



Design and Manufacture of a Monitoring Tool for the Amount of Water Usage Using IoT-Based NodeMCU ESP32

Miftahatur Rizqi, Zulwisli, S.Pd., M. Eng

Department of Electronics Engineering, Faculty of Engineering

Padang State University, Indonesia

ABSTRACT

Various public service institutions are increasingly improving the quality of their services. One of the public services is the clean water system at PDAM. Currently, the information on bills charged by customers from water consumption is still minimal, customers only know the bill when they want to make payments. Not only that, sometimes the fees charged are very large because customers cannot see how much water is used. Thus, the making of this Final Project aims to apply Internet of Things-based technology to design a control and monitoring system for the amount of water usage with water input detected by the Water Flow Sensor which is controlled by the ESP32 Wifi microcontroller in the form of a large reading output of water volume, water discharge, tariff fees that can be monitored on a smartphone. Testing can be done by comparing the volume of water measured and read on a measuring cup, LCD and smartphone. With this system we can easily monitor the use of water every day in real time via a smartphone.

Keywords: Water, Monitoring, Wifi ESP32, Internet of Things, Smartphones

1. Introduction

A water meter is a type of instrument for measuring the volume of drinking water in a pipeline network to serve both individual and group use with due regard to technical and non-technical aspects, so that people can easily obtain a certain amount of water.

However, what is currently running at PDAM Tirta Benteng Kota Tangerang in calculating the costs that must be incurred by customers from water use is still using manual procedures, namely officers must visit one customer at a time and check the value of the volume/pressure issued each month and then convert it into a value. cost/litre. This causes inefficient service. Another problem that arises is the lack of information to customers how much it costs to pay each month which results in a lack of information when they want to make payments for water usage fees.

Customers can directly see their water usage and staff automatically get customer data without having to visit customers one by one. The microcontroller used in this Monitoring System is NodeMCU which is based on ESP32. The flow rate and water volume measurement system that is designed basically consists of two main parts, namely the flow rate sensor or *water flow sensor* and a microcontroller for processing the measured signal. Several other components used as system control functions use relays and *water pumps* as automatic valves or faucets.

Here I want to design a tool that is able to monitor digitally-based (IoT) usage. The entire system can be accessed using a smartphone on the MIT App Inventor application and is expected to be able to provide convenience for all parties, not only for agencies such as water companies, but also for the benefit of the whole community.

2. Research methods

With the Prototype Model approach, the description or stages of the research are related to the activities carried out in this study. There are several more detailed explanations regarding the methods used in this study.

Prototype is the initial stage of the system used for design trials, design concepts and finding problems and possible solutions. The *prototyping* method starts with the needs of the user to determine the purpose of designing a tool or software and identify rules. Then an overview of the tools and applications is created which is then visible to the user.

2.1 Research Materials

The materials needed in designing and manufacturing a monitoring tool for the amount of water use using the IoT-based NodeMCU ESP32 are as follows.

- | | |
|----------------------------------|-----------------------|
| a. Waterflow Sensors | h. 220V DC Water Pump |
| b. NodeMCU ESP32 microcontroller | i. Buckboost |
| c. LEDs | j. LCD 16 x 2 |
| d. RTC | k. Powersupply12V |
| e. Relays | l. FUSE 10A |
| f. Arduino IDE | m. Jumper Cable |
| g. Solenoid Valve | |

2.2 Water Discharge Calculation System by Tool

The inflow channel that is read by the sensor is done by turning on and turning off the flowing *water* pump, as a substitute for the process of opening and closing the water faucet that

will go through the *solenoid valve* which is used to lock and open the water that will flow to the *water flow* sensor . *waterflow* sensors which acts as a water discharge measuring sensor. In the calculation, the *interrupt* in the sensor reading produces a *Flowrate* (current speed) which is then converted into the water discharge calculation formula and repetition occurs so that the discharge increases (Counter ++) to produce units of liters. In the process and when the pump is turned off the last usage discharge data is entered *online* via the ESP32 Wifi module and stored in the *database*.

2.3 Tool Design Drawings

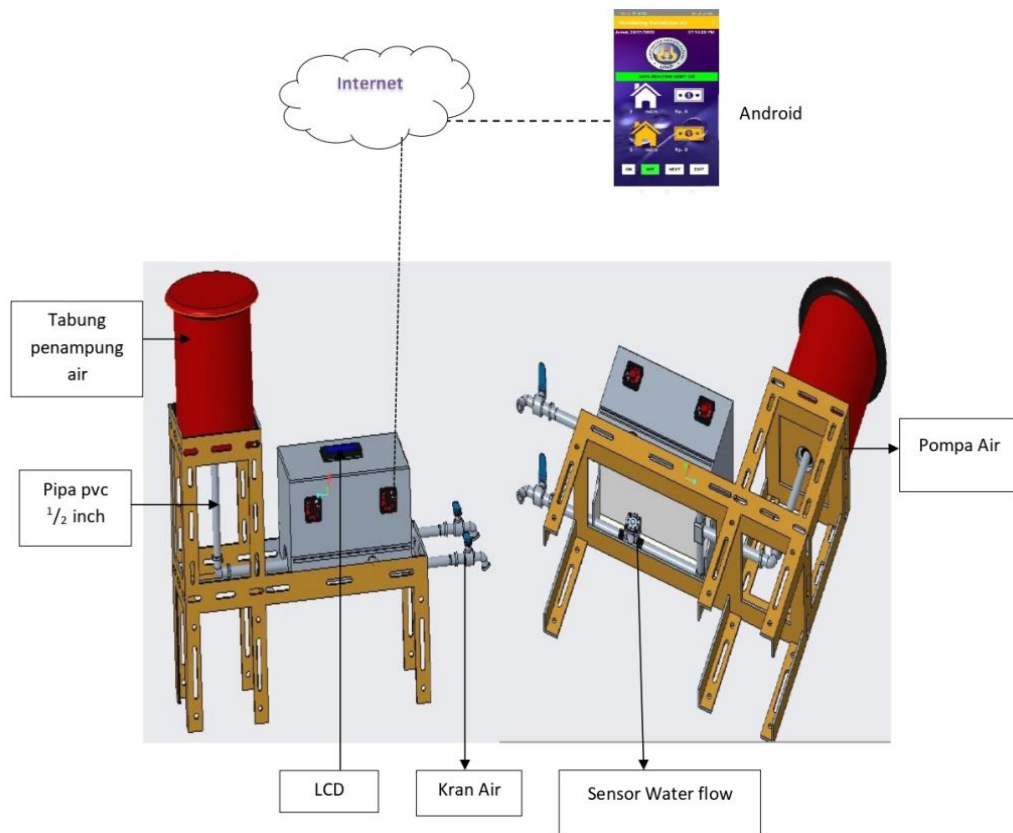


Figure 1. Tool Design

On the side view of the tool design there is a water reservoir tube, inside the water reservoir there is a *water pump* . Furthermore, there is a 16 X 2 LCD above the box measuring 30 X 20 cm

in which there are components such as NodeMCU esp32, relay, Auto buckboost, switches and others. In the picture below there is a *water flow* sensor which is attached to a $1/2$ inch pipe, then there is a faucet where water will flow in a container for a certain time and Wifi esp32 will connect to the Internet and also monitored through the MIT App Inventor application.

2.4 Hardware Flowchart Tool

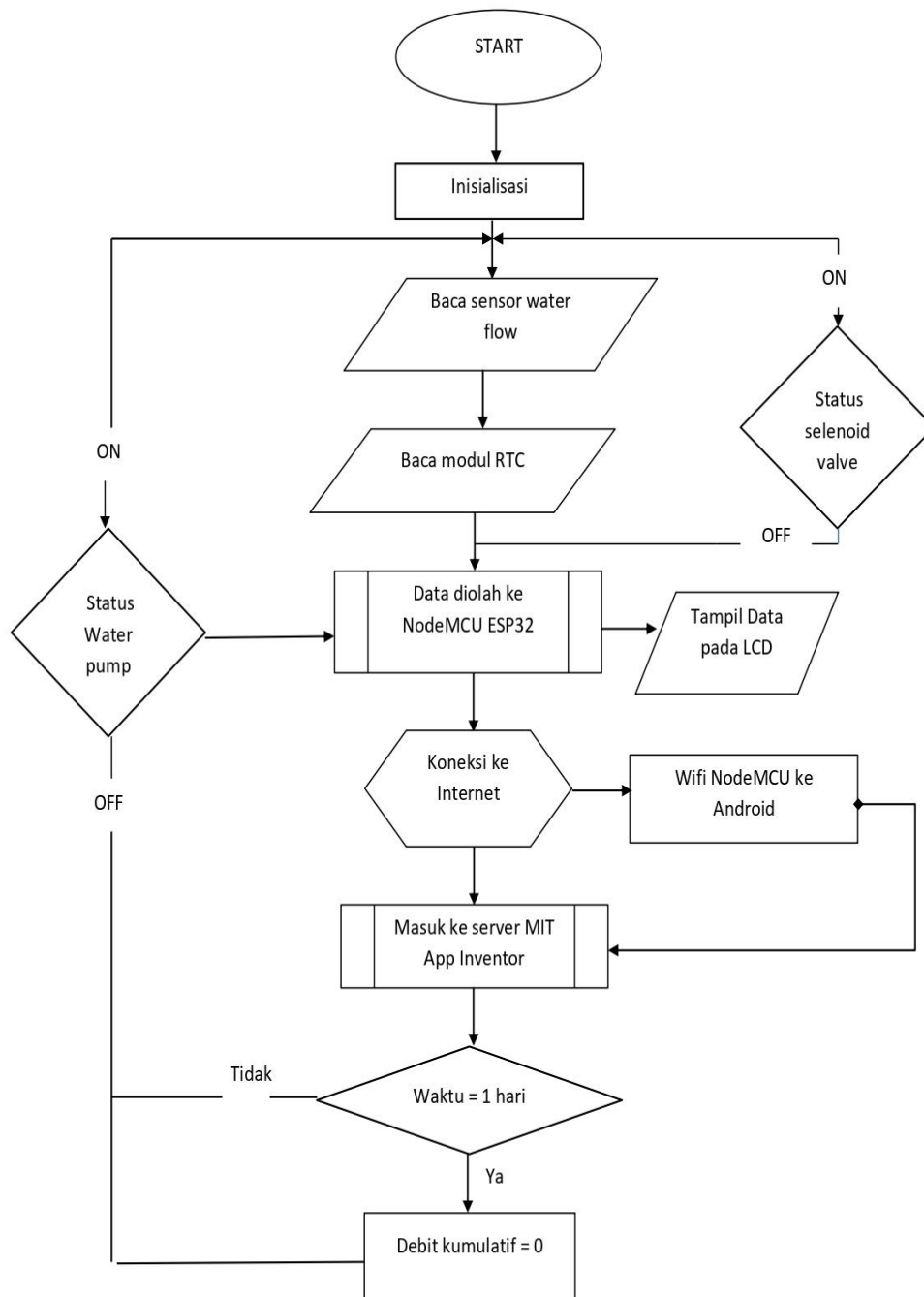


Figure 2. Tool Design Flowchart

Explanation of Flowchart above _ that is as following :

The inflow channel that is read by the sensor is done by turning on and turning off the flowing *water* pump, as a substitute for the process of opening and closing the water faucet that will go through the *solenoid valve which is used to lock and open the water that will flow to the water flow sensor* . *waterflow* sensors which acts as a water discharge measuring sensor. NodeMCU ESP32 will process water discharge data in the form of digital signals to be displayed on the LCD. The LCD displays water discharge data from water usage as well as the date and time of measuring the water discharge. Date and time data can be displayed on the LCD with a real time clock or RTCinput on the Arduino IDE. RTCis a real time clock that can store data of seconds, minutes, hours, date, month, day of the week, and year. Serially then look for a WiFi connection according to what has been programmed. Once connected, ESP32 sends data from the *water flow sensor readings* to the MIT App Inventor server to be displayed on the MIT App Inventor Website on Android.

2.5 Tool Working Principle

The working principle of the control and monitoring above is that all electrical components in this system are connected to a power supply so that this system can only work when it gets electrical voltage. The water supply will flow through the PVC pipe channel when the relay is Normally Close (NC) or in the On state which is controlled through a realtime monitoring data base software using water discharge so that the water pump is active and flows water to the solenoid valve which is used to lock and open the water to be flow to the water flow sensor. The condition of the water pump is controlled by the NodeMCU esp32 microcontroller which obtains input from a smartphone, the water pump is connected to the relay driver, and the NodeMCU esp32. Commands from the smartphone will be sent to NodeMCU esp32, which will then be sent

to the water pump via a relay driver so that the water pump is in the condition according to the desired command.

The water flow that has passed through the water flow sensor will be read by the sensor. The data read by the water flow sensor is the volume and discharge of the water flow. The read data will be sent and processed on the NodeMCU ESP32 microcontroller. From the microcontroller then the data can be monitored on the LCD. Apart from that, data from the microcontroller is also sent to NodeMCU ESP32 which is a Wifi module so that the data can be monitored through the MIT App Inventor application.

3. Results and Discussion

From research observations, testing and analysis aims to find out whether the system created is in accordance with what was planned or not, if there are differences, then the reasons or causes can be known through data analysis. Tests carried out on each design block on the system that has been made. The measuring instrument used for testing this system is a multimeter.

3.1 Measurement Results of one power (Power Supply)

One power measurement is done by measuring several measurement points on the power supply. By using a multimeter measuring tool. Measurement of the power supply voltage when working is carried out at the measurement point as shown in the following figure.

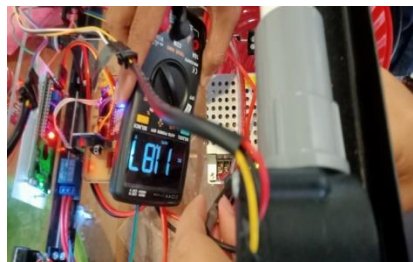


Figure 3. Power supply measurement

Measurement of the power supply using a multimeter, namely the measured voltage of 11.87 V indicates that the power supply is functioning properly.

3.2 Measurement Results Auto buckboost

The buckboost voltage measurement when working is carried out at the measurement point as shown in the following figure.



Figure 4. Measurements on Buckboost

The DC supply provides a voltage to the gate driver of 12V and the Buck Boost Converter of 11.83V. The monitoring system starts from the output signal of the Buck Boost Converter which will be connected first to the ESP32 and programmed on the Arduino IDE.

3.3 Solenoid Valve Test Results

Solenoid valve testing aims to determine the conditions and working principles of the Solenoid valve. Solenoid valve testing is carried out by pairing the Solenoid valve with a pipe then the pins of the Solenoid valve will be connected to the relay and NodeMCU ESP32.

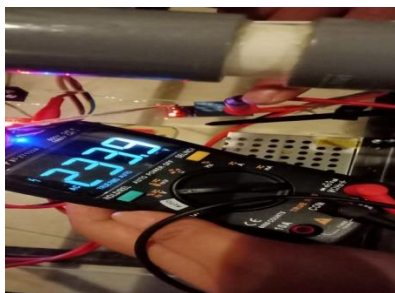


Figure 5. Testing Solenoid Valve measurements when NC is on the relay

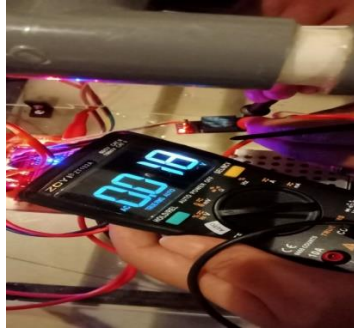


Figure 6. Testing the Solenoid Valve measurement when NO is on the relay

Based on the results of the measurement data above, it is known that when the relay is normally closed (NC) or is on, the AC supply provides a voltage to the gate driver of 233.9 V indicating that the solenoid valve can function properly.

This test is carried out by providing input in the form of a pressurized water supply. The following is a picture of the test results from the Solenoid Valve.



Figure 7. Low level solenoid valve test results



Figure 8. Medium level Solenoid Valve test results

Based on the results of the test data above, it is known that when the solenoid valve is supplied with low pressure water, the solenoid valve can still flow water smoothly so it can be concluded that the solenoid valve can work properly at all pressure levels.

3.4 RTC testing

The RTC DS3231 test in this design aims to store time and date data according to international standards. The water flow sensor will read the flow of water flowing through the rotor. real time clock or RTC will read the time and date of water volume measurement. In the system, the RTC will provide time input to the microcontroller to be displayed on the LCD. This process will be carried out for a number of days so that when the time is right, the system will start again to calculate the accumulative volume of the tool from 0 liters.



Figure 9. Time testing on RTC

3.5 Testing LCD 16 X 2

The LCD test aims to ensure that the time and volume of water data that has been read by the water flow sensor can be displayed on the available 16x2 LCD. The program has been set so that the first line displays the time value of the DS3231 RTC and the second line displays the data value of the water volume variable. The data of the two variables displayed will be updated every second.



Figure 10. LCD test results 16 x 2

3.6 Water Flow sensor testing

In this test there are two water flow sensors to be tested. To make it easier to retrieve data, the first water meter is named meter A, and the second water meter is named meter B. Retrieval of each data will appear on the serial monitor display in the MIT App Inventor application.

3.6.1 Water flow sensor testing (meter A)

Table 1. test results (meter A) with a 500ml measuring cup

Test	Measuring cup (ml)	Sensor reading(ml)	Difference	Error %
1.	500 ml	 Vol = 0.611	111 ml	22,2
2.	500ml	 Vol = 0.521	21 ml	4,2
Average				13.2%

Table 2. Test results (Meter A) 1000 ml measuring cup

Test	Measuring cup (ml)	Sensor reading(ml)	Difference	Error %
1.	1000ml	 Vol = 1.051	41 ml	4,1
2.	1000ml	 Vol = 1.041	51 ml	5,1
Average				4.6%

Table 3. Test results (meter A) 1500ml measuring cup

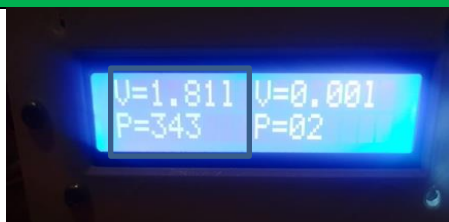
Test	Measuring cup (ml)	Sensor reading(ml)	Difference	Error %
1.	1500ml	 Vol = 1.811	311 ml	20,7
2.	1500ml	 Vol = 1.611	111 ml	7,4
Average				14.05%

Table 4. Test results (meter A) 2000 ml measuring cup

Test	Measuring cup (ml)	Sensor reading(ml)	Difference	Error %
1.	2000ml	 Vol = 1.961	39ml	1.95
2.	2000ml	 Vol = 2.021	21 ml	1.05
Average				1.5%

Based on Table 1. to Table 4. the following conclusions will be obtained:

1. In the meter experiment A with a 500 ml measuring cup, an average *error* of 13.2% was obtained
2. In the meter experiment A with a 1000 ml measuring cup, an average *error* of 4.6% was obtained
3. In the meter experiment A with a 1500 ml measuring cup, an average *error* of 14.05% was obtained
4. In the meter experiment A with a 2000 ml measuring cup, an average *error* of 1.5% was obtained

3.6.2 Water flow sensor testing (meter B)

Table 5. Test results (meter B) 500ml measuring cup

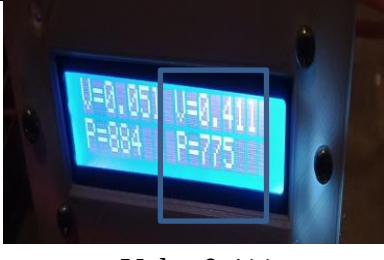
Test	Measuring cup (ml)	Sensor reading(ml)	Difference	Error %
1.	500ml	 Vol = 0.431	69 ml	13,8
2.	500ml	 Vol = 0.411	89 ml	17,8
Average				15.8%

Table 6. Test results (meter B) 1000 ml measuring cup

Test	Measuring cup (ml)	Sensor reading(ml)	Difference	Error %
1.	1000ml	 Vol = 1.331	331 ml	33,1
2.	1000ml	 Vol = 1.121	121 ml	12,1

Average	22.6%
---------	-------

Table 7. Measurement results (meter B) 1500ml measuring cup

Test	Measuring cup (ml)	Sensor reading(ml)	Difference	Error %
1.	1500ml	 Vol = 1.631	131 ml	8,73
2.	1500ml	 Vol = 1.521	21 ml	1,4
Average				5.065%

Table 8. Test results (meter B) 2000ml measuring cup

Test	Measuring cup (ml)	Sensor reading(ml)	Difference	Error %
1.	2000ml	 Vol = 2.021	21 ml	1.05
2.	2000ml		59 ml	2.95

		Vol = 1.941		
Average				2%

Based on Table 5. to Table 8. the following conclusions will be obtained:

1. In the B meter experiment with a 500 ml measuring cup, an average error of 15.8% was obtained
2. In the B meter experiment with a 1000 ml measuring cup, an average error of 22.6% was obtained
3. In the B meter experiment with a 1500 ml measuring cup, the average error was 5.065%.
4. In the B meter experiment with a 2000 ml measuring cup, an average error of 2% was obtained

Based on the experimental table above, it can be seen that the water flow sensor can read the large volume of water and water discharge with different water pressures. From the tests carried out the higher the water pressure value, the greater the value of the water discharge that comes out and the faster the time taken in the process of filling the water volume. From the volume of water that comes out, the average percentage error accuracy of the water flow sensor (meter A) is 8.3% and the water flow sensor (meter B) is 11.3% so it can be concluded that the water flow sensor functions properly according to the principle its work so that it can be used in making the final project as a sensor measuring the volume and discharge of water that comes out.

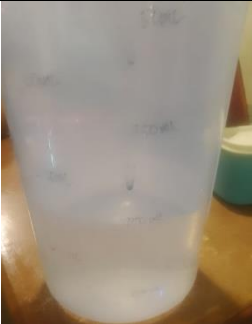
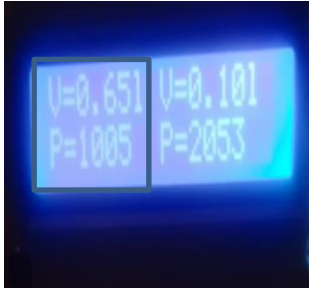
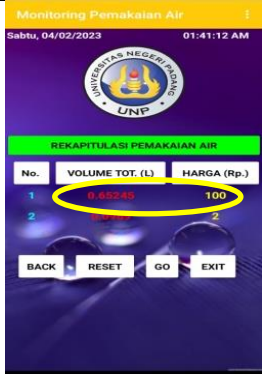
3.7 Automatic Control Test Results and Monitoring the amount of water usage

Automatic control of water volume is a process of controlling water expenditure based on the input provided via a smartphone application. Which, if the volume of water that comes out matches the volume of water input via the smartphone application, the water tap will automatically turn off. With this control, the customer can set a target for dispensing the volume of water as desired. If the desired target volume of water has been met, the water faucet will automatically be locked.



Figure 11. Automatic Control Testing Process

Table 9. Monitoring results of the amount of water usage at Faucet A

No	Input Sliders Water Volume	Comparison of the volume of water in the measuring cup (liters)	Water volume reading on LCD (liters)	Water volume reading on Smartphone
1.	500 ml	 Vol = 651 ml	 Vol = 651 ml	 Vol = 651 ml

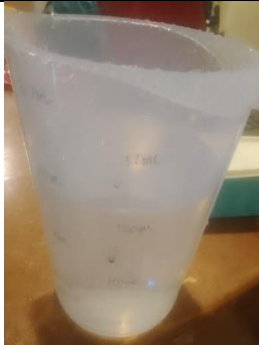
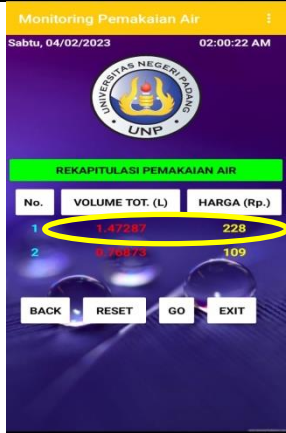
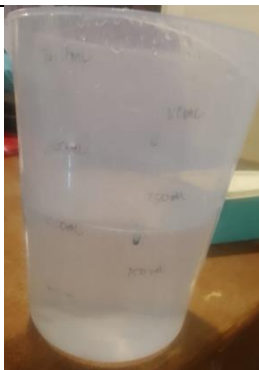
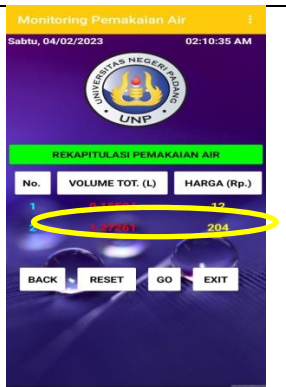
2.	1000 ml	 Vol = 1.471 ml	 Vol = 1.471 ml	 Vol = 1.471 ml
Amount		Vol = 2.122 ml	Vol = 2.122 ml	Vol = 2.122 ml

Table 10. Monitoring results of the amount of water usage at Faucet B

No	Input Sliders Water Volume	Comparison of the volume of water in the measuring cup (liters)	Water volume reading on LCD (liters)	Water volume reading on Smartphone
1.	500 ml	 Vol = 821 ml	 Vol = 821 ml	 Vol = 821 ml
2.	1000 ml	 Vol = 1.271 ml	 Vol = 1.271 ml	 Vol = 1.271 ml
Amount		Vol = 2.121 ml	Vol = 2.092 ml	Vol = 2.092 ml

Based on Table 9. and table 10. the following conclusions will be obtained:

1. In the water faucet experiment A with a 500 ml measuring cup, an average error of 30.2% was obtained
2. In the water faucet experiment A with a 1000 ml measuring cup, an average error of 47.1% was obtained.
3. In the B faucet experiment with a 1750 ml measuring cup, an average error of 64.2% was obtained.
4. In the B faucet experiment with a 2000 ml measuring cup, an average error of 45.6% was obtained.

From the volume of water that comes out, the average percentage error accuracy of the water flow sensor (faucet A) is 38.6% and the sensor water flow (faucet B) is 27.1% so it can be concluded that the difference in the measuring cup and LCD with the input in the Arduino IDE program due to delays or time when outputting data the water flow sensor functions properly according to its working principle as a sensor measuring the volume and discharge of water that comes out.

But the opposite also happens, namely in the ratio of the percentage of water flow discharge errors. This is also influenced by some measurement limitations. One that is considered quite influential is that the measurement is carried out with variations in flow rate which are not constant considering that the rotation of the tap from the water source is still done manually even though an automatic valve (solenoid valve) has been installed on the outlet valve. The limitations of the volume measuring instrument in this measurement are felt to have quite an effect on the results of measurements and calculations.

4. Conclusion

The monitoring tool for the amount of water usage based on the internet of things can work well, that is, it can measure the volume of water by testing water use and payment fees can be displayed properly on the MIT App Inventor application so that users can observe via Android or smartphone.

ACKNOWLEDGEMENTS

In completing this design and final project, the author has been assisted by several parties. By several parties. Therefore, the author would like to thank to Allah SWT who has given life, safety, and health both jasani and spirit. Prophet Muhammad SAW who is always our role model.

REFERENCES

- (1) Burange, Anup & Misalkar, Harshal. (2015). Review of Internet of Things in development of smart cities with data management & privacy. 189-195. 10.1109/ICACEA.2015.7164693.
- (2) Dong, Dai and Li, Xiaoning. (2017). November. *Simulation and Experimental Research on the Response of a Novel High-pressure Pneumatic Pilot-operated Solenoid Valve*. 19th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), Auckland, New Zealand.
- (3) Doshi, Jash, Tirthkumar Patel, and Santosh kumar Bharti. 2019. "Smart Farming Using IoT, a Solution for Optimally Monitoring Farming Conditions." *Procedia Computer Science* 160:746–51
- (4) Efendi, Y. (2018). *Internet Of Things (Iot) Sistem Pengendalian Lampu Menggunakan Raspberry Pi Berbasis Mobile*. *Jurnal Ilmiah Ilmu Komputer*, 4(2), 21–27.

