Jurnal Teknik Informatika Vol. 16 No. 2, October 2023 (133-140)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v16i2.32161

JURNAL TEKNIK INFORMATIKA

Homepage : http://journal.uinjkt.ac.id/index.php/ti

IoT Based Early Flood Detection System with Arduino and Ultrasonic Sensors in Flood-Prone Areas

Muhammad Darwis[1], Hafiizh Asrofil Al Banna[2], Setiawan Restu Aji[3], Dinda Khoirunnisa[4], Nakia Natassa[5]

[1,2,3,4,5] Informatics Engineering Study Program, Faculty of Engineering Sciences, Paramadina University
[1,2,3,4,5]Jl. Raya Mabes Hankam No. Kav 9, Cipayung, East Jakarta, DKI Jakarta 16680.

Article:
Accepted: September 13, 2023
Revised: May 14, 2023
Issued: October 28, 2023

***Correspondence Address:**
muhammad.darwis@paramadina.ac.id

ABSTRACT

IoT is one of the focuses of application development carried out by various developers today. The aim is to enable various devices and work independently to meet the various needs of their users. The flood early warning system is one of the much-needed IoT-based applications, enabling users to quickly obtain water level information in an area. This application can help people to be more aware of flood disasters, especially during the rainy season. This research develops a flood early warning system application by utilizing Arduino and ultrasonic sensors installed in flood-prone areas. The sensor is used to measure the water level at a time based on the distance from the water surface to the sensor. When the distance between the water surface and the sensor is less than or equal to the set threshold, the sensor will send data and alerts to the user via email. This research applies the IoT design and development method. In addition, this research also used the C and Python programming language for application prototypes and the MySQL database to store the data. the application in this study was tested using the blackbox method and the results showed that all application functions could run properly.

Keywords: *IoT, early flood detection system, arduino, ultrasonic sensor, python*

Jurnal Teknik Informatika Vol. 16 No. 2, October 2023 (133-140)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v16i2.32161

1. INTRODUCTION

Internet of Things (IoT) allows users to connect objects to the internet with the aim of collecting data and controlling processes or machines remotely. IoT allows the physical world to be integrated into the digital world to optimize time, save costs and simplify the way humans work [1]. IoT applications have been widely applied to various areas of life to help humans complete their work, such as smart home, smart city, agriculture and other fields [2]. To maximize how IoT applications work, various studies are continuously being carried out to ensure supporting technologies such as devices, networks, data and security when implemented [3] [4] [5]. This aims to ensure that the IoT applications developed are appropriate and in accordance with human needs [6].

One IoT application that has begun to be developed in the environmental field is a flood detection or control application. This has been developed using various methods such as measuring rain intensity, water discharge height or water volume in the river. [7] [8].

By implementing IoT technology in these applications, humans will be greatly helped in dealing with flood disasters that can occur at any time, especially during the rainy season. As a tropical country with a long rainy season, Indonesia often experiences flooding and is vulnerable to it. Therefore, flood detection applications are very much needed, both for the community and the government in dealing with this annual disaster. Based on this background, this research focuses on developing a flood early warning system application by utilizing ultrasonic sensors installed in flood-prone areas, this aims to make users or the surrounding community can easily get information related to flooding so that they can increase awareness.

In this study, the authors developed a flood early warning system application by utilizing ultrasonic sensors installed in flood-prone areas. The ultrasonic sensor was chosen because of its ease of use and accuracy in measuring water level distances [9]. The sensor is used to measure the water level at a time based on the distance from the water surface to the sensor. When the distance between the water surface and the sensor is less than or equal to the set threshold, the distance between the water level and the sensor is 10 cm in this study, the sensor will send data to database and alerts to the user via email.

During the implementation of this research, the authors thought that the application of the flood early warning system would be very appropriate and help users obtain information on water levels in their area, especially at night. As a prototype and experiment, the authors use email to send data captured by sensors. However, of course, the method of sending data can use other communication methods such as SMS or WhatsApp chat applications.

To facilitate the application development process in this study, the authors use the IoT design and development method proposed by Arshdeep Bahga and Vijay Madisetti in their book [10]. With this method, the stages of developing IoT applications in this study are more structured starting from the collection and analysis of needs to the implementation process. This method has also become a reference for many researchers in the development of IoT applications.

With this method, the author can easily define the entire application development framework, starting from objectives to equipment requirements and configuration. Thus, the scheme including the costs required for completing the application can be clearly known. This aims to enable future users and researchers to know their needs, including initial application development costs.

In addition, in this study, the authors also use the Python and C programming languages to develop application prototypes. Python and C are used to control the Arduino and ultrasonic sensors used in this research. The prototype is sufficient to provide an overview of how the IoT devices implemented in the research work. Also in this study, the MongoDB database is used to store data recorded by IoT devices.

Based on the background and problems above, here are some technical questions that will be a reference for the purpose of this research is how to create a flood early warning application that can be easily applied in flood-prone areas and What is the scheme for a flood early warning application that can be developed and used easily by users.

To test the application prototype produced in this study, the authors use the blackbox testing method to determine its functionality. As a result, it is known that all functions that have been designed and defined

Jurnal Teknik Informatika Vol. 16 No. 2, October 2023 (133-140)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v16i2.32161

can run well. The author hopes that this application can be developed further so that it can provide access to information quickly and easily for its users regarding the condition of the water level in their area. Thus, they can be more alert and prepared for flood disasters that can occur at any time.

Several studies with the same object of study have previously been carried out by various researchers, including [11] who developed an Android-based application for flood warnings, [12] and [13] who made a flood warning application using ATmega and buzzer and SMS, [14] who also developed an IoT-based flood warning application in Manado City and [15] who utilized raspberry pi to create a flood early warning application.

In this research, an early flood warning system was developed specifically to be installed in flood-prone areas. Users who live in the area have their email registered so that when the water level exceeds the limit, everyone will receive a notification. In previous research, the focus was generally on specific users. Apart from that, all the needs and equipment used in this research are easily accessible and the development techniques are not complicated. This is to provide flexibility for users who want to develop similar applications further. However, one thing that needs to be considered in the application development process in this research is security issues specifically physical security. This is because the application prototype produced in this research uses simple sensors with non-industrial equipment specifications. For further development, this problem can be underlined while still paying attention to needs including development costs.

2. METHODS

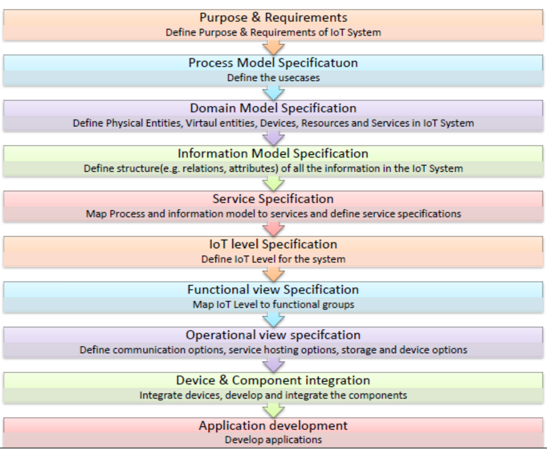

Figure 1.Research stages [10]

The application in this study was developed based on the IoT application design and development methodology proposed by Arshdeep Bahga and Vijay Madisetti. The stages of the research are as shown in Figure 1.

Based on Figure 1, the research stages in this study are:

1. Purpose & Requirements

This stage is the beginning of the whole series in research. In this phase, the author begins to conduct an analysis of the goals and needs in developing this flood early warning application.

2. Process and Domain Model Specification

Process and domain models are the stages of defining the use case and workflow of the application to be developed. In this phase, the components needed to develop the application in this research are also defined.

3. Information Model, Service and IoT Level Specification

This stage is carried out by determining the IoT Level by looking at the structure and technology used. In addition, this phase also defines what models and services will be contained in the application.

4. Functional and Operational View Specification

At this stage, the author begins to map the structure of the IoT application, especially from a functional and operational perspective. In this phase, an overview of the application and the technology used will be displayed.

5. Device and Component Integration

This phase contains the integration design of all devices and components that have been defined previously. The detailed physical design of the application will be drawn at this stage and will serve as a reference in the development process.

6. Application Development

At this stage, the flood early warning application is implemented, both in the form of code and user interface. All services that have been previously defined, will be developed and integrated with each other.

7. Testing

The last stage in this research is to test the resulting application. Testing is carried out using the blackbox method which aims to ensure that all application functionality can run properly.

Jurnal Teknik Informatika Vol. 16 No. 2, October 2023 (133-140)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v16i2.32161

3. RESULTS AND DISCUSSION

3.1. Purpose & Requirement

The purpose of this research is to develop an IoT-based flood early warning application that can be easily applied in flood-prone areas and can be developed and used easily by users. Based on the results of the analysis, identification of application needs in this study are:

a. HCSR04 Ultrasonic Sensor
b. Alarms (Buzzers)
c. Arduino UNO R3
d. Breadboard 400 Points
e. Jumper Cable
f. SMTP e-mail accounts
g. MongoDB databases

To meet needs and support the completion of this research, the author used equipment that was easily accessible and spent around IDR 300,000 for application development. This is of course very affordable to realize. However, as mentioned in the Introduction chapter, to minimize security issues, more appropriate equipment may be needed that meets industry specifications but is still affordable.

3.2. Process and Domain Model Specification

Process specifications and application domains in this study are illustrated through block diagrams as shown in Figure 2.

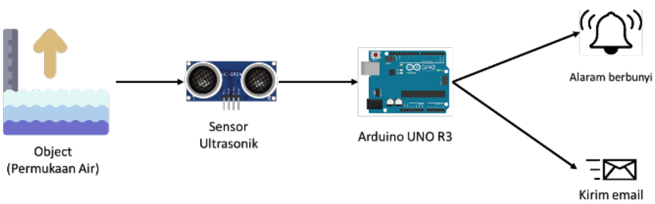

Figure 2.Application block diagram

Based on Figure 2, the ultrasonic sensor connected to the Arduino will read the distance between the sensor and the object (water surface), when the water level approaches the sensor it is a sign that the water level has less than equal to 10 and will send the status of the water level via email and will sound buzzer.

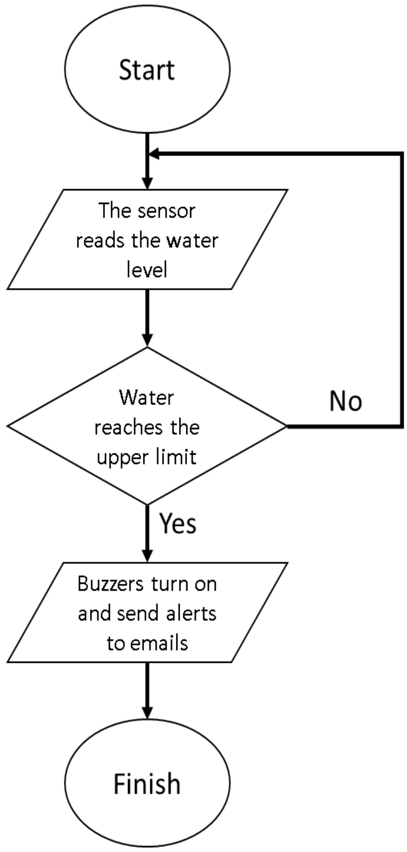

Figure 3.Application flowchart

The flowchart of the IoT-based flood early warning application in this study is shown in Figure 3. In general, when a user uses the application in this research, the user interface and how it works are:

a. The device was developed and installed in areas prone to flooding. It is recommended in areas where water first passes, for example ditches or rivers.
b. All users who need to receive water level notifications are registered by email.
c. The sensor will work by reading the distance from the water level to the sensor. If the distance between the water and the sensor is less than or equal to 10 cm, the sensor will save the data to the database and send notifications to registered users.
d. The sensor will also provide notification to the buzzer to sound an alarm.
e. When the user has received a notification and heard the alarm sound, they can immediately check the condition of the water level. If the water level is indeed high, they can prepare for the next anticipatory steps.

Jurnal Teknik Informatika Vol. 16 No. 2, October 2023 (133-140)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v16i2.32161

3.3. Information Model, Service and IoT Level Specification

As a prototype, the flood early warning application in this study is IoT Level 1 because the application is developed with a single node where the data and analysis are carried out on a local host. The application information model is illustrated through a class diagram as shown in Figure 4:

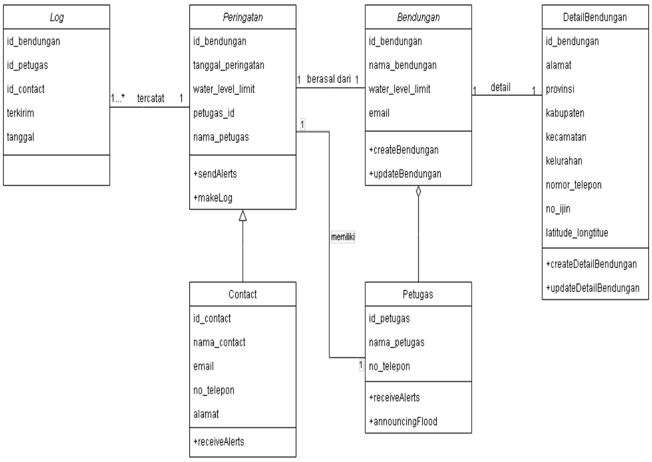

Figure 4.Design class diagrams

3.4. Functional and Operational View Specification

The functional and operational views of the applications developed in this study are shown in Figure 5:

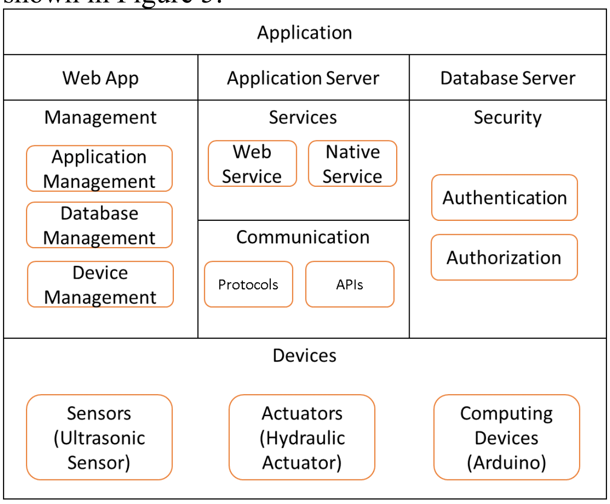

Figure 5.Functional and operational view

3.5. Device and Component Integration

In this research, a schematic circuit of tools is made that describes the integration of the devices and components used. The goal is to make it easier when connecting each electronic module with the Arduino Uno microcontroller board according to the pins used and ensure they are connected properly. The details are as in Figure 6.

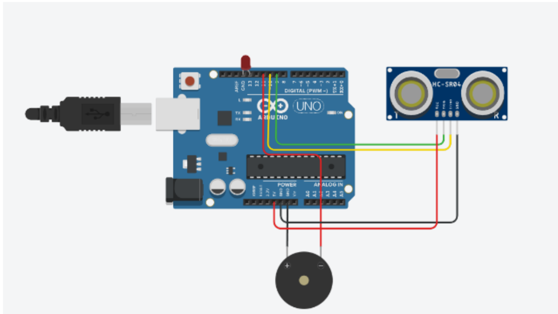

Figure 6.Schematic circuit of device and component

3.6. Application Development

The results of the application implementation in this study are shown in Figure 7. All existing devices and components have been integrated and run using C and Python languages.

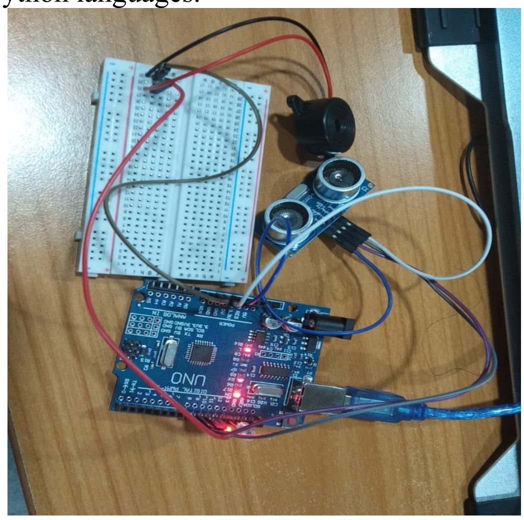

Figure 7.Application implementation results

The flood early warning application in this study was developed using C and Python languages. The implementation process is carried out by creating object-oriented program code. Some important points in the implementation process are shown in the program code snippets as shown in Figure 8.

```
void callBuzzer(int safeDistance, int turnOff){
    if(safetyDistance <= 10){
        digitalWrite(ledPin, HIGH);
        tone(buzzer, 10000, 500);
        // noTone(buzzer);
        delay (1000);
    } else {
        digitalWrite(buzzer, LOW);
        digitalWrite(ledPin, LOW);
    }
}
```

Figure 8.SafetyDistance variable code

Jurnal Teknik Informatika Vol. 16 No. 2, October 2023 (133-140)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v16i2.32161

In Figure 8, there is a safetyDistance variable coding which indicates the safe distance of the water surface to the ultrasonic sensor if the safe distance is exceeded which is less than equal to 10 then an alarm will sound and the application will automatically send an email. Furthermore, there is coding to show user or officer email settings in flood-prone areas. The contacts registered in the database will be sent a warning email when the water level has exceeded the safe limit. The code snippet is as shown in Figure 9.

```
if(now >= dangerTimeFirst or now <= dangerTimeSecond):
  server = smtplib.SMTP_SSL('smtp.gmail.com', 465)
  server.ehlo()
  server.login(gmail_user, gmail_password)
  server.sendmail(sent_from, to, email_text)
  server.close()
```

Figure 9.Email setting code

In addition, there is also a code for setting the contents of a graphical diagram which shows the results of reading the water level by the sensor at one time. The code is illustrated in Figure 10.

```
if __name__ == "__main__":
    #port arduino
    ser = serial.Serial("COM4")
    ser.flushInput()
    #ukuran window grafik
    plot_window = 20
    y_var = np.array(np.zeros([plot_window]))
    plt.ion()
    fig, ax = plt.subplots()
    line, = ax.plot(y_var)
    massage_sent = 0
```

Figure 10.Water level graphic diagram code

Apart from that, one of the core program codes in this research is the command code for sending an email to an existing contact and it will be recorded into the database when the distance between the sensor and the water surface exceeds the threshold. The code snippet is as shown in Figure 11.

```
if(distance > 0 and distance <= 10): #water level
  client
=pymongo.MongoClient("mongodb://root119203008:password
119203008@ac-78tbgc3-shard-00-
00.zau7bk7.mongodb.net:27017,ac-78tbgc3-shard-00-
01.zau7bk7.mongodb.net:27017,ac-78tbgc3-shard-00-
02.zau7bk7.mongodb.net:27017/?ssl=true&replicaSet=atla
s-47g18x-shard-
0&authSource=admin&retryWrites=true&w=majority",
tls=True, tlsAllowInvalidCertificates=True) #isi pesan
ke email
   db = client["db_arduino"]
   collection = db["data"]
   data = { "name": "Waduk Karangkates ", "address":
   "Karangkates, Kec. Sumberpucung, Kabupaten Malang,
Jawa Timur 65165","status":"Danger" }
   x = collection.insert_one(data)
   gmail_user =
'setiawan.aji@students.paramadina.ac.id'
   gmail_password = 'xjlngcrxczkxnfhi'
   sent_from = gmail_user
   to = ['mlg.hafiizh@gmail.com']

   try:
   #ketika jam lokal lebih dari 18.00 dan kurang dari
08.00
   #secara otomatis system akan mengirim email
peringatan
      if(now >= dangerTimeFirst or now <=
dangerTimeSecond):
      server = smtplib.SMTP_SSL('smtp.gmail.com', 465)
      server.ehlo()
      server.login(gmail_user, gmail_password)
      server.sendmail(sent_from, to, email_text)
      server.close()
   except:
      print("Something went wrong...")
      massage_sent += 1
```

Figure 11.Email sending code

According to Figure 10, if the sensor has recorded data on the distance of the water surface to the sensor, the user can see a graph of the recorded distance through the application, as shown in Figure 12.

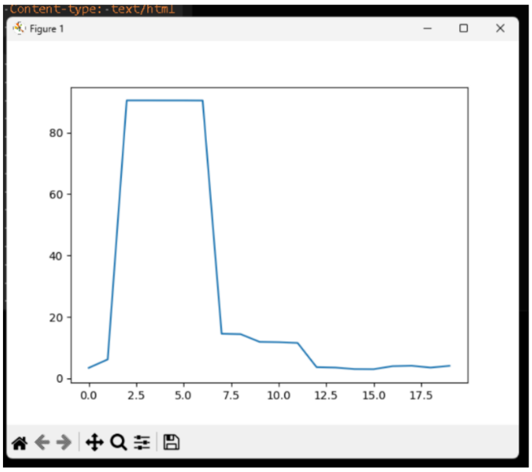

Figure 12. Graph monitor display of water surface distance to sensor

In general, the flood early warning application in this study can run well. When the water level reaches the threshold value and the ultrasonic sensor reads it, the application will send a notification by email to the user, as shown in Figure 10.

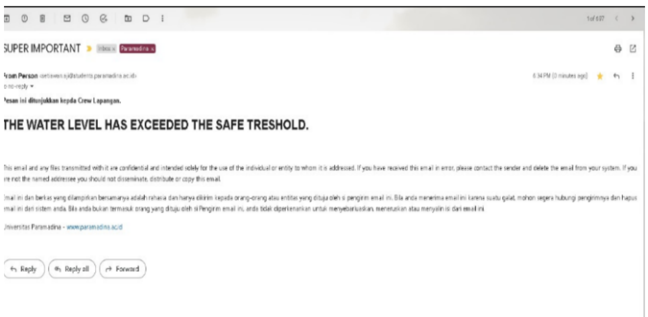

Figure 10.Display of water level notification via email

3.7. Testing

After the entire application implementation process is complete, the author tests the application using the blackbox method. The early flood warning system developed in this research was installed in the area around the campus, especially in ditches, for a period of time. To make it easier, the author gave water to the test area with different intensities. This was done considering the limitations of the existing prototype application. There are several features and processes that are tested to ensure the application can work according to the initial design. The test results are as shown in Table 1.

Table 1.The results of application testing with the blackbox method

ID	Test Description	Expected Result	Test Result	Conclusion
A1	There is water at a certain height and close to the sensor.	The system will read the water level distance data and save it to the database	The system reads the water level distance data and saves it to the database	Success
A2	Clicking on the menu displays the water distance recording graph	The system will display a graph of the distance and water level	The system displays a graph of the distance and water level	Success
A3	The water level exceeds the threshold.	The system will sound a buzzer and send information to the registered email and record the history to the database	The system sounds a buzzer and sends information to the registered email and records its history to the database	Success

Table 1 shows some important points in application testing. Based on all testing activities, the authors found that the entire process was successfully executed. This shows that the functionality of the application in this study is successful and can be applied. Further improvements may be made by further researchers, such as the use of the latest tools and sensors, so that the results of this study can be truly useful for the community, especially those in flood-prone areas.

As a discussion, one of the things that needs to be underlined regarding this research is the issue of security, especially the physical security of the device. If the physical IoT early flood warning system device is not designed in such a way, it may be hampered by weather problems that cause the device not to last long. This is because the device was developed using simple equipment so it is prone to damage. For further development, perhaps a similar device could be used but with industrial specifications that are more resistant to weather conditions. However, the cost factor also needs to be considered, because the price of devices with industrial specifications is higher. This is because the target of this research is the end user. Regarding data security and privacy, the application developed in this research is quite reliable. Users and researchers can then carry out further development for better results.

CONCLUSION

The conclusions in the research on IoT based early flood detection system with arduino and ultrasonic sensors in flood-prone areas are:

1. The IoT based early flood detection system in this research was developed using the IoT design and development method so that users can understand the needs and initial development scheme including costs easily. The devices and tools used in application development in this research are also easy to reach.
2. Arduino Microcontroller IoT Devices and Ultrasonic Sensors can be applied in flood early detection systems. The application produced in this study can record the distance between the water level and the sensor and save it to the database. In addition, the application can send notifications to registered users when the water level has exceeded a threshold.

3. All the functionality of the flood early detection system in this study has been tested using the blackbox method. The results show that all features and processes can run well so that they can be implemented.

REFERENCES

[1] B. Mazon-Olivo and A. Pan, "Internet of Things: State-of-the-art, Computing Paradigms and Reference Architectures," *IEEE Lat. Am. Trans.*, vol. 20, no. 1, pp. 49–63, 2022, doi: 10.1109/TLA.2022.9662173.

[2] J. Wang *et al.*, "IoT-Praetor: Undesired Behaviors Detection for IoT Devices," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 927–940, 2021, doi: 10.1109/JIOT.2020.3010023.

[3] L. Chettri and R. Bera, "A Comprehensive Survey on Internet of Things (IoT) Toward 5G Wireless Systems," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 16–32, 2020, doi: 10.1109/JIOT.2019.2948888.

[4] H. Kim *et al.*, "IoT-TaaS: Towards a Prospective IoT Testing Framework," *IEEE Access*, vol. 6, no. c, pp. 15480–15493, 2018, doi: 10.1109/ACCESS.2018.2802489.

[5] G. Bedi, G. K. Venayagamoorthy, R. Singh, R. R. Brooks, and K. C. Wang, "Review of Internet of Things (IoT) in Electric Power and Energy Systems," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 847–870, 2018, doi: 10.1109/JIOT.2018.2802704.

[6] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, "Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT Scenarios," *IEEE Access*, vol. 8, pp. 23022–23040, 2020, doi: 10.1109/ACCESS.2020.2970118.

[7] K. Vinothini and S. Jayanthy, "IoT based flood detection and notification system using decision tree algorithm," *2019 Int. Conf. Intell. Comput. Control Syst. ICCS 2019*, no. Iciccs, pp. 1481–1486, 2019, doi: 10.1109/ICCS45141.2019.9065799.

[8] A. A. Rashid, M. A. M. Ariffin, and Z. Kasiran, "IoT-Based Flash Flood Detection and Alert Using TensorFlow," *2021 11th IEEE Int. Conf. Control Syst. Comput. Eng. (ICCSCE), Penang, Malaysia*, pp. 80–85, 2021, doi: 10.1109/ICCSCE52189.2021.9530926.

[9] F.- Puspasari, I.- Fahrurrozi, T. P. Satya, G.- Setyawan, M. R. Al Fauzan, and E. M. D. Admoko, "Sensor Ultrasonik HCSR04 Berbasis Arduino Due Untuk Sistem Monitoring Ketinggian," *J. Fis. dan Apl.*, vol. 15, no. 2, p. 36, 2019, doi: 10.12962/j24604682.v15i2.4393.

[10] A. Bahga and V. Madisetti, *Internet of Things: A Hands-on Approach*, 1st Editio. New Delhi: Universities Press, 2015.

[11] F. Susilawati, Z. Haritsah, and Safwan, "Aplikasi Peringatan Bencana Banjir Berbasis Android," *J. J-Innovation*, vol. 9, no. 1, pp. 7–10, 2020.

[12] Sumarno, B. Irawan, and Y. Brianorma, "Sistem Peringatan Dini Bencana Banjir Berbasis Mikrokontroler Atmega 16 Dengan Buzzer Dan Short Message Service (SMS)," *J. Coding Sist. Komput. Univ. Tanjungpura*, vol. 1, no. 1, 2013, [Online]. Available: http://jurnal.untan.ac.id/index.php/jcskommipa/article/view/2317.

[13] A. Ahlul, K. Ramadhan, E. Kurniawan, and A. Sugiana, "Perancangan Sistem Peringatan Dini Banjir Berbasis Mikrokontroler Dan Short Message Service (SMS)," *e-Proceeding Eng.*, vol. 7, no. 1, pp. 178–186, 2020.

[14] P. J. Nainggolan, M. Najoan, and S. Karaow, "Pengembangan Sistem Informasi Peringatan Dini Banjir di Kota Manado Berbasis Internet of Things," *Tek. Inform.*, vol. 15, no. 1, pp. 65–74, 2020, [Online]. Available: https://ejournal.unsrat.ac.id/index.php/informatika/article/view/29064.

[15] A. Prasetyo and R. Rahmat, "Rekayasa Sistem Peringatan Dini Bencana Banjir Berbasis IoT Menggunakan Raspberry Pi," *J. Teknol. Technoscientia*, vol. 15, no. 1, pp. 29–35, 2022, doi: 10.34151/technoscientia.v15i1.4035.