Android-Based Reptile Incubator System as a Genders Determinant at Inferno Reptile

Ahmad Rozak Setia Nugraha[1], Azam Muzakhim Imammuddin[2*], Isa Mahfudi[3]

1,2 Digital Telecommunication Network Study Program, Department of Electrical Engineering, State Polytechnic of Malang, 65141, Indonesia

3 Telecommunication Engineering Study Program, Department of Electrical Engineering, State Polytechnic of Malang, 65141, Indonesia

[1]2041160152@student.polinema.ac.id , [2]azam@polinema.ac.id , [3]isa_mahfudi@polinema.ac.id

***Abstract*— The Inferno Reptile is currently experiencing problems with its incubator, which is used to hatch female Leopard Geckos because the temperature in the incubator can drop below the ideal temperature range. In this case, researchers developed an Android-based incubator that can provide information on temperature, humidity, water level, and can notify the user when the water level is less than 2 cm and it can be used to incubate Leopard Gecko, Ball Python, and Bearded Dragons with user-defined genders by selecting on the app. Based on the black box performance results, the system works 100% smoothly and can maintain the ideal temperature for each type of reptile. Based on the tests that have been carried out, the average delay in sending data to the incubator is 6.283 seconds, and the average delay in receiving the application is 622 ms and the tests Quality of Service, the throughput obtained is 579.317 bit/s, and 583.685 bit/s are included in the 'Poor' category, Delay 1022.220 ms and 1013.256 ms as 'Very Poor', Packet Loss 0.0% as 'Very Good', and Jitter 20.572 ms and 7.914 ms as 'Very Good'. As a result, Inferno Reptile can hatch three types of reptiles.**

Keywords*— *Application, Incubator, Internet of Things, Quality of Service, Reptiles.

I. INTRODUCTION

Reptiles are one type of animal that is used as a pet. The reason people keep reptiles is because of their exoticism, fairly easy maintenance, calm nature, and not easily sick [1]. One example of a reptile that is easy to keep is the Leopard Gecko. Leopard Gecko has advantages in terms of diverse colors and motifs, easy maintenance and prices ranging from hundreds of thousands to millions of rupiah [2]. The sex of the Leopard Gecko can be determined by adjusting the temperature of the incubator. The temperature used to hatch male and female Leopard Geckos is different. The temperature for hatching male Leopard Gecko ranges from 30-32°C whereas, female Leopard Gecko ranges from 26-28°C [3]. Besides being used to hatch Leopard Geckos, this incubator can also be used to hatch Ball Pythons and Bearded Dragons. The temperature used to hatch female Ball Pythons is 29-30°C and males 31-32°C [4]. Meanwhile, for hatching female Bearded Dragons, the required temperature is 28–29°C and for males, it is 30-31°C [5].

Because of the many enthusiasts, many people have started breeding reptiles, one of which is the Leopard Gecko, such as at Inferno Reptile. Inferno Reptile is a Micro, Small and Medium Enterprises (UMKM) engaged in the sale of reptile animals, reptile maintenance accessories, and reptile farming, namely Leopard Gecko. The Inferno Reptile is located at Perum. Griyashanta Blok. K, Malang City. At Inferno Reptile, there is a problem with one of the incubators whose temperature is unsTable because the temperature can drop the ideal temperature, causing eggs to fail to hatch. In addition to causing material losses, if eggs that fail to hatch rot or are affected by mould, it is feared that other eggs will be affected, such as mould, causing greater losses.

To solve the problems that exist in the Inferno Reptile, an incubator is needed that can be adjusted and maintain the stability of the temperature in the incubator [6]. The incubator temperature is set using an application. Besides being used to regulate the temperature of the incubator, the application is useful for monitoring temperature, humidity, water level, and providing information in the form of notifications on mobile phones when the water level in the incubator is less than 2 cm. The purpose of this research is to make an incubator that can be adjusted according to the type of reptile and the desired gender [7]. In addition to being used to hatch Leopard Gecko eggs, this incubator can also be used to hatch eggs of other reptiles, such as Ball Python and Bearded Dragon, by adjusting temperature requirements. Incubator setups for sexing reptiles can be used for selective breeding if needed to breed a specific morph or gender of three reptiles. And notifications are needed so that users always know the availability of water in the incubator, which is always filled to maintain air humidity in the incubator. If the incubator lacks moisture, it will have an impact on the incubated reptile eggs.

The tests carried out on this system are functionality tests using the Black Box method, system delay, and Quality of Service (QoS). The Black Box method focuses on system performance and a list of existing requirements. It is used to determine whether the system can run according to user requirements [8]. The data collected in the system delay test is

*Corresponding author

the delay in sending sensor data to Firebase and the reception delay by reducing the time the data is received in the application with the sensor sending time to Firebase [9]. In QoS, there are four parameters used for testing, namely throughput, delay, packet loss, and jitter [10]. Throughput is the total number of successful packet arrivals observed at a destination during a given time interval, divided by the duration of that time interval [11]. Delay is the time delay of a packet that occurs due to the process of sending from one point to another point as the destination of the packet [12]. Delay can be affected by distance, physical media, congestion, or also long processing times [13]. Packet loss is the failure of a data packet to reach its destination [14]. Jitter is the time shift between packets that occurs in IP networks [15]. QoS testing was carried out on two providers to determine the quality of service of the two internet services used.

II. METHOD

A. Research Design

The type of research conducted was manufacturing and development research. To solve these problems as shown in the Figure 1, the stages of system design, tool design, application design and development, parameter determination, design implementation, functionality of the system, until the formation of the system, and the system is suitable for use are needed.

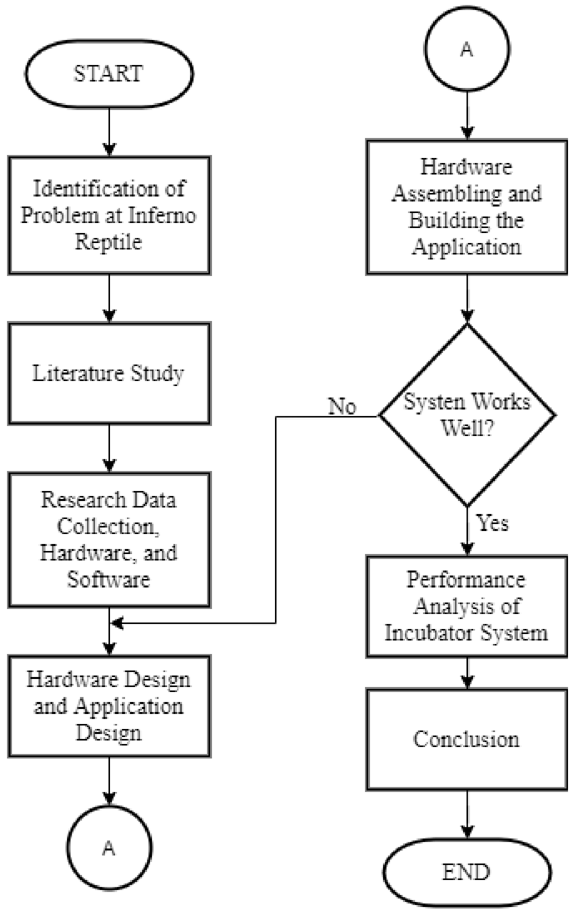

Figure 1. Research design

B. System Circuits Design

Figure 2 is a circuit of the reptile incubator system that is made. The microcontroller board used is Node MCU ESP8266, the sensor used is DHT22 to detect temperature and humidity in the incubator, and the HY-SRF05 is an ultrasonic sensor used to measure the water level. The actuators used are PTC Air Heater as heater and Peltier as cooler. All sensor data is sent to Firebase via an internet connection. The data on Firebase will be displayed on the application. The relay used is a 1-channel relay to control the PTC and Peltier because the PTC and Peltier work at 12V while the microcontroller and sensors used work at 5V. The relay is also connected to the DC fan. In Peltier, the DC fan is used to cool the Peltier and blow cold air into the incubator, while in PTC, the DC fan is used to blow hot air. Temperature control in the incubator is also carried out in the application.

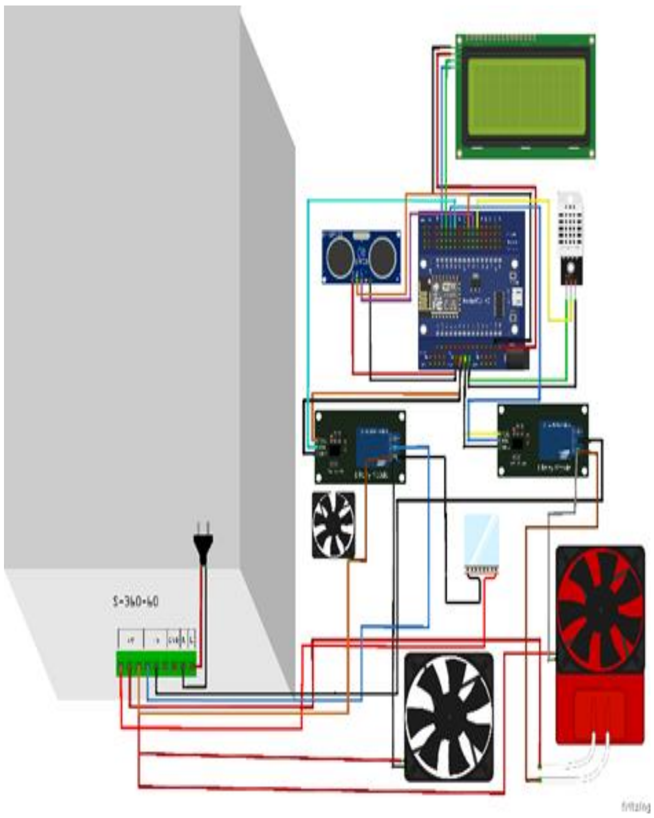

Figure 2. System circuits design

C. System Usage Flow Diagrams

Figure 3 shows the flow of the reptile incubator. The following is an explanation of the flow of using the incubator.

1. User places the reptile eggs into the incubator.
2. User selects the type of reptile.
3. User selects the sex of the reptile they want to hatch by pressing a button in the application.
4. The incubator will adjust the temperature based on the type of reptile and the sex of the reptile selected by the user.
5. There is water inside the incubator to stabilize the humidity. If the water level is less than 2cm, the system will send a notification.

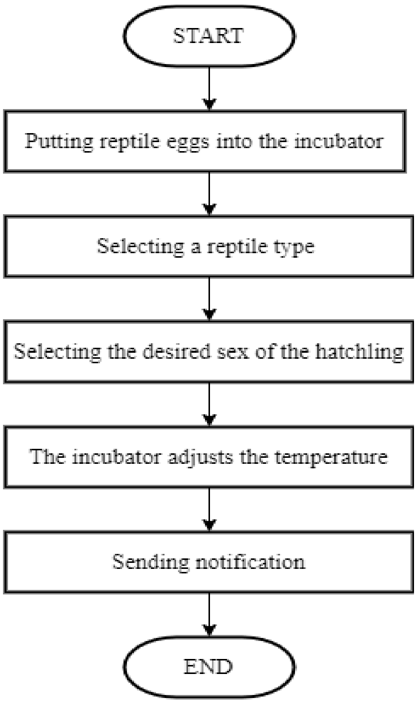

Figure 3. System usage flow diagram

D. *Working Flowchart System*

Figure 4 is a flowchart of how the incubator works. Starting from the DHT22 sensor, calculate the temperature and humidity in the incubator, display the temperature and humidity data on the LCD, and then send it to Firebase so that it can be displayed in the application. Then, the temperature adjustment process in the incubator is based on the type of reptile and gender selected by keeping the temperature at a predetermined limit. Until the calculation of the water level is displayed on the LCD and sent to Firebase to be displayed on the application, if the water level is less than two, there will be a notification received by the user.

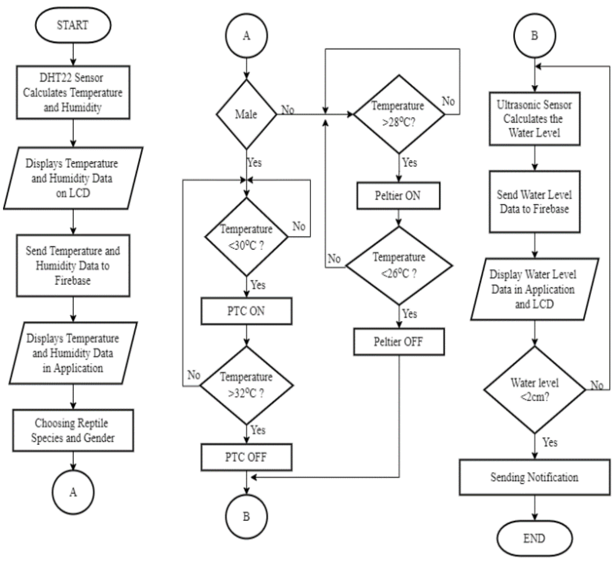

Figure 4. Flowchart system

III. RESULTS AND DISCUSSION

A. *Evaluation Blackbox Testing*

This evaluation is done by testing the incubator system that has been made with reference to the functionality of the application made and the ability of the incubator to maintain temperature.

This test uses the Black Box method, this evaluation is used to determine whether the system can run as needed. Evaluation results are shown in Table 1.

TABLE I
EVALUATION BLACK BOX TESTING

Name	Test Case	Expected Result	Evaluation
Sending data sensor	The ESP8266 MCU node sends sensor data in the form of temperature, humidity, and water level to the Firebase.	Data successfully sent to the Firebase Realtime Database	In accordance
Display the sensor data	The app display sensor data from Firebase.	The application successfully displays sensor data in Firebase.	In accordance
Sending data from app	When the button is pressed, the string data is sent to Firebase.	The application successfully sent string data to Firebase	In accordance
Reading string data from Firebase	Node MCU Esp8266 reads string data from Firebase.	Node MCU Esp8266 successfully read string data from Firebase to execute temperature regulation on the incubator	In accordance
Testing the temperatu re setting	The incubator runs the temperature settings according to the settings that have been selected in the application.	The incubator successfully executes the temperature settings according to the options in the app.	In accordance
Testing the temperatu re stability of the incubator	Testing the temperature stability of the incubator corresponding to the temperature setting for 180 minutes.	The temperature in the incubator can be maintained according to the temperature setting selected in the app.	In accordance

B. *Implementation of Incubator System*

Figures 5 to 7 present the implementation results of the incubator system that was successfully developed, including both the physical reptile incubator and the Android-based monitoring application. These figures illustrate the integration between hardware and software components within the system. The implementation demonstrates that the incubator system is capable of operating as intended, with the Android application providing real-time monitoring and control support. This integration enables users to observe system performance and interact with the incubator efficiently, confirming the practicality of the proposed design.

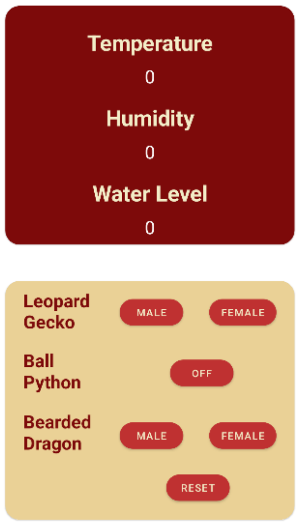

Figure 5. Android 6App

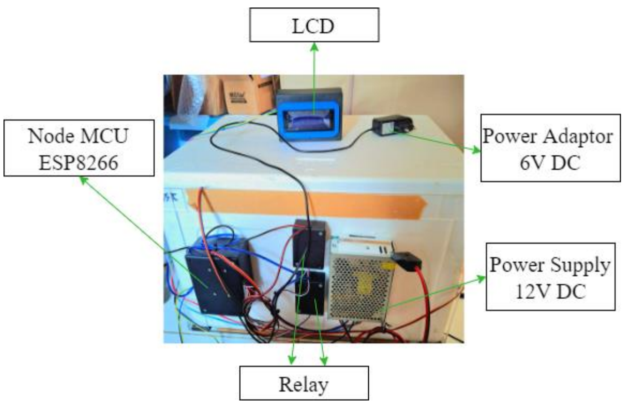

Figure 6. Incubator outside view

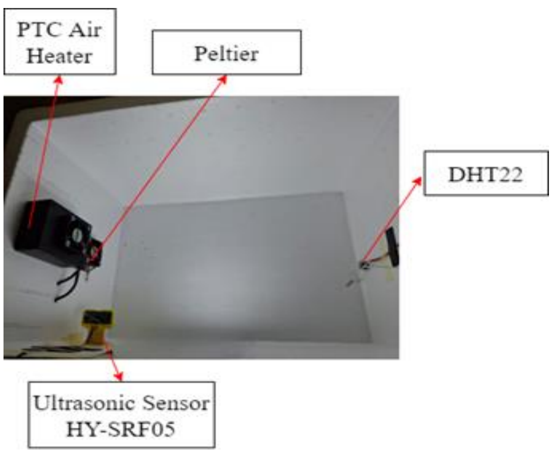

Figure 7. Incubator inside view

The dimensions of the incubator that have been made are 52cm long, 37cm wide, and 32cm high. The thickness of the Styrofoam material is 2,5cm.

C. Temperature Test Result

Are the ideal temperature test results for each type of reptile.

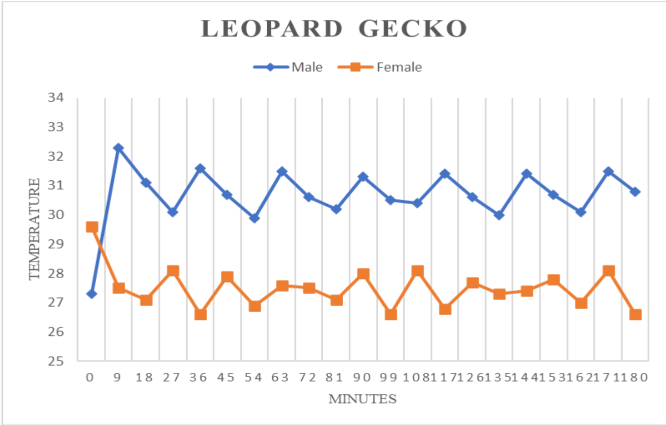

Figure 8. Temperature testing graph for leopard gecko

In Figure 8, the blue line represents the temperature setting for male leopard geckos, while the orange line indicates the temperature setting for female leopard geckos. During the initial period, from minute 0 to minute 9, the incubator system works to adjust and stabilize the internal temperature until it reaches the predetermined ideal range. The graph shows that under the male temperature setting, the incubator is able to maintain a stable temperature range of 30-32°C for a duration of up to 180 minutes. Meanwhile, for the female temperature setting, the incubator successfully maintains the temperature within the range of 26-28°C over the same time period. These results demonstrate that the incubator system is capable of controlling and sustaining the required temperature conditions consistently for both male and female leopard gecko incubation.

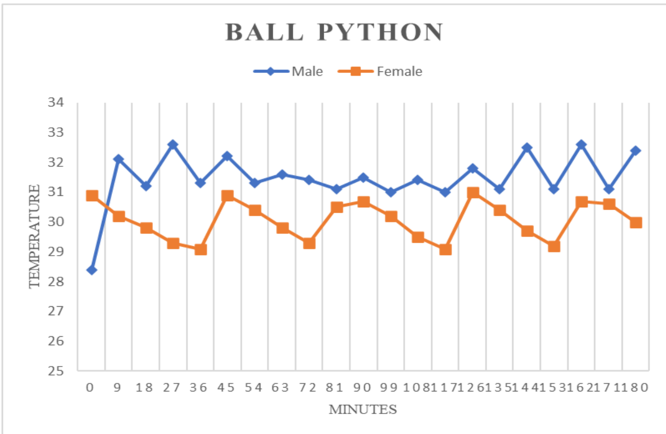

Figure 9. Temperature testing graph for ball python

Figure 9 shows the temperature measurements for Ball Python in the male temperature setting taken over 180 minutes.

The temperature in the incubator set at 31–32°C, and the female temperature set at 29–30°C. At minutes 0 to 9, there was an increase in temperature from 28°C to 32°C. This is due to the incubator adjusting the room temperature to the ideal temperature for male Ball Pythons.

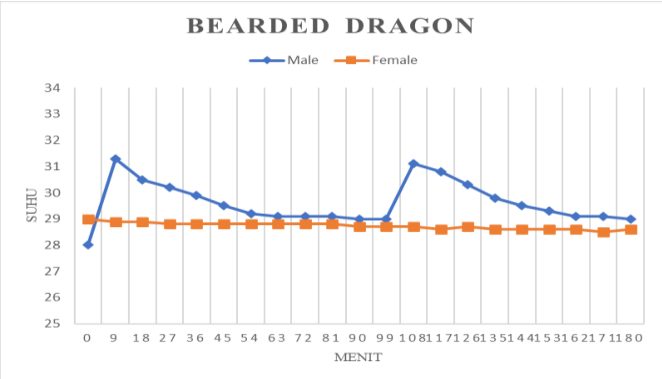

Figure 10. Temperature testing graph for bearded dragon

Figure 10 shows that when the temperature for male Bearded Dragon is not ideal, the incubator will condition the ideal temperature in the range of 29–31°C. The temperature in the incubator can also be maintained for 180 minutes without any change in temperature outside the range that has been set. For females, the incubator temperature remains in the 29–30°C range for 180 minutes.

The three graphs show that the temperature in the incubator can maintain a stable temperature according to the specified settings. The temperature in the incubator for 180 minutes did not change significantly outside the range of temperature settings for each type and sex of reptile.

D. Data Transmission Delays

Delivery delay testing is carried out to determine the delay in sending sensor data to Firebase from the NODE MCU in each loop of the running programme. The method used is to reduce the delivery time of two with the delivery time of one based on the time displayed on the Arduino IDE Serial Monitor, as shown in Figure 11.

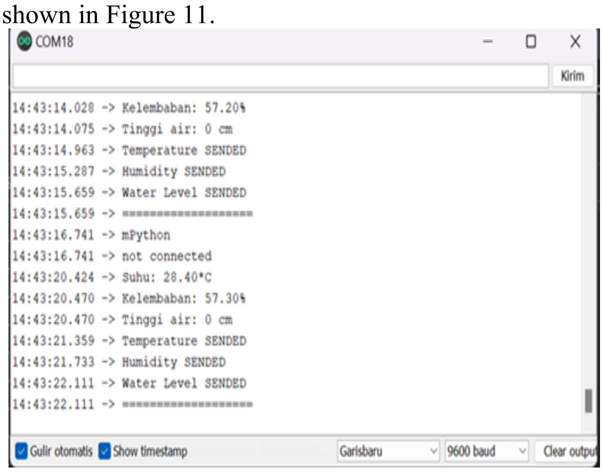

Figure 11. Timestamp on serial monitor

The results of the delivery delay data are shown in Figure 12.

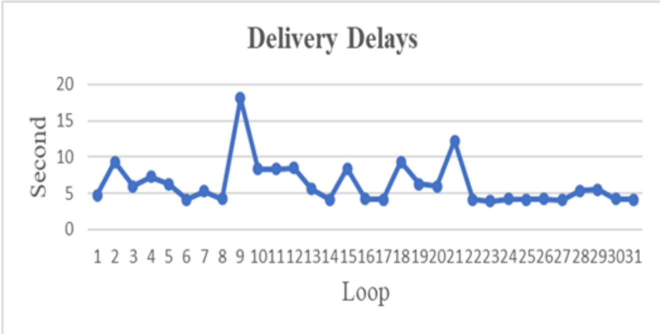

Figure 12. Delay in transmission data from 31 programme loop runs

The average delay in sending data from the NODE MCU to Firebase is 6.283 seconds. This delay occurs because the programming code on the NODE MCU includes a predefined time interval for sensor data acquisition, data transmission to Firebase, and reading the Realtime Database to determine whether the incubation mode is activated. These sequential processes contribute to the overall transmission delay observed in the system. However, as shown in Figure 12, there are instances where the transmission delay exceeds 10 seconds. This condition occurs when the NODE MCU experiences a reconnection process to the Firebase Realtime Database, which temporarily interrupts data transmission and increases the overall delay time.

E. Delay In Receiving Data

To get the receiving delay data, it is done by measuring the data transmission time displayed on the serial monitor and the reception time in the application displayed on Logcat Android Studio, as shown in Figure 13.

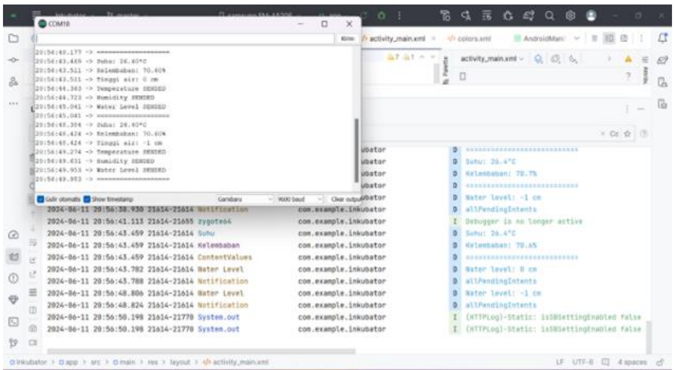

Figure 13. Timestamp in serial monitor and logcat

Based on the calculation of the data reception delay, the results obtained are presented as shown in Figure 14. This figure illustrates the variation in delay values during data transmission, which reflects the system's performance in receiving data under the tested conditions.

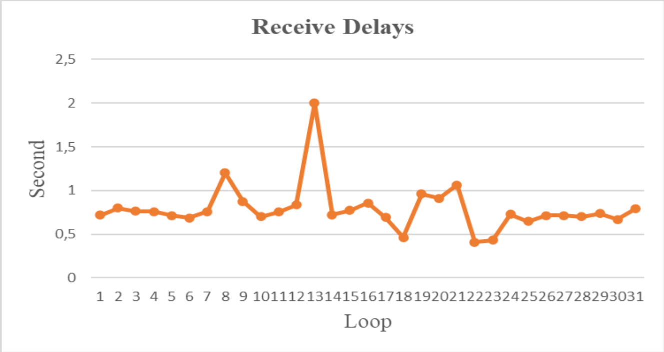

Figure 14. Received delay from serial monitor time calculation with logcat

The receiving delay data is obtained from reading the Firebase Realtime Database on the Android application that has been created. The average reception delay value obtained is 622 milliseconds. In Figure 14, there is a delay time that reaches 2 seconds. This can occur due to the influence of the Node MCU, which reconnects to Firebase, so that the data sent is delayed. In addition, when data has the same value as the previous data when received, Logcat does not display the reception time, which affects the calculation of the reception delay.

F. Throughput Test Result

Throughput data was collected by continuously pinging the Firebase server using the Command Prompt (cmd) for a duration of 30 minutes to observe the data transmission performance of the system. During this process, network traffic was captured and analysed using Wireshark software by applying filters based on the network source being used. From the captured traffic, the packet transmission results could be identified and measured to determine the system's throughput performance. The results of the packet traffic used for throughput testing are presented in Figure 15.

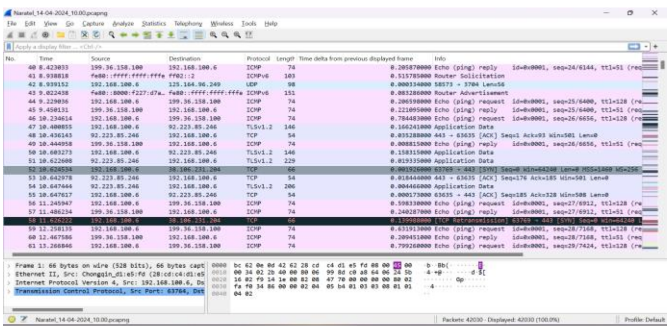

Figure 15. Packet traffic data

After the data packet is obtained, the Wireshark capture process is stopped and further analysis is carried out by applying a filter on the destination IP address. This filtering step is important to focus only on packets sent to the intended server and to eliminate unrelated network traffic, allowing more accurate throughput analysis. The filtering results are illustrated in Figure 16.

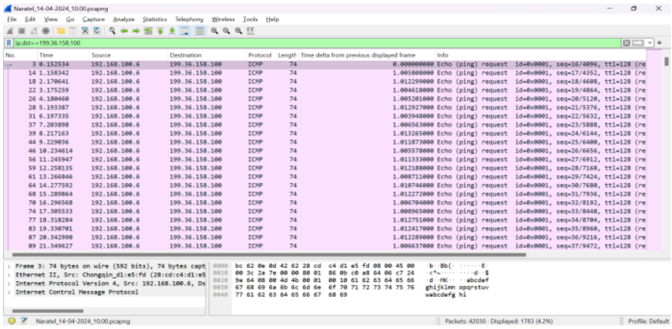

Figure 16. Filtered data packets

After filtering, select the Capture menu File Properties menu, as shown in Figure 17.

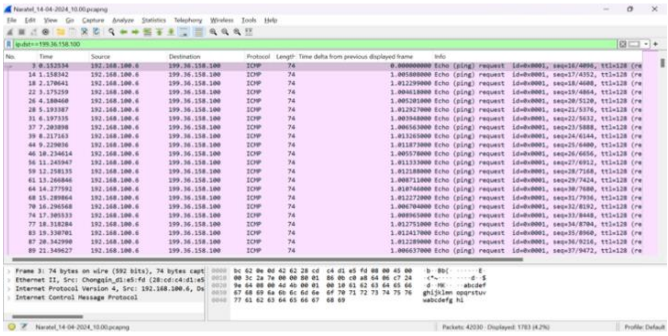

Figure 17. Statistics menu and capture file properties

After clicking the Capture menu File Properties menu, a new window will appear containing the details of the network used, as shown in Figure 18.

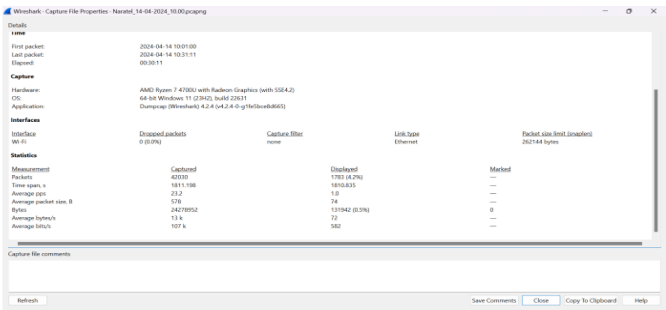

Figure 18. Detail window

From Figure 18, we can calculate the throughput. The Equation (1) used to calculate throughput value.

$$Throughput = \frac{Bytes}{Span\ Time} \quad (1)$$

Based on Figure 18 in the statistics section, it is known that the packet received is named Bytes by 131868 Bytes. As for the length of observation time, it can be seen in the Time Span section. The observation time displayed in Figure 18 is

1814,707 Byte. From the results of Figure 18, throughput can be calculated as follows:

$$Throughput = \frac{131868 \text{ Bytes}}{1814{,}707\ s}$$

$$= 72{,}666 \text{ Byte/s}$$

$$= 72{,}666 \text{ Bytes} \times 8$$

$$= \ 581{,}328 \text{ bit/s}$$

Tables 2 and 3 show the results of the throughput data that has been collected. The data taken comes from two different internet providers.

TABLE II
THROUGHPUT DATA FROM NARATEL PROVIDER

Date	Naratel		
	05.00	10.00	20.00
14/04/2024	581,330	582,900	583,771
15/04/2024	582,355	581,670	583,382
16/04/2024	582,257	582,392	581,298
17/04/2024	581,574	582,415	581,226
18/04/2024	582,276	583,529	581,182
19/04/2024	580,810	582,168	536,042
20/04/2024	570,846	581,060	581,168
Average	580,207	582,305	575,438

TABLE III
THROUGHPUT DATA FROM TELKOMSEL PROVIDER

Date	Telkomsel		
	05.45	10.45	20.45
14/04/2024	584,020	585,326	584,362
15/04/2024	584,092	585,113	581,346
16/04/2024	583,156	584,405	582,254
17/04/2024	583,973	583,147	580,280
18/04/2024	584,449	583,458	583,584
19/04/2024	584,673	583,020	583,957
20/04/2024	584,608	584,921	583,239
Average	584,139	584,199	582,717

The average throughput value obtained for provider Naratel is 579,317 bit/s, while for provider Telkomsel it is 583,685 bit/s. These values indicate that both providers deliver relatively similar throughput performance during the testing process. Based on the throughput index classification, the throughput performance of both providers falls into the bad category, indicating that the data transmission rate is still below the optimal standard for stable and efficient network communication.

G. Delay Test Result

Delay data collection is performed using Wireshark software by filtering the destination IP by writing ip.dst==199.36.158.100, which is the IP of Firebase as shown in Figure 19.

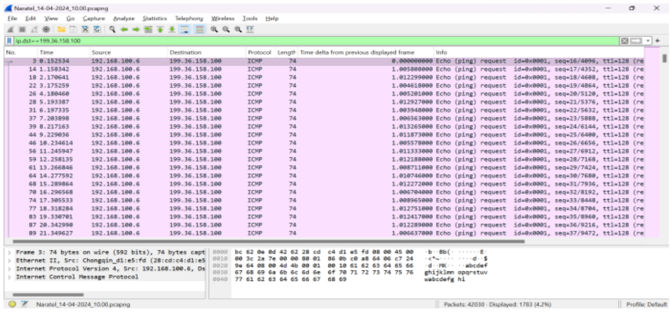

Figure 19. Filtered packets were used to calculate the average delay

After filtering, the parameter Time delta from the previous displayed frame is selected and applied as a column to display the time difference between sent and received packets, as illustrated in Figure 20. This step allows for precise observation of packet transmission intervals and delay variations. Subsequently, the filtered data is exported in CSV format, as shown in Figure 21, to facilitate further calculation and analysis of the delay values using spreadsheet or statistical tools.

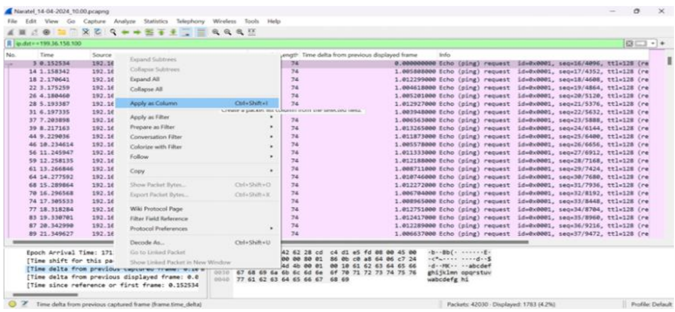

Figure 20. Time delta from previous displayed to indicate the time of sending and receiving the packet

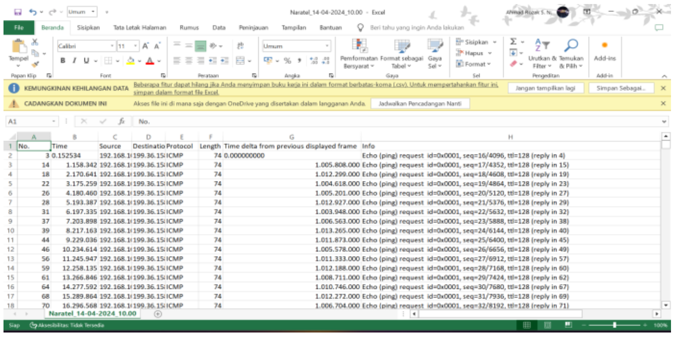

Figure 21. Exported data in CSV format

To determine the total number of packets transmitted during the testing process, the statistics menu in Wireshark is accessed and the Capture File Properties option is selected. This action opens a new window that displays comprehensive information related to the capture session, including the total packet count, capture duration, and data size. The detailed network information obtained from this window is used as a reference for further analysis of network performance, as shown in Figure 22.

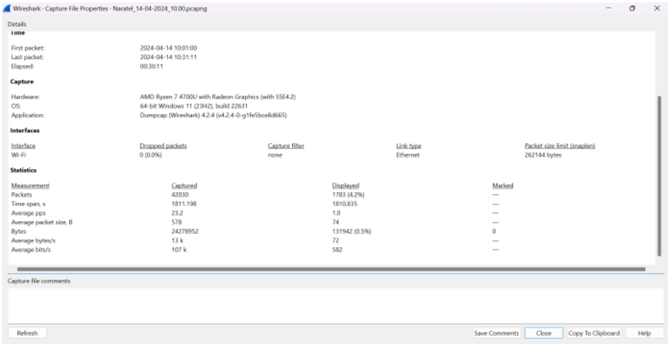

Figure 22. Detail window that has been filtered to calculate average delay

From Figure 22, it can be seen that the total delay shown in the Time Span, s Displayed section is 1810,835 and the number of packets shown in Bytes Displayed is 131942 (0.5%). Then, to calculate the average delay, use the following formula in Equation (2).

$$Average\ Delay = \frac{Total\ delay}{Total\ packet\ received} \quad (2)$$

$$= \frac{1810,835}{1783}$$

$$= 1,0156113292\ s = 1015,6113292\ ms$$

Tables 4 and 5 show the results of the average delay data that has been collected. The data taken comes from two different internet providers.

TABLE IV
AVERAGE DELAY DATA FROM NARATEL PROVIDER

Date	Naratel		
	05.00	**10.00**	**20.00**
14/04/2024	1,018354	1,015611	1,014097
15/04/2024	1,016563	1,017759	1,014773
16/04/2024	1,016734	1,016497	1,018411
17/04/2024	1,017927	1,016457	1,018538
18/04/2024	1,016700	1,014517	1,018613
19/04/2024	1,019267	1,016889	1,104391
20/04/2024	1,037058	1,018827	1,018638
Average	1,020372	1,016651	1,029637

TABLE V
AVERAGE DELAY DATA FROM TELKOMSEL PROVIDER

Date	Telkomsel		
	05.45	**10.45**	**20.45**
14/04/2024	1,013664	1,011401	1,013071
15/04/2024	1,013539	1,011771	1,018326
16/04/2024	1,015166	1,012997	1,016738
17/04/2024	1,013746	1,015181	1,020197
18/04/2024	1,012919	1,014640	1,014421
19/04/2024	1,012531	1,015403	1,013774
20/04/2024	1,012644	1,012102	1,015021
Average	1,012644	1,012102	1,015021

Based on Tables 4 and 5, it is obtained that the average delay of the Naratel provider is 1022,220064 ms, while for the Telkomsel provider it is 1013,255692 ms. Based on the index, the average delay of the two providers is included in the excellent category because the average delay of the two providers is >450 ms.

H. Packet Loss

Packet loss data collection is done by looking at the "Packet Drops" information in the Wireshark detail window, as show in Figure 23. By looking at this caption, it is enough to know the packets lost during data packet transmission.

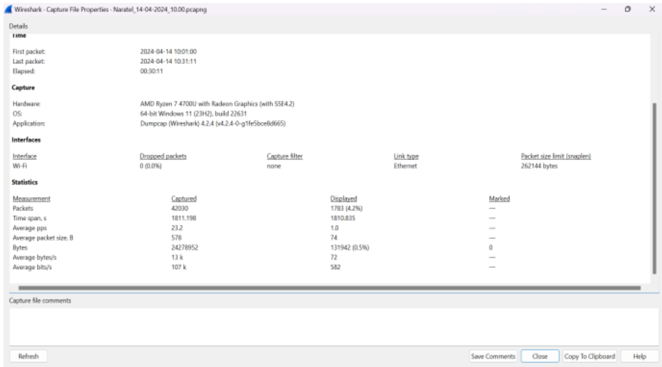

Figure 23. Dropped packets in the detail window to find out packet loss

The formula for calculating packet loss is as follows in Equation (3):

$$Packet\ loss = \frac{Packet\ send - Packet\ Receive}{Packet\ send} \times 100\% \quad (3)$$

$$= \frac{1778-1778}{1788} \times 100\% = 0,0\ \%$$

TABLE VI
AVERAGE PACKET LOSS DATA FROM NARATEL PROVIDER

Date	Naratel		
	05.00	**10.00**	**20.00**
14/04/2024	0,0	0,0	0,0
15/04/2024	0,0	0,0	0,0
16/04/2024	0,0	0,0	0,0
17/04/2024	0,0	0,0	0,0
18/04/2024	0,0	0,0	0,0
19/04/2024	0,0	0,0	0,0
20/04/2024	0,0	0,0	0,0
Average	0,0	0,0	0,0

TABLE VII
AVERAGE PACKET LOSS DATA FROM TELKOMSEL PROVIDER

Date	Telkomsel		
	05.45	**10.45**	**20.45**
14/04/2024	0,0	0,0	0,0
15/04/2024	0,0	0,0	0,0
16/04/2024	0,0	0,0	0,0
17/04/2024	0,0	0,0	0,0
18/04/2024	0,0	0,0	0,0
19/04/2024	0,0	0,0	0,0
20/04/2024	0,0	0,0	0,0
Average	0,0	0,0	0,0

Based on Tables 6 and 7, the average packet loss on Naratel and Telkomsel providers is 0,0%. Based on the packet loss index for Naratel and Telkomsel providers, it is included in the very good category because there is no packet loss at all.

I. Jitter Test Result

To get jitter data, after exporting the data in CSV format, add delay variations based on time delta by subtracting delay time 1 from delay time 2, as can be seen in Figure 24.

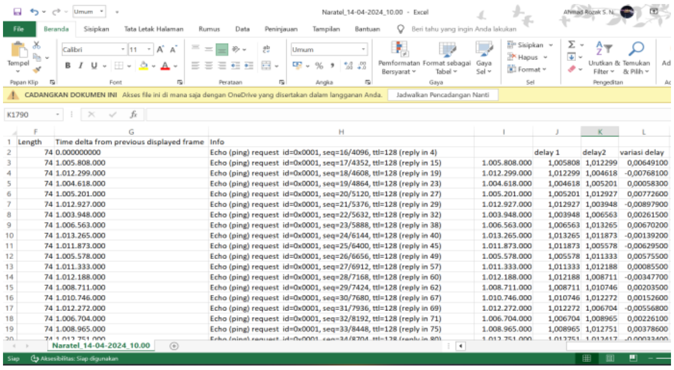

Figure 1. Data that has been added with a delay variation

After the delay variations are obtained, the next step is to calculate the total delay by summing all of the delay variation values. Before performing this calculation, any negative values are converted into positive values using the ABS command in Microsoft Excel to ensure accurate results. Once all values have been adjusted, the entire set of delay variations is added together to obtain the total delay value used for further analysis. The formula for calculating average jitter is as follows in Equation (4):

$$Average\ Jitter = \frac{Total\ Jitter}{Packet} \quad (4)$$

$$= \frac{18{,}976826}{1775} = 0{,}01069117\ ms$$

Tables 8 and 9 show the results of the average jitter data that has been collected. The data taken comes from two different internet providers.

TABLE VIII
AVERAGE JITTER DATA FROM NARATEL PROVIDER

Date	Naratel		
	05.00	**10.00**	**20.00**
14/04/2024	13,181	9,382	7,255
15/04/2024	10,691	11,581	8,948
16/04/2024	10,086	10,090	13,522
17/04/2024	11,041	9,697	14,697
18/04/2024	9,800	7,742	13,920
19/04/2024	14,196	9,817	169,387
20/04/2024	49,062	14,229	13,696
Average	16,865	10,363	34,489

TABLE IX
AVERAGE JITTER DATA FROM TELKOMSEL PROVIDER

Date	Telkomsel		
	05.45	**10.45**	**20.45**
14/04/2024	6,302	4,820	7,020
15/04/2024	7,825	4,320	17,054
16/04/2024	9,652	5,828	12,173
17/04/2024	7,391	9,535	11,989
18/04/2024	6,061	8,685	7,592
19/04/2024	4,861	10,286	6,704
20/04/2024	4,843	4,250	8,999
Average	6,705	6,818	10,219

Based on Tables 8 and 9, the average value of jitter for Naratel and Telkomsel providers obtained is included in the good category because the jitter value is between 1 and 75 ms. The average value of Telkomsel jitter is greater than Naratel jitter because the average jitter for Naratel at three times is 20,572 ms, while Telkomsel is 7,914 ms.

IV. CONCLUSION

Based on the description that has been presented, it can be concluded as follows: Based on the results of the Black Box Testing Evaluation that has been done, the Reptile Incubator System as an Android-Based Reptile Incubator System as a Sex Determinant at Inferno Reptile has a system that runs 100% for all functions and purposes. system that runs 100% for all functions and purposes. The incubator has the ability to stabilise the temperature in accordance with the selected settings because the temperature does not experience significant changes from the ideal temperature range for each sex. from the ideal temperature range for each species in each type of reptile. Data on temperature, humidity, and water level are also successfully sent to Firebase and can be displayed on the Android application. In addition, the incubator can also run temperature settings based on data sent by the application to Firebase. based on the data sent by the application to Firebase. In the Quality of Service test the throughput obtained is 579.317bit/s for the Naratel provider and 583.685bit/s for the Telkomsel provider. The Telkomsel provider falls into the 'Bad' category because the throughput for both providers is less than 25 kb/s. The value is less than 25 kb/s, and the index is 1. Delay for provider Naratel obtained a value of 1022.220064 ms, and for Telkomsel, 1013.255692 ms is included in the 'Bad' category because the value is more than 25 kb/s and the index is 1. 'Bad' because the value is more than 450 ms and the index value is 1. In packet loss, a value of 0.0% is found for providers. Naratel and Telkomsel are included in the 'Very Good' category because the value is 0% and the index is 4. On average, Jitter obtained a value of 20.572 ms from the provider. Naratel and 7.914 ms for Telkomsel providers are included in the 'Good' category because the value is in the range of 1 to 75 ms, and the index is 3.

REFERENCES

[1] M. Mamalia and A. Reptil, “46 Ug Jurnal Vol.13 No.12,” vol. 13, no. 12, pp. 46–57, 2022.

[2] Y. R. D. Wahana, “SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN MORPH LEOPARD GECKO UNTUK PEMULA MENGGUNAKAN METODE SIMPLE ADDITIVE WEIGHTING,” Politeknik Negeri Jember, 2021.

[3] R. M. Septian Rizki Agrianto,Purba Daru Kusuma, “SIMULASI DAN ANALISIS KINERJA QOS (QUALITY OF SERVICE) JARINGAN BERBASIS SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP),” vol. 5, no. 1, pp. 131–144, 2019.

[4] F. Di Ianni et al., “Demonstration of Parthenogenetic Reproduction in a Pet Ball Python (Python regius) through Analysis of Early-Stage Embryos,” 2023. doi: 10.3390/genes14091744.

[5] D. Recardsari, “Inilah Fakta Menarik Tentang Bearded Dragon, Reptil Eksotis Yang Bisa Ganti Kelamin!,” 2021. Accessed: Jun. 05, 2024. [Online]. Available: https://www.kawanhewan.com/post/bearded-dragon-reptil-eksotis

[6] M. Caesar, A. Raya, and B. Saputra, “Rancang Bangun Inkubator Anakan Burung Parkit Free Flight,” J. Elektro dan Teknol. Inf., vol. 1, no. 2, pp. 38–46, 2022.

[7] M. Y. Kusuma, Sugiono, and M. J. Afroni, “RANCANG BANGUN IoT UNTUK SISTEM INKUBATOR TELUR PENYU,” J. Ilm. Mhs., vol. 16, no. 3, pp. 1–4, 2023.

[8] V. Febrian, M. R. Ramadhan, M. Faisal, and A. Saifudin, “Pengujian pada Aplikasi Penggajian Pegawai dengan menggunakan Metode Blackbox,” J. Inform. Univ. Pamulang, vol. 5, no. 1, p. 61, 2020, doi: 10.32493/informatika.v5i1.4340.

[9] A. Miranto and E. Reynaldi, “Perancangan Dan Implementasi Antarmuka Pengguna Sistem Pemantauan Kualitas Udara Berbasis Aplikasi Android,” Cybersp. J. Pendidik. Teknol. Inf., vol. 7, no. 1, p. 46, 2023, doi: 10.22373/cj.v7i1.17491.

[10] R. M. C. M. Septrio Rafinaldo, Iwan Iskandar, Nazruddin safaat harahap, “Analisis Kualitas Jaringan Internet pada SMK Menggunakan Metode Quality of Service,” KLIK Kaji. Ilm. Inform. dan Komput. ISSN 2723-3898 (Media Online) Vol 3, No 6, Juni 2023, Hal 977-984 DOI 10.30865/klik.v3i6.903 https//djournals.com/klik, vol. 3, no. 6, p. 977, 2023, doi: 10.30865/klik.v3i6.903.

[11] B. Pamungkas and F. A. Sutanto, “Penerapan Metode Quality of Service (QoS) dalam Analisis Kualitas Jaringan Internet Burjo Pantry Semarang,” J. JTIK (Jurnal Teknol. Inf. dan Komunikasi), vol. 8, no. 2, pp. 367–373, 2024, doi: 10.35870/jtik.v8i2.1932.

[12] A. Mikola and M. Sari, “Analisis Sistem Jaringan Berbasis QoS untuk Hot-Spot Di Institut Shanti Bhuana,” J. Inf. Technol., vol. 2, no. 1, pp. 31–35, 2022, doi: 10.46229/jifotech.v2i1.398.

[13] I. P. Sari, “Evaluasi Kualitas Jaringan Internet Pemerintah Daerah Kota Padang Panjang Menggunakan Metode Quality of Service,” J. Sistim Inf. dan Teknol., vol. 4, pp. 25–29, 2022, doi: 10.37034/jsisfotek.v4i1.116.

[14] M. Ryan Kamil, F. Arzalega, and A. Sani, “JBPI-Jurnal Bidang Penelitian Informatika Ciptaan disebarluaskan di bawah Lisensi Creative Commons Atribusi 4.0 Internasional Analisis Kualitas Layanan Jaringan Internet Wifi PT.XYZ dengan Metode QoS (Quality of Service),” pp. 77–88, 2023, [Online]. Available: https://ejournal.kreatifcemerlang.id/index.php/jbpi

[15] A. Dyan Ramadhan and I. Iskandar, “KLIK: Kajian Ilmiah Informatika dan Komputer Evaluasi Peforma Jaringan Internet Menggunakan Metode QoS,” Media Online), vol. 3, no. 6, pp. 996–1004, 2023, doi: 10.30865/klik.v3i6.892.