Vol 01 2023, pp 30-36
e-ISSN: XXXX-XXXX
DOI: https://doi.org/10.54482/SPECTRUM/

Software Tool for Monitoring the Amount of Water Usage Using Iot-Based Mcu Nodes

Rini Marniati AN, Geovanne Farell SPd, M.Pd.T.

Department of Electronics Engineering, Faculty of Engineering

Padang State University, Indonesia

ABSTRACT

Making this Final Project aims to apply Internet of Things-based technology that is designed by making hardware and software. The process of designing and building the system as a whole follows several stages, namely: 1) creating an application design as a tool controller, 2) creating a design tool program to create a water monitoring tool that can measure water usage that works automatically using an application that has been designed using the MIT APP Inventor and make the menu as the main controller for the tool, after designing the controller application we also enter program code using the Arduino Ide application using the C++ programming language.

Keywords : Internet Of Things, MIT APP Inventor, Arduino Ide, Firebase,Smartphone

1. Introduction

water _ is source Power good vital nature For flora, fauna and life man in advance earth nor For need man in need daily at various sector life . Every human being on this earth will be very dependent on water. Without water maybe humans can not live. There are many activities related to water such as drinking, bathing and washing or other things in daily life. Humans often use water irregularly . so that can cause water pollution and will decrease water on future to prevent this, the government creates a company that can provide and distribute clean water to the community.

This water monitoring can identify the water usage of each house with the results of an analog meter display that always changes the numbers every time and there are other supporting indicators such as a compass that is always rotating which shows that water is flowing . The problem that often arises is reading the water debit is done manually and the costs incurred are not in accordance with the distributed water discharge, with the incompatibility of the data obtained then can the occurrence of various frauds in reading usage data, accompanied by unknown damage and so on, apart from that the water discharge readings used at this time still use analog so that method is less efficient in today's globalization era.

A water meter is a type of instrument for measuring the volume of drinking water in a pipeline network to serve the use of both individuals and groups with due regard to technical and non-technical aspects, so that people can easily obtain a certain amount of water. Currently, many water companies are still using manual water meter equipment systems. measuring the amount of water use is widely used in an analogue manner so that water usage data is difficult for customers to know. This tool is designed to use a *water flow* sensor to

measure the discharge of water flowing into the pipe and the measurement results will be processed by the NodeMCU microcontroller. The entire system can be accessed using Mit App Inventor 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. Method

2.1.Research material

The materials needed in making IoT-based water usage monitoring are as follows.

1. NodeMCU ESP32 microcontroller
2. Water flow sensor
3. Relays
4. Arduino pcb board
5. power supply
6. 220V DC water pump
7. Jumper Cable
8. Solenoid valve
9. Outo buckboos
10. Led
11. LCD 16 x 2

2.2.Research Tools

The tools needed to make monitoring the amount of water usage based on IoT consist of software

a. Software

   1. Laptops

2. Arduino ide software
3. MIT APP Inventor
4. Soilwork software

2.3.Tool design drawing

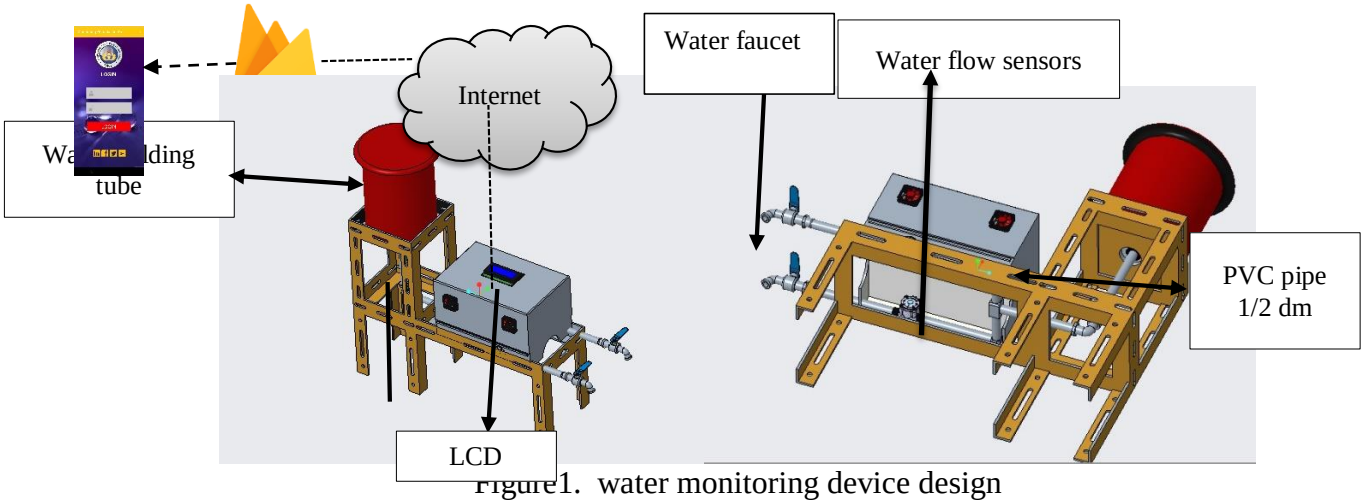


Figure1. water monitoring device design

2.4.Application Flowcharts

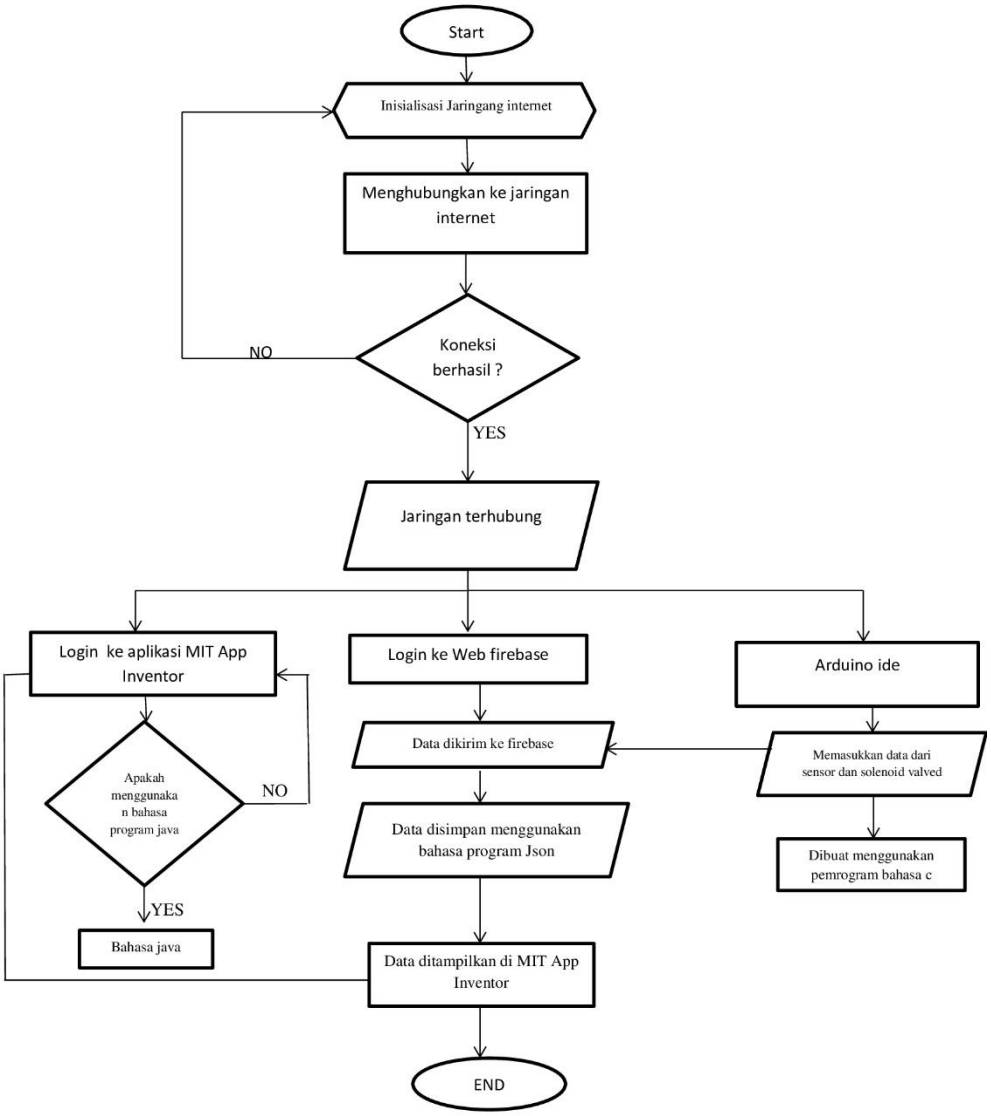


Figure 1Flow Chart

Explanation of the Flow Chart above is as follows:

Where in the start section it is done to start executing commands, after that it is given an initial value to activate the internet network by describing a condition if the internet connection is connected then the network will be connected temporarily if the internet connection is off the network is not connected. After the network is connected, the next step is to enter the *Mit* App Inventor application by stating a statement about using or not

using the Java language. c++ to *firebase* which will be stored using the Json programming language (JavaScript Object Notation) . After the Arduino ide program has been created and *inputted* , the next step is to connect the tool with the application by logging into the firebase web that we have created so that we can copy the firebaseURL address and *firebase* token to the proportions menu on the firebaseDB1 device. Data sent by NodeMCU to firebase will be displayed on to *firebase* then the data will be displayed by the Lcd and the Mit App Inventor application.

2.5.Tool Working Principle

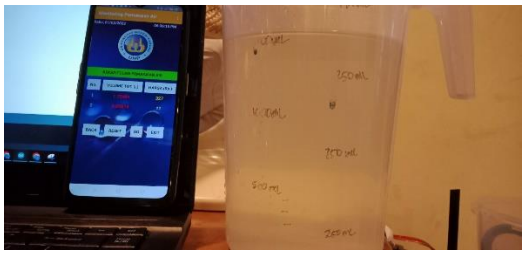
Software design for making applications on *smartphones* is made using the Mit App Inventor application. Making application designs in Mit App Inventor, namely in the design designer section by selecting devices related to the application to be made on the user interface menu located on the left side of the Mit App Inventor display which is then drawn in and arranged in such a way in the display phone. The properties menu to the right of the Mit App Inventor display functions to manage the devices used. After the design is made, the next step is to make a program or coding for each screen on the blocks section. After setting the appearance of the application is complete, to activate this application on a smartphone, select build on the menu and select (app provide QR code for.apk).

Next, we will create a program for each of the tool components that we use in making the program using the C ++ language which is made in the Arduino ide software . After the program is finished, the program will saved and compiled will be in a file with a hex extension which will then be the program sent to the NodeMcu Esp32 microcontroller so it can manage existing data sent after the data is processed on the microcontroller so that it is connected to the internet.

To connect the water usage monitoring tool with the application, we must copy the *firebaseURL* and *firebase* Token addresses to the proportion menu on the *firebaseDB1 device* . The data sent by NodeMCU to *firebase* will will saved as JSONs And synced in a manner *realtime* to every client connected and displayed on the Mit App application according to a predefined virtual pin so that the application on *the smartphone* works as desired.

3. Results and Discussion

This research has been successfully developed to produce an IoT-based water usage monitoring tool. The tool program for monitoring the amount of water usage based on IoT can be seen in the image below



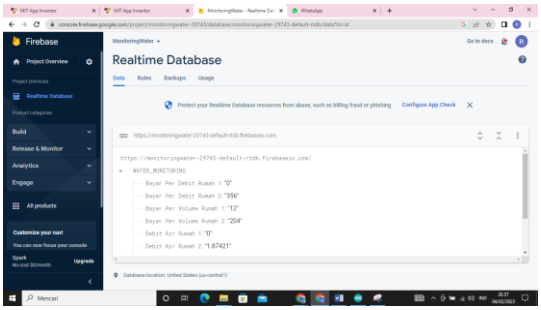


Figure 2when the tool has been monitored as a whole

3.1. Application Testing at MIT App Inventor

a. Application testing when it's not

when android is not connected to wifi then the application does not run properly and will not connect to the database.

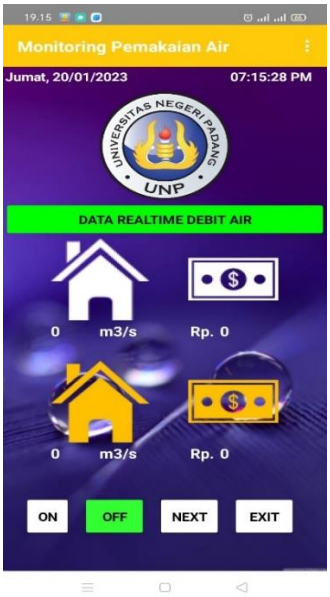


Image 3. When the application is not connected to the internet

Seen in the picture above when the application is connected to the ESP32 module, the application can receive data from the database and will display water debit data and the price that must be paid per debit

b. Application testing when connected

when android is connected to wifi, the application runs well and will connect to the database.

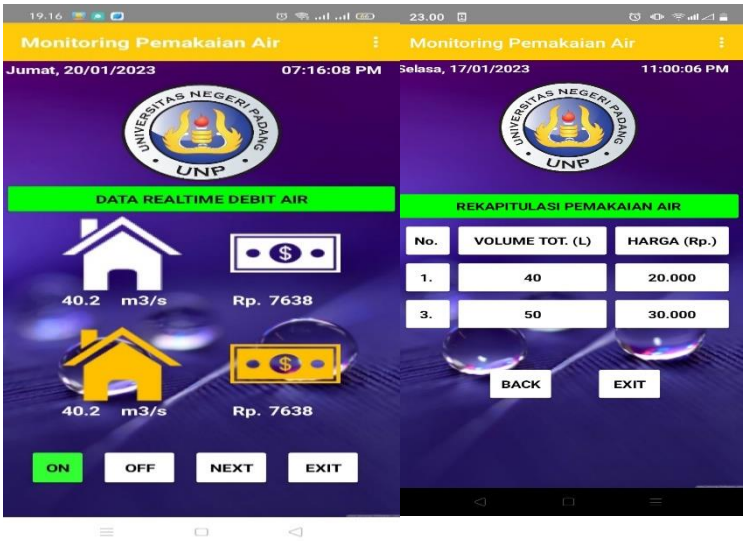


Figure 4applications that are connected to the internet network

Seen in the picture above when the application is connected to the ESP32 module, the application can receive data from the database and will display water debit data and the price that must be paid per debit

3.2.Data testing in firebase

a. When not connected wifi

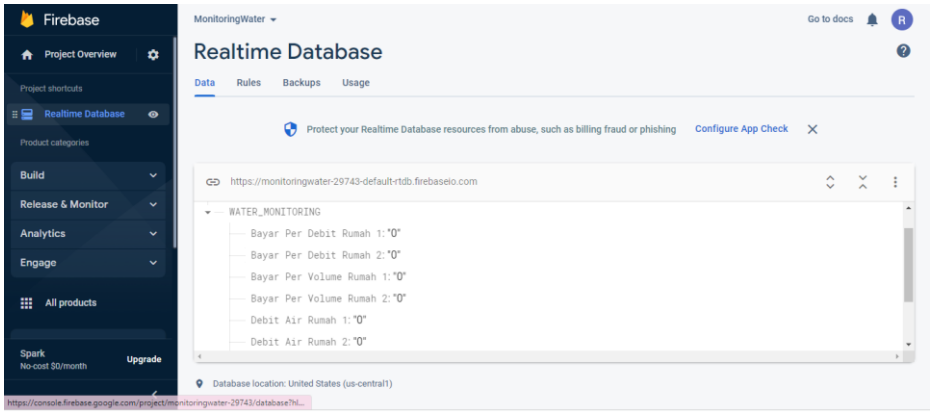


Figure 5. when data is not sent to firebase

When the PC or laptop used to access Firebase is not connected to WiFi, Firebase will not be able to receive and send data from ESP32 or send and receive data to applications that have been made.

b. When connected to wifi

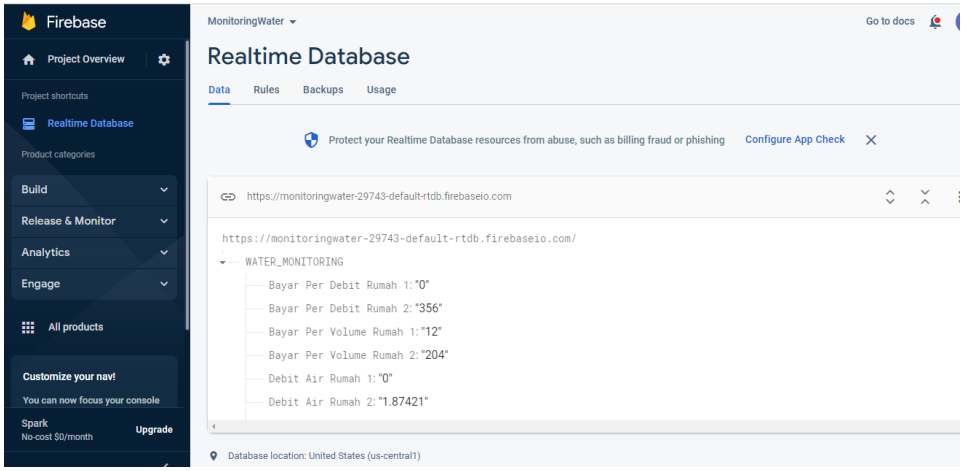


Figure 6When data from the tool is read to Firabase

when the PC or laptop used to access Firebase is connected to wifi, Firebase will be able to receive and send data from ESP32 or send and receive data to applications that have been made.

3.3.Arduino IDE program testing

In this Arduino IDE program, it discusses reading data from sensors sent to Firebase with a comprehensive program listing as follows.

In this Arduino IDE program, it discusses reading data from sensors sent to Firebase with a comprehensive program listing as follows.

```
//Memanggil Library
#include <Arduino.h>
#include <Wire.h>
#include <WiFi.h>
#include "time.h"
#include <LiquidCrystal_I2C.h>
#include <FirebaseESP32.h>
#include <FlowMeter.h>
```

The Symtax program above functions to call the time library, wifi, lcd i2c, firebase and flow meter.

```
//Memasukkan URL dan Token Firebase
#define FIREBASE_HOST "https://monitoringwater-29743-default-rtdb.firebaseio.com/"
#define FIREBASE_AUTH "CurT2qYC91C8dzIdjgvyjWi0q989mrocvVKAKnnG"
```

Symtax The above program functions to input URLs and firebase tokens

```
//Memasukkan Usernamae & PW wifi
#define ssid "WIFI-UNP"
#define password ""
```

Symtax Diats program functions to input the Username and password used.

```
//Deklarasi Waktu
const char* ntpServer = "pool.ntp.org";
const long  gmtOffset_sec = 7 * 3600;
const int   daylightOffset_sec = 3600;
```

Symtax The Diats program functions to declare the time function. The number 7 functions for the declaration of West Indonesian Time (WIB). The number 3600 indicates the value of seconds per hour.

```
//Deklarasi Variabel dan pin ESP-32
const int relay = 2;
String sistem;
float deb = 0; double vol = 0; float deb2 = 0; double vol2 = 0; double voltot; double voltot2;
unsigned int pay = 0; unsigned int paytot = 0; unsigned int pay2 = 0; unsigned int paytot2 = 0;
```

Symtax The above program functions to declare the variables and pins used. system variables are used to store values retrieved from the database, while the variables deb , vol, deb2, vol2, pay, paytot, pay2, paytot2 are used to store values sent to firabase .

```
//Mulai koneksi wifi
WiFi.begin (ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  delay(300);
}
```

Symtax The program above functions to start a wifi connection

```
//Memulai firebase
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
Firebase.reconnectWiFi(true);
```

Symtax The program above functions to start firebase

```
//Deklarasi sensor water flow pada pin 15 dan pin 18 ESP-32 sebagai Interupt
Meter = new FlowMeter(digitalPinToInterrupt(interruptPin), mySensor, handleInterrupt, FALLING);
Meter2 = new FlowMeter(digitalPinToInterrupt(interruptPin2), mySensor2, handleInterrupt2, FALLING);
```

Symtax The above program functions to declare the pin of the water flow sensor and the pin of the ESP-32 as an Interrupt.

```
// Menampilkan data Debit, Volume dan Nilai Bayar di Serial Monitor
Serial.println("Volume Rumah 1:" + String(vol) + " l total.");
Serial.println("Volume Rumah 2:" + String(vol2) + " l total.");
Serial.println("Bayar Per Volume Rumah 1: " + String(paytot));
Serial.println("Bayar Per Volume Rumah 2: " + String(paytot2));
```

Symtax The above program functions to display debit, volume and payment value data on the serial monitor

```
//Mengirim data Voolume dan Nilai Bayar ke Firebase
Firebase.setString(firebaseData, "/WATER_MONITORING/Volume Air Rumah 1", vol);
Firebase.setString(firebaseData, "/WATER_MONITORING/Volume Air Rumah 2", vol2);
Firebase.setString(firebaseData, "/WATER_MONITORING/Bayar Per Volume Rumah 1", paytot);
Firebase.setString(firebaseData, "/WATER_MONITORING/Bayar Per Volume Rumah 2", paytot2);
```

The Symtax program above functions to send volume and payment value data to firebase so that it can be stored as a database.

4. Conclusion

The program that has been made has worked successfully, but the actual results are not in accordance with the values set in the program, this is caused by several problems such as the internet speed that is used is not good, the components of the tool that are used for too long become hot so that the performance or work of the tool slowing down, error reading measurement results and so on.

ACKNOWLEDGEMENTS

Thank you profusely to my parents and family who have given prayers and encouragement and love to the author. As well as my D3 Electronic Engineering friends class of 2019 who have provided support and always motivated me.

REFERENCES

[1]. Hakim, D. B. (2009). PDAM pada Rumah Tangga Menggunakan Mikrokontroler NODEMCU Berbasis Smartphone ANDROID, J. IPTEK,. 45–52.

[2]. Hardisal. (2022). monitoring dan stimulasi detak jantung dengan murottal al-q'an berbasis internet of things. 22.

[3]. Ramadhan, A. A. (2021). Sistem Kontrol Dan Monitoring Meteran Air Perusahaan Daerah Air Minum (Pdam) Berbasis Iot.

[4]. Riyanto, T. (2018). Rancang Bangun Sistem Keran Air Dan Monitoring Jumlah Pemakaian Air Berbasis IOT Via Aplikasi Android (Doctoral dissertation, Universitas Komputer Indonesia.