DESIGNING IOT SYSTEM USING REAL TIME DATABASE FIREBASE FOR SMART DRIP IRRIGATION URBAN FARMING APPLICATION

David Firman Manggasa'[1], Erna Sri Sugesti[2], Ing Paulus Suwanto[3]
[1]international Telecommunication Engineering, School of Electrical Engineering , Telkom University Bandung, West Java, Indonesia
[1]**firmanmanggasa@student.telkomuniversity.ac.id**, [2]**ernasugesti@telkomuniveristy.ac.id**, [3]**suwantodr@gmail.com**

Abstract- The traditional manual method of watering plants in *Urban Farming* is being replaced with *Drip Irrigation* systems, which use low water intensity. However, these systems are not always easy to control. To address this issue, an *Internet of Things (IoT)*-based drip irrigation control system was designed, which uses the *Firebase RTDB* as the application development database and the *Arduino Mega 2560* as the microcontroller. The system monitors plant conditions and controls water distribution via the internet with smartphones as controllers. The *Arduino Mega 2560* receives data from soil humidity, pH and temperature sensors and sends it to the smartphone via the Wi-Fi module ESP8266. The smartphone can then control the opening or closing of the drip irrigation based on the data received. The system was tested and performed well, with automatic watering according to time and successful reading and sending of pH and humidity values. The error rate value between the temperature sensor and room temperature was 2.22%, and the Quality of Service (QoS) delay and throughput values were also good. Overall, the system provides a reliable and efficient way to control drip irrigation in urban farming.

Keywords: *Drip Irrigation, Internet of Things (IoT), Urban Farming, RTDB Firebase, Arduino Mega 2560.*

I INTRODUCTION

Drip irrigation is an efficient method for providing water to plants, particularly in areas with low rainfall or when regular humidity is required. However, traditional irrigation methods are often ineffective and wasteful. To address this, an Internet of Things (IoT) based drip irrigation control system has been developed, which can be controlled through a smartphone and is suitable for use in urban farming. The system allows for precise control of water distribution and can also mechanically treat plants with organic fertilizer. This innovative system offers advantages over traditional methods, particularly in terms of water conservation and ease of use.

Conventional irrigation has problems such as excessive water use and inability to control remotely. To address this, an IoT-based drip irrigation system was designed to monitor and control temperature, humidity, and pH remotely. The system's accuracy and measuring instrument quality were also evaluated. With the implementation of this research, the aim is to obtain and prototype an IoT-based drip irrigation control and monitoring system using RTDB Firebase as a database that is accessed using a Smartphone.

Based on the description of the problem formulation, there are several limitations of the problem that focus on several things in this final project.

a. The system built only focuses on building an IoT system and connecting it to mechanical/hardware devices.
b. The Smartphone application used is limited only to Android users.
c. In temperature data collection there is no logger data, data collection is discrete and only approaches.

The QoS value on the network is observed only on Throughput and Delay.

Research Method

To get a good tool in terms of quality by considering the QoS value and Error Value, the design steps are described in Figure 1.1 and the explanation is as follows below.

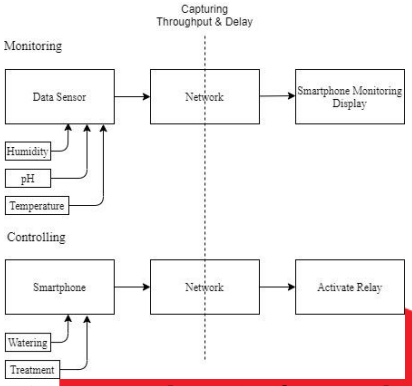

Figure 1.1. Schematic of Research Method

a. To Determine, The design includes system software, smartphone applications and hardware in the form of drip irrigation mechanics and the controller.
b. Data Collection, At this stage, the process of collecting information data in the form of performance, throughput and delay is carried out and will be analyzed later.
c. Performance and QoS Analysis, At this stage analysis and comparison are carried out based on the parameters of QoS Throughput, Delay and Performance control/monitoring from the results of data collection.

II BASIC THEORY

2.1. Drip Irrigation

Plants have different water requirements according to environmental conditions and the time of their growth. Therefore, precision irrigation is needed that can drain water according to environmental conditions and plant needs[2]. Drip irrigation is an example of precision irrigation in urban farming, which is a water irrigation application system by flowing water to an emitter drip point that is located above or below the ground surface with a small operating pressure and low discharge so that water only wets part of the soil surface[1].

2.4. RTDB Firebase

Google Firebase is a real time database service provider currently owned by Google. Firebase is a solution offered by Google to simplify the work of Mobile Apps Developers. With Firebase, application developers can focus on developing applications without having to put a lot of effort into backend matters. One of the features owned by Google Firebase is the Firebase Real Time Database which uses the Java Script Object Notation or JSON format which is the format used to store and transfer data on the Firebase RTDB [7].

2.5. NodeMCU ESP8266

NodeMCU is a kit that functions to assist in designing tools based on the Internet of Things (IoT). NodeMCU was developed to make it easier to operate API (Application Programming Interface) for IO hardware. NodeMCU is designed like an Arduino Microcontroller which has other hardware Input Outputs. NodeMCU has also been embedded with an ESP8266 WIFI Module.

2.6. Arduino Mega2560

Arduino Mega 2560 microcontroller is a microcontroller on ATMEGA 2560 which has 54 digital input/output pins of which 16 pins are used as PWM outputs, 16 analog inputs, and inside there are 16 MHZ crystal oscillators, USB connections, power, ICSP, and reset buttons. [6]. This arduino performance can be programmed with Arduino IDE software or Visual Studio Code which has an IO IDE Platform by connecting it to a computer with a USB cable. To turn it on using AC or DC current and can also use a battery.

2.7. DHT22 Sensor

DHT22 is a complex temperature and humidity sensor with digital signal output calibration with a temperature measuring range of 0-50ºC and a relative humidity measuring range of 20%-90%, its small size so it only requires 5v of power. DHT22 is claimed to have good reading quality, judging by the fast response of the data acquisition process and its minimalist size, as well as the relatively low price of the thermohygrometer[17].

In the DHT22 Implementation, the room temperature read is not the same as the temperature read by the room temperature meter. The occurrence of this difference can be influenced by the Sensing Element factor that is not good or of a different type. For DHT22 use Polymer Capacitor as Sensing Element [17]. To calculate the accuracy can be used the formula:

$$e_H = \frac{|H_S - H_H|}{H_H} x100\% \qquad (2.1)$$

$$\chi_{e_H} = \frac{\sum e_h}{n} \qquad (2.2)$$

Description :

e_H = Measurement Error value of room temperature.

H_S = Measurement of Room

temperature using a DHT22 sensor.

H_H = Measurement of Temperature using Room Temperature Meter.

χ_{e_H} = Average value error of measurement room temperature.

n = Number of Trials.

2.8. Quality of Service (QoS)

2.8.1. Delay

Delay is the time it takes for data to start when it is sent until the data reaches its destination. In this research, the delay is calculated from 2 observations, namely when the system is controlling from Smartphone to Hardware and monitoring sensor data until when the data is displayed on the Smartphone [15].

$$D = \frac{\sum \Delta_t}{\sum P} \quad (2.3)$$

Description :

D = Delay value.

$\sum \Delta_t$ = The amount of difference in the delivery time of each package.

$\sum P$ = The number of all successfully received packetss.

Tabel 2.1. Delay Category based on iTU-T

Category Delay	Delay Quantity (ms)
Good	<150 ms
Enough	150 ms s/d 400 ms
Bad	> 400 ms

2.8.2. Throughput

Throughput is the amount of data that can be sent over a communication network. Throughput can also be interpreted as the total number of observed packet arrivals [14].

$$T = \left(\frac{\frac{\sum b}{t_S}}{1000} \right) \times 8 \quad (2.4)$$

Description :

T = Throughput value in bps unit.

$\sum b$ = Total Number of Bytes successfully received.

t_S = Time interval for observing packet delivery.

By using Wireshark Software to observe QoS on the network, the results of the Throughput value will be directly displayed in either bytes/s or bits/s.

2.9. Performance Parameter

Performance is the ability of the system to be able to fulfill a given function with the aim of knowing how capable the system can be relied on in carrying out its functions. In this research, the performance that will be tested is in the form of the system's ability to read sensor values and display them on Smartphone monitoring, system performance when the user gives orders to control the watering point and the performance of reading room temperature values which will be compared with the original temperature [16].

III EXPERIMENT DESIGN

3.1. General Description of the System

IoT-based drip irrigation works similarly to other IoT plant irrigation, but with less water usage and data sourced from soil pH and humidity sensors. The data is sent to the Cloud for processing and to schedule watering times for dry soil at 08.00 and 16.00. The Cloud data is displayed on a mobile app for monitoring and manual watering is also possible through the app.

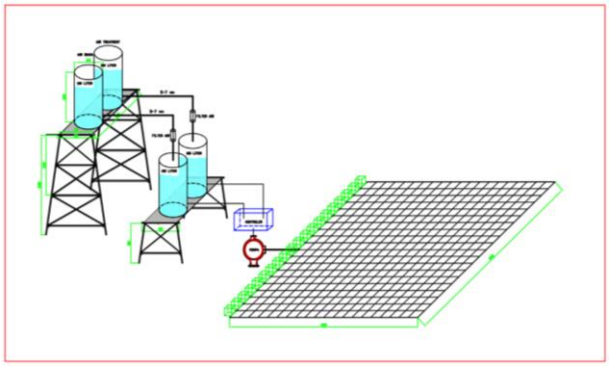

Figure 3.1. Drip Irrigation model realization

3.2. Working Principle

The Arduino Mega2560 microcontroller regulates values from 20 Humidity sensors and 12 pH sensors, which requires additional pins using the ADC ADS1115 device. The sensor values are used to control the Pump and Solenoid via Relays. The values are also displayed wirelessly on a Smartphone app.

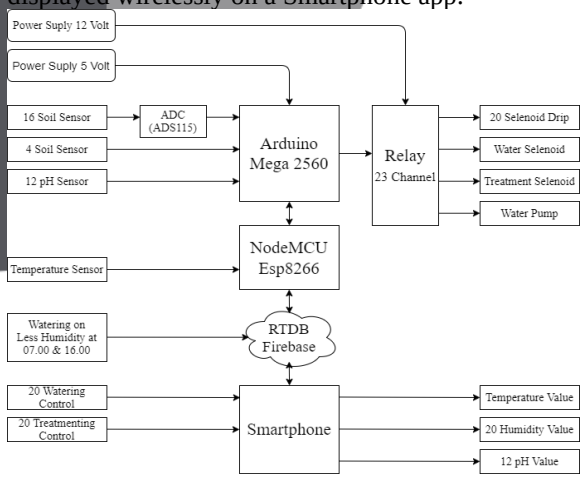

Figure 3.2. Block Diagram of IoT based Drip Irrigation

NodeMCU is a microcontroller that utilizes the ESP8266 chipset with built-in Wi-Fi capabilities, allowing easy connection to a Wi-Fi network and TCP/IP connections through simple commands.

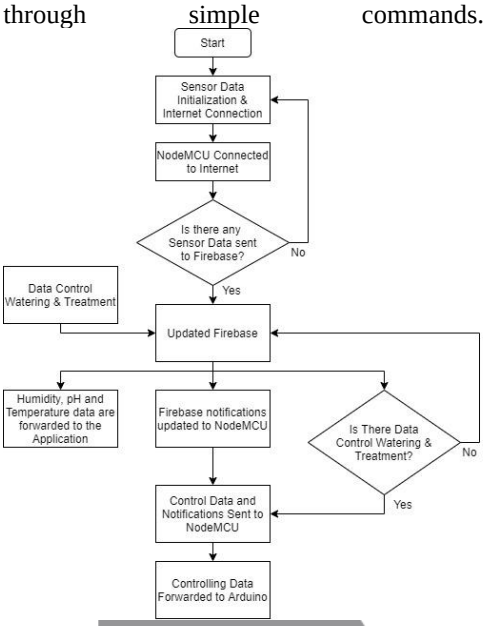

Figure 3.3. Firebase and NodeMCU Connection Flowchart

NodeMCU adds a Room Temperature Sensor input to the system, which is sent to the Firebase RTDB Cloud along with Humidity and pH sensor values. NodeMCU is also connected to a Smartphone app which sends commands to the Tool via the Cloud and the Arduino Mega2560 activates Relays accordingly. The Firebase RTDB Cloud is used to schedule watering and treatment and serves as an important database between Smartphones and Tools.

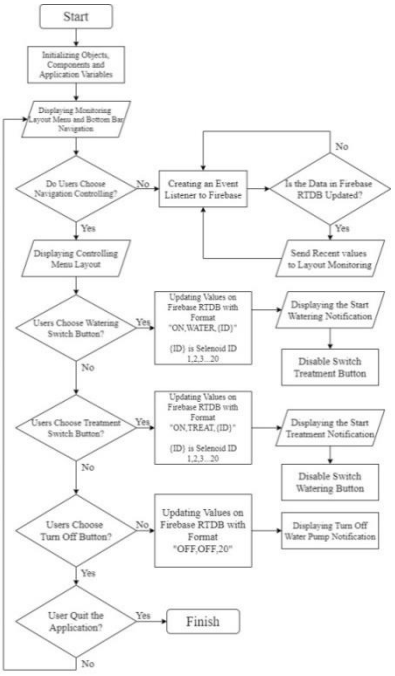

Figure 3.4. Firebase and Application Connection Flowchart

Smartphones, especially in this study using the Android type, can carry out several orders, such as watering or treatment at certain drip irrigation points. In addition, the Smartphone will display the data sensor that have been sent by ESP8266 and received on Firebase which are then processed so that they can be displayed in detail on the Smartphone.

3.3. NodeMCU Esp8266 Configuration

NodeMCU Esp8266 on this tool has a function as a sender and receiver of data with firebase, besides that NodeMCU will be connected to Arduino as the main microcontroller. Therefore, it is necessary to configure the system so that it can run.

3.3.1. EPS8266 NodeMCU Configuration Displays DHT22 Sensor Value

3.3.2. Configure NodeMCU connection with Arduino

3.3.3. Configure NodeMCU connection with Firebase

3.4. Firebase and Android Apps Configuration

Here are the steps in creating Google Firebase as a database or cloud and connecting it to Android Application and Tools:

a. Real Time Database Creation on Firebase Cloud

b. Development of a Drip Irrigation Mobile Application

IV RESULTS AND ANALYSIS

Figure 4.1. Application of Drip Irrigation on Plants

Figure 4.2. Primary Selenoid in each Gutter

Figure 4.3. Pump and Water Tank Implementation

Figure 4.4. Housing

4.1. Implementation of IoT-based drip Irrigation System

In this test, the observed output is the success of the tool in carrying out each task or its implementation. The Table 4.1 below shows the test results for each. From the results of the IoT-based Drip Irrigation implementation in the table above, the system can work according to its function and be integrated with each other starting from the Application, RTDB Firebase as Cloud and Hardware.

Table 4.1. List of System Implementation Test Results

No.	Implementation Results	Information
1.	The Analog Humidity Sensor value is read and classified into Low, High and Normal on the NodeMCU Esp8266	Succeed
2.	Analog values for pH and temperature sensors are read on the NodeMCU Esp8266	Succeed
3.	The data on the NodeMCU Esp8266 is sent to the Database and reads with the same value.	Succeed
4.	The data on the Firebase RTDB is read by the Android Application and displayed on the monitoring layout.	Succeed
5.	The Watering Control of the app activates the relay that sets the watering point	Succeed
6.	Control Treatment from the application activates the relay that sets the treatment point	Succeed
7.	Automatic Watering at 08.00 and 16.00	Succeed

4.2. Performance Testing

4.2.1. Control Watering From Smartphone

This test is carried out to see the performance of the Application and Hardware whether the Watering Switch Control on the Application has triggered the relay on the Tool which regulates each watering point along with the pump and ensures that the commands in the database are appropriate. Based on the results of the Watering Control observations in Table 4.2 below it can be concluded that the watering control of each Planting Medium on the application has been connected according to what is triggered on the hardware and application. Hardware in doing its job requires delay in processing the given command. When executing a watering command, the system will activate the pump, solenoid water and activated primary solenoid.

Table 4.2. Watering Accuracy Test Results

Watering Switch ON	Firebase Comand	Relay Primer, Relay Water and Pump	Delay
Solenoid Primer 1	“ON,Water,1”	Active	2
Solenoid Primer 2	“ON,Water,2”	Active	2
Solenoid Primer 3	“ON,Water,3”	Active	3
Solenoid Primer 4	“ON,Water,4”	Active	2
Solenoid Primer 5	“ON,Water,5”	Active	4
Solenoid Primer 6	“ON,Water,6”	Active	3
Solenoid Primer 7	“ON,Water,7”	Active	3
Solenoid Primer 8	“ON,Water,8”	Active	3
Solenoid Primer 9	“ON,Water,9”	Active	4
Solenoid Primer 10	“ON,Water,10”	Active	5
Solenoid Primer 11	“ON,Water,11”	Active	3
Solenoid Primer 12	“ON,Water,12”	Active	2
Solenoid Primer 13	“ON,Water,13”	Active	2
Solenoid Primer 14	“ON,Water,14”	Active	2
Solenoid Primer 15	“ON,Water,15”	Active	5
Solenoid Primer 16	“ON,Water,16”	Active	6
Solenoid Primer 17	“ON,Water,17”	Active	2
Solenoid Primer 18	“ON,Water,18”	Active	3
Solenoid Primer 19	“ON,Water,19”	Active	3
Solenoid Primer 20	“ON,Water,20”	Active	4

4.2.2. Control Treatment From Smartphone

This test measures the delay performance of the Application and Hardware in triggering the relay for treatment control. The delay is manually measured using a timer from the command given via Smartphone until the relay gives an active notification.

Table 4.3. Treatmenting Accuracy Test Results

Treatment Switch ON	Firebase Comand	Relay Primer, Relay Water and Pump	Delay
Solenoid Primer 1	“ON,Treat,1”	Active	2
Solenoid Primer 2	“ON,Treat,2”	Active	3
Solenoid Primer 3	“ON,Treat,3”	Active	5
Solenoid Primer 4	“ON,Treat,4”	Active	3
Solenoid Primer 5	“ON,Treat,5”	Active	4
Solenoid Primer 6	“ON,Treat,6”	Active	2
Solenoid Primer 7	“ON,Treat,7”	Active	2
Solenoid Primer 8	“ON,Treat,8”	Active	2
Solenoid Primer 9	“ON,Treat,9”	Active	2
Solenoid Primer 10	“ON,Treat,10”	Active	3
Solenoid Primer 11	“ON,Treat,11”	Active	4
Solenoid Primer 12	“ON,Treat,12”	Active	2
Solenoid Primer 13	“ON,Treat,13”	Active	7
Solenoid Primer 14	“ON,Treat,14”	Active	3
Solenoid Primer 15	“ON,Treat,15”	Active	4
Solenoid Primer 16	“ON,Treat,16”	Active	3
Solenoid Primer 17	“ON,Treat,17”	Active	2
Solenoid Primer 18	“ON,Treat,18”	Active	2
Solenoid Primer 19	“ON,Treat,19”	Active	4
Solenoid Primer 20	“ON,Treat,20”	Active	3

Based on the results of the Treatment Control observations in Table 4.3 it can be concluded that the treatment control of each Planting Medium on the application has been connected according to what is triggered on the hardware and application. Hardware in doing its job requires delay in processing the given command. When executing a Treatment Command, the system activate the pump, treatment solenoid and activated primary solenoid.

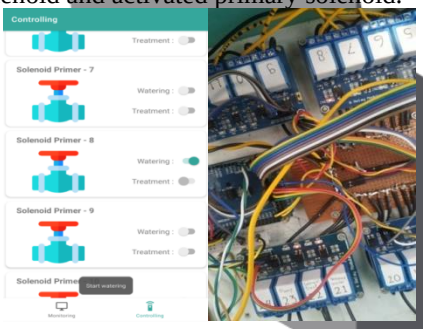

Figure 4.6. Example of watering control test results.

4.2.3. Automated Watering Test

The Automatic Watering test results showed that the system activated and read the TA relay alternately from 1 to 20, based on humidity sensor values declared "High" on Firebase which activated the Pump, Relay Water, and Relay Primer for 30 seconds at 08.00 am and 04.00 pm. Humidity sensor values declared "Low" and "Normal" were not activated and the system proceeded to the next TA.

Table 4.4. Automatic Watering

TA/Relay Primer	Sensor Firebase Data and Automatic Watering at 08.00		Duration	Sensor Firebase Data and Automatic Watering at 16.00		Duration
	Sensor Value	Relay Primer		Sensor Value	Relay Primer	
1	Low	Deactive	-	High	Active	30 s
2	High	Active	30 s	High	Active	30 s
3	Low	Deactive	-	Normal	Deactive	-
4	High	Active	30 s	Low	Deactive	-
5	Normal	Deactive	-	High	Active	30 s
6	High	Active	30 s	Low	Deactive	-
7	High	Active	30 s	High	Active	30 s
8	Low	Deactive	-	Low	Deactive	-
9	Low	Deactive	-	Low	Deactive	-
10	Low	Deactive	-	Low	Deactive	-
11	Low	Deactive	-	Low	Deactive	-
12	Low	Deactive	-	Low	Deactive	-
13	Low	Deactive	-	Low	Deactive	-
14	Low	Deactive	-	Low	Deactive	-
15	Low	Deactive	-	Low	Deactive	-
16	Low	Deactive	-	Low	Deactive	-
17	Low	Deactive	-	Low	Deactive	-
18	Low	Deactive	-	Low	Deactive	-
19	Low	Deactive	-	Low	Deactive	-
20	Low	Deactive	-	Low	Deactive	-

4.2.4. Testing the reading of Humidity value

This test is carried out by testing the performance of the Humidity Sensor installed at each point of the watering Planting Medium. The humidity sensor analog value (0-32767) for ADS1115 and (0-1024) for Arduino analog that is read converted into percent and then classified into 3 levels. Sensor values (0-30%) are classified as “Low”, (31-46%) are classified as “Normal”, (47-100%) are classified as “High”. Classification of humidity sensor values is done to simplify the process of sending data and Automatic Watering.

Tabel 4.5. Test results of Soil Moisture Value Reading

Planting Medium	Soil Moisture Test Results 1			Soil Moisture Test Results 2			Soil Moisture Test Results 3		
	Arduino	Node MCU	App Results	Arduino	Node MCU	App Results	Arduino	Node MCU	App Results
1	17952	H	High	19136	H	High	21752	H	High
2	8592	L	Low	14480	N	Normal	17387	H	High
3	8608	L	Low	12416	N	Normal	16395	H	High
4	15248	N	Normal	16480	H	High	18540	H	High
5	9344	L	Low	10160	N	Normal	13502	N	Normal
6	12688	N	Normal	16144	H	High	17932	H	High
7	11024	N	Normal	14400	N	Normal	16375	H	High
8	18096	H	High	20480	H	High	22487	H	High
9	14256	N	Normal	17408	H	High	20113	H	High
10	21664	H	High	23040	H	High	25098	H	High
11	11584	N	Normal	13712	N	Normal	16749	H	High
12	8224	L	Low	9216	L	Low	13754	N	Normal
13	11488	N	Normal	13248	N	Normal	14966	N	Normal
14	8816	L	Low	9824	L	Low	11369	N	Normal
15	2976	L	Low	3616	L	Low	6790	L	Low
16	9840	L	Low	11248	N	Normal	15780	N	High
17	20643	H	High	22487	H	High	26297	H	High
18	16543	H	High	18540	H	High	23094	H	High
19	18243	H	High	20136	H	High	24758	H	High
20	21597	H	High	24017	H	High	25938	H	High

From the results of reading the soil humidity value in the table, it can be seen that the classification of the configured values has been successful and the sending of soil humidity data on the tool has been successfully read in the application. Where the classification of sensor reading values for ADC1115 is 0-10156 (0-30%) reads "Low", 10158-15072 (31-46%) reads "Normal" and 15073-32767 (47-100%) reads "High". And for Analog Arduino, 0-316 (0-30%) reads "Low", 317-470 (31-46%) reads "Normal" and 471-1024 (47-100%) reads "High".

4.2.5. Testing the reading of pH value

This test is carried out by testing the performance of the pH sensor which is installed only at several points of the Planting Medium. Unlike the implementation of the Soil Moisture sensor, the analog value that is read and will be transferred to the NodeMCU is not classified because each pH analog value has its own influence for the user in determining treatment.

Tabel 4.6. Test results for reading pH 1.

Results of reading data on Planting Medium	pH Sensor in Planting Medium											
	1	2	3	4	5	6	7	8	9	10	11	12
Arduino	6	6	6	6	6	7	N/A	N/A	N/A	N/A	N/A	N/A
NodeMCU	6	6	6	6	6	7	N/A	N/A	N/A	N/A	N/A	N/A
Application	6	6	6	6	6	7	N/A	N/A	N/A	N/A	N/A	N/A

Tabel 4.7. Test results for reading pH 2.

Results of reading data on Planting Medium	pH Sensor in Planting Medium											
	1	2	3	4	5	6	7	8	9	10	11	12
Arduino	6	6	7	6	6	7	N/A	N/A	N/A	N/A	N/A	N/A
NodeMCU	6	6	7	6	6	7	N/A	N/A	N/A	N/A	N/A	N/A
Application	6	6	7	6	6	7	N/A	N/A	N/A	N/A	N/A	N/A

Tabel 4.8. Test results for reading pH 3.

Results of reading data on Planting Medium	pH Sensor in Planting Medium											
	1	2	3	4	5	6	7	8	9	10	11	12
Arduino	6	6	7	6	6	7	N/A	N/A	N/A	N/A	N/A	N/A
NodeMCU	6	6	7	6	6	7	N/A	N/A	N/A	N/A	N/A	N/A
Application	6	6	7	6	6	7	N/A	N/A	N/A	N/A	N/A	N/A

In this 3 time test the results of sending pH data from Arduino to the application are the same. From the reading value of the pH sensor, valid results were obtained on plant media 1-6 because the tool (pH sensor) was installed and calibrated correctly and for planting media 7-12 the results are not available because the sensor is not installed. The test point is the success of sending pH data from Arduino to the application.

4.2.6. Accuracy of reading the DHT22 Temperature Sensor value

For reading the room temperature sensor value in the application, a comparison will be made between the DHT22 Sensor Reading with the original room temperature according to the Analog Temperature sensor. This test was conducted to see the performance of the DHT22 Sensor.

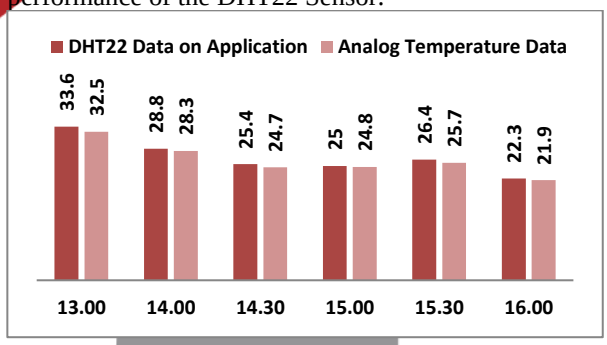

Figure 4.7. Temperature Value comparison chart

The DHT22 temperature sensor was tested for accuracy and compared to an analog room temperature meter. The results showed a good accuracy of 2%-5% error value, according to the DHT22 datasheet in Appendix A.4. The difference in accuracy can be influenced by the sensing element factor, such as a different type or not being good, where DHT22 uses a polymer capacitor as the sensing element.

Tabel 4.9. DHT22 and room temperature comparison and error value results.

No.	Measurement Time	H_S	H_H	$H_S - H_H$	e_H (%)
1.	13.30	33,6 °C	32,5 °C	1,1 °C	3,38
2.	14.00	28,8 °C	28,3 °C	0,5 °C	1,77
3.	14.30	25,4 °C	24,7 °C	0,7 °C	2,83
4.	15.00	25,0 °C	24,8 °C	0,2 °C	0,81
5.	15.30	26,4 °C	25,7 °C	0,7 °C	2,72
6.	16.00	22,3 °C	21,9 °C	0,4 °C	1,83
	χ_{e_H}				2,22

From the data in the Table 4.9, it can be seen that the reading of the DHT22 temperature value is higher than the original room temperature. Based on the results of calculating the average error value from

testing the temperature readings between the DHT22 sensor and the standard room temperature meter, the average error value is 2.22%. Based on the data sheet in appendix A.4. This result shows good accuracy because it is in the range of the average error value of 2% -5%.

4.3. Quality of Service

Wireshark is used to test the quality of service (QoS) for IoT-based drip irrigation by analyzing delay and throughput parameters. Testing is done by changing the distance between the device and the access point for 4 sessions at 0, 5, 10, and 15 meters with an internet connection.

4.3.1. Delay

In this test, it will be tested how long the delay occurs during data communication between Hardware and Applications. In this test, it will be tested how long the delay in sending data from the hardware to the application or control data from the application to the Esp8266 will be tested. This test is done by processing the vulnerable time of sending some data randomly between the Application and Hardware.

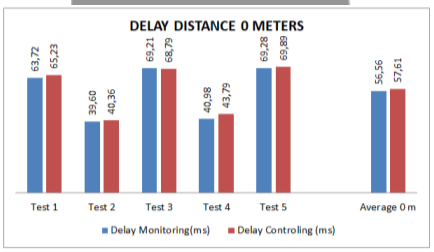

Figure 4.8 Delay measurement chart at 0 meters

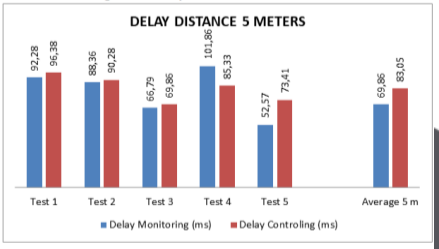

Figure 4.9 Delay measurement chart at 5 meters

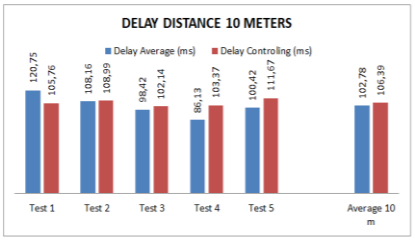

Figure 4.10 Delay measurement chart at 10 meters

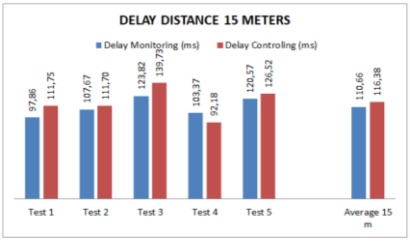

Figure 4.11 Delay measurement chart at 15 meters

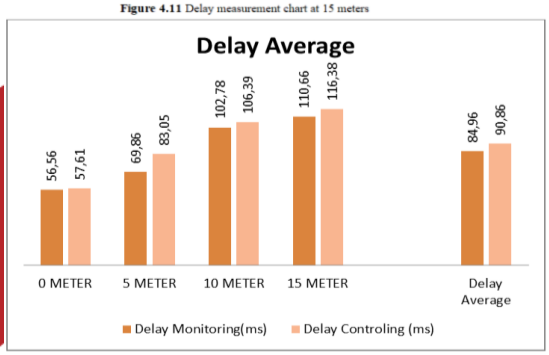

Figure 4.12 Average Total Delay

Figure 4.12 shows the average delay at each distance of the tool's range. The best average delay is at 0 meters, while the worst is at 15 meters. The delay increases as the distance from the access point increases. The delay measurement results are still considered very good, as they are all below 150 ms.

4.3.2. Throughput

Throughput is the total number of observed packet arrivals at the destination during a certain time interval divided by the duration of that time interval. Just like the calculation of delay, the calculation of the average throughput is done by doing 5 experiments at each distance of 0, 5, 10 and 15 meters with a pause in each session of 1 minute. From each experiment each distance taken for each average result.

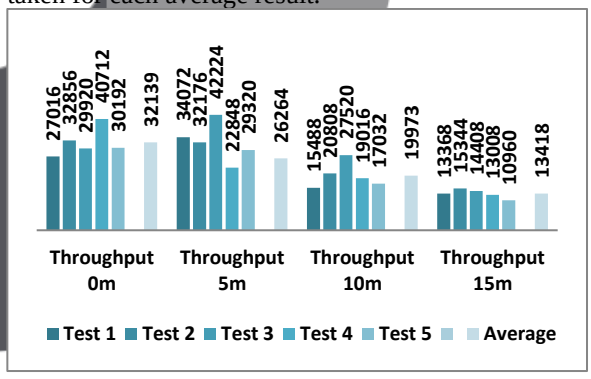

Figure 4.13. Average throughput per session results.

The graph in Figure 4.13 shows the average throughput at different distances from the internet access point. The highest average throughput is at 0 meters with 32139 bps, followed by 5 meters with 26264

bps, 10 meters with 19973 bps, and the lowest is at 15 meters with 13418 bps. Graph shows that throughput decreases as distance from internet access point increases. Average throughput for all distances tested is 19594 bps.

IV CONCLUSION

After implementing and testing this Drip Irrigation System and Hardware Application, the following conclusions can be drawn:

1. The irrigation system works according to the functions and points of watering or treatment in each Cannals.
2. Automatic Watering at a predetermined hour is successful based on the classification data.
3. The reading and sending of pH, humidity and temperature data on the device was successfully sent to Database and display on Application.
4. DHT22 Room Temperature data reading obtained good quality results.
5. The smallest average delay value is 56.56 ms at a distance of 0 meters. While the largest average delay value is 116.38 ms at a distance of 15 meters. Best throughput value is at a distance of 0 meters (32,139 bps) bigger then value at a distance of 15 meters (19,594 bps).
6. When the distance from the access point to the device is getting closer, the network quality is getting better.

REFERENCES

[1] S. Dasberg and Dani Or, “Applied Agriculture Drip Irrigation” Springer-Verlag Berlin Heidelberg GmbH vol. 161, 2019.

[2] M. Rivai, Suwito, M. Ashari and M. A. Mustaghfirin, "Drip Irrigation System using BLDC Motor-driven Direct Pumping and Soil Moisture Sensor," 2019 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE), Jember, Indonesia, 2019, pp. 221-226,doi:10.1109/ICOMITEE.8921024, 2019.

[3] F. M. Anggrayni, D. R. Andrias, M. Adriani “Ketahanan Pangan Dan Coping Strategy Rumah Tangga Urban Farming Pertanian Dan Perikanan Kota Surabaya” Jurnal Media Gizi Indonesia, Program Studi S1 Ilmu Gizi Fakultas Kesehatan Masyarakat, Universitas Airlangga, Surabaya, Indonesia, Vol 10, No 2, 2015.

[4] G. H. Cahyono, “Internet of Things (Sejarah, Teknologi dan Penerapannya)”, sp, vol. 6, no. 3, Dec. 2016.

[5] L. K. Saputra and Y. Lukito, "Implementation of air conditioning control system using REST protocol based on NodeMCU ESP8266," 2017 International Conference on Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS), Yogyakarta, , pp. 126-130, doi: 10.1109/ICON-SONICS.2017.8267834. 2017.

[6] Oktariawan, Imran, Martinus and Sugiyanto, “Pembuatan Sistem Otomasi Dispenser Menggunakan Mikrokontroler Arduino Mega 2560”. Jurnal Ilmiah Teknik Mesin, Vol 1, No 2, April 2013.

[7] E. B. Lewi, U. Sunarya, D. N. Ramadan, “Water Level Monitoring System Based On Internet Of Things Using Google Firebase”. Universitas Telkom, D3 Teknik Telekomunikasi, e-Proceeding of Applied Science : Vol.3 . ISSN : 2442-5826. 2017.

[8] G. Bedi, G. K. Venayagamoorthy, R. Singh, R. R. Brooks and K. Wang, "Review of Internet of Things (IoT) in Electric Power and Energy Systems," in IEEE Internet of Things Journal, vol. 5, no. 2, pp. 847-870, April 2018.

[9] Wilianto, A. Kurniawan. “Sejarah, Cara Kerja Dan Manfaat Internet Of Things,” Matrix : Jurnal Manajemen Teknologi dan Informatika, [S.l.], v. 8, n. 2, p. 36-41,. ISSN 2580-5630. July 2018.

[10] G. M. Madhu and C. Vyjayanthi, "Implementation of Cost Effective Smart Home Controller with Android Application Using NodeMCU and Internet of Things (IOT)," 2018 2nd International Conference on Power, Energy and

Environment: Towards Smart Technology (ICEPE), Shillong, India, pp. 1-5, doi: 10.1109. 2018.
[11] D. Pujilestari, “Manfaat dan Cara Kerja IoT” December, 2020. https://www.smartcity indo.com /2020/12/manfaat-dan-cara-kerja-iot.html. SmartCityIndo, Accesed 2021.
[12] H. Husdi, “Monitoring Kelembapan Tanah Pertanian Menggunakan Soil Moisture Sensor Fc-28 Dan Arduino Uno,” Ilk. J. Ilm., vol. 10, no. 2, pp. 237–243, 2018, doi: 10.33096/ilkom.v10i2.315.237-243.
[13] G. D. Ramady, R. Hidayat, “Sistem Monitoring Data pada Smart Agriculture System Menggunakan Wireless Multisensor Berbasis IoT,” Pros. Semin. Nas. Teknoka, vol. 4, no. 2502, pp. E51–E58, 2019, doi: 10.22236/teknoka.v.
[14] R. Wulandari, “Analisis Quality of Service (QoS) pada jaringan internet studi kasus : UPT Lokauji Teknik Penambangan Jampang Kulonprogo - LIPI,” vol. 2, pp. 162–172, 2016.
[15] P. Arief, "Alat Monitoring Hemoglobin Menggunakan Algoritma Jaringan Saraf Tiruan Propagasi Kembali Berbasiskan Internet of Things", 2019. https://docplayer.info/208380555-Sistem-pintar-berbasis-internet-of-things-iot-untuk-kolam-ikan-koi.html. Accesed 2021.
[16] Ratnasih, D. Perdana and Y. G. Bisono, “Performance Analysis and Automatic Prototype Aquaponic of System Design Based on Internet of Things (IoT) using MQTT Protocol,” Jurnal Infotel, Vol. 10. No. 3. 2018.
[17] F. Puspasari, T. P. Satya and U. Y. Oktiawati, “Analisis Akurasi Sistem Sensor DHT22 berbasis Arduino terhadap Thermohygrometer Standar”, Jurnal Fisika Dan Aplikasinya, Volume 16, Nomor 1, 2020.
[18] A. Andhini, I. Ibrahim, dan Y. Saragih, “Implementasi Aplikasi Styins Home pada Smart Home Security Menggunakan Real-Time Database Firebase”, JurnalEcotipe, vol. 7, no. 2, hlm. 117-126, Okt 2020.
[19] E. Kumara “Analisis Paket Data dengan Mengunakan Wireshark dan Command Prompt” Palembang, Jurnal Sistem Komputer Fakultas Ilmu Komputer Universitas Sriwijaya vol. 1, no. 2, 2017.
[20] N. Naraswari, F. Imansyah and F. T. Pontia “Analisis Uji Kuat Sinyal Terhadap Jarak Jangkau Maksimal Sistem Penerimaan Sinyal Internet berbasis Edimax Hp-5101ack” Pontianak, Jurnal Teknik Elektro Fakultas Teknik, Universitas Tanjungpura, 2017.