Jurnal Jaringan Telekomunikasi E-ISSN: 2654-6531 P- ISSN: 2407-0807 Vol. 12, No.1 (2022)5–11

Weather Condition Monitoring System as A Floods Prevention in Malang using Android Application

Audia Yumna Dita Pramesti [1], Farida Arinie Soelistianto[2], Lis Diana Mustafa[3]
[1,2] Digital Telecommunication Networks Study Program
[3] Telecommunication Engineering Study Program
Department of Electrical Engineering, State Polytechnic of Malang, Malang, Indonesia

[1] audiayumnadita@gmail.com, [2]farida.arinie@polinema.ac.id, [3]lis.diana@polinema.ac.id

***Abstract*— On January 2021 rain often splashing Malang City with intensity high and deep period a long time, so some areas experience flood. BMKG is an Indonesian official institution in charge to give information forecast weather good through online media nor offline. However, they do not give specific information about real-time location weather. This study aims to make it easy for user in accept information on bulk rain, water level, wind speed and temperature as effort to prevent flood. This study proposed prototype using bulk sensor rain, water level, wind speed and temperature as well as using ESP32 as microcontroller with monitoring via android. This tool could put in general place or in a vulnerable flood area. Results show that sending data from ESP32 microcontroller to firebase database is successful sent through "AVRO POLINEMA" Wi-Fi connection. QoS performance between ESP32 to Firebase shows throughput = 7.076368 Kb / s (poor), packet loss = 0.5% (very good), delay = 250,582 s (good), jitter = 0.0299139 ms (good) and between Application to Firebase got results throughput = 32.24112 (medium), packet loss = 0.4% (very good), delay = 118.009 ms (very good), jitter = 0.0903578947 ms (good).**

Keywords*— Android, BMKG, Firebase, Rainfall, Monitoring.

I. INTRODUCTION

Malang City is a a region located in the Province of East Java. In general, the Malang Raya area is surrounded by mountains fiery in the east and west, while in the South consist from hills aged tertiary and southern Indonesian seas. Various condition this disaster-prone, including eruptions Mountain fiery, potential land landslide, flood flash, and earthquake earth [1] .

The Meteorology, Climatology and Geophysics Agency (BMKG) is a official Indonesian institution which serves information for monitor weather and climate in Indonesia. For weather information, BMKG work together with a number of weather station monitor in a number of important point in Indonesia. Information about forecast weather and climate every day will be delivered to public through online media such as websites, apps weather on smartphone , radio, television even offline media as newspapers and some type of print media other [2].

Because of the existing system now not yet apply real-time information, moreover the registration of vulnerable flood locations still use manual map[3] or should look for information through online and offline media. The development existing technology is expected to help public by giving information about limit of heavy rain intensity, water level, wind speed and environment temperature.

Based on the above problem, then Writer will propose “Weather Condition Monitoring System as A Floods Prevention in Malang using Android Application”. The system could send information on smartphones applications in the form of map location of the tested area, bulk rain intensity, wind speed, water level, and environment temperature around location. The system will be used to monitor weather condition at the rain and drought season. As for the important parameters is bulk rain intensity, wind speed and water level. For the coordinate of tested area, is taken from Google Maps

II. LITERATURE REVIEW

A. Microcontroller (ESP32)

ESP32 Wi-Fi module is a latest wifi development board of ESPs that use the ESP32 chip. Besides give connection wifi, module it also gives Bluetooth connection BLE equipped with a dual core 32bit MCU. This Wi-Fi Module has voltage 3.3V so that to make electronic using ESP32 user must notice that the electricity supply in the circuit cannot more than 3.3V [4].

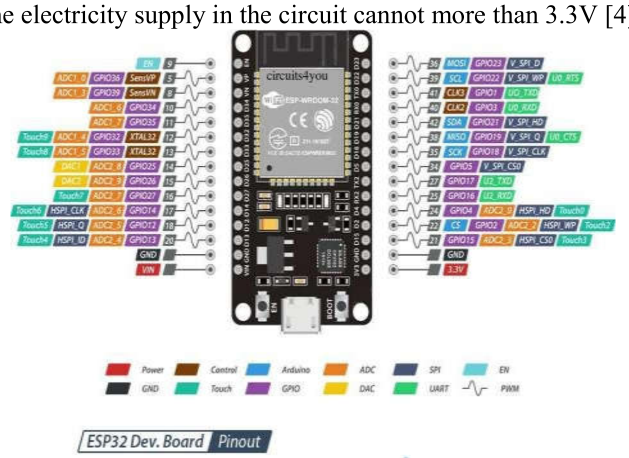


Figure 1. Microcontroller (ESP32)

B. *Anemometer Sensor*

The anemometer sensor is a sensor to measure speed the wind around and is used on several station measurement weather. One of the wind speeds measurement that can be used is count the time it happened in each appearance signal pulse [5].

Figure 2. Anemometer Sensor

C. *Rainfall Sensor*

Automatic Rain Gauges Type Tipping (Tipping Bucket) is rain meter that uses weight principle, that is when rain down, then the water will enter in to the heavy rain funnel accommodated in the bucket. When the rain water is full equivalent with 0.5 mm tall (or in accordance with sensor specifications) will tiptoes and the water is removed, then one of the bucket side will tiptoe [6][7].

D. *Proximity Sensor with Ultrasonic (HC-SR04)*

Ultrasonic sensor is a sensor that works with the principle of sound wave reflection and is used to detect the presence of a particular object or object with a working frequency in the area above the sound wave of 20 kHz to 2 MHz[8]. This sensor consists of two units, namely a transmitter unit and a receiver unit. The working principle of this sensor is the transmitter unit sends an ultrasonic wave and then measures the time it takes for a reflection to come from an object. This length of time is proportional to twice the sensor distance [9].

Figure 3. Proximity Sensor with Ultrasonic HC-SR04

E. *Temperature Sensor (DS18B20)*

The DS18B20 sensor provides 9 bits to 12 bits (can be configured) reading temperature showing device temperature. Information sent to/from DS18B20 via 1-wire interface, so that only one necessary wires and ground connected from microprocessor center to DS18B20. The DS18B20 sensor can also powered from external *supply* 3-5.5V [10].

Figure 4. Temperature Sensor (DS18B20)

F. *Firebase*

Firebase is an API platform provided by google for Android, iOS, and Web applications. This platform offer analytics, development, cloud messaging, real-time database, database authentication, hosting, and other features [11].

G. *Android Studio*

Android studio is an official platform of the IDE (Integrated Development Environment) used for development android application and is open source (free). The launch of Android Studio was announced by Google on May 16, 2013 at the I/O Conference event and at the it Android Studio replaces Eclipse as official IDE for develop Android application [12].

H. *Wireshark*

Wireshark is a network protocol structure or protocol (sniffer) analyzer provided for free. Wireshark can operate on the Microsoft Windows, Linux, and Unix operations. Wireshark also supports Graphical User Interface (GUI) and has feature filtering information [13].

III. RESEARCH METHODS

A. *System Block Diagram*

Block diagram of this study shown in figure 5. Based on fig 5 there are four sensors; temperature sensor (DS18B20), wind speed sensor (anemometer), bulk rain gauge sensor (rain gauge), and water level sensor (ultrasonic). Data from all these sensors will be sent to database through ESP32 microcontroller which has been included with Wi - Fi module. Data will be saved to databases (firebase), after that data will be shown to user via android application.

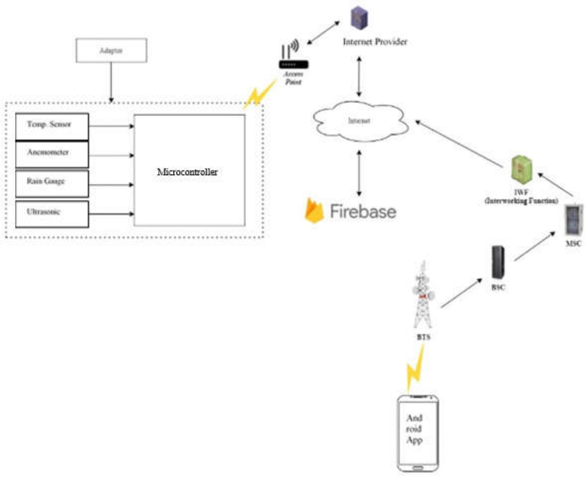


Figure 5. System Block Diagram

B. *Flowchart*

Complete flowchart from this study is as following:

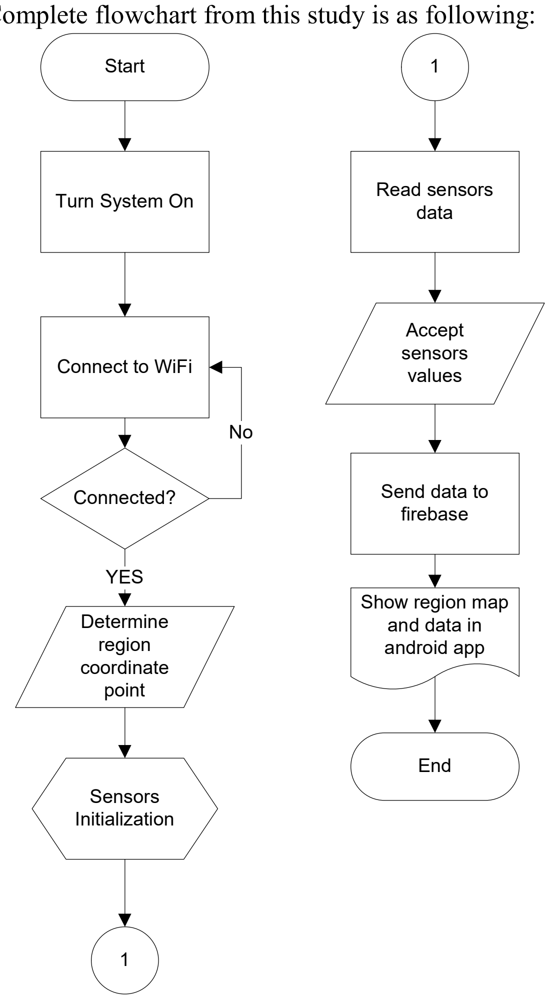


Figure 6. Complete System Flowchart

The first stage when start is turn on system. System will be turned on for 5 days for knowing weather condition in the region. Regional differences sometimes causing same bad weather condition. Second stage is connecting system to wifi. Wi-Fi Network in use is the closest from the place this prototype put and confirmed could be connected. Third stage, if system could connect to Wi-Fi network, is determining the coordinates point of the area that have been inputted into the android application determined or taken from Google Maps location where testing will be done. However, if could not connected to network then system will return to second stage. Fifth stage is initialization of temperature sensor, wind speed sensor, rainfall sensor, and water level sensor. These four sensors will be turned on for testing.

After system read temperature sensor, wind speed sensor, rainfall sensor, and water level sensor, then microcontroller receive data from the four sensors and will be sent to Firebase via Wi-Fi and the process of sending data to Firebase will take place. The last stage is showing map of the area and data from the sensors to android application to be monitored by the user.

C. *Test Parameters*

Parameters used for study this are:

1. Anemometer sensor accuracy for knowing wind speed.
2. Rain gauge sensor accuracy to measure bulk rain that falls on the rain gauge sensor funnel.
3. Ultrasonic sensor accuracy to measure water level.

D. *Mechanical Planning*

This step plans device for make it manufacturing easier on next stage. The mechanical plan is illustrated in the image below.

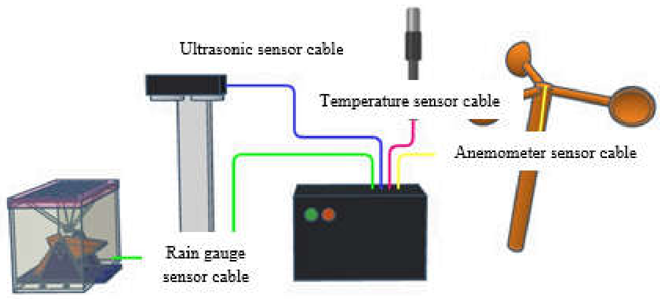


Figure 7. Mechanical Planning Illustration

E. *Electrical and Circuit Schematic Planning*

Figure 8 shows the complete schematic of this paper.

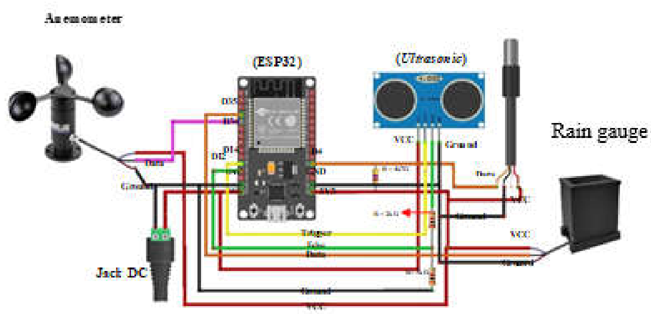


Figure 8. Complete Circuit Schematic

IV. RESULTS AND DISCUSSION

A. *Hardware Results*

Hardware Results is shown in the image below

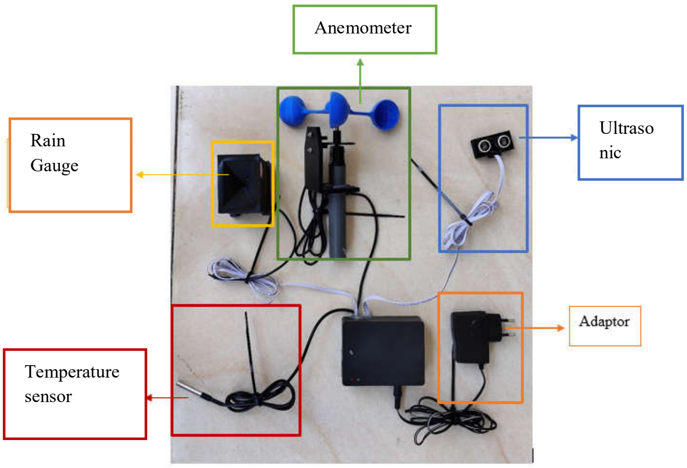


Figure 9. Hardware Results

B. *Software Results*

1) *Information Page*

On page information shows Google Maps map where the prototype put. This prototype placed in the Kedungkandang, Lowokwaru, and Sukun District. This page containing an information of weather condition including rain rate (mm), wind speed (m/s), water level (cm), and temperature (° C) and equipped with time and real-time graph.

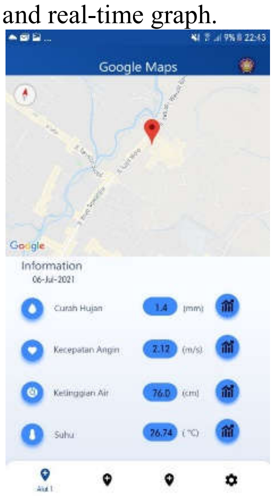


Figure 10. Information Page

2) *Real-time Chart Page.*

This page shows chart of each data that has been monitoring in a period of time.

Figure 11. Real-time Chart Page

3) *Settings page*

This page is used to enter or change data. Data that can be changed covers prototype location, its latitude and longitude, and placement height.

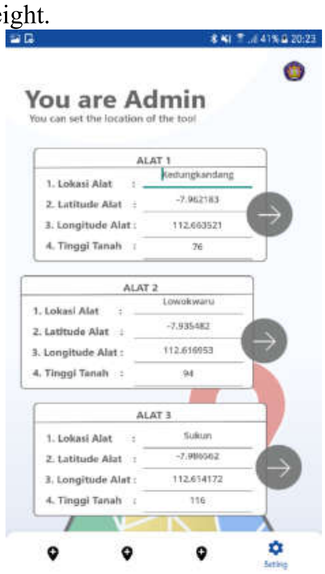


Figure 12. Settings page

C. *Database Test*

1) *Microcontroller Connectivity Test*

In this test, ESP 32 will be connected to the internet by adding Wi-Fi password and linked SSID [14]. Fig 13 shows the Arduino IDE program on entering Wi-Fi password and SSID.

```
FirebaseDemo_ESP32 §
#include <WiFi.h>
#include <IOXhop_FirebaseESP32.h>

// Set these to run example.
#define FIREBASE_HOST "https://ditaapk-20abe-default-rtdb.firebaseio.com/"
#define FIREBASE_AUTH "aSNWOCQmzfRMfX4NPV7D2oDaEmQf9nVgJtPhOv2Q"
#define WIFI_SSID "AVRO POLINEMA"
#define WIFI_PASSWORD "avro@123"
```

Figure 13. Password and SSID in Arduino IDE Program

When connection is successful, Arduino IDE serial monitor will show the notification.

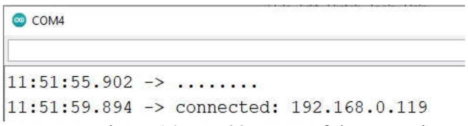


Figure 14. ESP32 Successful Connection Display

2) Sending Data to Firebase Test

Test done when ESP32 has been connected to internet by adding Firebase Authentication and Firebase Host on Arduino IDE program [15].

Figure 15. Firebase Authentication and Firebase Host Code

Data that will be sent could be seen on the Arduino IDE serial monitor. In Fig. 16, the data has been successfully stored in database.

Figure 16. Data Successfully Stored in Real-time Firebase Database

D. Quality of Service (QoS) Test

1) Testing Between ESP32 to Firebase

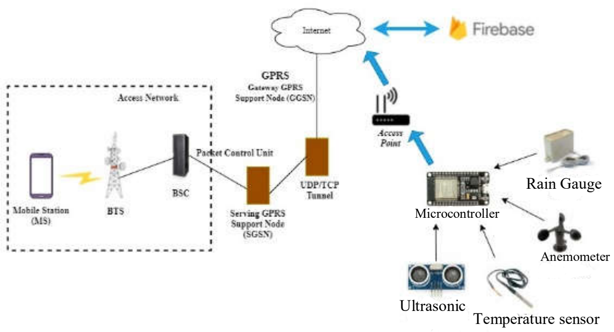


Figure 17. QoS Testing Between ESP32 to Firebase (Blue Colored Path)

a) Throughput Test

Measurement	Captured	Displayed	Marked
Packets	193	193 (100.0%)	—
Time span, s	57.772	57.772	—
Average pps	3.3	3.3	—
Average packet size, B	265	265	—
Bytes	51102	51102 (100.0%)	0
Average bytes/s	884	884	—
Average bits/s	7076	7076	—

Figure 18. Display of Capture File Properties in Wireshark

$$Throughput = \frac{jumlah\ bytes}{time\ span}$$

$$= \frac{51102\ bytes}{57.772\ s}$$

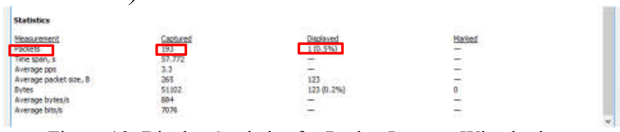
= 884,546 bytes/s

= 0.884546 KB/s

= 0.884546 x 8

= 7.076368 Kb / s (Poor)

a) Packet Loss Test



Statistics

Measurement	Captured	Displayed	Marked
Packets	193	1 (0.5%)	—
Time span, s	57.772	—	—
Average pps	3.3	—	—
Average packet size, B	265	123	—
Bytes	51102	123 (0.2%)	0
Average bytes/s	884	—	—
Average bits/s	7076	—	—

Figure 19. Display Statistics for Packet Loss on Wireshark

Packet Loss = (((Packet Sent – Package Received) / Package Sent) x 100)

= (193 – 192) / 193) x 100)

= (1/193) x 100

= 0.518

= 0.5% (Good)

a) Delay Test

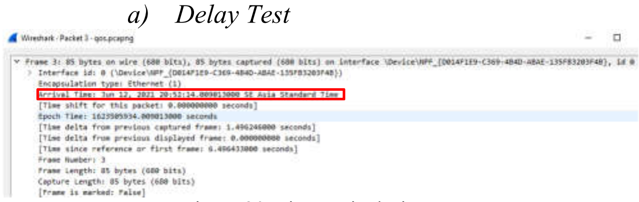


Figure 20. First Arrival Time

Figure 21. Second Arrival Time

Package delay

= (Second time – First time)

= (14.09486000 - 14.009013000)

= 0.000473 s

Total delay = total whole delay package

= 38.08841 s

Average delay

= total number of delays / received packets

= 38.08841 / 152

= 0.250582 s

= 250,582 ms (Good)

b) Jitter Test

Total jitter = sum of whole jitter pack

= 0.004517 s

Total delay variation

= Delay - (average delay)

= 38.08841 – (0.250582)

= 37.837828 s

Jitter

= Total delay variation / Total packets received

= 37.837828 / 151

= 0.250581 s

= 250,581 ms

Average jitter
= total number of jitters / packets received
= 0.004517 s / 151
= 0.0000299139 s
= 0.0299139 ms (Good)

2) QoS Testing Between Applications to Firebase

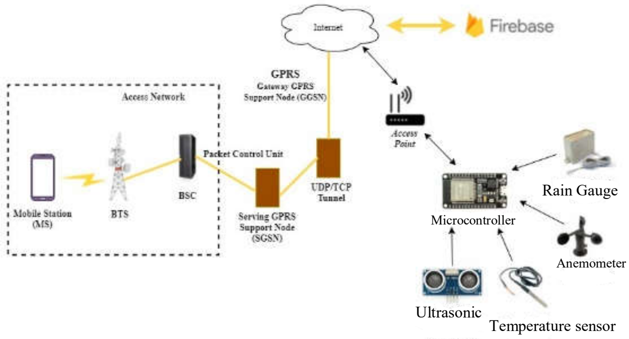


Figure 22. QoS Testing Between Applications to Firebase (Yellow Colored Path)

a) Throughput Test

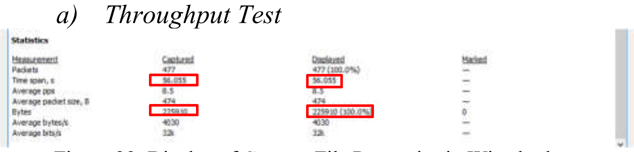


Figure 23. Display of Capture File Properties in Wireshark

$$Throughput = \frac{jumlah\ bytes}{time\ span}$$
$$= \frac{225910\ bytes}{56.055\ s}$$

= 4030.14 bytes/s
= 4.03014 KB/s
= 4.03014 x 8
= 32.24112 Kb / s (Medium)

b) Packet Loss Test

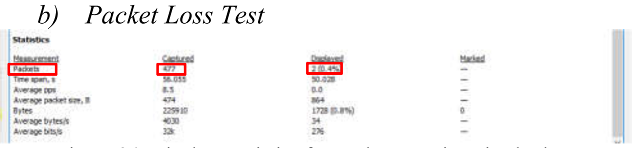


Figure 24. Display Statistics for Packet Loss in Wireshark

Packet Loss = (((Packet Sent – Package Received) / Package Sent) x 100)
= (477 – 475) / 477) x 100)
= (2/477) x 100
= 0.419
= 0.4% (Very Good)

c) Delay Test

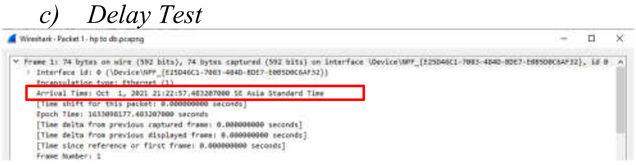


Figure 25. First Arrival Time

Figure 26. Second Arrival Time

Packet delay
= (Second time – First time)
= (57.526123000 – 57.483207000)
= 0.042916 s

Total delay = total of whole delay packets
= 56.05457 s

Average delay
= total number of delays / received packets
= 56.05457 / 475
= 0.118009 s
= 118,009 ms (Very Good)

d) Jitter Test

Jitter packet
= (Second delay – First delay)
= (0.001228 - 0.042916)
= -0.041688 s

Total jitter = sum whole jitter pack
= -0.04292 s
Total delay variation = Delay - (average delay)
= 56.05457 – (0.118009)
= 55.936561 s

Jitter
= Total delay variation / Total packets received
= 55.936561 / 475
= 0.11776 s
= 117.6 ms

Average jitter
= total number of jitters / packets received
= - 0.04292 s / 475
= -0.0000903579 s
= -0.0903578947 ms
= 0. 0903578947 ms (Good)

V. CONCLUSION

The proposed prototype uses ESP 32 as main controller, bulk rain sensor, anemometer, ultrasonic, and temperature DS18B20 as end device to collect data and processed on the ESP 32. Sending data from ESP32 microcontroller to the firebase database is successful, as evidenced in Figure 15 with connection *Wi-Fi* "AVRO POLINEMA" to database successfully sent. *Real-time* data reading mechanism to android using *Quality of Service* (QoS) test which is influenced by parameters of throughput, packet loss, delay, and jitter. From the results of these parameters show that QoS performance between ESP32 to Firebase obtained results *throughput* = 7.076368 Kb / s (poor), *packet loss* = 0.5% (very good), *delay* = 250,582 s (good), *jitter* = 0.0299139 ms (good) and between Application to Firebase got results *throughput* = 32.24112 (medium), *packet loss* = 0.4% (very good), *delay* = 118.009 ms (very good), *jitter* = 0.0903578947 ms (good).

REFERENCES

[1] KF Sulistyani, GD Pandulu, ST Civil, F. Teknik, U. Tribhuwana, and T. Malang, "Mapping of critical irrigated areas in Uptd Irrigation Pujon, Malang Regency," vol. 2, no. 1, pp. 1–10, 2017.

[2] R. Noviarti, Asrizal, and Yohandri, "Development of a 2.4 Ghz Xbee Wireless Communication System Using Sent Data to Support the Tipping Vessel Rainfall Gauge at BMKG Sicincin," Pillar Phys. , vol. 9, pp. 1–15, 2017.

[3] W. Hariyani, Y. Efenie, Muhsi, JT Informatics, F. Teknik, and UI Madura, "Geographic Information System for Mapping of Natural Disaster-Prone Locations by Season in Pamekasan Regency Based on Android," vol. I, pp. 327–332, 2015.

[4] Nurhastuti, “Forest Fire Detection Using Lora (Long Range) Wireless Network Communications,” 2019.

[5] Fannida Sheila Please, “Measurement And Testing Speed Wind With Using Arduino Uno R3 Based Anemometer Sensor," Pp. 44–48, 2018.

[6] R. Bangun et al., "Design and build a weather monitoring system and automatic rainfall gauge based on iot blynk," Fak. Tech. Eletro Univ. Islam Malang, vol. 3, p. 9, 2021.

[7] H Hudiono, M Taufik, RHY Perdana, and AE Rakhmania. "Telemetering of Rainfall Measurement Results Using 433 MHz Wireless Transmission." JURNAL INFOTEL 13, no. 3 (2021): 143-150.

[8] D Defnizal, and RN Ernes. "The Implementation of Artificial Intelligence in Charity Box at Mosque and Musholla as RFID Based Security System." Sinkron: jurnal dan penelitian teknik informatika 5, no. 1 (2020): 35-42.

[9] B. Arsada, "Application of Ultrasonic Sensors for Detecting Distance Positions in Space Using Arduino Uno," J. Tek. Electrical , vol. 6, no. 2, pp. 1–8, 2017.

[10] HP Ramadhan, C. Kartiko, and A. Prasetiadi, “Monitoring of Shrimp Pond Water Quality Using Data Logging Methods,” J. Tek. information. and Sis. inf. , vol. 6, no. April, pp. 102–114, 2020.

[11] B. Mehta, N. Madhani, and R. Patwardhan, “Firebase: A Platform for your Web and Mobile Applications,” Int. J. Adv. res. science. eng. , vol. 6, no. 4, pp. 45–52, 2017.

[12] A. Juansyah, "Development of Assisted - Global Positioning System (A-GPS)-Based Child Tracker Application with Android Platform," J. Ilm. computer. and Information. , vol. 1, no. 1, pp. 1–8, 2015, [Online]. Available: elib.unikom.ac.id/download.php?id=300375.

[13] HP Widhiatmoko, H. Susilawati, F. Teknik, and UJ Sudirman, “Comparation of Voice Call Quality Throughput,” pp. 485–490, 2020.

[14] Hasibuan, Arnawan, A Qodri, and M Isa. "Temperature Monitoring System using Arduino Uno and Smartphone Application." Bulletin of Computer Science and Electrical Engineering 2, no. 2 (2021): 46-55.

[15] Maharani, A Ninda, and B Handaga. "Rancang Bangun Aplikasi Pengontrol Sistem Penyiram Tanaman Berbasis Arduino Dan Android." PhD diss., Universitas Muhammadiyah Surakarta, 2021.