

MONITORING SERVER ROOM TEMPERATURE REMOTELY IN REAL TIME USING RASPBERRY PI AND FIREBASE



**This Final Project is Compiled as a Condition to Complete Bachelor Degree Program at
Informatics Department in Faculty of Communication and Informatics**

Submitted by:

ADE GUNTORO PUTRA ANGGORO

L 200 134 015

DEPARTMENT OF INFORMATICS

FACULTY OF COMMUNICATION AND INFORMATICS

UNIVERSITAS MUHAMMADIYAH SURAKARTA

2018

APPROVAL PAGE

MONITORING SERVER ROOM TEMPERATURE REMOTELY IN
REAL TIME USING RASPBERRY PI AND FIREBASE

SCIENTIFIC PUBLICATION

by:

ADE GUNTORO PUTRA ANGGORO

L 200 134 015

Has been reviewed and approved for testing by::

Supervisor

A handwritten signature in blue ink, consisting of a stylized 'B' followed by a series of loops and a final flourish.

Dr. Ir. Bana Handaga, M.T.

NIK. 793

APPROVAL PAGE

MONITORING SERVER ROOM TEMPERATURE REMOTELY IN
REAL TIME USING RASPBERRY PI AND FIREBASE

BY

ADE GUNTORO PUTRA ANGGORO

L 200 134 015

It has been retained in front of the Board of Examiners
Faculty of Communication and Informatics
muhammadiyah Surakarta university
On 27 October 2017

Board of Examiners:

1. Dr. Ir. Bana Handaga, M.T.
(Ketua Dewan Penguji)
2. Dr. Heru Supriyono, M.Sc.
(Anggota I Dewan Penguji)
3. Aris Rakhmadi, S.T, M.Eng.
(Anggota II Dewan Penguji)

(.....)
(.....)
(.....)

This scientific publication has been accepted as one of the requirements
To earn a Bachelor's Degree

Date 13 Agustus 2018

Knowing,



Dean of Study
Faculty of Communication and Informatics

Nurghiyatna, S.T., M.Sc., Ph. D.
NIK : 881



Program Leader
Informatics

Dr. Heru Supriyono, M.Sc.
NIK:970

DECLARATION

I declare that this scientific publication does not contain works that have been proposed to obtain a degree in college and throughout my knowledge also does not contain work or opinions that ever written or published by another person, except being referred to in the text and mentioned in the bibliography.

If it is proven later that there is untruth in my statement above, I will be fully responsible.

Surakarta, 27 October 2017

Author

A handwritten signature in black ink, appearing to read 'GAP', is shown on a light gray background.

Ade Guntoro Putra Anggoro

L 200 134 015



**UNIVERSITAS MUHAMMADIYAH SURAKARTA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
PROGRAM STUDI INFORMATIKA**

Jl. A Yani Tromol Pos 1 Pabelan Kartasura Telp. (0271)717417, 719483 Fax (0271) 714448
Surakarta 57102 Indonesia. Web: <http://informatika.ums.ac.id>. Email: informatika@ums.ac.id

SURAT KETERANGAN LULUS PLAGIASI

No Surat 030/A.3-IL.3/Inf-FKI/1/2018

Assalamu'alaikum Wr. Wb

Biro Skripsi Program Studi Informatika menerangkan bahwa :

Nama : Ade Guntoro
NIM : **L200134015**
Judul : Monitoring Server Room Temperature Remotely in Realtime
Using Raspberry Pi And Firebase
Program Studi : Informatika
Status : **Lulus**

Adalah benar-benar sudah lulus pengecekan plagiasi dari Naskah Publikasi Skripsi, dengan menggunakan aplikasi Turnitin.

Demikian surat keterangan ini dibuat agar dipergunakan sebagaimana mestinya.

Wassalamu'alaikum Wr. Wb

Surakarta, 24 Oktober 2017

Biro Skripsi Informatika

Ihsan Cahyo Utomo, S.Kom., M.Kom.



UNIVERSITAS MUHAMMADIYAH SURAKARTA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
PROGRAM STUDI INFORMATIKA

Jl. A Yani Tromol Pos 1 Pabelan Kartasura Telp. (0271)717417, 719483 Fax (0271) 714448
Surakarta 57102 Indonesia. Web: <http://informatika.ums.ac.id>. Email: informatika@ums.ac.id

feedback studio MONITORING SERVER ROOM TEMPERATURE REMOTELY IN REAL TIME USING RASPBERRY PI AND FIREBASE -- /0 < 1 of 41 > ?

MONITORING SERVER ROOM TEMPERATURE REMOTELY IN REAL TIME USING RASPBERRY PI AND FIREBASE

Abstract

Temperatur monitoring in real time is one of the challenges in the world of IoT today, plus in monitoring it, it must be through the internet so it can be accessed from anywhere. The main purpose of this research is to be able to read the data generated by the sensor, and in the final step is to display the data generated into a graphical user interface (GUI) in real time with the update data interval for 60 seconds. Temperature sensors are used to sense the temperature changes that occurs around it with a data update interval of approximately 1 second. Raspberry Pi 3 model B is used to read the data changes generated by the temperature sensor, and then send it to Firebase every 60 seconds. Firebase itself is a service from Google which provides online database facility that has real time features. The main function of the GUI is to display data that has been collected in Firebase. The GUI display is website-based so it is capable of being used on all devices that have internet browser features that support javascript. In this system, an email notification is added if temperature has exceeded the specified limit from 18 degrees Celsius to 27 degrees Celsius. The entire system has been tested directly and proven capable of being used to monitor temperature in UMS server room.

Keyword : Firebase, Raspberry Pi, Real Time, Temperature, UMS.

Abstrak

Pemantauan suhu secara real time merupakan salah satu tantangan di dunia IoT hari ini, plus dalam pemantauan, harus melalui internet sehingga bisa diakses dari mana saja.

Match Overview

6%

1	www.safaribooksonline...	1%
2	techieroom.in	1%
3	eprints.ums.ac.id	1%
4	Submitted to National ...	1%
5	Submitted to Manage...	<1%
6	Submitted to Bolton In...	<1%
7	Submitted to SHAPE (V...	<1%

Page: 2 of 15 Word Count: 4677

MONITORING SERVER ROOM TEMPERATURE REMOTELY IN REAL TIME USING RASPBERRY PI AND FIREBASE

Abstract

Temperatur monitoring in real time is one of the challenges in the world of IoT today, plus in monitoring it, it must be through the internet so it can be accessed from anywhere. The main purpose of this research is to be able to read the data generated by the sensor, and in the final step is to display the data generated into a graphical user interface (GUI) in real time with the update data interval for 60 seconds. Temperature sensors are used to sense the temperature changes that occurs around it with a data update interval of approximately 1 second. Raspberry Pi 3 model B is used to read the data changes generated by the temperature sensor, and then send it to Firebase every 60 seconds. Firebase itself is a service from Google which provides online database facility that has real time features. The main function of the GUI is to display data that has been collected in Firebase. The GUI display is website-based so it is capable of being used on all devices that have internet browser features that support javascript. In this system, an email notification is added if temperature has exceeded the specified limit from 18 degrees Celsius to 27 degrees Celsius. The entire system has been tested directly and proven capable of being used to monitor temperature in UMS server room.

Keyword : *Firebase, Raspberry Pi, Real Time, Temperature.*

Abstrak

Pemantauan suhu secara real time merupakan salah satu tantangan di dunia IoT hari ini, plus dalam memantaunya, pasti melalui internet sehingga bisa diakses dari mana saja. Tujuan utama dari penelitian ini adalah untuk dapat membaca data yang dihasilkan oleh sensor, dan pada langkah terakhir adalah menampilkan data yang dihasilkan ke dalam antarmuka pengguna grafis (GUI) secara real time dengan update data interval selama 60 detik. Sensor suhu digunakan untuk merasakan perubahan suhu yang terjadi disekitarnya dengan interval update data sekitar 1 detik. Raspberry Pi 3 model B digunakan untuk membaca perubahan data yang dihasilkan oleh sensor suhu, lalu mengirimkannya ke Firebase setiap 60 detik. Firebase sendiri merupakan layanan dari Google yang menyediakan fasilitas database online yang memiliki fitur real time. Fungsi utama dari GUI adalah untuk menampilkan data yang telah terkumpul di Firebase. Tampilan GUI berbasis situs sehingga bisa digunakan pada semua perangkat yang memiliki fitur browser internet yang mendukung javascript. Dalam sistem ini, pemberitahuan email ditambahkan jika suhu melebihi batas yang ditentukan dari 18 derajat celcius sampai 27 derajat celcius. Seluruh sistem telah teruji secara langsung dan terbukti mampu digunakan untuk memantau suhu di ruang server UMS.

Keyword : *Firebase, Raspberry Pi, Real Time, Suhu.*

1. INTRODUCTION

University of Muhammadiyah Surakarta or commonly also known as UMS is one of the best private universities in Indonesia according to the results of research from QOS University and it's located in Pabelan, established since 1981. Technological developments that occurred in the scope of the University of Muhammadiyah Surakarta developed quite good from year to year. This is to answer the needs of technology that continues to grow at any time. With a total of active students who reach thousands of students, hundreds of employees including lecturers who teach at UMS, this can cause a very crucial problem in itself. Most UMS services, such as lecture schedules, financial payments and student administration, are all processed centrally on the server computer.

As we know, computers in general can generate heat and need to be cooled so as not to interfere with the performance of the computer. Computer sever is able to continue to live for 24 hours to serve most of the services in UMS. Currently UMS has two server rooms, the first server room is under the UMS library, while the second or the last server room is located on the 5th floor of Siti Walidah building. The biggest problem that occurs is in the first server space that is related to cooling. For now cooling on the server room using Air Conditioner or commonly known as AC. Checking UMS server room temperature conditions is usually done manually. This is a consideration in this study to monitor the condition of UMS server room temperature in real time from a distance.

In this study, the author will use a temperature sensor and Raspberry Pi to monitor the temperature of the server room. Temperature sensor is a tool that serve to feel the temperature changes that occur in the surrounding environment, both changes in liquid, solid, even in the form of air. Small changes can not usually be felt directly by humans, it will be difficult for humans to feel, for that temperature sensors are used. Results or data captured by this sensor is the thing will be changed into the form of the display, making it easy to read by humans.

Real time technology is widely utilized in various fields and is very useful in the development of life on Earth. Some of them are the monitor in real time conditions of CO₂ using Arduino and Raspberry (Wiska et al., 2016). In Philippines there is a study of the condition of the Ph land monitors using Raspberry Pi but still using a local server, but not connected to the internet (Flores et al., 2016). Then like we see in hospital, there is a tool that is capable of displaying heart rate condition in real time.

Some of the above research still have some problem, one of which is not yet able to display data in real time via the internet, so that data is still not accessible from anywhere. To fix the issue, researchers will use the Firebase as a storage outputs or results data in real time. The main goal of this research is monitor sensors in real time via the internet, so the results are read to be able to read from anywhere and from all devices. One of the advantages of Firebase is being able to store and display it in real time without having to refresh your website or app pages. The focus of this research is the development of website-based software, because this is why Firebase is chosen as the online database service used.

2. METHOD

In this research will be developed server room temperature monitoring system, there are some things to note. some of which are the types of sensors that will be used, how to read the temperature sensor, and how to make the temperature can be read in real time remotely. This system will be able to measure the temperature between 17 degrees celsius up to 60 degrees celsius. Base on Thermal Guidelines for Data Processing Environments (ASHRAE, 2011), this system will will send a temperature notification when the room temperature has reached the lowest and highest specified limits, minimum is 17 celsius and highest is 27 celsius. Based on the room plan, the server room area is 22 square meters.

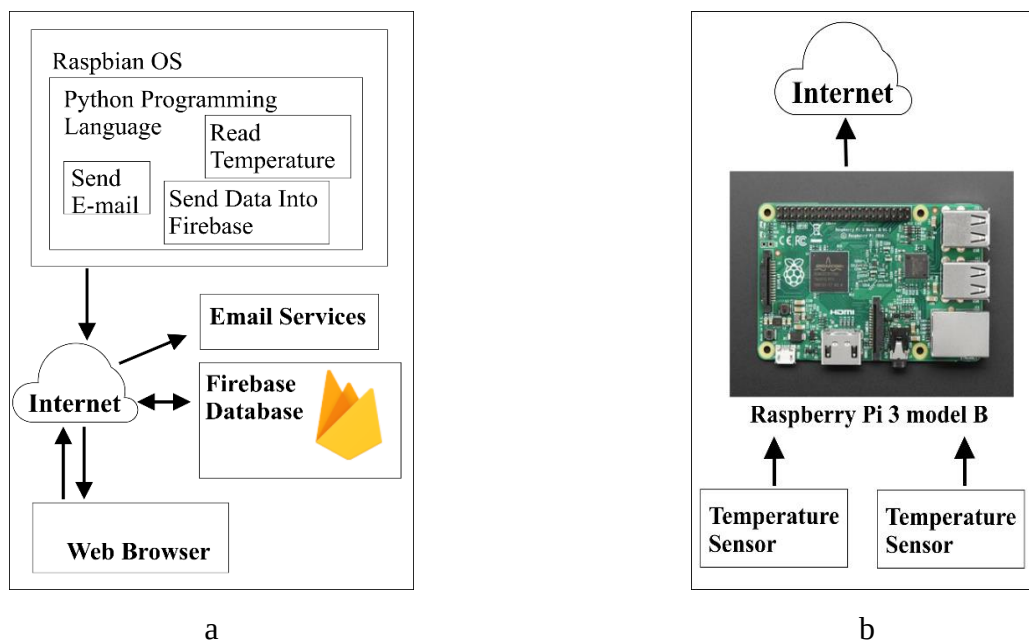


Figure 1. (a) the initial architecture of the software to be developed, (b) the hardware architecture to be used.

The data from the sensor will be stored in a real time database service from Firebase. In the future, this data can be accessed from a web browser that supports Javascript.

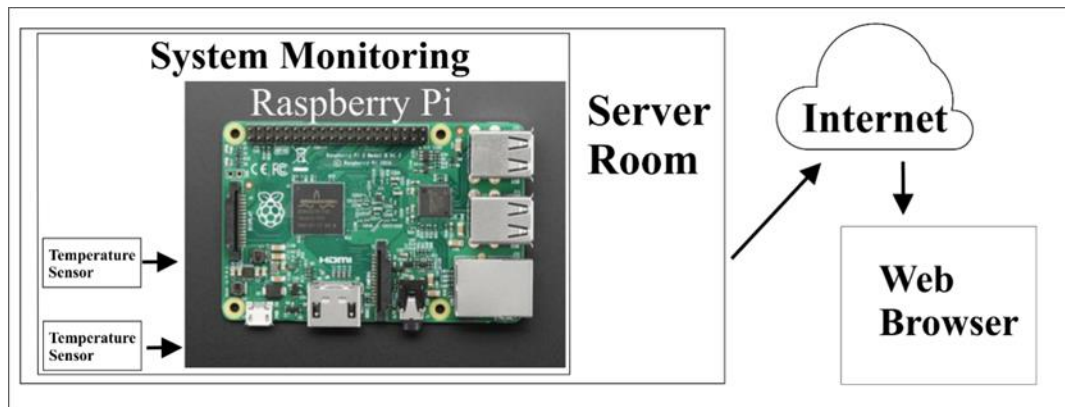


Figure 2. The design of the server room monitoring system in general.

2.1 Hardware Analysis

The main purpose of this research is to monitor the temperature in real time which can be accessed from anywhere using internet connection. It is necessary to have a device that has several connectivity devices such as Ethernet Card or Wi-Fi card so that the device can be connected to the internet. In addition to some types of connectivity, it is also necessary a tool that is able to read sensors, accommodate or collect data, and is able to transmit data from sensors to the internet. In addition, the tool must be able to run some kind of programming language to read the data generated by the sensor.

Looking at the hardware analysis briefly above, then selected Raspberry Pi 3 model B as a tool to be used, this is because Raspberry Pi 3 model B is able to read data from the sensor, and send it to the internet using some type of connectivity. Raspberry Pi 3 is a mini computer, almost the same size as a credit card. Raspberry Pi has fulfilled the necessary requirements, since Raspberry Pi already has an ethernet card with speeds up to 100Mbps, Wifi card, 1GB LPDDR2 RAM and an ARM type processor with 1.3 GHz speed with 4 cores.

The next step is the selection of temperature sensors to be used. For the temperature sensor selected temperature sensor DS18B20, this sensor is very appropriate for use with Raspberry Pi. This sensor already supports one of the features of Raspberry Pi, which is 1-Wire. Simply by enabling the 1-Wire feature in Raspberry Pi, adding a resistor with a value of 4.7K Ohm, and installing a python library then this sensor is able to work like a temperature sensor in general.

The DS18B20 temperature sensor is capable of measuring temperatures from -55°C to 125°C with an accuracy level of approximately 0.5°C at a temperature of -10 to 85°C

a. Connect Raspberry Pi with temperature sensor

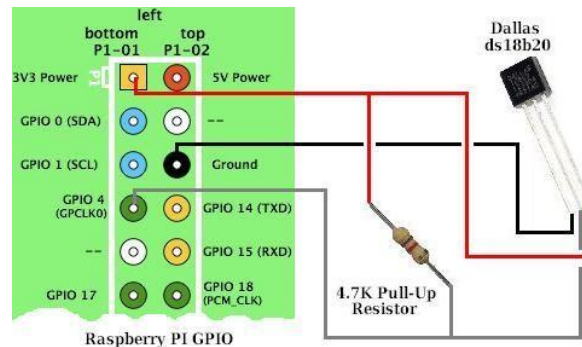


Figure 3. DS18B20 temperature sensor installation configuration on Raspberry Pi 3 model B.

DS18B20 sensor has 3 pins and requires only 3 pieces of cable. Pin VDD functions as a power source required by the sensor, VDD will be connected to 3.3V power pin number 1 which is in Raspberry Pi. Pin DQ as input / output data on 1-Wire, DQ will be connected to GPIO4 pin residing in Raspberry Pi. Next is the GND pin or commonly called ground will be connected at the ground pin number 6. While the 4.7K Ohm resistor is connected with VDD and DQ. The picture below is the configuration pins found on DS18B20 and Raspberry Pi.

2.2 Software Analysis

As mentioned above, Raspberry Pi 3 model B is a mini computer. Just like a computer in general, to run the function as a computer then needed an OS (operating system) to operate it, Raspbian Jessie chosen desktop version. Raspbian Jessie desktop version is another variant of the operating system developed specifically for Raspberry Pi. Another version of Raspbian Jessie desktop version is Raspbian Jessie Lite.

Once the OS is installed, the next step is to choose the programming language that will be used in Raspberry Pi. Python is chosen because by default Python is a programming language supported by Raspberry Pi. Python itself has several advantages, one of which is the existence of a module that can be used to read various types of sensors, one of which is “w1thermsonor”. “w1thermsonor” is a library developed specifically for reading DS18B20 temperature sensors.

To save the database, this research selects Firebase services. because Firebase provides some much needed features in this research. Some of them are Firebase auth, fiebase database,

and Firebase hosting. a connector is required that serves to transmit data from Raspberry Pi to Firebase, the connector is Pyrebase. Pyrebase is a library capable of manipulating data from Raspberry Pi to Firebase using the Python programming language. Pyrebase has the ability to save new data, edit, or even delete existing data.

In the development of the server room temperature monitor system, there are two software to be created. The first is software that functions to read the temperature and send data from Raspberry Pi to the Firebase database using the Python programming language. The second software is a web-based user interface that uses HTML and Javascript programming languages. Selection of software development using HTML and Javascript, because the two programming languages are easy to understand and the results can be displayed on devices that have Internet browsing services. While Javascript itself is mandatory to use if you want to use concurrently with HTML, because only with Javascript is the database capable to display in web-based services

a. Firebase

Firebase is a mobile and web application platform with tools and infrastructure designed to help developers build high-quality apps. Firebase is made up of complementary features that developers can mix-and-match to fit what their needs. Firebase has a unique database structure that is different from other databases. The data stored in the database is stored in a JSON object (Cheng, 2017). The data is accessible from all the platforms (Singh, 2016). Firebase provides several services that will be used in this research, including Firebase Realtime Database, Firebase Auth, and Firebase Hosting. With Firebase Realtime database, data is synchronized between different clients and with the backend server almost instantaneously. This implies that if the underlying data is changed by one of the clients, it gets streamed in real time to every connected client; hence, all the other clients automatically get updates with the newest set of data (Waikar, 2015).

To perform integration stages between Firebase and python and web-based programs, researchers first create a project within Firebase. This project can be built into Firebase Console, or import from projects already available in Google Developer. After the project is created, the next step is to integrate and initialize Firebase with the application. Once the project has been created, we will be given three choices, integration with iOS, Android, or Website. Researchers choose the integration with the website, because it is easy to

understand and can be used almost in all devices that provide web browser services. next is the integration with Python that serves to transmit data to Firebase.

b. Reading temperature using Python programming language

The next step is to read the data generated by the sensor using “w1thermsensor”. To read data using “w1thermsensor” can by reading the commands that have been provided.

```
1 from w1thermsensor import W1ThermSensor
2 sensor = W1ThermSensor(W1ThermSensor.THERM_SENSOR_DS18B20, "0116252dc0ee")
3 hasil = sensor.get_temperature(unit=1)
4 print hasil
```

Figure 4. The code used to read the temperature sensor using “w1thermsensor”.

This is a brief explanation of the existing code in figure number 4. Line number 1 serves to import modules from “w1thermsensor”. Line number 2 is reading the temperature sensor with id "0116252dc0ee" stored in a variable with the name "sensor". Line number 3 is to read the temperature generated by the "sensor" which is subsequently converted into units of celsius with "(unit = 1)" is the modifier. While the 4th row is to display the results into the Python Shell or terminal that is on Raspberry Pi.

```
Python 2.7.9 (default, Sep 17 2016, 20:26:04)
[GCC 4.9.2] on linux2
Type "copyright", "credits" or "license()" for more :
>>> ===== RESTART =====
>>>
28.75
```

Figure 5. Output when the code is executed in Python.

c. Sending data from Raspberry Pi to Firebase.

When the data generated from the sensor is capable of being read, the next step is to send data to Firebase. All data sent to Firebase will be placed in the same Child, ie "sensor". To avoid duplication of data, Push method will be used which will generate Unique ID. Every 60 seconds data will be sent by Firebase using the Sleep function available in Python. Before submitting data to Firebase, however, the application must be pre-installed with Firebase before it is used.

This is brief explanation on Figure 6. The first line serves to import "pyrebase". The API key contained in line 4 is the key that allows Python to access the database. The 5th line is the Firebase Hosting address. The 6th line is the URL address of the database, can be used to access the database in the form of JSON directly from the web browser. The 8th row is the

Service Account, only with the Inti Pyrebase Service Account capable of performing activities such as updating data, deleting data, and creating new data in Firebase. line 10 and 11 function to initialize Firebase, and especially on line 11 serves to call the database feature.

```
1 import pyrebase
2
3 ▾ config = {
4     "apiKey"      : "AIzaSyAd6c3KPZA4iG069kmMwxRUHb40ZhjZhkQ",
5     "authDomain"  : "skripsifix-69e9a.firebaseio.com",
6     "databaseURL" : "https://skripsifix-69e9a.firebaseio.com",
7     "storageBucket" : "skripsifix-69e9a.appspot.com",
8     "serviceAccount" : "skripsifix-6d0bb45895e2.json"
9 }
10 firebase = pyrebase.initialize_app(config)
11 db = firebase.database()
```

Figure 6. Integration and initialization Firebase with Python language.

After the integration and Firebase initialization is complete, the next step is to send data from Raspberry Pi to Firebase. The following image will show the data will be sent to Firebase.

```
--
19 ▾ while True:
20     sensor1 = W1ThermSensor(W1ThermSensor.THERM_SENSOR_DS18B20, "021621d197ee")
21     hasil1 = sensor1.get_temperature(unit=1)
22     now = datetime.datetime.now()
23     locale.setlocale(locale.LC_TIME, "id_ID")
24     date = now.strftime("%d/%m/%Y")
25     clock = now.strftime("%H:%M:%S")
26     data2 = {"temp": hasil1, "date": date, "time": clock}
27     db.child("sensor").push(data2)
28     sleep(60)
```

Figure 7. Code for upload data from Raspberry Pi to Firebase. On line 27, Line 26 and 27 is code that serves to upload data from Raspberry Pi to Firebase.

2.3 Reading data from Firebase and display it into Web

The example view of the website used by the client to monitor in real-time, made with as simple as possible before doing development with Bootstrap.

a. Reading data from Firebase database

To read data from Firebase to html, the first thing to do is integrate with html and do feature calling using javascript like the following figure 8. After initialize, the next stage is to create a selector that serves to display data in HTML. Then the selector will be called in javascript to display data from Firebase. Before displaying data from Firebase, the researcher determines which child or database location will be selected to display the data just like in figure 8.

```
const dbRef      = firebase.database().ref('sensor1');
|
```

Figure 8. Sensor1 is the child that contains the data that will be displayed in HTML

```
5 <script src="https://www.gstatic.com/firebasejs/4.1.5/firebase.js"></script>
6 <script>
7 var config = {
8   apiKey: "AIzaSyAd6c3KPZA4iG069kmMWXRuHb40ZhjZhkQ",
9   authDomain: "skripsifix-69e9a.firebaseio.com",
10  databaseURL: "https://skripsifix-69e9a.firebaseio.com",
11  projectId: "skripsifix-69e9a",
12  storageBucket: "skripsifix-69e9a.appspot.com",
13  messagingSenderId: "629214868537"
14 };
15 firebase.initializeApp(config);
16 </script>
```

Figure 9 Integration and initialization Firebase with HTML and Javascript.

After selecting Child database, next is to call the selector in javascript which function to display data to HTML.

```
const firesuhu    = document.getElementById('suhu');
const firejam     = document.getElementById('nama');
const firetanggal = document.getElementById('objek');
```

Figure 10. Selecting a selector from HTML.

Next is to display data from Firebase in accordance with the instructions on the Firebase documentation page.

```
43 dbref.limitToLast(1).on('child_added', function(snap){
44   var data = snap.val();
45   firesuhu = data.temp;
46   firetanggal = data.date;
47   firejam = data.time;
48 });
```

Figure 11. Displaying Value from Firebase to HTML via the selected selector.

On line 43 "dbref" is called to read the data in Child. "LimitToLast (1)" serves to limit the amount of data to be read that is 1 and is most recent. While "child_added" is reading the newly added data and then stored temporarily in the "snap" function. The 44th row creates a variable containing data from "snap" which will be called on the 45th - 47th line. ".temp", ".date", ".time" is the object that contains the data. The following figure is a selector located in HTML that serves to display data. "Id = temperature", "id = date", "id =

hour" is an id or a selector that serves to display javascript data to HTML. And in the figure 13 is the result of the development.

```
20 <div style="clear: both">
21   <h2 style="float: left">Suhu :&nbsp;&nbsp;&nbsp;</h2>
22   <h2 style="float: left" id="suhu"></h2>
23 </div>
24 <div style="clear: both">
25   <h2 style="float: left">Tanggal :&nbsp;&nbsp;&nbsp;</h2>
26   <h2 style="float: left" id="tanggal"></h2>
27 </div>
28 <div style="clear: both">
29   <h2 style="float: left">Jam :&nbsp;&nbsp;&nbsp;</h2>
30   <h2 style="float: left" id="jam"></h2>
31 </div>
```

Figure 12. id is a selector contained in HTML.

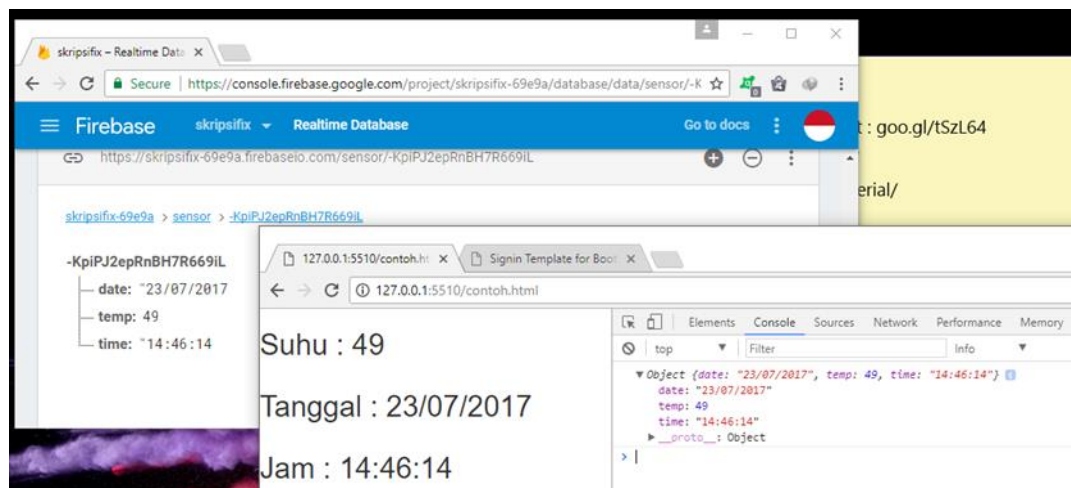


Figure 13. Sample view from Firebase and console to prove that the data has been entered.

b. Displays server room temperature history in realtime

To display data from Firebase database into chart form, ChartJS is selected. As for displaying data as a whole in the form of tables used Datatable. To use ChartJS, data from Firebase is first converted into array form. The following example is the code to convert the "temp" data into array form which can then be applied to other data.

```
function suhuToArray(snapshot) {
  const returnArr = [];
  snapshot.forEach(function(childSnapshot) {
    const item = childSnapshot.val().temp;
    returnArr.push(item);
  });
  return returnArr;
};
```

Figure 14. Changing object into array in javascript.

Once the data has changed into an array, then enter the data into ChartJs as shown figure 15. In Firebase documentation, “limitToLast (10)” serves to limit the amount of data to be displayed from Firebase by 10 data. 10 data is the data just entered, because by default Firebase will add new data entered under existing data or at the back of the order.

```
firebase.database().ref('sensor').limitToLast(10).on('value', function(snapshot) {
  const suhu = suhuToArray(snapshot);
  const jam = jamToArray(snapshot);
  const tanggal = tanggalToArray(snapshot);
  var ctx = document.getElementById("canvas").getContext("2d");
  var mychart = new Chart(ctx,{
    type: 'bar',
    data: {
      labels: jam,
      datasets: [{
        label: "Suhu",
        backgroundColor: "rgba(220,220,220,1)",
        fill: false,
        data: suhu
      }]
    }
  });
});
```

Figure 15. Entering data from array into CharJS.



Figure 16. ChartJs in Line and Bar form with data from Firebase.

3. RESULTS AND DISCUSSION

3.1 Results

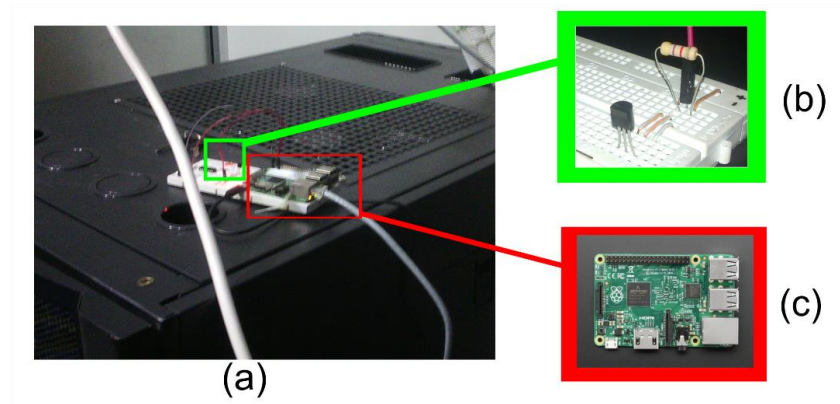


Figure 17. (a) The monitoring system placed in the server room above a box. (b) Temperature sensors and resistors used to monitor server room temperature. (c) Raspberry Pi 3 B model used to monitor the temperature of the server room connected with the LAN cable.

From the results of research that has been done, the device model that uses still looks very simple, because it is only used in a closed room or indoor so it's no need to use other additional protector like a case or protector for Raspberry Pi, temperature sensors and some other devices are tied onto the Project Board with Cable Tie. For website pages, two different pages will be created with two different functions, login page and dashboard. To ensure that the model has been made to function properly, the model will be placed in a room for an hour or more to ensure that the model is working properly. Pyrebase will only send data with Push motode, while Chartjs will display data with graphics will change every new data entered.

3.2 Testing

Some of the aspects tested in this testing phase are as follows :

a. Testing network connection

Testing the network connection is intended to Raspberry Pi get bandwidth that is capable to be used in sending data to Firebase and send email. Testing is done using an access point and LAN router connected to the internet in a space. The results of internet connection testing can be seen as in table 1. From the results of tests performed, Raspberry Pi is able to send data to Firebase and email without your interference means. It is recommended to use speeds of more than 1 Mbps so as not to interfere with data transmission.

Table 1. Testing the network connection with various settings.

Number	Distance	Network Interface	Speed	Status
1	2 Meters	LAN cable	100 Mbps	Good
2	6 Meters	Wi-Fi	1 Mbps	Good
3	8 Meters	Wi-Fi	1-3Mbps	Good
4	10 Meters	Wi-Fi	1 Mbps	Good

From the results of the tests performed, Raspberry Pi is able to send data to Firebase and send e-mail without any significant interruption. It is strongly recommended to use internet speed more than 1 Mbps so that there is no interference in data transmission, and distance between access point and raspberry pi not more than 10 meters.

b. Email testing

Notification tests are performed using simulated data by using values that exceed the room temperature. Email is sent via Raspberry Pi, with 3 recipients at the same time.

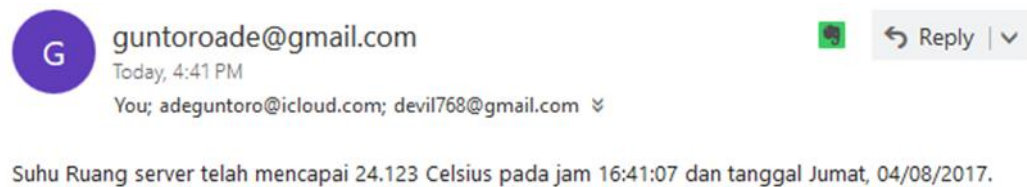


Figure 18. Email sent from Raspberry Pi when the temperature has exceeded the specified temperature.

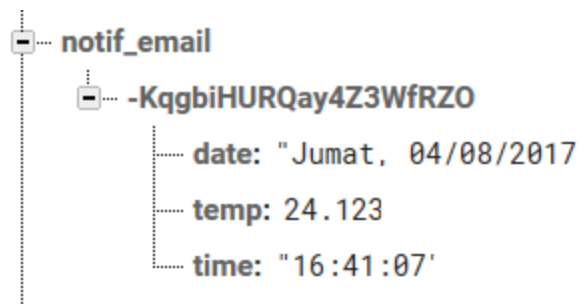


Figure 19. Temperature data exceeding the limit on Firebase

Figure 18 is an email sent when the temperature has exceeded the specified limit, while figure 19 is the temperature data in Firebase. Clock description of sent emails and data recorded in Firebase, almost no pause as it was sent on the hour of 16, minute 41. Likewise

with the contents of the email, no pause occurred, because the temperature exceeded the limit at 16:41:07

c. Sensor calibration and testing

Temperature calibration is used because each sensor has a low level of accuracy or is incompatible with the measuring instrument. The function of calibrating the temperature on the sensor is to get a temperature close to the temperature present in the thermomete. Using the quadratic interpolation function in Excel, a formula can be generated as follows:

$$y = -0,0016x^2 + 1,0021x + 1,3636$$

In the next stage is to do a comparison with Linear Interpolation with the formula also obtained from Excel using the same data, and the result of the formula is like this :

$$y = 0.9451x + 1.8273$$

After the 2 formulas were found, then the next is to compare the results. Linear interpolation formula produces a small error value, that's why the linear interpolation formula is used in the temperature measurement. Variable 'x' is the temperature value generated by the sensor and stored in the variable 'y' which can then be used to display the calibrated temperature using the above formula. Calibration and testing of sensors using additional cable along 1 meter and the thermometer used is the alcohol room thermometer.

d. Testing system in UMS server room

In the testing phase, the model will be placed in server room and will be left for several hours to ensure that the model is working properly. Models will be connected with the internet through LAN cable. The python program for reading and transmitting data from the sensor will run automatically since Raspberry Pi booting. Setting the number of bandwidth is also required to ensure that Raspberry Pi gets enough bandwidth to transmit data. To ensure that the data generated by the temperature sensor is sent, we must first login to the Firebase Console using an account from Google. If any data goes to Firebase database, then root child will change color to yellow and back again change color to black if no data entered.

The next stage is to monitor the temperature of the room through an application that has been uploaded to Firebase Hosting. Do not forget the user must first to login using Firebase Auth feature using email and password made first in Firebase Console. After login, the user will be directed directly to the dashboard view to monitor the temperature data sent to

Firebase. the only way to read data from Firebase using a web application that has been created is to login, because the default Rules contained in Firebase only allow registered users to read the data. If the user is willing to get data and used for offline, the client can import data into JSON via Firebase Console.

Users will get notified via email if the temperature has exceeded the normal limit or limit that has been determined. And Research add sound like the alarm sound will come out if the user monitor the temperature through the web browser application.

4. CONCLUSIONS

After going through several stages such as design, developing and testing, can be drawn conclusions with some points like the following : Raspberry Pi 3 B model can be used to send data to Firebase and email notification by using LAN cable owned by UMS with 100 Mbps bandwidth. Raspberry Pi 3 B model can also send data to Firebase and email notification using wifi connection with a maximum speed of 1 Mbps. There is a temperature difference ranging from 1C to 2C. The length of the cable used by the temperature sensor affects the temperature sensor test results. The connection speed required by Raspberry Pi to transmit data to Firebase is quite varied, preferably the speed of connection using a LAN or wireless cable above 1 Mbps with a distance of not more than 10 meters.

References

- ASHRAE (American Society of Heating, Refrigeration and Air-Conditioning), guideline, third edition, 2011, Technical Committee 9.9 (TC 9.9).
- Cheng, F. (2017). Build Mobile Apps With Ionic 2 and Firebase Hybrid Mobile App Development. doi:10.1007/978-1-4842-2737-4.
- Childs-Maidment, J. (2017, January 07). Pyrebase. Retrieved August 09, 2017, from <https://github.com/thisbejim/Pyrebase>.
- DS18B20 Programmable Resolution 1-Wire Digital Thermometer - Maxim. (n.d.). Retrieved August 09, 2017, from <https://www.maximintegrated.com/en/products/analog/sensors-and-sensor-interface>.
- Flores, K. O., Butaslac, I. M., Gonzales, J. E., Dumlao, S. M., & Reyes, R. S. (2016). Precision agriculture monitoring system using wireless sensor network and Raspberry Pi local server. *2016 IEEE Region 10 Conference (TENCON)*. doi: 10.1109/tencon.2016.7848600.
- Installation & Setup in JavaScript | Firebase. (n.d.). Retrieved August 09, 2017, from <https://firebase.google.com/docs/database/web/start>.

- Singh, N. (2016). Study of Google Firebase API for Android. *International Journal of Innovative Research in Computer and Communication Engineering*, 4(9). doi: 10.15680/IJIRCCE.2016.0409133.
- Waikar, M. (2015). *Data-oriented Development with AngularJS*. Birmingham, UK: Data-oriented Development with AngularJS.
- Wiska, R., Habibie, N., Wibisono, A., Nugroho, W. S., & Mursanto, P. (2016). Big sensor generated data streaming using Kafka and Impala for data storage in Wireless Sensor Network for CO2 monitoring. *2016 International Workshop on Big Data and Information Security (IWBIS)*. doi:10.1109/iwbis.2016.7872896.