Journal of Computation Physics and Earth Science
ISSN: 2776-2521 (online)
Volume 4, Number 1, April 2024, Page 6-14
https://journal.physan.org/index.php/jocpes/index

Integrated of a Real-Time Flood Monitoring System with AI-Based Sensors in North Pantura Java

Ahmad Dinan Irsyadi[1]
[1]Undergraduate Program in Applied of Instrumentation Meteorology, Climatology Geophysics (STMKG)

Article Info

Article history:

Received March 5, 2024
Revised March 15, 2024
Accepted March 23, 2024

Keywords:

Tidal flooding,
IoT,
AI,
Real-time monitoring,
Prediction Tidal Flooding.

A B S T R A C T

This research focuses on the development and implementation of an Internet of Things (IoT)-based system for predicting tidal flood (banjir rob) using sensor data and machine learning techniques. The system utilizes sensors such as ultrasonic sensors (HC-SR04), DHT11 (for temperature and humidity), connected to an ESP32 module for real-time data collection. The collected data is sent to the ThingSpeak platform for storage and analysis. A machine learning model, specifically a Random Forest Regressor, is trained on historical data from ThingSpeak to predict the flood height based on environmental factors such as temperature and humidity. To enhance the practicality of the system, a Telegram bot is integrated to provide real-time flood predictions directly to users. The system fetches the latest sensor data, predicts the flood height, and sends this information via the Telegram bot. The machine learning model is evaluated using metrics such as R2 score and Mean Squared Error (MSE), ensuring accurate and reliable predictions for flood monitoring. This approach presents a low-cost, real-time, and scalable solution to predict tidal floods in coastal regions. The system's integration of IoT, cloud computing, and machine learning offers a powerful tool for local authorities, disaster management teams, and residents to monitor and prepare for potential flood events. The research highlights the potential of combining IoT technology with AI to enhance environmental monitoring and early warning systems in flood-prone areas.

Corresponden Author:

Ahmad Dinan Irsyadi,
Undergraduate Program in Applied of Instrumentation Meteorology, Climatology Geophysics (STMKG)
Tangerang City, Banten, Indonesia
Email: ahmaddinan35@gmail.com

1. INTRODUCTION

In North Java, there are have a highway that makes the route of Java easier and faster than the road usually. It names is Pantura, Pantura have a fast highway in Java, but in North Java there are many tidal floods by BNPB. There are many risks and any disaster with floods.

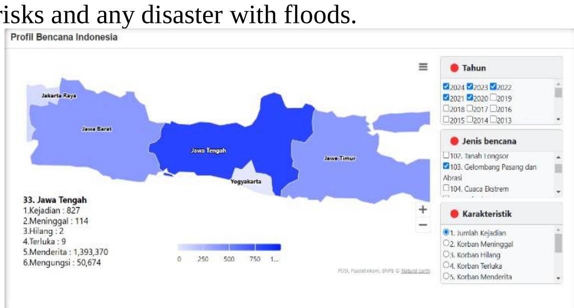


Fig 1. Floods Disaster Marks map in 5 years [1]

Tidal floods are floods that occur on the coast due to rising sea levels. Tidal floods are characterized by high sea waves that overflow onto land. Damage to coastal ecosystems is one of the causes of tidal floods. Many factors can influence the occurrence of tidal floods, including global warming, land subsidence, regional topography, abrasion and sea tides. The system built in this journal can monitor tidal floods/sea water by utilizing the concepts of Internet of Things (IoT) and Artificial Intelligence (AI). The system is built with an Arduino 8266 microcontroller which will be connected to ultrasonic and humidity sensors as well as radar, the system will be connected to a Wifi modem which will act as a link between the system and Realtime web monitoring. In this final assignment, the author only focuses on creating, designing and testing websites and the speed of information used to monitor tidal floods/sea water.

The Internet of Things (IoT) is a network of objects the physical or “thing” that is embedded with electronics, software, sensors and connectivity to enable it to achieve value and service larger ones by exchanging data with manufacturer, carrier, and/or other connected devices. AI is programming that learns through events and reprogram the device accordingly with the last visible pattern and not with pre-programmed instructions [2]. The combination of the two Artificial Intellegence (AI) has a significant impact on society by quickly making predictions and making it easier to do work. The aim of this research is to develop an AI-based tidal flood monitoring system that is integrated with sensors to provide real-time data and increase the speed and accuracy of early warning. And the expected results of this research are a prototype tidal flood monitoring system that can provide real-time data, increase the accuracy of flood predictions by up to 85-95%, a faster early warning system, allowing communities to prepare earlier.

2. RESEARCH THEORY

2.1 Monitoring System

Monitoring systems represent one of the most important tools in certain areas, allowing for continuous observation and analysis of specific parameters. Most modern monitoring systems include IoT technology, which makes it possible to acquire and process data in real time. IoT-enabled monitoring systems are particularly effective, as they can gather large volumes of data from multiple sources at the same time. These systems provide accurate and timely information, which is essential for mitigating environmental hazards and supporting sustainable practices. The Monitoring System is a system that is used continuously to monitor and collect data of a particular object or process. Data collected from This system is then analyzed with the aim of produce crucial information in support decision making or necessary action. This system not only provides real-time images about the condition of the object or process being monitored, but It also allows for in-depth analysis to the data collected [3].

2.2 Internet of Things

The IoT has revolutionized monitoring systems by allowing smooth connectivity and the acquisition of data in real time. IoT-based monitoring systems are finding more applications in a wide array of fields, including environmental management, healthcare, agriculture, and industrial processes. This section presents a review of the recent literature concerning IoT-based monitoring systems, describing their architecture, applications, and challenges. These sensors collect data from the surroundings and send it wirelessly using Wi-Fi, ZigBee, or LoRa protocols. Edge computing or cloud computing platforms are employed for the processing of the collected data for real-time analytics and decision-making. IoT- based monitoring systems ensure timely, precise, and actionable insights across industries in real time. More improvement in IoT technologies leads to increases their domain of application and thus challenges, toward a world full of connected and intelligible devices.

Internet of Things (IoT) devices include: four interdependent and interdependent core layers support. These four layers include sensors and monitoring devices, communication networks for transferring data, platforms that manage and analyze the collected data, as well as the interface users that facilitate human interaction with IoT system. Taken together, these layers working to optimize IoT performance inside various application contexts, from smart homes to industrial automation, creating a connected environment and smart. Reference to "figure 2.1" indicates that this illustration can provide a more in-depth visual view of the structure IoT complex [3].

Fig 2. IOT Architecture Layer [4]

2.3 ESP 32

ESP32 is a powerful and economical platform for IoT applications, particularly in monitoring systems. Equipped with integrated Wi-Fi and Bluetooth capabilities, together with its low power consumption and high processing performance, the ESP32 has been gaining rapid attention among researchers and developers. The section will discuss the use of ESP32 in IoT-based monitoring systems, its architecture, applications, and advantages.

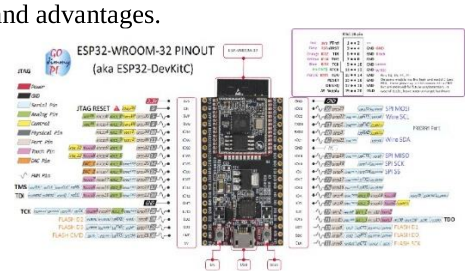


Fig 3. ESP Module Kit [5]

ESP32 is a module that can used in many projects and also is a complete module with integrity microcontroller that can work simultaneously independent. ESP32 provides WiFi networking independently as a bridge from the microcontroller existing to your WiFi network. ESP32 using a dual-core processor running on Xtensa LX16 instructions. ESESP32 has 34 general purpose input/output pins (GPIO) whose function can be configured by configuring the appropriate registers. The ESP32 GPIOs are divided into four categories: Digital Only, Analog Enabled, Capacitive Touch Enabled, and Other [6].

2.4 Ultrasonic Sensor HC-SR04

The HC-SR04 ultrasonic sensor is a widely used and cost-effective device for measuring distance. Its applications span robotics, automation, and embedded systems due to its simplicity, reliability, and affordability. This review aims to summarize key studies and developments related to the HC-SR04 and its application in a wide range of fields. The HC-SR04 ultrasonic sensor works on the principal of sending ultrasonic waves at a frequency of 40 kHz and measures the time required for the echo to be received after it bounces off an object [7]. The module calculates the distance using the formula (1):

$$Distance = \frac{speed\ of\ sound\ X\ Time}{2} \tag{1}$$

Fig 4. Ultasonic Sensor HC-SR04 [8]

The sensor has two major elements: Transmitter: This sends out the waves and Receiver: This detects the reflected waves. Its working range lies between approximately 2 to 400 cm, and with a ±3mm accuracy, the module fits all sorts of uses for short and middle-range distances.

2.5 Sensor DHT11

The DHT11 is a low-cost digital sensor for temperature and humidity measurement. The low cost, ease of use, and reasonable accuracy have made it a widely used sensor in various applications, such as environmental monitoring, smart agriculture, and IoT systems. The DHT11 is a capacitive digital sensor capable of measuring:

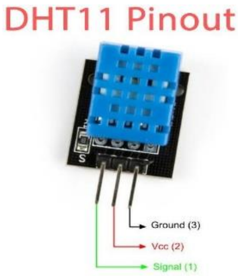


Fig 5. Sensor DHT11 [9]

Temperature: The range is from -40°C to 80°C, with an accuracy of ±0.5°C. Humidity: Ranges from 0% to 100% relative humidity (RH), with an accuracy of ±2% RH [10]. This sensor works by sending on a single wire and enables easy interfacing with Arduino, or other microcontrollers such as ESP32 and Raspberry Pi. It outputs periodic data temperature and humidity, usually every 2 seconds, which is faster for practical applications in almost all applications of real-time monitoring.

2.6 AI predictions System

This tutorial describes how to implement an AI-based prediction system using data from ThingSpeak, which is a cloud system for IoT analytics. The system uses the data gathered in ThingSpeak to train an ML model for making predictions. ThingSpeak A platform for real-time data collection and analysis, ideal for IoT devices. AI prediction system It works on ML algorithms to predict future data trends on the basis of historical data retrieved from ThingSpeak.

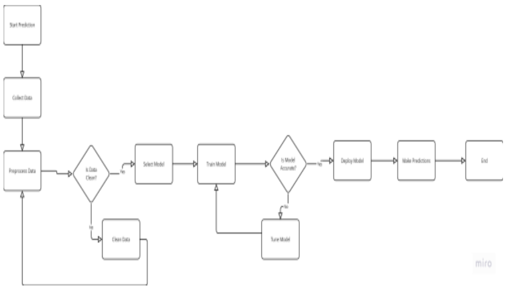


Fig 6. Diagrams Flow AI [11]

3. RESEARCH METHOD

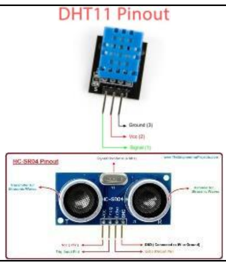
3.1 Data Collection

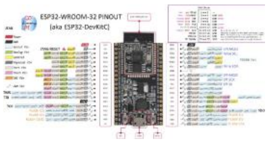
1) *Direct measurement*
Direct measurement in prototype sample using connected sensors with an IoT system it is possible direct real-time data collection from the coastal environment. Measurement of this data focused on the height of sea waves caused by nighttime abrasion and also sea level rise during rain and other disaster symptoms [3] [12].
2) *Obsevation*
Make direct observations in sample prototype to weather conditions to ensure accuracy of data collected from sensors and monitoring systems to compare the values of one or more variables at different samples or times [12].

3.2 System Framework

Table 1. System Framework

Input	Process	Output





The Framework depicted in the table above begins with ultrasonic sensors and humidity and temperature sensors. These sensors are tasked with recognizing conditions surrounding environment. Data collected from sensors this is then processed by the nodemcu esp32. Nodemcu esp32 acts as a transmission device and link between various components in the system. After that, the processed data will be sent to database for subsequent display on the website monitoring. If the system detects the characteristics of a tidal flood, the system will do so automatically sending warning messages via the platform WhatsApp.

3.3 AI System Workflow

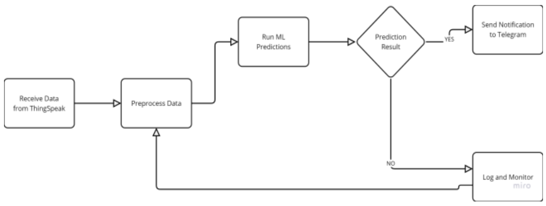


Fig 7. Workflow AI

The flowchart illustrates the process of an AI prediction system using data from ThingSpeak, with the following steps: This step involves fetching real-time data from the ThingSpeak IoT platform. ThingSpeak channels store IoT sensor data (e.g., temperature, humidity). Data is retrieved via ThingSpeak's REST API using an HTTP request. The raw data collected from ThingSpeak is cleaned and prepared for machine learning. After preprocessing, the cleaned data is fed into a machine learning model to generate predictions. The model could be trained for tasks like anomaly detection, forecasting, or classification based on the data. The predictions are evaluated for their outcome: If the prediction is significant (e.g., a temperature threshold is exceeded), it triggers a notification. Otherwise, the system moves to logging and monitoring. Alerts or notifications are sent to a Telegram channel or user if a specific condition is met. This is achieved using Telegram's Bot API, which can deliver real- time messages. All data, predictions, and actions (such as notifications) are logged for monitoring and future analysis. This ensures the system's performance can be tracked and any issues identified over time.

4. RESULTS

4.1 Monitoring system

The publicly accessible monitoring system provides real-time data on humidity, temperature, and water levels, offering a comprehensive overview of environmental conditions. This system ensures transparency, enabling the public to stay informed about factors that may impact daily activities, such as fishing or outdoor events. By making this information readily available, the system empowers users to make informed decisions, whether they are fishermen needing accurate weather data or individuals preparing for outdoor activities. The system helps users adapt to tidal and weather changes, promoting safety and reducing risks from unexpected environmental shifts. Regular updates to the system's database ensure that the information remains current, providing a reliable resource for planning and decision-making. The research results are presented through various formats, including figures, graphs, and tables, to enhance clarity and understanding.

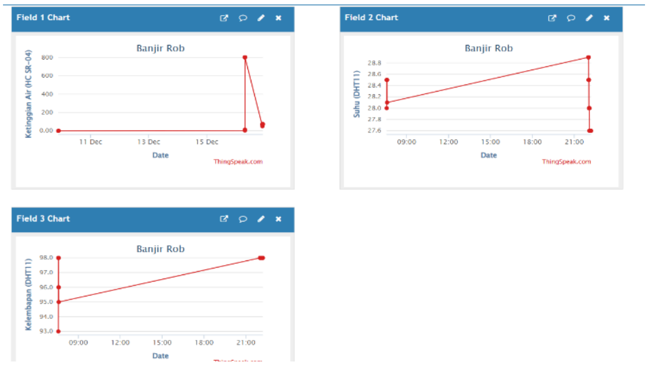


Fig 8. Dashboard Monitoring

4.2 IOT Device

IoT devices with WiFi-enabled sensors collect real-time data on environmental parameters like temperature, humidity, and water levels. This data is transmitted to a cloud platform for further analysis, enabling fast and efficient monitoring and response to changes. The use of IoT not only improves data accuracy but also speeds up reaction times to environmental events, which is crucial for various industries and public applications. The system, built with an ESP32, HC-SR04 ultrasonic sensor, and DHT11 sensor, monitors tidal flooding. The HC-SR04 measures the distance to the water surface, detecting rising water levels, while the DHT11 records temperature and humidity. The ESP32 processes the data and sends it to ThingSpeak, a cloud platform for IoT data analytics. The system updates data every 2 seconds using a URL generated by ThingSpeak's API key, ensuring accurate, near-instant updates. While the free ThingSpeak service limits updates to every 15 seconds, this system achieves fast data transmission with minimal latency. This solution is ideal for coastal areas like Semarang, which often experience storm surges.

.

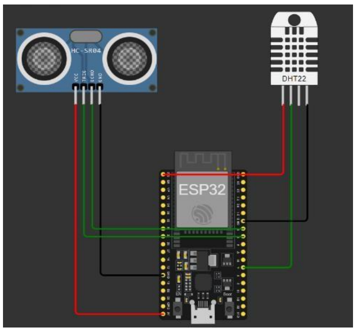


Fig 9. Block Diagram System

4.3 System and Device Testing

a. *IoT Testing*

After several days of testing the existing tool, it was found that the sensor could reach a height of 8 meters and had an accuracy of up to +- 0.01 cm and was able to calculate at a speed of 2 seconds to transmit data, however there were several corrections from the tool up to 0.032 cm. and can be calculated through coding in the Arduino IDE. Can we know in Fig 10 below.

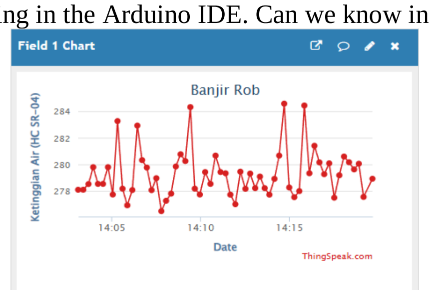


Fig 10. Water Level Testing

After several days of testing the existing equipment, it was found that the sensor could read the surrounding conditions at a certain temperature and humidity. temperature and humidity by DHT 11 is able to produce good data at certain intervals in a faster time. In fig 11 and 12 shows that there is a correlation between temperature and humidity so that when this sensor is used and is able to work better it will send valid and well-tested data.

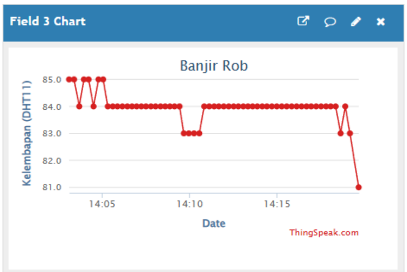


Fig 12. Humidity Testing

b. AI Predictions System Testing

After obtaining sufficient data for training and testing by the AI system, the next step is to clean the data from unnecessary teasers and also decide to use as much data as possible for prediction training. As seen in the image below, we have cleaned the data from the sensors above, which will then use a simple regression model with a data separation of 80% versus 20% data. Next, we prepared a model evaluation with MSE and R2 Score to approach the best prediction model. The following is a table of the data to be trained.

created_at	entry_	field1	field2	field3
2024-12-16 00:34:28 UTC	96	2.96	28.00	93.00
2024-12-16 00:34:44 UTC	97	801.43	28.50	98.00
2024-12-16 00:35:01 UTC	98	13.23	28.50	98.00
2024-12-16 00:35:18 UTC	99	801.43	28.50	98.00
2024-12-16 00:35:36 UTC	100	801.47	28.50	96.00
2024-12-16 00:35:53 UTC	101	801.53	28.50	96.00
2024-12-16 00:36:10 UTC	102	801.52	28.10	95.00
2024-12-16 15:01:11 UTC	103	49.50	28.90	98.00
2024-12-16 15:01:27 UTC	104	67.90	28.90	98.00
2024-12-16 15:01:42 UTC	105	67.88	28.90	98.00
2024-12-16 15:02:00 UTC	106	67.90	28.50	98.00
2024-12-16 15:02:17 UTC	107	67.90	28.50	98.00
2024-12-16 15:02:35 UTC	108	67.90	28.50	98.00
2024-12-16 15:02:53 UTC	109	68.31	28.50	98.00
2024-12-16 15:03:08 UTC	110	67.88	28.50	98.00
2024-12-16 15:03:25 UTC	111	67.90	28.00	98.00
2024-12-16 15:03:43 UTC	112	67.90	28.00	98.00
2024-12-16 15:03:59 UTC	113	67.90	28.00	98.00
2024-12-16 15:04:16 UTC	114	67.90	28.00	98.00
2024-12-16 15:04:32 UTC	115	67.90	28.00	98.00
2024-12-16 15:04:49 UTC	116	67.90	28.00	98.00
2024-12-16 15:05:05 UTC	117	68.32	28.00	98.00
2024-12-16 15:05:20 UTC	118	67.92	28.00	98.00
2024-12-16 15:05:36 UTC	119	67.90	27.60	98.00
2024-12-16 15:05:51 UTC	120	67.90	27.60	98.00
2024-12-16 15:06:06 UTC	121	75.62	27.60	98.00
2024-12-16 15:06:22 UTC	122	72.59	27.60	98.00
2024-12-16 15:06:37 UTC	123	72.57	27.60	98.00
2024-12-16 15:06:55 UTC	124	72.57	27.60	98.00
2024-12-16 15:07:12 UTC	125	72.59	27.60	98.00
2024-12-16 15:07:27 UTC	126	72.59	27.60	98.00
2024-12-16 15:07:43 UTC	127	72.59	27.60	98.00
2024-12-16 15:08:00 UTC	128	72.57	27.60	98.00
2024-12-16 15:08:16 UTC	129	73.03	27.60	98.00
2024-12-16 15:08:32 UTC	130	72.59	27.60	98.00
2024-12-16 15:08:48 UTC	131	72.59	27.60	98.00
2024-12-16 15:09:03 UTC	132	72.57	27.60	98.00
2024-12-16 15:09:18 UTC	133	72.57	27.60	98.00
2024-12-16 15:09:36 UTC	134	72.57	27.60	98.00
2024-12-16 15:09:51 UTC	135	72.57	27.60	98.00
2024-12-16 15:10:07 UTC	136	72.57	27.60	98.00
2024-12-16 15:10:24 UTC	137	72.59	27.60	98.00
2024-12-16 15:10:40 UTC	138	72.59	27.60	98.00
2024-12-16 15:10:56 UTC	139	72.59	27.60	98.00
2024-12-16 15:11:12 UTC	140	73.03	27.60	98.00
2024-12-16 15:11:28 UTC	141	72.57	27.60	98.00
2024-12-16 15:11:44 UTC	142	72.59	27.60	98.00
2024-12-16 15:12:01 UTC	143	73.03	27.60	98.00

Fig 13. Cleaned Data

Next, the data will go through a coding process to send early warning notifications according to the model that has been trained and will then be linked according to the desired prediction model. The following is the coding and results of the ai prediction system.

```
Data Head:
               created_at  entry_id  field1  field2  field3
0  2024-12-16 00:34:28 UTC        96    2.96    28.0    93.0
1  2024-12-16 00:34:44 UTC        97  801.43    28.5    98.0
2  2024-12-16 00:35:01 UTC        98   13.23    28.5    98.0
3  2024-12-16 00:35:18 UTC        99  801.43    28.5    98.0
4  2024-12-16 00:35:36 UTC       100  801.47    28.5    96.0
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48 entries, 0 to 47
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   created_at  48 non-null     object
 1   entry_id    48 non-null     int64
 2   field1      48 non-null     float64
 3   field2      48 non-null     float64
 4   field3      48 non-null     float64
dtypes: float64(3), int64(1), object(1)
memory usage: 2.0+ KB

Info Data:
 None

Evaluasi Model:
Mean Squared Error (MSE): 0.2689600000000002
R-squared (R2): 0.2528888888888883

Model berhasil disimpan sebagai model_banjir_rob.pkl
```

Fig 14. Model Prediction using MSE and R^2 Score

After that, we try to send the data to the Telegram bot and immediately provide an early warning and provide several weather indices and water levels at that time. With this design, the public can monitor more quickly and directly integrate via direct interest and if there is no internet, it can be accessed via a web IP address or sensor directly. Code for transmission in thingspeak to telegram.

```
import requests
import time

# Replace with the correct Channel ID and API Key from ThingSpeak
channel_id = "2778994"  # Replace with your Channel ID
thingSpeakAPIKey = "YJCNE1O9SWH9AJWQ"  # Replace with your API Key

# URL to fetch the latest data from ThingSpeak
thingSpeakURL = f"https://api.thingspeak.com/channels/{channel_id}/feeds.json?api_key={thingSpeakAPIKey}&results=1"

# Telegram Bot Token
telegram_token = "7607703134:AAEIJuubHsePT_uPhzN13QkGR4bE7Pye-jE"  # Replace with your Telegram bot token
chat_id = "1243984526"  # Replace with your Telegram chat ID

# Function to fetch sensor data from ThingSpeak
def get_sensor_data():
    try:
        response = requests.get(thingSpeakURL)
        response.raise_for_status()  # Ensure successful response

        # Get data in JSON format
        data = response.json()
        print("Received Data:", data)  # Debug: see the received data

        if 'feeds' in data:
            return data['feeds'][0]  # Return the latest feed data
        else:
            print("Data does not contain 'feeds' key")
            return None
    except requests.exceptions.RequestException as e:
        print(f"Error while fetching data: {e}")
            return None
    except ValueError as e:
        print(f"Error parsing JSON: {e}")
        return None

# Function to send message to Telegram
def send_telegram_message(message):
    try:
        telegram_url = f"https://api.telegram.org/bot{telegram_token}/sendMessage"
        payload = {
            'chat_id': chat_id,
            'text': message
        }
        response = requests.post(telegram_url, data=payload)
        response.raise_for_status()  # Ensure successful response
        print("Message sent successfully to Telegram!")
    except requests.exceptions.RequestException as e:
        print(f"Error while sending message to Telegram: {e}")

# Fetch data and send to Telegram periodically
while True:
    # Fetch sensor data from ThingSpeak
    sensor_data = get_sensor_data()

    if sensor_data:
        # Get the sensor data you want, e.g., field1, field2, etc.
        field1_value = sensor_data.get('field1', 'Data not available')
        field2_value = sensor_data.get('field2', 'Data not available')

        # Create the message to send to Telegram
        message = f"Sensor Data:\nField 1: {field1_value}\nField 2: {field2_value}"

        # Send message to Telegram
        send_telegram_message(message)

    # Wait a few seconds before fetching data again
    time.sleep(60)  # Wait for 60 seconds (1 minute) before fetching data again
```

Fig 15. Code Transmission to Telegram

CONCLUSION

In closing, the conclusion of this research is that the work idea is not yet perfect, therefore in the tool testing there has been no data validation or calibration. In future research, the researcher hopes that there will be additions such as remote sensing or signal transmission that can work well. However, in this research the researcher is proud of being able to produce work at the speed desired by the researcher according to the introduction.

REFERENCE

[1] BNPB, "profil bencana indonesia," BNPB. [Online]. Available: https://dibi.bnpb.go.id/home/index2

[2] S. Gitakarma, "Peranan Internet of Things Dan Kecerdasan Buatan Dalam Teknologi Saat Ini," *J. Komput. dan Teknol. Sains*, vol. 1, no. 1, pp. 1–8, 2022.

[3] Hafidin Bangun Widyanto, "Sistem Monitoring Kecepatan Dan Arah Angin Berbasis Internet of Thing (Iot) Sebagai Peringatan Dini Bencana Alam," *Univ. Brawijaya*, no. 1, pp. 58–66, 2024.\

[4] "pengertian-arsitektur-IoT-menurut-para-ahli-1024x632."

[5] E. Systems, "E-BOOK ESP32 Series Datasheet," *Espr. Syst.*, pp. 1–60, 2021, [Online]. Available: https://www.espressif.com/sites/default/files/documentation/es p32_datasheet_en.pdf

[6] S. Anwar, "Rancang Bangun Sistem Deteksi Perubahan Signifikan Pasang Surut Air Laut Berbasis Internet of Thing (Iot)," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 3, 2024, doi: 10.23960/jitet.v12i3.5109.

[7] J. Smot, "The Basics of Ultrasonic Sensors," same sky.

[8] E. J. Morgan, "HC-SR04 ETC2 | Alldatasheet," pp. 1–6, 2014, [Online]. Available: https://pdf1.alldatasheet.com/datasheet/pdf/view/1132203/ETC2/HC-SR04.html

[9] L. Guangzhou Aosong Electronics Co, "Modul Suhu dan Kelembapan DHT11", [Online]. Available: www.aosong.com

[10] T. Liu, "Aosong Electronics Co ., Ltd," *Digit. Relat. humidity Temp. sensor/module(DHT22)*, vol. 22, pp. 1–10, 2013.

[11] MIro, "Diagram Flows," Miro. [Online]. Available: https://miro.com/welcomeonboard/M0RacFowWVlUU2puc1NFQXBTSUJrbE5wWGdRSHhWR2U2MDF2cHdOb0lneUdne FBTeTI4TFU5cmdwTFZxSXpsblp0YXVYZ1FyRDVCNVhwS0pOMk5BdXRKMzlVVXdHZGw4NVc1dFhQaXY5dFFRZXFTWmRjNDZsM1NLR3ovUHpKaTchZQ==?share_link_id=346092488239

[12] M. M. Ali, T. Hariyati, M. Y. Pratiwi, and S. Afifah, "Metodologi Penelitian Kuantitatif dan Penerapannya dalam Penelitian," *Educ. Journal.2022*, vol. 2, no. 2, pp. 1–6, 2022.