilitates sustainable urban planning by providing information for future infrastructure planning and risk mitigation strategies.

4. Conclusions

This research employs a technological research approach, utilizing ESP8266 programming on Arduino IDE to collect data from ultrasonic sensors measuring water levels. The data transmission process to ThingSpeak is utilised for recording and analysis. The core system involves ultrasonic sensors, ESP8266 microcontrollers, and the ThingSpeak platform as the foundation for data storage and analysis.

To enhance the comprehensiveness of flood monitoring, future research might consider integrating additional sensors such as temperature, humidity, or atmospheric pressure. Incorporating communication modules like GSM or LoRa enables real-time data transmission in remote areas without Wi-Fi, extending the monitoring range.

Rigorous validation of this system is conducted through field tests across diverse real-world scenarios. Comparisons with existing methods showcase the system's advantages and potential for enhancements.

Leveraging the ThingSpeak React app to dispatch notifications via Telegram or other platforms empowers the system to issue alerts when water levels surpass specific thresholds. This allows users to promptly respond, facilitating timely preventive measures or evacuations.

Acknowledgements

We express our profound gratitude to the authors whose generous support made this publication possible. Our heartfelt thanks also go to the BPBD (City Disaster Management Agency) of Medan for their invaluable partnership and collaboration.

References

[1] R. I. Hapsari and M. Zenurianto, "View of flood disaster management in Indonesia and the key solutions," *American Journal of Engineering Research*, vol. 5, no. 3, pp. 140–151, 2016.

[2] S. Syafrizaldi, S. Kholil, H. E. Samosir, and A. T. Sikumbang, "The disaster communication system in the city of Medan," *Jurnal Studi Komunikasi*, vol. 6, no. 2, pp. 655–678, 2022.

[3] J. Sutton and K. Tierney, "Disaster preparedness: Concepts, guidance, and research," *Colorado: University of Colorado*, vol. 3, no. 1, 2006.

[4] K. M. Carley, M. M. Malik, M. Kowalchuck, J. Pfeffer, and P. Landwehr, "Twitter usage in Indonesia," *Available at SSRN 2720332*, 2015.

[5] J. Abrahams, "Disaster management in Australia: The national emergency management system," *Emergency Medicine*, vol. 13, no. 2, pp. 165–173, 2001.

[6] S. K. Kafle, "Disaster early warning systems in Nepal: Institutional and operational frameworks," *Journal of Geography & Natural Disasters*, vol. 7, no. 2, pp. 2167–0587, 2017.

[7] V. Clerveaux, T. Katada, and K. Hosoi, "Information simulation model: Effective risk communication and disaster management in a mixed cultural society," *Journal of Natural Disaster Science*, vol. 30, no. 1, pp. 1–11, 2008.

[8] T. Pujadi, "Early warning systems using dynamics system for social empowerment society environment," presented at the 2017 International Conference on Information Management and Technology (ICIMTech), IEEE, 2017, pp. 304–309.

[9] E. J. Raker, M. C. Arcaya, S. R. Lowe, M. Zacher, J. Rhodes, and M. C. Waters, "Mitigating Health Disparities After Natural Disasters: Lessons From The RISK Project: Study examines mitigating health disparities after natural disasters.," *Health Affairs*, vol. 39, no. 12, pp. 2128–2135, 2020.

[10] M. Esposito, L. Palma, A. Belli, L. Sabbatini, and P. Pierleoni, "Recent advances in internet of things solutions for early warning systems: A review," *Sensors*, vol. 22, no. 6, p. 2124, 2022.

[11] C. Moreno *et al.*, "RiverCore: IoT device for river water level monitoring over cellular communications," *Sensors*, vol. 19, no. 1, p. 127, 2019.

[12] E. Sitinjak, B. Meidityawati, R. Ichwan, N. Onggosandojo, and P. Aryani, "Enhancing urban resilience through technology and social media: case study of urban Jakarta," *Procedia engineering*, vol. 212, pp. 222–229, 2018.

[13] D. K. Fisher and R. Sui, "An inexpensive open-source ultrasonic sensing system for monitoring liquid levels," *Agricultural Engineering International: CIGR Journal*, vol. 15, no. 4, pp. 328–334, 2013.

[14] K. Chetpattananondh, T. Tapoanoi, P. Phukpattaranont, and N. Jindapetch, "A self-calibration water level measurement using an interdigital capacitive sensor," *Sensors and Actuators A: Physical*, vol. 209, pp. 175–182, 2014.

[15] X. Zhang *et al.*, "Self-powered distributed water level sensors based on liquid–solid triboelectric nanogenerators for ship draft detecting," *Advanced Functional Materials*, vol. 29, no. 41, p. 1900327, 2019.

[16] G. Arun Francis, M. Arulselvan, P. Elangkumaran, S. Keerthivarman, and J. Vijaya Kumar, "Object detection using ultrasonic sensor," *Int. J. Innov. Technol. Explor. Eng*, vol. 8, pp. 207–209, 2020.

[17] A. T. Chatfield and U. Brajawidagda, "Twitter early tsunami warning system: A case study in Indonesia's natural disaster management," presented at the 2013 46th Hawaii international conference on system sciences, IEEE, 2013, pp. 2050–2060.

[18] P. T. Thuy, S. Van Geluwe, V.-A. Nguyen, and B. Van der Bruggen, "Current pesticide practices and environmental issues in Vietnam: management challenges for sustainable use of pesticides for tropical crops in (South-East) Asia to avoid environmental pollution," *Journal of Material Cycles and Waste Management*, vol. 14, pp. 379–387, 2012.

[19] J. Chen, T. H. Y. Chen, I. Vertinsky, L. Yumagulova, and C. Park, "Public–private partnerships for the development of

disaster resilient communities," *Journal of contingencies and crisis management*, vol. 21, no. 3, pp. 130–143, 2013.

[20] E. Yulianto, P. Utari, and I. A. Satyawan, "Communication technology support in disaster-prone areas: Case study of earthquake, tsunami and liquefaction in Palu, Indonesia," *International journal of disaster risk reduction*, vol. 45, p. 101457, 2020.

[21] G. A. K. Surtiari *et al.*, "An augmented building urban community resilience to floods through social learning: Case studies of Surabaya and Medan Cities," *PROCEEDING BOOK VOL.*, p. 206, 2019.

[22] I. T. Harnanda, "Peran Badan Penanggulangan Bencana Daerah Dalam Pengendalian Kebakaran Hutan Di Provinsi Riau (Studi Kabupaten Rokan Hilir)," 2022.

[23] D. I. Putra and M. Matsuyuki, "Disaster management following decentralisation in Indonesia: Regulation, institutional establishment, planning, and budgeting," *Journal of Disaster Research*, vol. 14, no. 1, pp. 173–187, 2019.

[24] A. Wania, I. Joubert-Boitat, F. Dottori, M. Kalas, and P. Salamon, "Increasing timeliness of satellite-based flood mapping using early warning systems in the Copernicus Emergency Management Service," *Remote Sensing*, vol. 13, no. 11, p. 2114, 2021.

[25] F. Ai, X. Xerandy, T. Znati, L. K. Comfort, and F. A. Ismail, "A Reliable, Timely Communication Application to Enhance Tsunami Preparedness," *Hazardous Seas: A Sociotechnical Framework for Early Tsunami Detection and Warning*, p. 61, 2023.

[26] A. Khan, S. Gupta, and S. K. Gupta, "Multi-hazard disaster studies: Monitoring, detection, recovery, and management, based on emerging technologies and optimal techniques," *International journal of disaster risk reduction*, vol. 47, p. 101642, 2020.

[27] O. Mendoza-Cano *et al.*, "Experiments of an IoT-based wireless sensor network for flood monitoring in Colima, Mexico," *Journal of Hydroinformatics*, vol. 23, no. 3, pp. 385–401, 2021.

[28] R. Tawalbeh, F. Alasali, Z. Ghanem, M. Alghazzawi, A. Abu-Raideh, and W. Holderbaum, "Innovative characterisation and comparative analysis of water level sensors for enhanced early detection and warning of floods," *Journal of Low Power Electronics and Applications*, vol. 13, no. 2, p. 26, 2023.

[29] P. SURYANARAYANA, M. D. S. KUMAR, A. V. KRISHNA, and B. K. VAMSI, "ULTRASONIC BASED MEASUREMENT SYSTEM FOR LIQUID LEVEL AND TYPE MONITORING".

[30] L. Osterweil, "Software processes are software too," *Engineering of Software: The Continuing Contributions of Leon J. Osterweil*, pp. 323–344, 2011.

[31] C. D. Okwudibe and B. Akinloye, "Design and simulation of temperature data logger," *American Journal of Engineering Research (AJER)*, vol. 6, no. 12, pp. 14–19, 2017.

[32] S. Mahajan, "Design and development of an open-source framework for citizen-centric environmental monitoring and data analysis," *Scientific Reports*, vol. 12, no. 1, p. 14416, 2022.

[33] T. W. Lai, Z. L. Oo, and A. Moe, "Real Time Water Level Monitoring for Early Warning System of Flash Floods Using Internet of Things (IoT)".

[34] S. Lumbangaol, R. Siagian, F. Siahaan, and E. Sitinjak, "IOT-Based Pollution and Air Quality Monitoring Equipment with ESP-8266," presented at the Proceedings of the 6th Batusangkar International Conference, BIC 2021, 11-12 October, 2021, Batusangkar-West Sumatra, Indonesia, 2022.

[35] Z. M. Hussin, S. S. Saaddin, S. Mohammad, N. A. M. Azmi, and S. Salim, "Development of Automated Drainage System," presented at the 2022 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), IEEE, 2022, pp. 72–77.

[36] N. A. Latha, B. R. Murthy, and K. B. Kumar, "Distance sensing with ultrasonic sensor and Arduino," *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 2, no. 5, pp. 1–5, 2016.

[37] M. Al-Kuwari, A. Ramadan, Y. Ismael, L. Al-Sughair, A. Gastli, and M. Benammar, "Smart-home automation using IoT-based sensing and monitoring platform," presented at the 2018 IEEE 12th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG 2018), IEEE, 2018, pp. 1–6.

[38] K. Peffers, M. Rothenberger, T. Tuunanen, and R. Vaezi, "Design science research evaluation," presented at the Design Science Research in Information Systems. Advances in Theory and Practice: 7th International Conference, DESRIST 2012, Las Vegas, NV, USA, May 14-15, 2012. Proceedings 7, Springer, 2012, pp. 398–410.

[39] Đ. Mijailović, A. Đorđević, M. Stefanovic, D. Vidojević, A. Gazizulina, and D. Projović, "A Cloud-Based with Microcontroller Platforms System Designed to Educate Students within Digitalization and the Industry 4.0 Paradigm," *Sustainability*, vol. 13, no. 22, p. 12396, 2021.

[40] R. K. Kodali and B. S. Sarjerao, "A low cost smart irrigation system using MQTT protocol," presented at the 2017 IEEE Region 10 Symposium (TENSYMP), IEEE, 2017, pp. 1–5.

[41] N. Barsan and U. Weimar, "Understanding the fundamental principles of metal oxide based gas sensors; the example of CO sensing with SnO2 sensors in the presence of humidity," *Journal of Physics: Condensed Matter*, vol. 15, no. 20, p. R813, 2003.

[42] A. I. Siam *et al.*, "Secure health monitoring communication systems based on IoT and cloud computing for medical emergency applications," *Computational Intelligence and Neuroscience*, vol. 2021, 2021.

[43] S. Fang *et al.*, "An integrated information system for snowmelt flood early-warning based on internet of things," *Information Systems Frontiers*, vol. 17, pp. 321–335, 2015.

[44] S. Tang, D. R. Shelden, C. M. Eastman, P. Pishdad-Bozorgi, and X. Gao, "A review of building information modeling (BIM) and the internet of things (IoT) devices integration: Present status and future trends," *Automation in Construction*, vol. 101, pp. 127–139, 2019.