IOT – BASED DEVICE FOR REAL TIME SERVER ROOM ENVIRONMENT MONITORING SYSTEM

IRFAN BIN IZAD

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

IOT – BASED DEVICE FOR REAL TIME SERVER ROOM ENVIRONMENT MONITORING SYSTEM

IRFAN BIN IZAD

This report is submitted in partial fulfillment of the requirements for the Bachelor of Computer Science (Computer Networking) with Honours.

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2021

DECLARATION

I hereby declare that this project report entitled

IOT – BASED DEVICE FOR REAL TIME SERVER ROOM ENVIRONMENT MONITORING SYSTEM

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT : Irfan Izad Date : 1 SEPTEMBER 2021

(IRFAN BIN IZAD)

I hereby declare that I have read this project report and found this project report is sufficient in term of the scope and quality for the award of Bachelor of Computer Science (Computer Networking) with Honours.

SUPERVISOR : _____ Date : 4/9/2021

(ASSOC PROF TS. DR ROBIAH YUSOF)

DEDICATION

I would like to dedicate this thesis to my beloved parents who have always give support and motivation to me unconditionally, my friends who give courage and help to me whenever I stumble into problem and lastly the highest appreciation goes to my supervisor, Assoc Prof Ts. Dr. Robiah Yusof for her patience and guidance in this thesis.

ACKNOWLEDGEMENTS

I would like to thank Assoc Prof Ts. Dr. Robiah Yusof for giving assistant to complete this project successfully. I highly appreciate her patience in guiding me throughout this thesis.

I would also like to thank my beloved parents who have been giving me support and motivation throughout my project. Their endless dedication gives me a lot of strength so that I could complete this project successfully.

ABSTRACT

A server room is a room used to store, power, and operate computer servers and their associated components. This room is part of a data center, which typically houses several physical servers lined up together in different form factors, such as rack-mounted, or in tower or blade enclosures. A data center might consist of several server rooms, each of which is used for separate applications and services. Server room can also be found in the campus of Universiti Teknikal Malaysia Melaka. As we all know, server rooms tend to be hot due to the performances and the usage of the devices that is stored it in. In order to maintain the temperature and humidity of the server room, an admin is assigned to the server room to monitor the condition of the server room and also the performances of the device that is used in it. A server room monitoring system can be used to monitor the humidity and temperature in the server room. In addition, if the temperature in the server room rises, it responds by lowering the temperature. This study makes use of the Internet of Things (IoT), which is a gadget that is employed by the Raspberry Pi and Pocket Lab wireless sensor to read the room's temperature and humidity. Temperature and humidity log data is saved on a cloud server and displayed in real time in a chart diagram. The IoT reaction is based on established temperature norms, and users are notified via the Telegram application on their mobile devices. By using this monitoring system, it will be easier for admins to monitor the server room without actually need to physically check the server room regularly. The solution that is proposed in this project can be useful for present time and in the future as it has added feature. The added feature that is proposed in this project is that we will set a certain threshold on the variables that is affecting the environment of the server room. Also, we proposed a monitoring system that will monitor the server room within the working hour with real time monitoring system and notifications alert. This may be proven useful from using this system as it has high usability, easy to access, and convenient to the people who are in charge of the server room.

ABSTRAK

Bilik Server adalah ruang yang digunakan untuk menyimpan, menggerakkan, dan mengendalikan server komputer dan komponennya yang berkaitan. Ruangan ini adalah bahagian dari pusat data, yang biasanya menempatkan beberapa server fizikal yang berbaris bersama dalam pelbagai faktor bentuk, seperti dipasang di rak, atau di kandang menara atau bilah. Pusat data mungkin terdiri dari beberapa ruangan bilik server, yang masing-masing digunakan untuk aplikasi dan perkhidmatan yang terpisah. Ruang server juga boleh didapati di kampus Universiti Teknikal Malaysia Melaka. Seperti yang kita semua ketahui, bilik server cenderung panas kerana persembahan dan penggunaan peranti yang disimpan di dalamnya. Untuk menjaga suhu dan kelembapan bilik server, seorang pentadbir ditugaskan ke ruang server untuk memantau keadaan bilik server dan juga prestasi peranti yang digunakan di dalamnya. Sistem pemantauan bilik server dapat digunakan untuk memantau kelembapan dan suhu di ruangan server. Selain itu, jika suhu di ruangan server meningkat, ia bertindak balas dengan menurunkan suhu. Kajian ini menggunakan Internet of Things (IoT), yang merupakan alat yang digunakan oleh sensor tanpa wayar Raspberry Pi dan Pocket Lab untuk membaca suhu dan kelembapan bilik. Data log suhu dan kelembapan disimpan di server awangan dan dipaparkan dalam masa nyata dalam gambarajah carta. Reaksi IoT didasarkan pada norma suhu yang telah ditetapkan, dan pengguna diberitahu melalui aplikasi Telegram pada peranti mudah alih mereka. Dengan menggunakan sistem pemantauan ini, lebih mudah bagi pentadbir untuk memantau bilik server tanpa perlu memeriksa bilik server secara berkala. Penyelesaian yang dicadangkan dalam projek ini boleh berguna untuk masa sekarang dan masa depan kerana ia mempunyai ciri tambahan. Ciri tambahan yang diusulkan dalam proyek ini adalah kita akan menetapkan ambang tertentu pada pemboleh ubah yang mempengaruhi lingkungan ruangan server. Kami juga mencadangkan sistem pemantauan yang akan memantau ruangan server dalam waktu bekerja dengan sistem pemantauan masa nyata dan pemberitahuan pemberitahuan. Ini mungkin terbukti berguna daripada menggunakan sistem ini kerana ia mempunyai kegunaan yang tinggi, mudah diakses dan senang digunakan oleh orang-orang yang menjaga ruangan server.

Table Of Content

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

FYP - **Final Year Project**

CHAPTER 1

INTRODUCTION

1.0 Introduction

The server room is a room, generally equipped with air conditioning, dedicated to the continuous operation of computer servers. However, it is necessary to always monitor the health of the server to ensure hardware security. We will solve the background of the project, which is the best condition for the server room. Research questions will be expressed as three research questions. Based on the research questions, it will clearly show the server room issues that users are currently facing. After that, research goals can be generated, and all the work of the project will be based on the research goals.

First, let's take a look at the background of the project in order to run this project. Humidity and temperature history is explained. Second, it summarizes the research questions. Then the research questions and goals are carried out. The scope and contribution of the project will also be discussed. Finally, we describe the organization of the report to ensure that the project runs in the correct process.

1.1 Project Background

A server room is a room used to store, power, and operate computer servers and their associated components. This room is part of a data center, which typically houses several physical servers lined up together in different form factors, such as rack-mounted, or in tower or blade enclosures. A data center might consist of several server rooms, each of which is used for separate applications and services. The server room provides the needed lodging for computer network infrastructures that support data transfer and storage. (Onibonoje, 2019)

Nowadays, a server room can be found from business companies to the campus of a university. As the core of the campus network, server rooms on the campus are equipped with complex and expensive equipment which are sensitive to the external environment (Chen, 2012). The server room temperature and humidity must be in the control range which is 10°C to 25°C (temperature) and 40% to 60 % (humidity). (Mohd Hakimi Zohari, 2019). In order to maintain the quality of the conditions for the equipment in the server room, they need to

monitor to make sure no equipment or hardware is overheating or too cold. One way that can be used to control and monitor the temperature and humidity of the server room is to use an Internet of Thing (IoT) based system. (Purwanto, 2018)

The Internet of Things (IoT) refers to a system of interrelated, internet-connected objects that are able to collect and transfer data over a wireless network without human intervention. With the help of IoT, all the physical objects can connect to the internet (Roy, 2017). To keep the server running properly, the temperature and humidity in the server room need to be considered because if the temperature or humidity in the server room is not suitable with standards it will disturb or damage the existing server. (T H Nasution, 2018).

Recently, with increasing living standards and expectations for comfortableness, the use of residential air conditioning was becoming widespread. The control and monitoring of indoor atmosphere conditions represent an important task with the aim of ensuring suitable working and living spaces for people. (Chung, 2006). This can be also applied to the server room as we need to maintain that certain temperature for that room. Many factors make the server air unfavorable for air circulation, such as the influence of temperature. Too high temperature will cause hardware damage, and too low temperature will consume a lot of power and waste power. Another factor is the air humidity. Too high humidity will damage the server hardware, while too low humidity will cause static electricity in the server room. (Moechammad Alvan Prastoyo Utomo, 2019). Moreover, according to (Rafizah Ab Rahman, 2019), humidity, relative to the temperature, will increase if the temperature is too cold causing corrosion on the equipment or condensation that can damage the equipment. Inversely, if the temperature is too high, decreasing the humidity leads to static build-up that can also damage the equipment. According to (Dwi Ely Kurniawan, 2019) the server room condition must always be aware of the temperature and humidity.

1.2 Problem Statement

An organization that requires a server room needs to maintain a room condition that does not compromise the hardware and equipment performance. Detecting the changes within the room must be done without outside interference, or it could contribute to the detriment of the servers. As such, the absence of such a system is a problem for organizations. The Research Problem (RP) is summarized in Table 1.1

Table 1. 1 Summary of Problem Statement

No	Research Problem
RP1	The low experience on how to maintain the quality of the conditions for the equipment in the server room as it must always be protected from humidity and temperature.

Thus, Research Questions (RQ) which are depicted in Table 1.2 is constructed to identify the research problem as discussed in the previous section.

Table 1. 2 Summary of Research Question

No	Research Question
RQ1	What is the optimal range of humidity and temperature for server room?
RQ2	How are we going to capture the data of the server room condition?
RQ3	How to validate the accuracy results of data and information that is received from the system?

RQ1: What is the optimal range of humidity and temperature for server room?

This research question is used to study about the optimal server room condition.

RQ2: How are we going to capture the data of the server room condition?

This research question is formulated by considering the method in how does the system can capture the data about the server room condition. Thus, it is important to know how to apply the system in a server room.

RQ3: How to validate the accuracy results of data and information that is received from the system?

This research question is to know how the comparison the accuracy result of the system.

1.3 Objective

Table 1. 3 Summary of Research Objective

RP	RQ	RO	Research Objective
RP1	RQ1	RO1	To study humidity and temperature in a server room.
	RQ2	RO2	To develop a system that can monitor the humidity and temperature of a server room by alert notification and statistical information.
	RQ3	RQ3	To provide accurate system for monitoring humidity and temperature of server room.

RO1: To study humidity and temperature in a server room

Firstly, we need to investigate what is humidity and temperature and how it can affect the hardware and equipment in a server room.

RO2: To develop a system that can monitor the humidity and temperature of a server room by alert notification and statistical information.

We discuss about the system that can monitor the optimal condition for a server room.

RO3: To provide accurate system for monitoring humidity and temperature of server room

After discussing the applied method for the system, we need to provide an alert notification of the statistical information about the discussed server room.

1.4 Scope

Scope of project is going to the conducted as follow:

1. The sample dataset from the system will be used in this project.
   - A dataset is obtained from the sensors of the system.
   - The sensors will capture the data of the server room and send it to the system
   - The system modifies the data into a graph and will send it to the user
   - The user obtains the data and compare it with the optimum condition of the server room

2. This project just focuses the methods used by the system to capture the information in a server room and also send notifications to the user about the conditions of the server room.
3. This project used sensors to get the information of the server room.

1.5 Project Contribution

The contribution of this project is summarized in table 1.4

Table 1. 4 Summary of project contributions

RP	RQ	RO	RC	Research Contribution
RP1	RQ1	RO1	RC1	The range of humidity and temperature for device and server room.
	RQ2	RO2	RC2	The methods to capture the data of the server room monitoring system.
	RQ3	RO3	RC3	The accurate server room monitoring system with alert notification and statical info

Table 1.4 shows the project contribution based on the research problem, research question and research objective.

1.6 Report Organization

Chapter 1: Introduction

This chapter will discuss the introduction, the background of the project, the problem statement, the research question, the objective of the research, the scope, the contribution of the project and the organization of the report.

Chapter 2: Literature Review

This chapter will study related work such as the optimal state of the server room, the importance of humidity and temperature in the server room, and the methods that will be used in this project. Related work will be contributed to the next chapter.

Chapter 3: Methodology

This chapter explains the method used for humidity and temperature analysis and the methodology of the project. The project schedule and milestones will be discussed in this chapter to complete the project on time.

Chapter 4: Analysis and Design

This chapter introduces problem analysis, software and hardware requirements. Experimental design such as logical design and physical design will also be conducted in this chapter. The environment will be configured as required.

Chapter 5: Implementation and Testing

This chapter will describe the data implementation method to implement the proof-support graph.

Chapter 6: Testing

This chapter will describe the test result and analysis about this project.

Chapter 7: Conclusion

This chapter will summarize all chapters in a concluding manner. Summary, limitations and future work of the project will be discussed.

1.7 Summary

In conclusion, this project aims to create a system capable of detecting the humidity and temperature of the server room for optimal performance. The problem statement, objective and scope of the research in this chapter is to define and discuss the problem statement, as well as to create a possible solution. A report organizer is also made in this chapter to show the sequence of the project.

CHAPTER 2

LITERATURE REVIEW

2.0 Introduction

This chapter will study about the related works such as the server room environment monitoring system and the IoT-based device environment monitoring system that will be used in this project.

2.1 Framework

Figure 2.1 shows the structure framework for this chapter. This framework is used to follow what will be done in this chapter. Other than that, this framework also very useful in this chapter to know what is contained in the title.

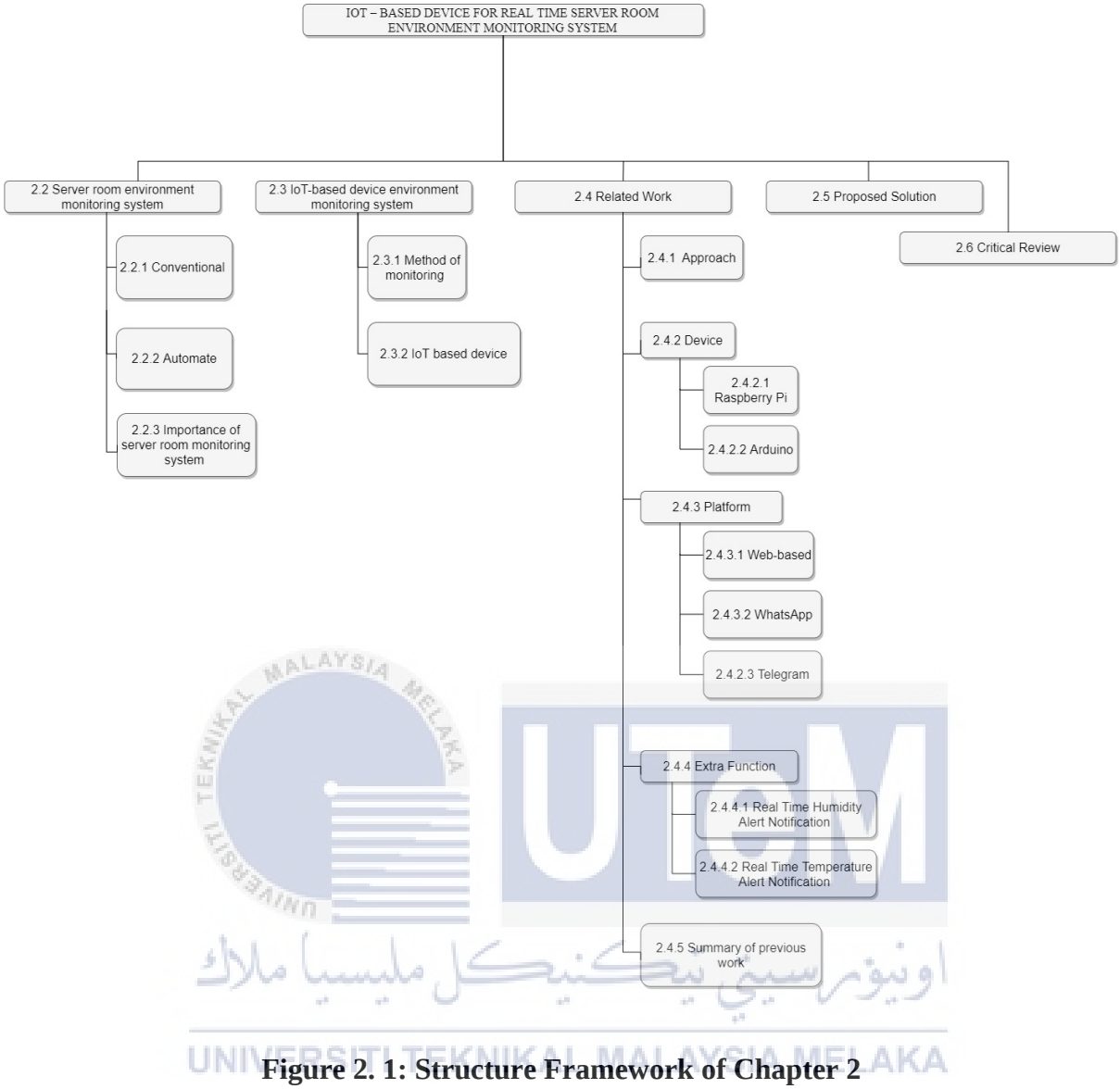

Figure 2. 1: Structure Framework of Chapter 2

2.2 Server Room Environment Monitoring System

This section we will discuss what is the server room and how does one take care of the server room. Server room plays a vital role in every company and universities. Monitoring and managing the environmental conditions-accurately will maximize the use of the equipment, extend its useful life, minimize operating cost and avoid expensive affair caused by downtime or unexpected equipment breakdown. Conventional and automate are one the monitoring system of the server room. Those two factors will be what we only focus on.

2.2.1 Conventional

According to (Bernard Loh, 2019), Servers are delicate machines that need precise temperature and humidity control in order to operate at their best and avoid errors or malfunction. As a result, temperature and humidity are the most important environmental variables for most IT professionals.

Since servers generate a lot of heat, the room has several air conditioning systems manually installed to keep the temperature between 18 and 27 degrees Celsius. When a large number of air-conditioning systems are mounted, a malfunction can occur at any time, resulting in a cooling system overload. A temperature sensor should be placed near each air conditioning unit to detect any problems.

These are the conventional way for a company to keep their humidity and temperature for the server room. There are no monitoring systems installed so they just manually check the room on a daily basis to keep in check. This may be a cost-effective way for the company to save cost but it is not an efficient as the technician in charge need to regularly and manually check the server room.

2.2.2 Automate

According to (Bernard Loh, 2019) also, having an automated monitoring system can be helpful as it can prevent any disastrous occurrence that might happen in the server room such as risk of data and financial loss and equipment malfunction due to not monitoring the room at all times. Monitoring a server room 24/7 is a must, as it contains crucial assets. Things can go wrong when the personnel in charge get busy or needed to leave for a few minutes or hours. An overnight downtime of the equipment can mean thousands of losses. Using a monitoring system, we can monitor the server room remotely even if we are not near it.

Having an automate monitoring system will provide a real-time alert system as it will monitor the server room and will send it through SMS and email. In just 2-3 seconds, we will be able

to receive alarm updates when something happened to one of the equipment and therefore preventive action can be taken care of immediately. Technician can now easily monitor the status of each device and equipment in real-time and remotely.

2.2.3 Importance of server room monitoring system

Server Room Monitoring System can provide alert status of disaster that threatens the availability and up-time. A monitoring system which is needed in a complex server environment and can be helpful even in a small server room of any facilities that requires it. According to (Philip de Freitas, 2019), setting up a server room temperature monitoring system is essential to ensure conditions are in check. Data loggers that provide alerts are the perfect solution to minimize risk and provide reassurance. Programming alerts for early warning thresholds should be put in place. The entire purpose of securing the server room is to protect valuable data and hardware assets.

Having a monitoring system of a server room is very helpful not just for the technician, but also for the equipment kept in there as well. If the condition of the server room is not optimal, it will have a negative impact on the way the hardware runs. Processors can get too hot to the point where they disfigure the chip and crash a system. Even if they're running too cold, processors finish tasks slowly and become less effective. Environmental issues that do not immediately impact the hardware can still cause a reduction in the total lifespan of the equipment.

With the help of the monitoring system, all of the problems that stated above can be avoided as we monitor the room frequently and will get notifications if there is any occurrence in the room. We can set an optimal temperature and humidity for the server room as our liking and make sure the equipment will not be damage in any way possible

2.3 IoT-based device environment monitoring system

In this section will discuss about IoT-based device environment monitoring system. This section only focuses on method of monitoring and IoT-based device. In this project, we are focusing on the best method for monitoring the server in a cost effective way and also the suitable IoT-device that can be used to monitor the server room.

2.3.1 Method of monitoring

As we all know, a server room contains many valuable equipment such as hardware like servers and data storage devices racks, cabling system, UPS, and many more. Due to this, server room equipment consumes more energy and generates more heat than ever before. High temperatures can make equipment malfunction or even cause damage. Internal parts can swell and pull apart, or they can simply burn. Temperature management and maintenance in the server room is important for keeping the equipment secure and performing at its best (Morgana Siggins, 2019). There are multiple strategies or method for monitoring the server room.

One method is to monitor areas of the room that are most likely to be subjected to extreme conditions. If the temperature, airflow, and humidity are all within the correct range in most server room locations. Using equipment cooling and distribution systems can help us stay comfortable with this method. The next method is to monitor the temperature, humidity, and airflow at the rack. If the ambient level is within the normal range at the server level, inside the rack, then it is clear that the heating, ventilation, and air conditioning equipment is functioning properly. The latter method is a combination of the two strategies stated to minimize the weakness of one or the other method. This is the most expensive route, but the most complete and the most recommended to ensure the safety of the device.

To make sure these methods work on their own, we should have a good remote terminal. According to (Morgana Siggins, 2019), there are 2 examples of an efficient remote terminal that can help us better monitor the server room. The first is “TempDefender” in which we can monitor multiple points in the server room from a single device. It is a small, rack-mountable remote terminal that can manage up to 16 analog sensors reporting all environmental factors in the server room. The next solution is the racktorack solution, where the author recommends using the NetGuardian 216 remote control. This unit has four analog inputs, by which we can measure the environment in the rack (or chain sensors on several racks) and 16 discrete alarm inputs, so that we can monitor Direct server and other devices.

However, when we combine the selected monitoring strategy, with TempDefender or NetGuardian, we get the most complete view of the server room status. It might seem like a big investment, but take it as insurance because we can't afford to leave the equipment vulnerable.

2.3.2 IoT-based device

Iot-based device is anything that has a sensor attached to it and can transmit data from one object to another or to people with the help of internet is known as data. (Shiv Nadar, 2020). IoT devices include wireless sensors, software, actuators, and computer devices. They are attached to a particular object that operates through the internet, enabling the transfer of data among objects or people automatically without human intervention. In this project, IoT-based device system in the server room that detects the humidity and temperature if it is in a danger state, the system will send out notifications to the person in charge.

There are multiple researches of the IoT-based device that is used for monitoring the humidity and temperature of the server room. Some examples are Raspberry Pi and Arduino. Those stated devices are microcontrollers that is connected to sensors that work on sensed data.

2.4 Related Work

In this section will discuss about the related works with this project. This section only focuses on monitoring, humidity, temperature and server room.

2.4.1 Approach

The first related worked was called "IoT based temperature and humidity monitoring framework" by (rafizah ab rahman et al, 2020). In this project, they focus on the approached of internet of things (IoT) in monitoring the temperature and humidity of a data center in real-time used a simple monitoring system to determine the relationship and difference between temperature and humidity with respect to the different locations of measurements. In this studied, they proposed for this project that consists of sensor node hardware used Arduino mega 2560 r3, Arduino-compatible ethernet shield, and DHT11 temperature and humidity sensors.

The second related worked was titled "monitoring of temperature and humidity in a smart server room used raspberry pi and WhatsApp notifications" by (dwi ely kurniawan et al. , 2019). In this project, they focus on used raspberry pi and WhatsApp notifications to intelligently monitor server room temperature and humidity. This studied used the internet of things (IoT), a device used by the raspberry pi wireless sensor, and the wemos DHT shield that

could read temperature and humidity conditions in a room. The temperature and humidity log data were stored in a MySQL database and then displayed as a real-time graph. When a notification was sent to the user via the WhatsApp mobile app, the IoT response was based on a predefined temperature standard.

The second related worked was called "smart monitoring" the third related worked was titled "server room temperature and humidity monitoring used lattepanda and thingspeak" by (th nasution et al. , 2019). In this project they were focusing on the design approached of the remote temperature and humidity monitoring system and the server room used the lattepanda sensor and thingspeak for DHT 11 via the Arduino gpio pins. Based on the test results, lattepanda managed to sent continuous data to thingspeak every 30 seconds. Data was sent as temperature in degrees Celsius and percent humidity.

The fourth related worked was called "internet of things (iot) based server room temperature and humidity monitoring" by (moechammad alvan prastoyo utomo et al, 2019). In this project, they were focusing on the approached of an iot-based system that provides information while simultaneously regulating the temperature and humidity inside the server room. For their lived monitoring, used a system equipped with features to provided notifications to users through dynamic telegram app. The prototype brain was a raspberry and Arduino module. The temperature and humidity sensor (dht11) were connected to the raspberry. Whenever these values exceed the threshold selected for each notification provided to the user through the telegram client used the telegram api. The system they offered with the ability to efficiently monitor and dynamically controlled orders through telegram applications or through web server applications.

The fifth related worked was titled “IoT-based approached for real-time conditioning and controlled in server room” (onibonoje et al. , 2019). In this project, they focus on the current approached of an internet of thing (IoT) based system to condition and controlled environmental factors in real time. The system's hardware and software units had been designed and deployed to condition the server room by monitoring power outages, heat, watered leaks, smoke, fire, and lights out; and initiate corrective action in real time to keep them within the respective defined thresholds.

2.4.2 Device

In this section, we will discuss about the devices that is used related in this project.

2.4.2.1 Raspberry Pi

According to (dwi ely kurniawan et al. , 2019), raspberry pi was a small computing device the size of a credit card. Liked ordinary computers on this device, we could install the linux operating system. Raspberry pi could be used as a web server, router, media center, etc. The inputs and outputs of the raspberry pi were very similar to USB, LAN, HDMI and 40 pinned gpio interfaces and could been optimized as needed. Raspberry used apache, mysql, a web server, and the python yowsup library service to ran. The raspberry pi was equipped with an internal Wi-Fi device to connected and sent data between sensors with the raspberry pi, the raspberry pi saves power, only needed 2. 5a and 5v to started. Storing data on the raspberry pi did not used a disk liked a computer hard drove. The device used a micro sd card, which was also where the data on the smartphone was stored.

2.4.2.2 Arduino

Arduino uno r3 was a microcontroller board that contains 14 digital input and output pins, of which 6 could been used as pwm outputs and the remaining 6 could been used as analog inputs. It also includes a 16 mhz crystal oscillator, a usb connection, a power connector, an icsp connector, and a reset button. It contains everything you needed to support the microcontroller, just connected it to your computer with a usb cable or used a dc adapter or battery to power it up and started used it. There were several types of Arduino boards, but the most used were the Arduino uno and mega. The difference between the two was just the number of input and output pins. Compared to Arduino uno 9, Arduino mega had a better number of input and output pins. In modern times, Arduino was commonly used to program microcontrollers. Among other things, due to its user-friendly settings, liked any microcontroller, the arduino was a small printed circuit board with a chip that could been programmed and executed. Mission. The information in the computer program was sent to the arduino microcontroller and ultimately sent to a specific circuit or multicircuit machine to execute a specific instruction. According to (yusuf abdullahi badamas, 2020), arduino could helped to read information from input devices such as sensors, antennas and trimmers, and could also sent information to output devices such as leds, built-in speaker, lcd display and dc motor. According to (shamsul aizal

zulkifli and mohd najib russin, 2020), arduino could also been used as a microcontroller for a three-phase inverter and the researcher (tiffany tang, 2019) had shown that there was another project to connected Arduino with Kinect connected for motioned controlled. This shows that there were many applications that could use Arduino. In this project, Arduino had been used to process the data obtained from the ph sensor before transmitting to the mobile phone. The Arduino would also been used to turn on the watered pump and motor.

2.4.3 Platform

In this section, we will discuss about the platforms that is used related in this project.

2.4.3.1 Web-based

Web-based monitoring system is a system that compiles all relevant information about every project in one system. These data can then be accessed by different users depending on the user rights of each user and what they need to know.

In the past related work, we can see that a research done by (Rico Wijaya et al, 2019) used a web-based monitoring system store data and display data that they got from their devices.

2.4.3.2 WhatsApp

WhatsApp is a messaging application that allows users to share media such as text messages, chats and voice messages and video messages with individuals or groups. WhatsApp relies on data to send messages such as iMessage and BBM.

In the studied conducted by ((dwi ely kurniawan et al. , 2019), they said that some of the studied related to this studied did not had a systematic integration into an issue, for example focusing on SMS notification but no WhatsApp controlled, controlled server on / off and showed server room temperature log history. They used WhatsApp as a way to get notifications about server room status liked the humidity and temperature.

2.4.3.3. Telegram

Telegram is an online messaging app that works just like popular messaging apps WhatsApp and Facebook Messenger. This means that we can use it to send messages to our friends when connected to Wi-Fi or the mobile data.

Following previous researched conducted by (moechammad alvan prastoyo utomo et al. , 2019), they proposed a system that could been used to monitor the temperature or humidity of a particular room or location . The proposed system continuously sends data to the cloud to monitor data from anywhere, and for lived monitoring and regulation, the system was equipped with features that provided notifications to users through the application dynamic telegram. They continuously monitor the server room by setting threshold values for every aspect of the server room, such as humidity and temperature, and whenever these values exceed the thresholds selected for each notification. provided to users through telegram application by telegram plc.

2.4.4 Extra Function

In this section, we will discuss about the platforms that is used related in this project.

2.4.4.1 Real Time Humidity Alert Notification

A studied by (moechammad alvan prastoyo utomo et al. , 2019) created an additional function in which they used real-time humidity alert notification. Their system was equipped with lived monitoring and throttling and provides notifications to users used deployed sensors.

2.4.4.2 Real Time Temperature Alert Notification

According to (dwi ely kurniawan et al. , 2019), the server room was a very sensitive space because there was a server computer and data center infrastructure in an enterprise. Therefore, when the server room administrator was not near the server room, there was a needed for a system that could provide notifications to the administrator about the real-time temperature conditions in the server room. The administrator could control, for example, to shut down the server when the room temperature was high and restart the server when the room temperature returned to normal. Table 2. 1 presents a summary of related worked.

Table 2. 1: Summary of Related Work

Journal Name/ Author	Device/ Sensor	Suitability for Monitoring System	Cost factor for Monitoring System
IoT based temperature and humidity monitoring framework (Rafizah Ab Rahman et al, 2020)	• Arduino Mega 2560 R3 • Arduino Compatible Ethernet Shield • DHT11 Temperature and Humidity Sensors	Yes	Medium
Smart Monitoring Temperature and Humidity of the Server Room using Raspberry Pi and WhatsApp notifications (Dwi Ely Kurniawan et al, 2019)	• Raspberry Pi • Whatsapp • Wemos DHT Shield Wireless Sensor • MySQL Database	Yes	Low
Monitoring temperature and humidity of server room using Lattepanda and ThingSpeak (T H Nasution et al, 2019)	• Lattepanda • ThingSpeak, DHT11 Sensor • Arduino GPIO Pin	Yes	Medium
Server Room Temperature & Humidity Monitoring Based on Internet of Thing (IoT) (Moechammad Alvan Prastoyo Utomo et al, 2019)	• Raspberry Pi • Arduino Module Protoype • DHT11 Sensor • Telegram API	Yes	Low
An IoT-Based Approach to Real-Time Conditioning and Control in a Server Room (Onibonoje et al, 2019)	• Transceiver • microcontroller, antenna and power source Digimesh 2.4GHz Xbee with a PCB antenna node • Grove DHT22 Temperature/Humidity sensor • MQ5 smoke sensor, Flame senor • Grove-GL5528 Light sensor	Yes	High

Based on table 2. 1, we would chose the most suitable equipment and technique for this project and we would had the lowest costed factor to built it. The raspberry pi device was the cheapest and most popular device available in the market because it was very easy to set up. In this project, we could use the device to realized the solution proposed in this project. The technique we would use was to used a real-time monitoring system in which we monitor the humidity and temperature of the server room. Since the equipment at our disposal was used in this project, we would collect the data and sent it to the system. The system then checks if the received data exceeds the threshold, we set in the server room. If the data was above the threshold, the system would notify the manager by sending a notification through platforms liked telegram or WhatsApp.

2.5 Proposed Solution

In this section, we will discuss about the parameters that will use in this project. These parameters are used to measure which solution is the best used for this project.

2.5.1 Parameters

Parameters were an essential component of any analysis. Simply put, a parameter was any numerical quantity that characterizes a given set or aspect of it. This means that the parameter tells us all about the population. It represents the true valued that would been obtained if a census was taken, not a sample (mahesh, 2019). In this project, the parameters were temperature, humidity, real-time monitoring, notification alert and also online platform. We would review and compare each related worked and offered solutions from this review. Table 2.2 will show the parameters that are used in each of the studies.

Table 2. 2: Summary of Parameters from Previous Research

Yes (√), No (**X**)

No.	Journal Name/ Author	Temperature	Humidity	Real-time Monitoring	Alert Notification	Online Platform
1.	IoT based temperature and humidity monitoring framework (Rafizah Ab Rahman et al, 2020)	√	√	√	X	X
2.	Smart Monitoring Temperature and Humidity of the Server Room using Raspberry Pi and WhatsApp notifications (Dwi Ely Kurniawan et al, 2019)	√	√	√	√	√
3.	Monitoring temperature and humidity of server room using Lattepanda and ThingSpeak (T H Nasution et al, 2019)	√	√	√	√	X
4.	Server Room Temperature & Humidity Monitoring Based on Internet of Thing (IoT) (Moechammad Alvan Prastoyo Utomo et al, 2019)	√	√	√	√	√
5.	An IoT-Based Approach to Real-Time Conditioning and Control in a Server Room (Onibonoje et al, 2019)	√	√	√	X	X

Based on Table 2.2, not all researchers done every single one of the parameters. Therefore, there is room for improvement in notification alert on an online platform. So, in this project, we are going to include notification alert in order to notify the user if there is any malfunction

or the data is above the threshold that we have set for the system. Setting a threshold for the humidity and temperature of the server room is also one of the improvements that we going to add in order to make server room more secure.

2.6 Critical Review

Table 2.3 will provide a critical review for each one of the related works.

Table 2. 3: Critical Review of Previous Research

No.	Journal Name/ Author	Sensor	Functionality	Real-time Humidity and Temperature Monitoring	Notification's alert
1.	IoT based temperature and humidity monitoring framework (Rafizah Ab Rahman et al, 2020)	Humidity and Temperature Sensors	In their project, he used sensors to monitor humidity and temperature in real time, but they did not provide notification alerts to notify users in the server room.	In their study, they monitor the humidity and temperature in different locations in order to get the data needed for their conclusion	They do not include notifications in their studies.
2.	Smart Monitoring Temperature and Humidity of the Server Room using Raspberry Pi and WhatsApp notifications (Dwi Ely Kurniawan et al, 2019)	Humidity and Temperature Sensors	In their project, they designed a system that could captured real-time humidity and temperature data and sent it to the database. Then they showed it in the report. Its system also includes a form to notify users when the temperature rose above a specified valued.	Their system includes real-time monitoring where the sensors capture the humidity and temperature	The notifications they use are sent via WhatsApp application on the smartphone
3.	Monitoring temperature and humidity of server room using Lattepanda and ThingSpeak (T H Nasution et al, 2019)	Humidity and Temperature Sensors	In their project, they designed a system that could captured the temperature and humidity of the server room. The data they captured had been sent every 30 seconds.	The system sends data every 30 seconds base on their design.	They do not include notifications in their studies.
4.	Server Room Temperature & Humidity Monitoring Based on Internet of Thing (IoT) (Moechammad Alvan Prastoyo Utomo et al, 2019)	Humidity and Temperature Sensors	In their project, the system they created retrieved humidity and temperature data from the server room and displayed it on the website in graphical form.	The sensors collect the data from the server room using real-time monitoring and sends it to a website so that the user can read the temperature and humidity in the server room	The system sends notifications on the telegram application if the setting temperature exceed the limit
5.	An IoT-Based Approach to Real-Time Conditioning and Control in a Server Room (Onibonoje et al, 2019)	Humidity and Temperature Sensors	In their project, they created a system with many nodes that could detect many variables, such as power monitor outages, thermal conditions, watered leaks, smoke or fires, extinguishing the lights in the monitored server room.	Their system monitor the server room in a time limit that the researchers set.	They do not send notifications in to any platforms in their studies

Based on the Table 2.3, using a real-time monitoring system has a lot benefits as the system will continuously sends data to the user. Real-time monitoring systems Monitoring your

network in real time was a great way to support security compliance. Real-time monitoring could help users, for example it departments, to immediately identified and resolve security issues when they arise. These problems could include an abnormal increase in data received from the system. In this project, it was also useful to had a notification alert, especially for mobile app notifications. The mobile app notifications allowed us to always stayed in touched with the system in a non-intrusive manner by providing timely news and useful related information in accordance with our settings. In this case, by setting a threshold to the system, if the humidity and temperature were higher than the threshold, it would sent a notification to the user as soon as possible, no matter where the user was located.

2.7 Conclusion

In conclusion, we will propose a system that will monitor the server room within the working hour with real time monitoring system and notifications alert. The added feature that is proposed in this project is that we will set a certain threshold on the variables that is affecting the environment of the server room. The variables are the humidity and the temperature. For example, we will set the threshold for the temperature between 18°-27°C / 64°-80°F. If the temperature goes below or above the threshold, they system will notify the user using the platform provided via notifications.

CHAPTER 3

METHODOLOGY

3.0 Introduction

This chapter will be focusing on the methodology that used to develop the IoT-based device for real time server room environment monitoring system. There are various type of software development model or method available and choosing an appropriate software development model are crucial as it will hugely impact the system development process.

In this project, the goal is to gather all information related to the title of the project, which is the IoT-based device for real time server room environment monitoring system through the journal, articles, web search and books. For early stages, this project starts by stating the introduction of the project includes the objectives, problems statements etc. The chapter then progressing by identifying and analysing all information gathered in the literature review. A detail explanation on the software development model chosen will be further elaborate and the project milestones and Gantt chart for this project will be attached in this chapter.

3.1 Methodology

The software development model that is chosen for this project is Prototype model that is categorized under Software Development Life Cycle (SDLC) approach. This model is suitable because the requirements are very well documented, clear and fixed.

3.1.1 Software Development Life Cycle (SDLC)

When it comes to System Development Life Cycle (SDLC), Waterfall is the most used, traditional and sequential methodology. Although most of the review of this methodology considered as an outdated method, it is very helpful to help the

user/author/developer to understand the system in a very detailed way. It requires plenty of structure and documentation in the upfront, which are divided into several stages, each with different approaches and information.

Unfortunately, this approach is lack of flexibility, which means every information and steps partake in this approach require a detailed explanation and justification at the early stages. Any changes in the middle of the implementation stages will require the approach to full restart, stating and analyzing the preinformation data again.

3.1.2 Prototype Model

The Prototype model divides project activities into linear sequential stages, each of which is dependent on the deliverables of the last one and corresponds to a task expertise. Because it is basic and straightforward to grasp and apply, this model was chosen as the methodology for building this project. This method is divided into six stages: requirement analysis, system design, implementation, testing, deployment, and maintenance.

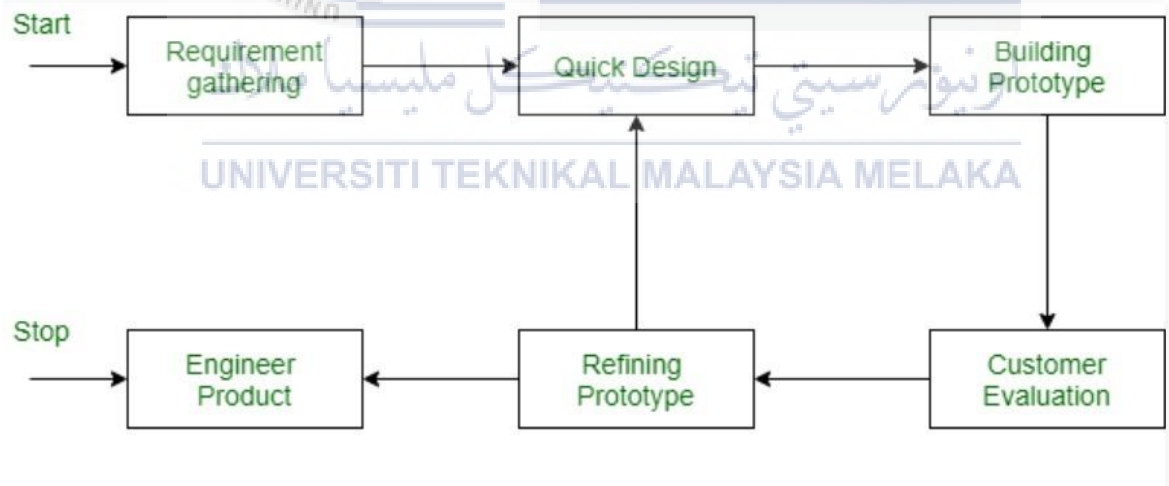

Figure 3. 1: Flow and Sequences of Prototype Model Phases

In this project, they use of Waterfall model is chosen due for the development of the proposed system. In general, each stage has to be finished in order to starts the next stages, which can help with the project's requirements and procedures. The approach of this model is highly suitable for this proposed system, allowing the progressions becomes more detailed, since all of the information and requirements are

stated at the beginning of the stages, enhancing the other stages' progress with all of the information stated and gathered.

3.1.2.1 Requirement Analysis Phase

The initial phase of prototyping model is requirement analysis. This phase is crucial as it analysing and acquire all the project requirements especially for functional requirements. All the functional and non-functional requirements must be match with the project objectives. Any unclear requirements will lead to poor defined project scope that might cause problem later in the development. The requirement of the project is discussed as below;

- System Requirements

  i) Notifications

  - A selected phone will pick to receive notifications from the system
  - If the system is enabled, it will display the status of the output devices, indicating it as ON or OFF.

  ii) Statistical usage of the component's consumption.

  - The system will produce a statistical data on the total usage of the component's consumption.
  - The system will be presented in form of table and graphs.

• Software Requirements

i) Raspberry Pi OS

- An open-source application used to write and upload code into the microcontroller.

ii) Pocket Lab

- A software that allows the data to be send/stores to cloud and able to be accessed on website.

• Hardware Requirements

i) Raspberry Pi

- A microcontroller kit to implement program to communicate with IoT devices.

ii) DHT 11 Sensor

- A sensor that detects the humidity and temperature in the specified detention area.

iii) Pocket Lab

- A sensor that can detect many things in the room environment.

3.1.2.2 Quick Design Phase

This stage discussed about the architecture of the project. The requirement specification from the first phase is studied to create a system design. The design will help the defining the overall system architecture using the requirement specify in the first phase. The prototype design is to use the sensors to detect the humidity and temperature of the server room and sends the data to Raspberry Pi. All the program code will be written specifically to each sensor that are involved. Then, a circuit design of the connection between the sensor, microcontroller and microchip on the breadboard is designed and drafted.

3.1.2.3 Build a Prototype Phase

Based on the information obtained in the previous phase, a real prototype is created during this phase. It is a scaled-down functioning model of the necessary system. The physical design specifications are converted into functional code. The Raspberry Pi is a cross-platform Windows tool that allows you to develop and upload programmes to compatible boards. The system is created in little program with input from the system design, which will be combined and evaluated in the following step.

During this phase, any code used in the system will meet the requirements and specifications.

3.1.2.4 Initial User Evaluation Phase

The proposed system is offered to the user for an initial evaluation at this phase. It aids in determining the working model's strengths and weaknesses. Feedback, including comments and suggestions, is gathered from testers or users for product evaluation and review.

3.1.2.5 Refining Prototype Phase

If the user is unsatisfied with the current prototype, it must be improved using the user's feedback and suggestions. This phase will not be finished until all of the user-supplied requirements have been met. Once the user is satisfied with the prototype, a final system is built on the approved final prototype.

3.1.2.6 Implement and Maintenance Phase

When the final prototype is created, the final system is thoroughly tested and put into production. Routine maintenance is conducted on the system to save downtime and avoid large-scale failures. Some flaws or problems have been identified based on user feedback in their surroundings. So, throughout this period, perform maintenance and resolve any difficulties that arise, ensuring that everything functions well. Maintenance is performed to guarantee that these modifications are delivered in order to improve the client environment.

3.2 Project Milestones

Project milestones will show the progress achievement based on a specific timeline in the project. Each of the task must be complete within the specific time to ensure the whole project can be finish within the given time. The milestones can serve

as a guidance for developer to identify whether the project is executed in good condition. Table 3.1 display the milestones of the project.

Table 3. 1: Milestone of the project

WEEK	ACTIVITY	NOTE / ACTION
< W0 **(< 21/3)**	Select a suitable project topic and potential Supervisor	• Action - Student
W 1 **(15/3 ➐ 21/3)** Meeting 1	Proposal PSM: Discussion with Supervisor	• Deliverable - Proposal • Action - Student
	Proposal assessment & verification	• Action - Supervisor
W2 **(22/3 ➐ 28/3)**	Proposal Correction/Improvement	• Action - Student
	Proposal submission to Committee via email	
	Proposal Approval	• Action - PSM/PD Committee
	List of Supervisor/Title	
W3 **(29/3 ➐ 4/4)**	Proposal Presentation & Submission via PSM ULearn	• Deliverable - Proposal Presentation (PP) and Completed Proposal Form • Action - Student
Meeting 2	Chapter 1 (System Development Begins)	• Action - Student
W4 **(5/4 ➐ 11/4)**	Chapter 1	• Deliverable - Chapter 1 • Action - Student, Supervisor
W5 **(12/4 ➐ 18/4)**	Chapter 2	• Action - Student
W6 **(19/4 ➐ 25/4)** Meeting 3	Chapter 2	• Deliverable - Chapter 2
	Project Progress	• Progress Presentation 1 (PK 1) • Action - Student, Supervisor
W7 **(26/4 ➐ 2/5)**	Chapter 3	• Action - Student
W8 **(3/5 ➐ 9/5)**	Chapter 3	• Deliverable: Chapter 3 • Action - Student, Supervisor
W9 **(10/5 ➐ 16/5)**	MID SEMESTER BREAK	
W10 **(17/5 ➐ 23/5)** Meeting 4	Chapter 4	• Action - Student
	Project Progress	• Progress Presentation 2 (PK 2) • Action - Student, Supervisor
W11 **(24/5 ➐ 30/5)**	Project Demo	• Action - Student, Supervisor
W12 **(31/5 ➐ 6/6)**	Project Demo PSM1 Report	• Action - Student, Supervisor
W13 **(7/6 ➐ 13/6)** Meeting 5	Project Demo PSM1 Report Schedule the Presentation	• Action - Student, Supervisor • Action - PSM/PD Committee • Presentation Schedule
W14 **(14/6 ➐ 20/6)**	Project Demo	• Deliverable - Complete PSM1 Draft Report • Action - Student, Supervisor
W15 **(21/6 ➐ 27/6)** Final Presentation	FINAL PRESENTATION Submission of the PSM1 Report onto the PSM ULearn.	• Action - Student, Supervisor, Evaluator, PSM/PD Committee

W16 **(28/6 ➔ 4/7)**	REVISION WEEK Correction on the draft report. Submit PSM1 Logbooks to PSM ULearn. Submit an EoS Survey form.	• Deliverable - Complete PSM1 Logbooks • Action - Student, Supervisor
		• EOS Survey • Action - Student
W17 & W18 **(5/7 ➔ 18/7)**	FINAL EXAMINATION WEEKS	

3.3 Project Gantt Chart

Gantt Chart is the simplify version of project milestones that display graphical portrayal of the project point of reference. The Gantt Chart serve as a guide for project completion each week using graphical way. Figure 3.2 display the project's Gantt chart.

Figure 3. 2: Project's Gantt chart

ACTIVITY	PERIODS (week)													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
PSM Proposal Submission														
PSM Proposal Correction														
Chapter 1														
Chapter 1 (Deliverable)														
Chapter 2														
Chapter 2 (Deliverable)														
Chapter 3														
Project Demo & Chapter 3 (Deliverable)														
Project Demo & Chapter 4														
Project Demo & Chapter 4 (Deliverable)														
Project Demo														
Project Demo & PSM Report														
Final Presentation														

3.4 Conclusion

In summary, this chapter explained the project methodology of the development project, include the chosen SDLC approach. Moreover, the chosen approach also elaborated by each stage, relating with the current development project on what and

how the stages will be established. This project also explore the importance of project milestones and project Gantt chart which are crucial element to ensure the project are develop in a structured and ordered manner. For next chapter, it will discuss about the analysis and design for the project, which includes the detailed problem analysis, hardware and software requirements that will be used in the project and the project design.

CHAPTER 4

ANALYSIS AND DESIGN

4.0 Introduction

In this chapter, we are elaborate some of the analysis that needs to be focus on which related to the title of the project, that is the IoT-based device for real time server room environment monitoring system. This chapter also presents the analysis requirement of the project to properly plan and prepare for the project. It also focusses on the problem before starting the requirement and design. This technique of analysis will help the development of the project. The next process is the design phase where it will contain the expected result of the project. The requirement of the hardware and software are all include in this chapter where it describes about the functionality and the architecture of the system and the hardware. The system design will be including in this chapter to view how the system will function and understanding the flow of the system.

4.1 Problem Analysis

The server room is a very sensitive space because the company has server computers and data center infrastructure. Therefore, server room conditions must always be protected from temperature and humidity. Of course, the server room is an air conditioner (AC), which plays an important role in maintaining the room temperature. If the air conditioner is damaged, it will increase the temperature and humidity of the room. However, the status of the server room should always be monitored and the temperature and humidity in the room should be maintained. Temperature and humidity are controlled by installing a thermometer and hygrometer in the server room. But if you enter the server room to monitor, it will be very troublesome because the temperature in the server room is very low. Also, if we want to monitor continuously, it will be more difficult to monitor directly in the computer room.

4.2 Requirement Analysis

This section we will discuss about the functional requirements that is included in this project. Requirement analysis consists of identifying, gathering, and understanding system

requirements and their characteristics. In this section, data requirement, software requirement and hardware requirement are the only parts that we are going to discuss in order to develop the project and to understand better what are the equipment that we are going to use in this project.

4.2.1 Data Requirement

The project has input and output data, namely the humidity and temperature of the server room. The monitoring system has sensors that receive and collect input data. The sensors used in this project convert stimuli such as heat and humidity into electrical signals. These signals are converted into binary codes through an interface and then passed to the microchip for processing. The processed signal will be displayed through a cloud platform application called Blynk because it will receive the data..

4.2.1.1 Humidity

Humidity is commonly understood as relative humidity. Relative humidity is the amount of water vapor actually contained in the air, expressed as a percentage of the maximum amount of water vapor that the air can hold at the same temperature. Humidity in server rooms and data centers should be maintained between 40% and 60% relative humidity (RH). The second range will help prevent electrostatic discharge, reduce the risk of corrosion (due to excessive condensation) and extend the life of the equipment.

4.2.1.2 Temperature

Temperature, a unit of measure for heat or cold expressed by one of many arbitrary scales and indicating the direction in which thermal energy will spontaneously occur. For server rooms, you should maintain a temperature range of 65 to 75 degrees F (18 to 24 degrees C).

4.2.2 Software Requirement

4.2.2.1 Raspberry Pi OS

Figure 4. 1: Raspberry Pi Operating System

Figure 4.1 shows an image of the Raspberry Pi operating system. Raspberry Pi OS is a free Debian-based operating system, optimized for Raspberry Pi hardware. To use the hardware, it is necessary to install the Raspberry Pi hardware. operation on the device.

4.2.2.3 Telegram

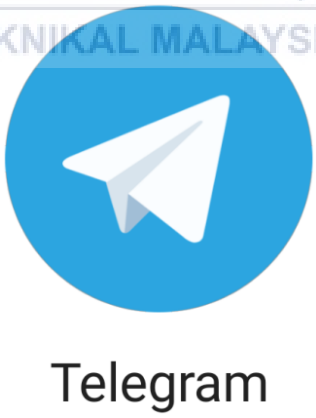

Figure 4. 2: Telegram Platform

Figure 4.2 shows the telegram icon. Telegram is a free and open source, cross-platform, cloud-based messaging service and instant messaging (IM) software.

4.2.3 Hardware Requirement

4.2.3.1 Raspberry Pi

Figure 4. 3: Raspberry Pi

Figure 4.3 shows an image of the Raspberry Pi. The Raspberry Pi is a cheap, cost-effective computer that can be plugged into a computer monitor or television, and uses a standard keyboard and mouse. It's a capable little device that allows people of all ages to explore computers and learn to program in languages like Scratch and Python. The SD card inserted into the card slot acts as a hard drive for the Raspberry Pi. It is powered by USB and the video output can be connected to a traditional RCA TV, a more modern monitor or even a TV using the HDMI port.

4.2.3.2 DHT 11 Sensor

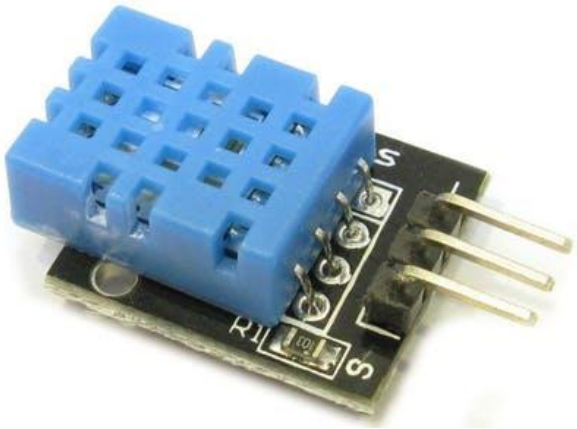

Figure 4. 4: Sensor

Figure 4.4 shows an image of the sensor used. It is called DHT11. The DHT11 is a basic and ultra-economical digital humidity and temperature sensor. It uses a capacitive humidity sensor and thermistor to measure ambient air and output a digital signal to the data pin (no analog input pin is needed). The DHT11 calculates relative humidity by measuring the resistance between two electrodes. The moisture sensing element of the DHT11 is a moisture-retaining substrate with electrodes applied to the surface.

4.2.3.3 Pocket Lab

Figure 4. 5: Pocket Lab

Figure 4.5 shows the pocket laboratory. Pocket Lab is a wireless sensor for exploring the world and building scientific experiments. Pocket Labs is a clever educational concept based on modern, small and inexpensive computing devices. The Pocket Lab is an ultra-durable wireless sensor that transmits data in real time and is capable of measuring acceleration, force, angular velocity, magnetic field, pressure, altitude and temperature.

4.2.3.4 Jumper Wire

Figure 4. 6: Jumper Wire

Figure 4.6 shows a Jumper Wire. A jumper wire is an electrical wire that consist a connected pin which used for interconnecting the items on the breadboard for prototyping or circuit testing with the need of soldering. There are three type of jumper wire which is male-to-male, female-to-female, and male-to-female.

4.3 Detailed Design

This section we will discuss what is the detailed design of this project. Detailed design deals with the implementation part of what is seen as a system and its sub-systems in the architectural designs. It is more detailed towards modules and their implementations. It defines logical structure of each module and their interfaces to communicate with other modules.

4.3.1 Circuit Design

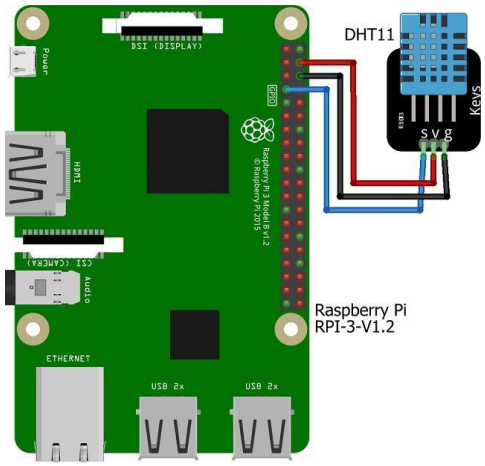

Figure 4. 7: Circuit Design

In figure 4.7, it is shown that the DHT11sensor has 3 pins, the signal pin is mounted on the GPIO 4 pin on the Raspberry PI, the pin ground (-) is installed on the ground pin on the Raspberry PI, and the Vcc (+) pin is attached to the power source pin with 5 volts on Raspberry Pi.

4.3.2 Architecture Design

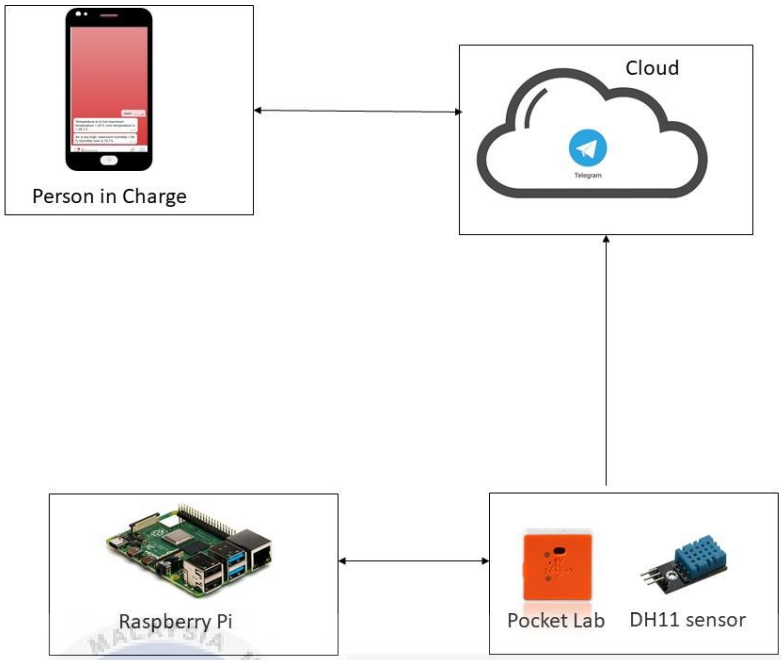

Figure 4. 8: Architecture Design

Figure 4.8 shows the architecture design of the system. The system design process consists of two parts, the first is the design of hardware reading data on the sensor. The second is sending data to Telegram Application capable to be resuscitated which is can be accessed via internet using a browser. Readings of temperature and humidity data using DHT sensor 11. DHT 11 sensor is a sensor module serves to censor temperature and humidity objects have analog voltage output and continues processed using a microcontroller. The DHT 11 sensor is connected to the GPIO port on Raspberry Pi. The data that is taken will then be send to the cloud that store the data that will be sent to the user. In this work, the data that will be sent is the humidity and temperature of the server room and will be sent in interval time. In time, a report will be generated to make sure the user is up to date of how the condition of the server room. This is to make sure the server room is always in maintenance even if its not in a critical condition.

4.3.3 Interface Design

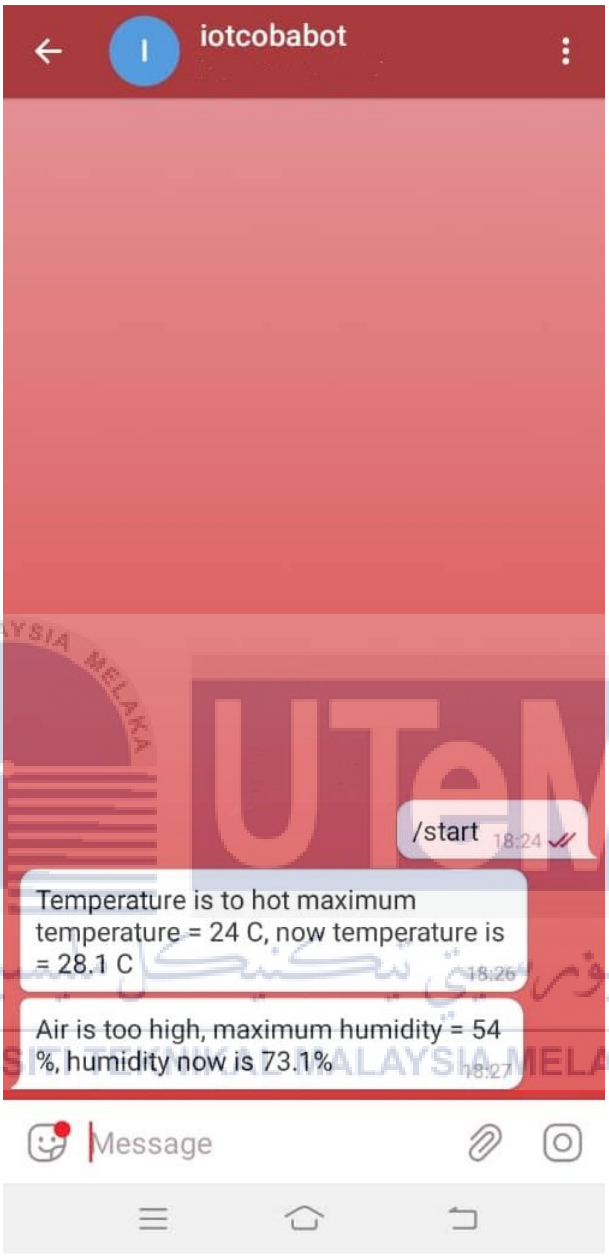

Figure 4. 9: Interface Design

Figure 4.9 shows the design of Telegram interface. Interface design is Design is the process of making interfaces with a focus on looks or style in software or computer devices. Good interface design is user-friendly, and the design is easy to use by whole community. The design will display the humidity and temperature of the server room.

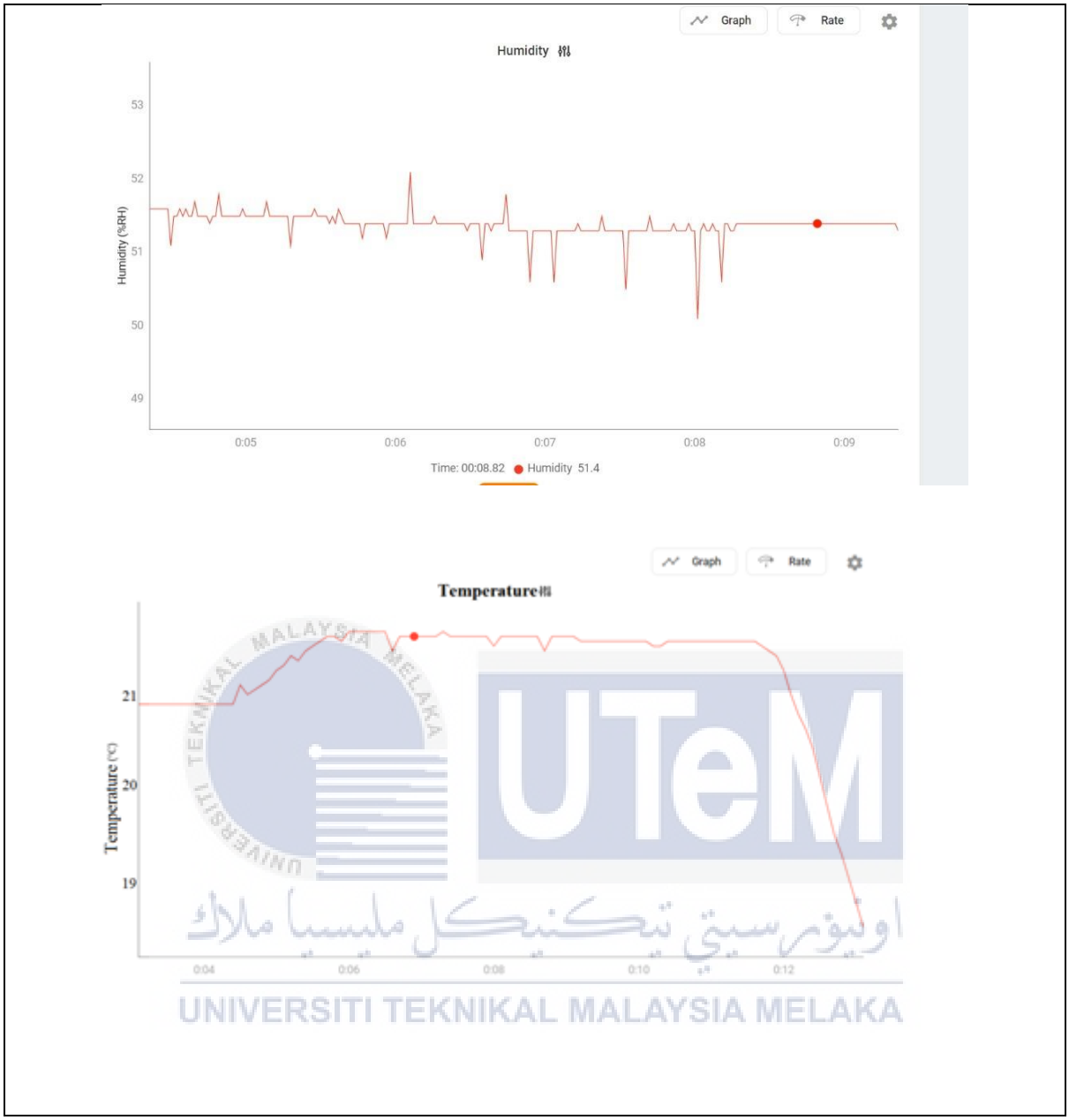

Figure 4. 10: Report Generated

Figure 4.10 shows the design of Pocket Lab interface. It shows the graphical design for the Pocket Lab sensor in which the received the data from the surrounding. Good interface design is user-friendly, and the design is easy to use by whole community. The design will display the humidity and temperature of the server room.

4.3.4 Flow Chart

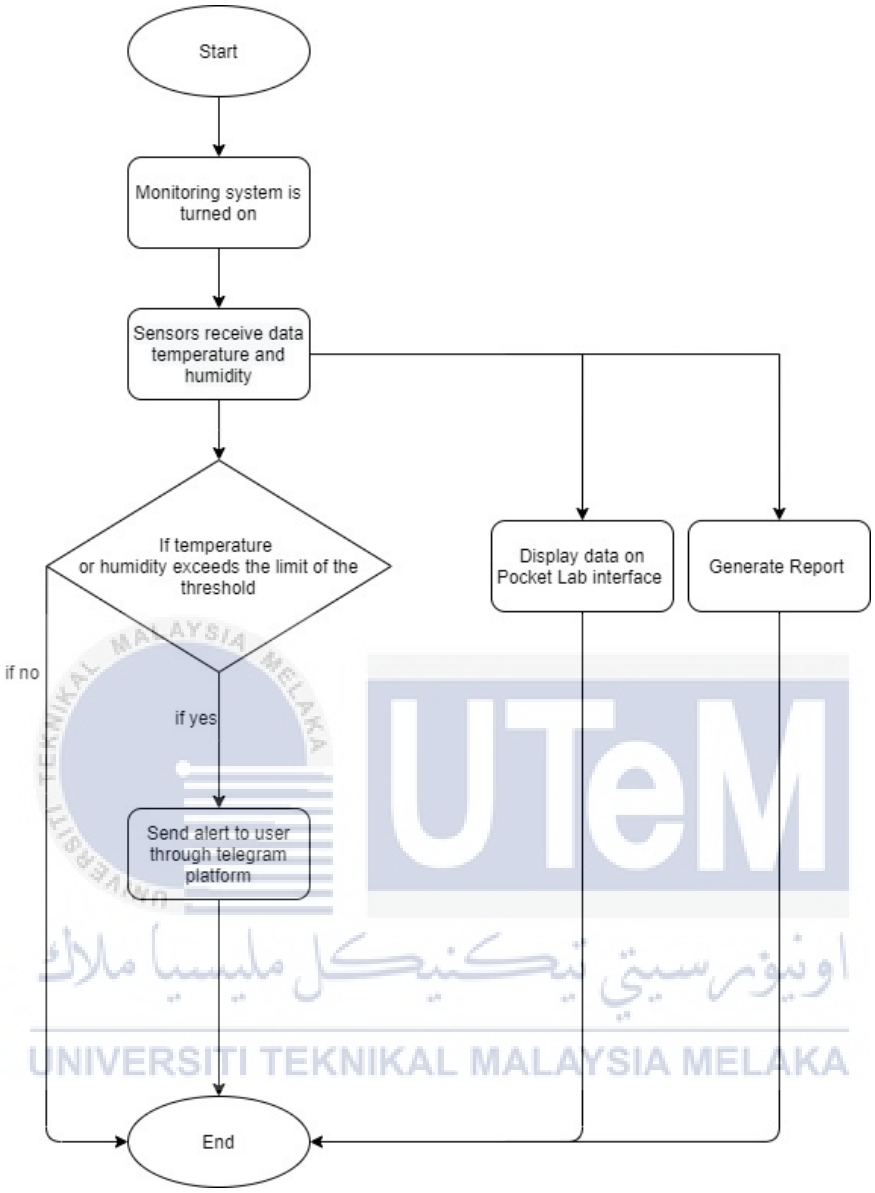

Figure 4. 11: Flowchart Process

Figure 4.13 shows the explanation of the system. The system starts when the monitoring system is started. Once the Raspberry Pi and all the other device are turn on, the sensors will start receiving the data that is the temperature and humidity in the server room. Once the sensors received the data, pocket lab will display those data on the pocket lab interface in which it can remotely be monitored on the person in charge device. Air temperature and humidity data are checked whether they exceed the temperature limit in which a minimum temperature is 18 ℃ and a maximum temperature is 27 ℃ or exceed the humidity limit of the air humidity of at

least 40% and maximum humidity of 55%. If the sensors detect the humidity or the temperature exceeds the threshold limit that was set in the system, it will send notification to the user through telegram platform. The monitoring system will send data to the user every 10 seconds so that the user will keep notice of the condition of the server room. Also, a report will be generated so that the user can keep monitoring the stability of the temperature and humidity for a certain length of time. This is to make sure if the air conditioner in the server room is running ok or not. If the report produces a suspicious or unwanted data, the person in charge can proceed to maintain or replace a new air conditioner for the server room

4.4 Conclusion

In conclusion, this chapter explained the importance of the requirement analysis and how it is essential to know the designs of every aspect for this project. Moreover, this chapter explained the importance of the flowchart and how the workflow of this project that will take into place. This chapter tackle about the analysis and design for the project, which includes the detailed problem analysis, hardware and software requirements that will be used in the project and the project design.

CHAPTER 5

IMPLEMENTATION

5.0 INTRODUCTION

This chapter describes the implementation of a project that contains the configuration of the development environment and the configuration of the project software. This chapter shows the implementation of sensors for monitoring the system and the representation of data through storage platforms.

There are two environments to consider in this project, but it is adjusted to the preference of the sensor and the monitoring environment. Each environment represents a variety of functions and hardware. Software implementations depend on each environmental configuration.

At the end of this chapter, the project should explain how the sensors and microcontroller interact with each other to monitor data through Telegram platform and present the data through the Telegram platform. This chapter also should show how the hardware and software implemented in each environment setup.

5.1 SOFTWARE DEVELOPMENT ENVIRONMENTAL SETUP

This part will go over the two-environment configuration, namely the sensor environment and the monitoring environment. The settings and variables used in each environment will be listed and explained in the environment setup. It will also walk through the steps of installing and configuring hardware or software.

First before the setup of the sensors, Raspberry Pi must be installed to enter configuration codes of the device and connection with the Raspberry Pi.

5.1.1 SENSOR ENVIRONMENTAL SETUP

DHT11, a temperature and humidity sensor will be connected to a Raspberry Pi using jumper cables. DHT11 is a low-cost digital sensor for sensing temperature and humidity. The sensor comprises of three pin which are VCC, Data and Ground. All of the pins are used to connect to the Raspberry Pi which it will powered and grounded by it.

The configuration code for each sensor is configured in the Python IDE that was installed manually into the Raspberry Pi. The microcontroller is able to read analog signal from the sensor. In this project, the sensor that was connected to the Raspberry Pi will be interpreted the signal from the analog pin to be converted to correct value.

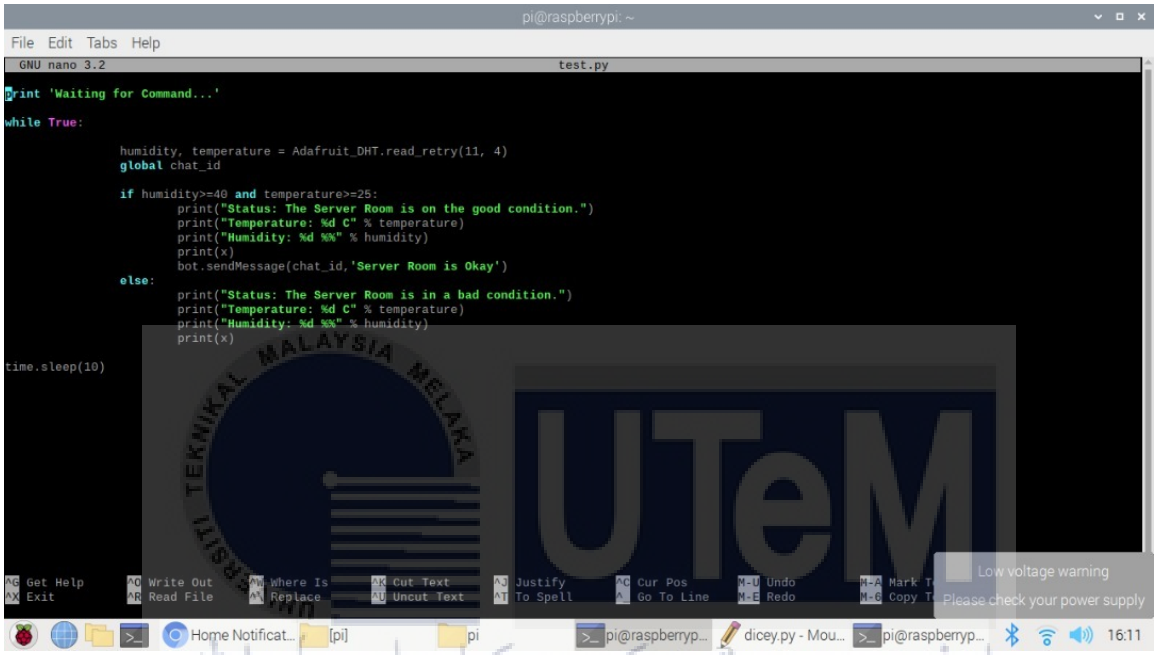

Figure 5. 1: Setup Code Implementation of the DHT11 sensor

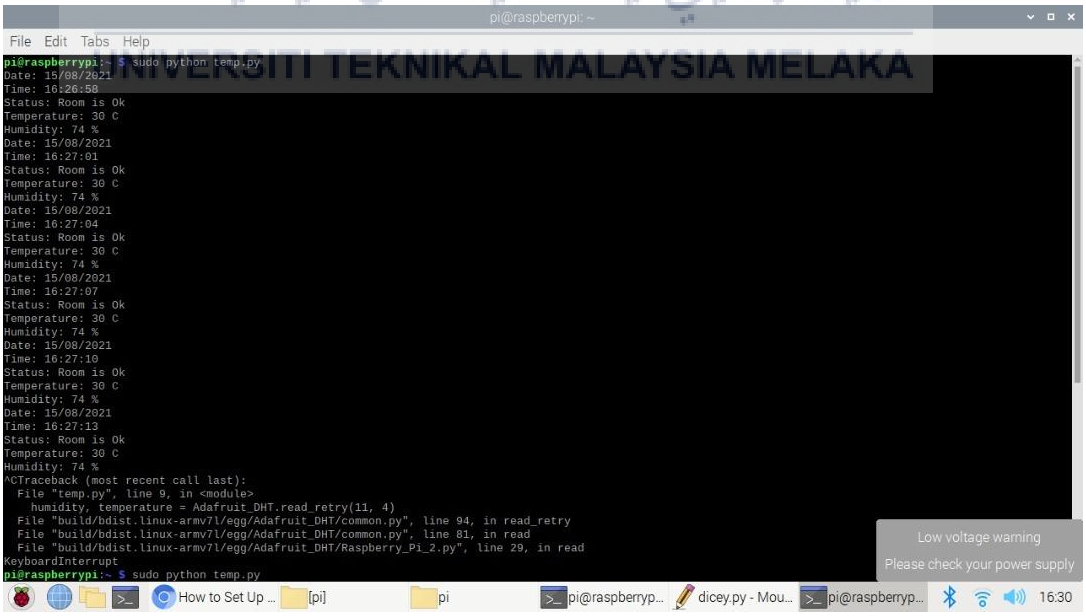

Figure 5. 2: Code Implementation on how the system react based on the signal from the DHT11 Sensor

Luckily, Raspberry Pi does provide a wireless capability, enabling it to connect to the nearest Wi-Fi network and connect it to the Internet. The schematic of the wiring is shown in Figure 1.3. These are the wiring pins that are used for this purpose:

- Data pin to pin GPIO4 of the Raspberry Pi.
- GND pint to the ground link on the Raspberry Pi.
- VCC pin to the 5V line on the Raspberry Pi.

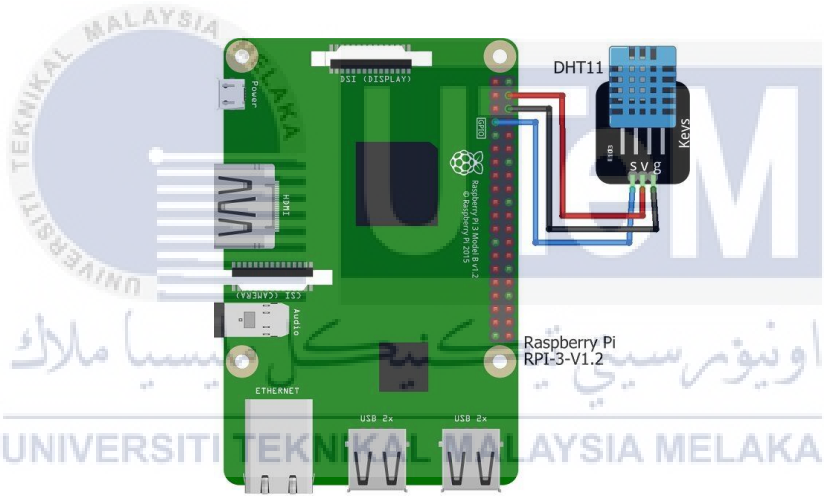

Figure 5. 3: Schematic of DHT11 wiring. Adapted from www.iotstarters.com on How to Connect an DHT11 to a Raspberry Pi

5.2 SOFTWARE ANALYSIS

This section will present how the data will be preview and monitor using various monitoring platform such as Telegram and Pocket Lab. This section will be also showing how device will communicate with Telegram platform and shows the expected data.

5.2.1 POCKET LAB APPLICATION

For this project, Pocket Lab is used to visualize and analyze data retrieved from the sensor.

The data or readings are stored in the platform and display in various form of visualization. Pocket Lab is an application and it can be monitored anywhere using the internet. It also takes a minimum of 15 seconds to update the readings. The step of configurations of Pocket Lab application and how the project is integrated into the application is shown below:

i. Download the application on a smartphone device from Apple Store or Google Play Store.

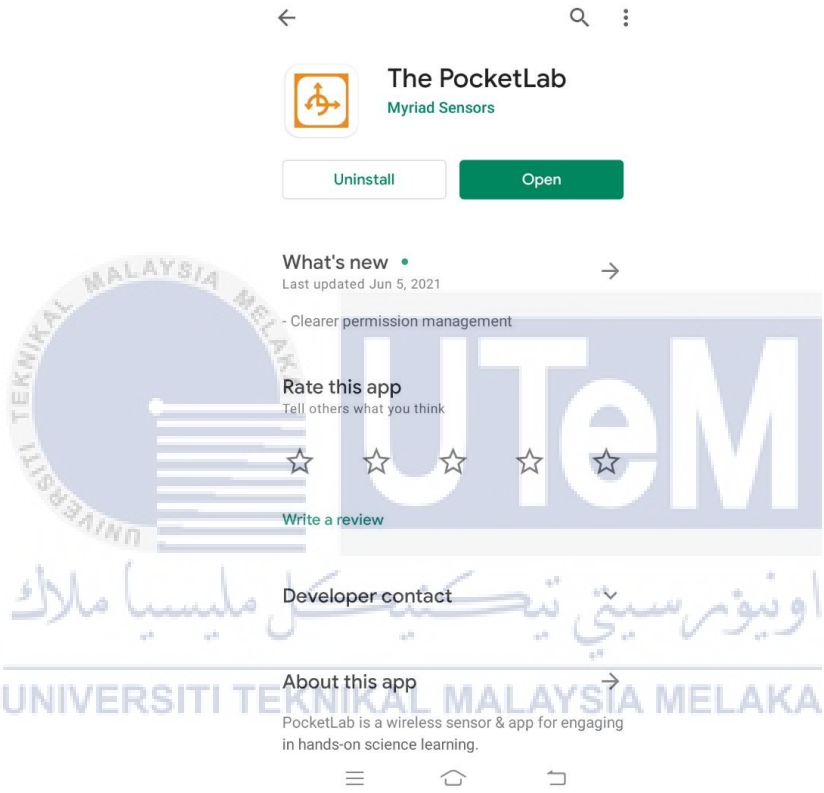

Figure 5. 4: Application on the App Store

ii. After download complete, turn the device's Bluetooth on in the settings menu.

iii. Press the button on the Pocket Lab sensor. The LED will flash alternating red and green.

iv. If the Pocket Lab sensor is in close range to your device, the sensor will connect automatically and the LED will flash blue.

v. Open the Pocket Lab application on your phone and then press the “Start Collecting Data” button.

Figure 5. 5: Main Interface of the Pocket Lab Application

vi. After that it will prompt to the connect to the Pocket Lab device page. In this page, we will need to press the “Connect a Pocket Lab”.

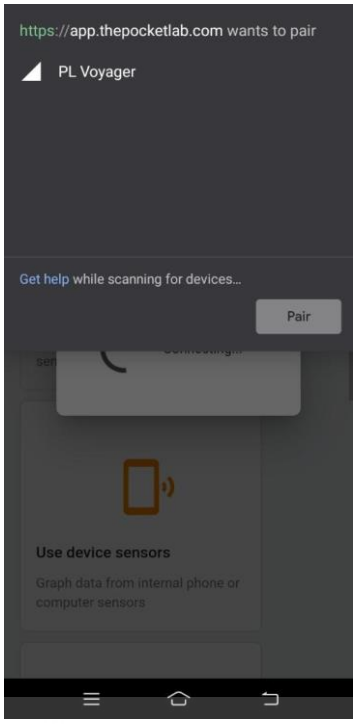

Figure 5. 6: Connect to the Pocket Lab device

vii. After it is connected, it will go the collect data page. Here, we can choose multiple data that can be collected from the Pocket Lab device. In this project, we will choose Humidity and Heat Index as our collected data. When it is chosen, it will visualize a graph that can be seen on the phone. From there, we can record the data that was collected.

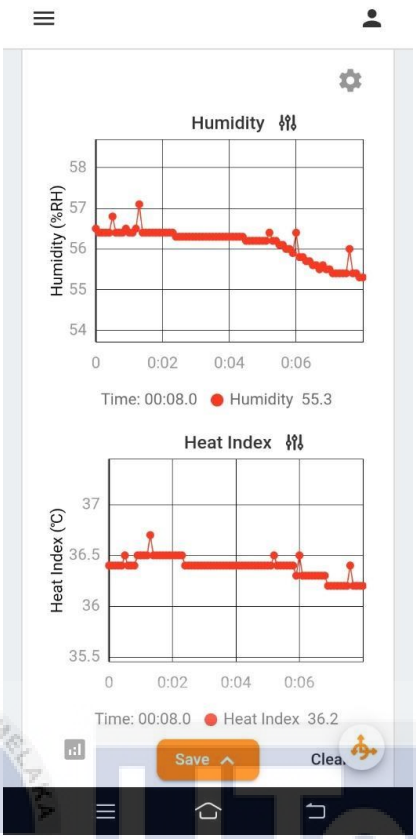

Figure 5. 7: Pocket Lab Device is recording the data that is being collected

Internet connection is crucial to the connection between the phone and the Pocket Lab.

5.2.2 TELEGRAM PLATFORM

For this project, readings from sensors are viewed from Telegram. It is a platform which allows a quick setup for control and monitor IoT projects. The platform is smartphone-based iOS and Android device. The step of configurations of Telegram platform and how the project is integrated into the application is shown below:

i. First, Install Telegram on our Phone and sign up for an account. After clicking on the sign-up button, fill in the details.

ii. Once the sign up and verification are completed, Now, Telegram Application is now accessible but not yet for the Raspberry Pi. Telegram reserves a

special kind of accounts for machines, called bot accounts. As the owner of your own Pi, you have to obtain a bot account for it.

iii. Now, text /newbot to BotFather

Figure 5. 8: Messaging the Telegram BotFather to start the Telegram Bot

iv. Next, we need to Install Telepot on Raspberry Pi. install telepot, a Python package that enables the Pi to speak Telegram Bot API. On the command line, run these two commands:

```
sudo apt-get install python-pip
sudo pip install telepot
```

v. After that, we need to test the Token that we received from the BotFather. On the command line, type python to enter the Python interpreter.

In the Python interpreter, enter these three lines, as in the screenshot above:

```
import telepot
bot = telepot.Bot('*** copy bot
token from browser ***')
bot.getMe()
```

Figure 5. 9: Lines of commands that is type into the Raspberry Pi to test the Token

vi. Now we can start messaging with the Telegram Bot that we have setup from using the BotFather. We can use it to get data from the Raspberry Pi into our phone via WiFi.

5.3 COMPLETE PROTOTYPE OF THE SYSTEM

This section will show the complete prototype of the IoT-Based Device For Real Time Server Room Environment Monitoring System. The prototype is built by assembling all of the listed hardware such as the microprocessor which is Raspberry Pi, DHT11 Sensor, Pocket Lab Voyager Sensor.

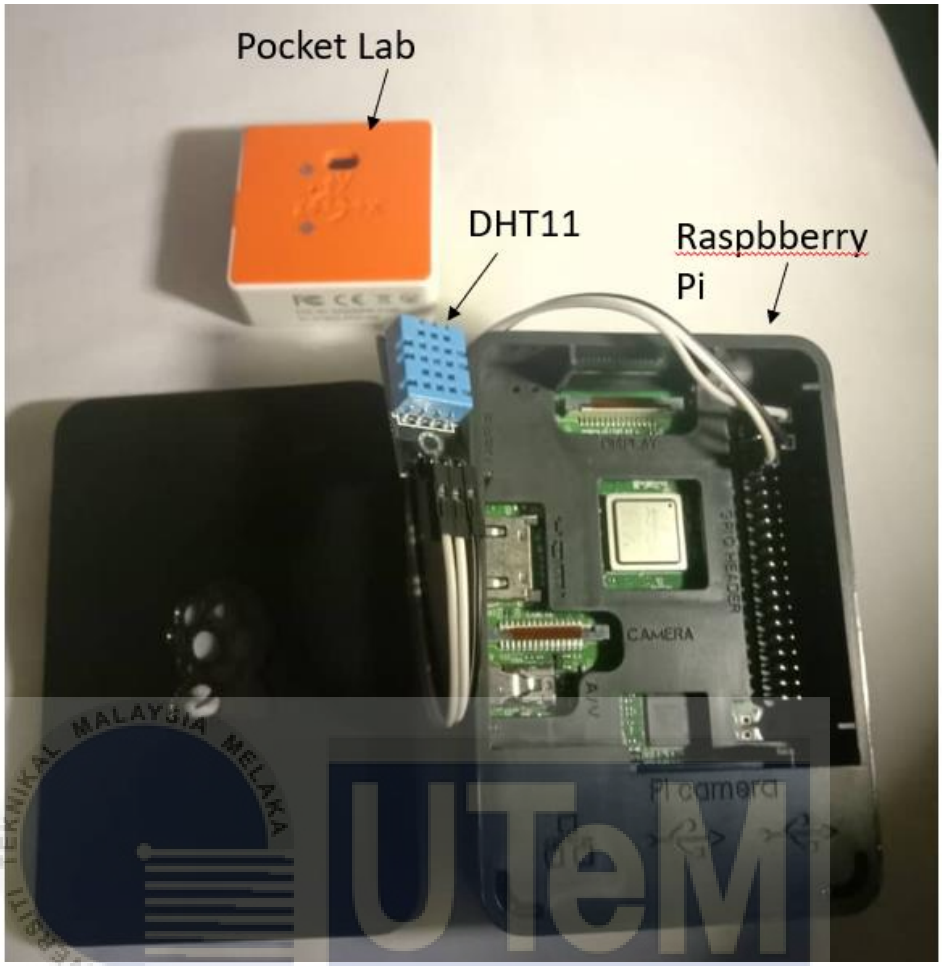

Figure 5. 10: Overall view of the prototype

5.4 IMPLEMENTATION STATUS

This section explains about the progress of the development status for each component and configurations.

Table 5. 1: Implementation Table Status

No	Component	Description	Duration
1	Assemble the hardware	Process of gathering all the required hardware that will be used for building the prototype device	7 days

2	Building the prototype	Process of connecting the Raspberry Pi and sensors by using jumper wires.	5 days
3	Configuration of software environment	Process of configuration the Raspberry Pi software with required libraries, Telegram and Pocket Lab Application.	15 days
4	Implementation of source code and integrating with prototype	Process of writing the source code in order to make the system align with the prototype device and functioning as the project proposed.	20 days

5.5 CONCLUSION

This chapter mainly shows the realization of the project, including the configuration of the sensor development environment and the monitoring part of the project. This setup clearly shows the connection between the sensor and the Raspberry Pi microcontroller. The software configuration of the sensor implemented in the Raspberry Pi.

The Telegram app and Pocket Lab app are also explained in detail in the chapters that help the device monitor and control sensor readings. At the same time, the Pocket Lab app collects data and displays the data in the form of graphs on the website. These two applications play an important role in the project by monitoring and analyzing sensor readings. In the next chapter, you will focus on testing equipment. The prototype will be used to test the functionality of the device.

CHAPTER 6

TESTING

6.0 INTRODUCTION

In this chapter, the various rules and activities performed on the fully deployed prototype will be explained as part of the IoT-Based Device for Real Time Server Room Environment Monitoring System. This chapter will describe the test method, analysis and test phase results according to the goals to be achieved by the test. This testing phase will focus on the goal to achieve the entire goal described in the previous chapter. The entire method will detail the conditions, the hardware and software involved, the expected results of the test, and the final result of the test.

6.1 TEST PLAN

In this section, we will explain all the testing requirements, that is, the organization, including developers and users, the test environment that will define the location to be tested, and the definition of hardware, firmware configuration, preparation and training before testing. This section will also define the test plan to define the test phase cycle and the duration of the test to be performed.

6.1.1 Test Plan

For this section, a brief description of the list of people involved in the testing process and the definition of who will be directly involved in testing the project during the testing process will be given. Specifically, in this project, there are two types of people directly involved in system testing, they are developers and users.

First of all, the developer is the person responsible for developing the entire system, managing the hardware, installing and configuring the software, ensuring that all system functions are working well, without errors or complexity.

In this testing phase, the developer will define the test subject that will be used and building a test case to ensure all of the testing case meets the expected results.

Second, the user is the person who tests the functionality of the prototype, acts as a test guide, defines the prototype, and gives feedback. The user will test the prototype, which will determine the test result, allowing the developer to identify errors and complications to correct.

6.1.2 Test Organization

In this project, the location and environment testing are to be carried out. The location of the testing is advised to be in a close and air-conditioned room as the sensors must capture the humidity and temperature of a server room. In this case, a room that is air conditioned is a suitable place to be a testing location. The room must also be in a place where there is a Wi-Fi connection nearby as to allow the Raspberry Pi to send the data to the user through Telegram Application. Therefore, a telecommunication signal availability must be considered.

For the hardware, the connectivity of the sensors and microcontroller must be tested in order to ensure the availability and the functionality of both hardware. In order for the sensor to record data properly, the microcontroller, Raspberry Pi, must be active to record and execute the configuration code of the IDE. Both of the sensors will work smoothly when the connection to the microcontroller is valid and tested.

6.1.3 Test Environment

Table 6. 1: List of Scenario

Scenario	Threshold	
	Temperature (C)	Humidity (%)
Scenario 1: Humidity and Temperature is in normal condition	18 - 27	40-55
Scenario 2: Humidity and Temperature is in high condition	27-65	35-60

Scenario 3: Humidity and Temperature is in low condition	0-18	10-35

Table 6.1 refer to the list of scenarios and the threshold of the humidity and temperature that was set for each of the scenarios. In this project, data transfer from prototype and telegram applications will be considered. Initially, the DHT11 sensor will send data from the Raspberry Pi every 10 seconds, which is essential for tracking the data recorded by the sensor. The data transmission cycle is once every 10 seconds, depending on the availability of the sensor. For any error in the sensor, the microcontroller will stop the data transmission cycle, allowing the developer to fix any errors in the prototype, and once the prototype is available, the cycle will start again.

In this project also, the data transfer from the Pocket Lab will also be considered. The sensor will send data every 10 seconds to the user phone. It can also send the data every 5 minutes as we are able to set the sensor settings as we see fit in this project. Any errors in sensors, the user's phone will stop the cycle of the data transfer, allowing the developer to troubleshoot any errors inside the prototype, starting the cycle back once the prototype is available.

In order to demonstrate the functionality of the prototype of the accuracy of data taken by the sensors, a series number of testings' is taken place. In each scenario, the data taken may various for each time. Based on the data taken, the status of the room and the alert notification may also change depending on the humidity and temperature. The alert notification is sent to the Telegram application from the user's phone. The status of the room is based on the threshold of humidity and temperature that is already discussed in the previous chapter for the optimum room condition. For a normal server room condition, the optimum humidity is between 40% to 55% and the temperature is between 18 C to 27 C. In a high temperature condition such as between 27 C to 65 C, the humidity will be shown between 35% to 60%. In a low temperature room such as between 0 C to 18 C, the humidity will be around 10% to 35%. According to (Makesh Balasubramani, 2017), if the air temperature is 24 °C (75 °F) and the relative humidity is zero percent, then the air temperature feels

like 21 °C (69 °F). Thus we can apply this threshold as our main point for the sensors' threshold

In this project, we made up into 3 scenarios for the testings' in order to fully operate on how the sensor works and test whether the threshold that the developer set up is in running or not. At the same time, we will also test the alert notifications on the user's phone. The first scenario will be in a normal condition where the humidity and temperature are in normal condition. Supposedly, the normal condition is within 18 ℃ to 27 ℃ of temperature and between 40% to 55% of humidity. The second scenario will be in a hot condition where the humidity and temperature is above the set threshold. The third scenario will be in a cold condition where the temperature and humidity will be below the threshold.

For the first scenario, we will test the alert notification of the user's phone based on threshold of the humidity and temperature that is set by the developer. The prototype will capture the humidity and temperature as per programmed. In this scenario, the temperature and humidity are within the range of threshold that is set by the developer. The test is conducted three times in order to get the consistency by the prototype result. The number of testing's is repeated three times in order to get the most precise data and so we can average out any data that we got from the sensors. In this scenario, we are going to see the alert notification from the user's phone and justify whether there is any change of output coming from the alert notification. The Pocket Lab sensor is inserted into the table as a range accuracy whether the data that we receive from the DHT11 sensor is working properly as intended or not. In this scenario also, the normal condition is within 18 ℃ to 27 ℃ of temperature and between 40% to 55% of humidity.

Table 6. 2: Scenario 1: Humidity and Temperature is in normal Condition

Number of testings'	Pocket Lab Sensor (Verification for data received)		Time Interval (seconds)	DHT 11 Sensors		Alert Notification by Telegram Application
	Temperature (Celsius)	Humidity (%)		Temperature (Celsius)	Humidity (%)	

1	26	46	10	25	46	Room Condition is normal
2	26	45	10	25	45	Room Condition is normal
3	27	44	10	25	46	Room Condition is normal

To visualize the data from the prototype based on this scenario, the prototype has been tested according to the data from the table 6.1. Below shows the result from the prototype's Terminal and Telegram Application.

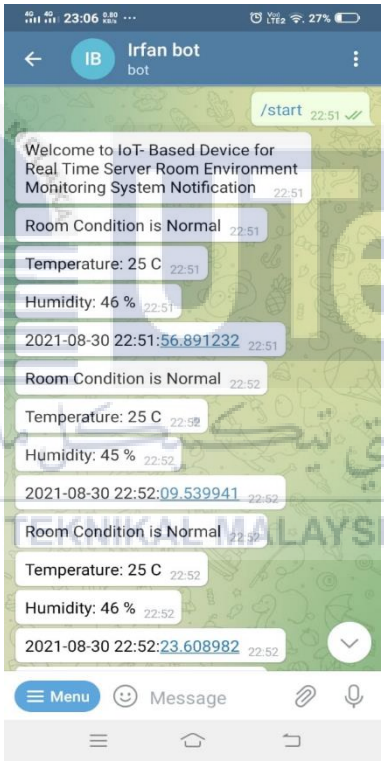

Figure 6. 1: Telegram Application

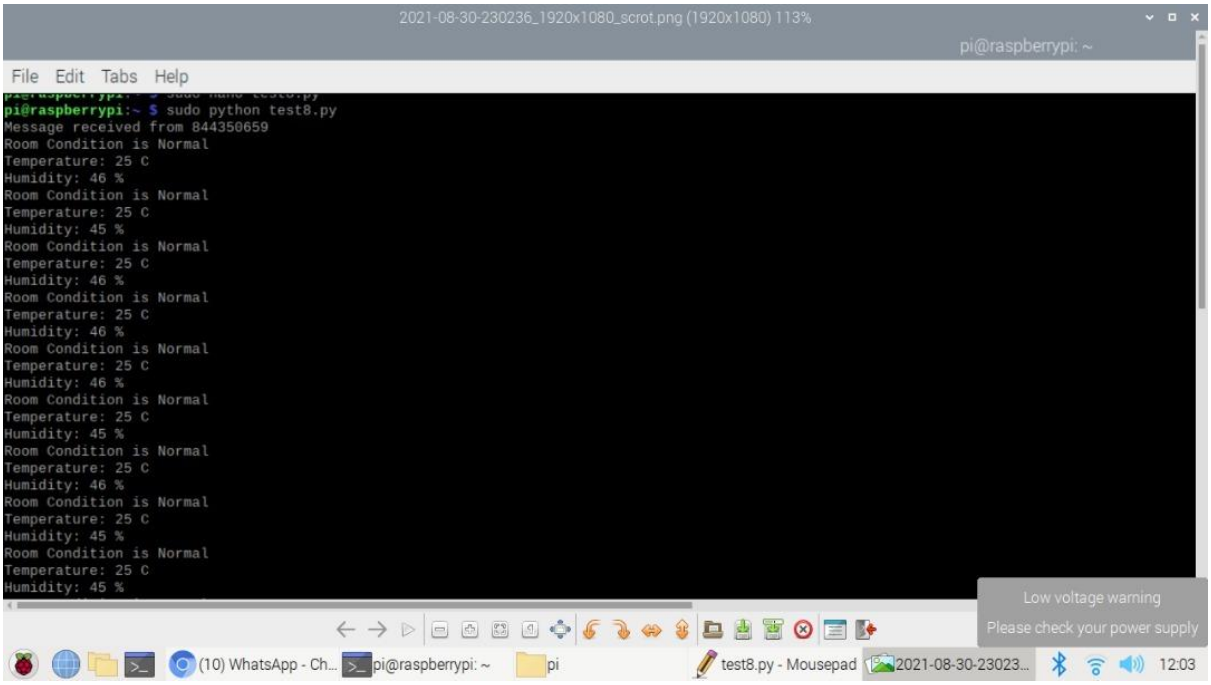

Figure 6. 2: Raspberry Pi Terminal

Based on the Figure 6.1 and Figure 6.2, it shows that both shows the correct Alert notification to the user.

- As we can see from table 6.2, the temperature is in a normal range and the humidity is also in normal range. Based on that, the temperature and humidity are within the threshold
- The temperature is above the 18 and below 27
- The humidity is above 40 and below 55.
- Because of it, the status of the room should be normal as it is in the normal condition of the room.
- Furthermore, the alert notification for the user's phone should be as expected per programmed by the developer.

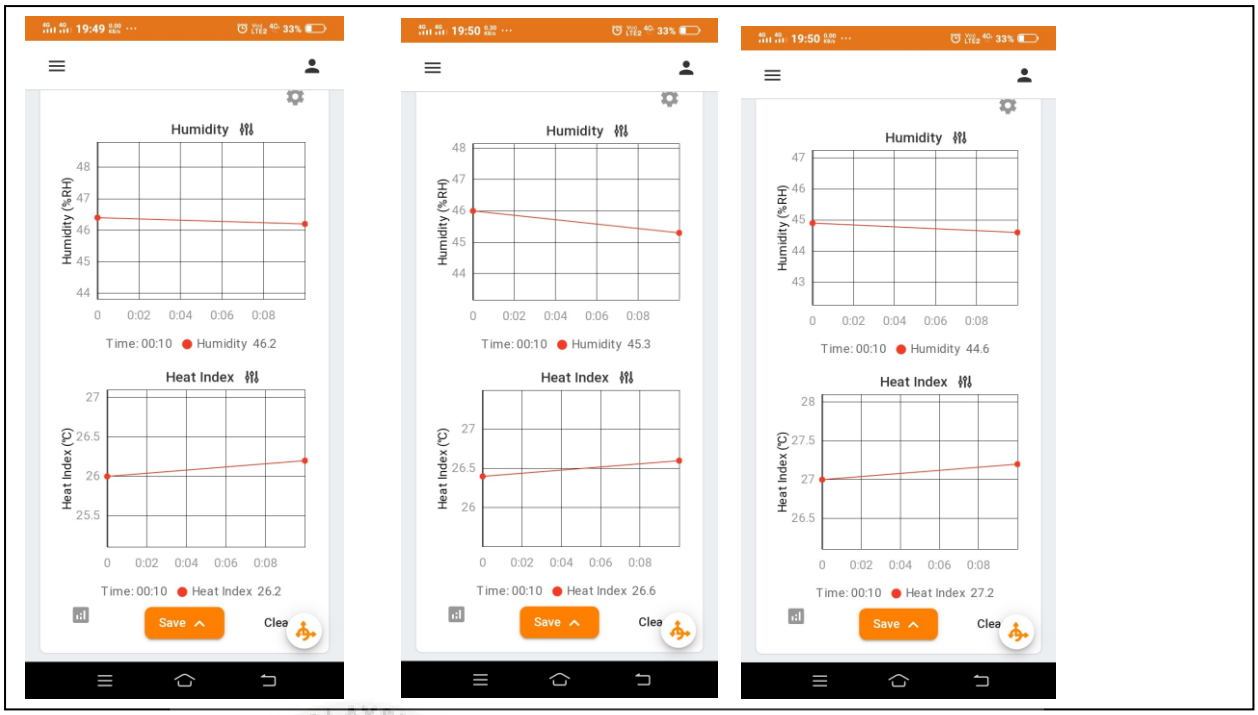

Figure 6. 3, Figure 6. 4, Figure 6. 5: Visualize data from the Pocket Lab Application

Figure 6.3, Figure 6.4 and Figure 6.5 shows the data that is received from the Pocket Lab and integrated into a graph to visualize in order for the user to understand easily what the data is received from the sensor. Figure 6.3 is from the scenario 1 testing 1. As we can see for the humidity graph, the graph shows the data is between 46% and 47%. This means during this time, the humidity is in the normal condition for a normal room condition. Meanwhile, the temperature is shown rising between 26 C to 26.5 C but as shown, it still within the threshold of an optimum condition for a normal condition room. Figure 6.4 is from scenario 1 testing 2. As we can see for the humidity graph, the graph is showing gradually falling and shows the data is between 45% and 46%. This means during this time, the humidity is still in the normal condition for a normal room condition. Meanwhile, the temperature is shown rising between 26.4 C to 26.6 C but as shown, it still within the threshold of an optimum condition for a normal condition room. And lastly Figure 6.5 is from scenario 1 testing 3. The humidity graph is shown gradually falling but it is still within the range of a normal condition for a normal room condition. Meanwhile, the temperature is shown rising between 27 C to 27.5 C but as shown, it still within the threshold of an optimum condition for a normal condition room.

For the scenario 2, the room is set in a certain condition to test the alert notification for the user's phone. In this case, the temperature and humidity are set high condition. The condition of the room is not optimum for the sake of testing the alert notification of the user's phone. In this scenario, the temperature should be between 27 C to 65 C, and the humidity will be shown between 35% to 60%.

Table 6. 3: Scenario 2: Humidity and Temperature is in High Condition

Number of testings'	Pocket Lab Sensor (Verification for data received)		Time Interval (seconds)	DHT 11 Sensors		Alert Notification by Telegram Application
	Temperature (Celsius)	Humidity (%)		Temperature (Celsius)	Humidity (%)	
1	67	35	10	64	33	Room Humidity is High Room Temperature is High
2	65	38	10	59	32	Room Humidity is High Room Temperature is High
3	64	39	10	59	32	Room Humidity is High Room Temperature is High

To visualize the data from the prototype based on this scenario, the prototype has been tested according to the data from the table 6.2. Below shows the result from the Telegram Application.

Figure 6. 6: Telegram Application

Based on the Figure 6.6, it shows that both shows the correct Alert notification to the user.

- As we can see, based on the table 6.3, we can that both the temperature and humidity is above the set threshold.
- The temperature is above 27.
- The humidity is above 35.
- Because of it, the status of the room should be not normal as it is not in the normal condition of the room.
- Furthermore, the alert notification for the user's phone should be as expected per programmed by the developer.

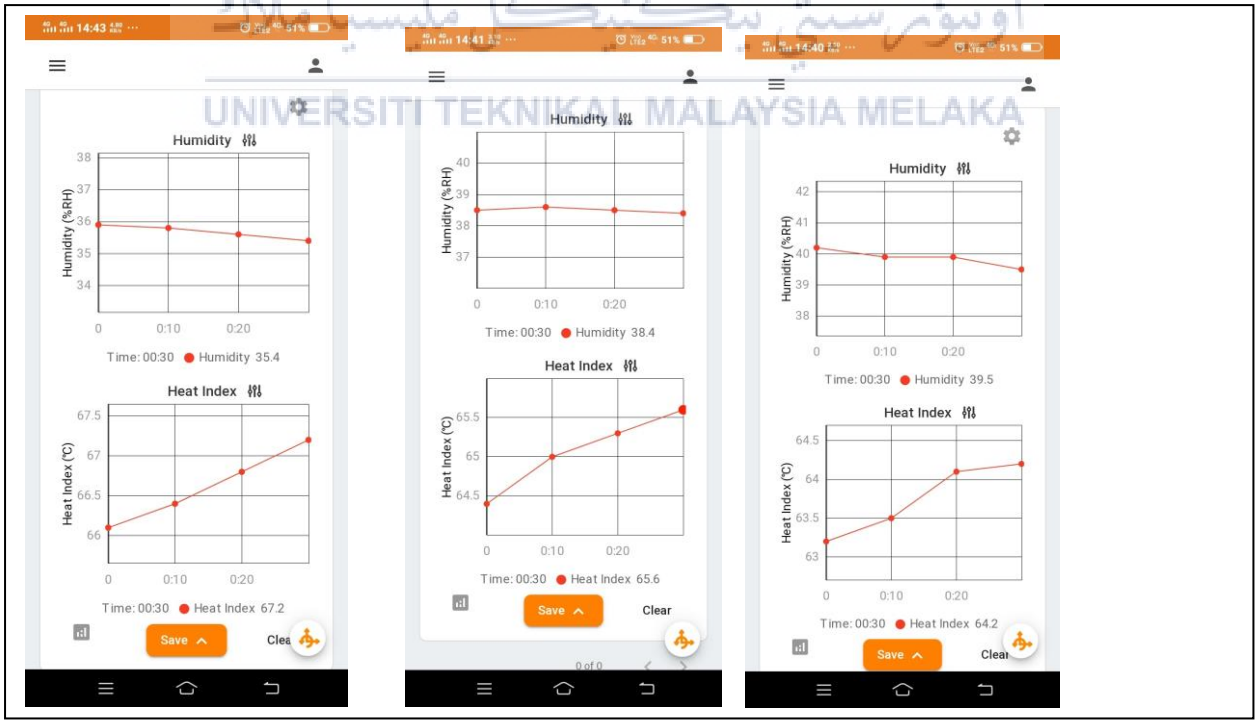

Figure 6. 7, Figure 6. 8, Figure 6. 9: Visualize data from the Pocket Lab Application

Figure 6.7, Figure 6.8 and Figure 6.9 shows the data that is received from the Pocket Lab and integrated into a graph to visualize in order for the user to understand easily what the data is received from the sensor. Figure 6.7 is from the scenario 2 testing 1. As we can see for the humidity graph, the graph is showing decreasing from 36% to 35%. This means during this time, the humidity is in low humidity as it is not normal for a normal room condition. Meanwhile, the temperature is shown increasing between 66 C to 67 C. With this graph, we can know that the temperature is increasing as it is telling that the room is in a high temperature. Figure 6.8 is from scenario 2 testing 2. As we can see for the humidity graph, the graph is still in between 38% and 39%. This means during this time, the humidity is still in low humidity as it is not normal for a normal room condition. Meanwhile, the temperature is shown is still falling between 64.5 C to 65.5 C. With this graph, we can know that the temperature is slowly increasing as it is telling that the room is in a high temperature. And lastly Figure 6.9 is from scenario 2 testing 3. As we can see for the humidity graph, the graph is showing gradually decreasing and shows the data is between 40% and 39%. This means during this time, the humidity is in low humidity as it is not normal for a normal room condition. Meanwhile, the temperature is shown falling between 63 C to 64 C. With this graph, we can know that the temperature is rising quickly as it is telling that the room is in a high temperature. The data given is separated into 3 trials in order to get the average number of humidity and temperature from the room.

For the third scenario, the room is set in a certain condition to test the alert notification for the user's phone. In this case, the temperature and humidity are set in a low condition. The condition of the room is not optimum for the sake of testing the alert notification of the user's phone. In this scenario, the room temperature should be between 0 C to 18 C, the humidity will be around 10% to 35%.

Table 6. 4: Scenario 3: Humidity and Temperature is in Low Condition.

Number of testings'	Pocket Lab Sensor (Verification for data received)	Time Interval (seconds)	DHT 11 Sensors	Alert Notification by Telegram Application

	Temperature (Celsius)	Humidity (%)		Temperature (Celsius)	Humidity (%)	
1	7	25	10	10	25	Room Humidity is Low Room Temperature is Low
2	9	26	10	12	27	Room Humidity is Low Room Temperature is Low
3	9	26	10	12	27	Room Humidity is Low Room Temperature is Low

To visualize the data from the prototype based on this scenario, the prototype has been tested according to the data from the table 6.3. Below shows the result from the Telegram Application.

Figure 6. 10: Telegram Application

Based on the Figure 6.10, it shows that both shows the correct Alert notification to the user.

- As we can see from table 6.4, the temperature and humidity is not in normal range.
- The temperature is below 18.
- The humidity is below 35.
- Because of it, the status of the room should be not normal as it is not in the normal condition of the room.
- Furthermore, the alert notification for the user's phone should be as expected per programmed by the developer.

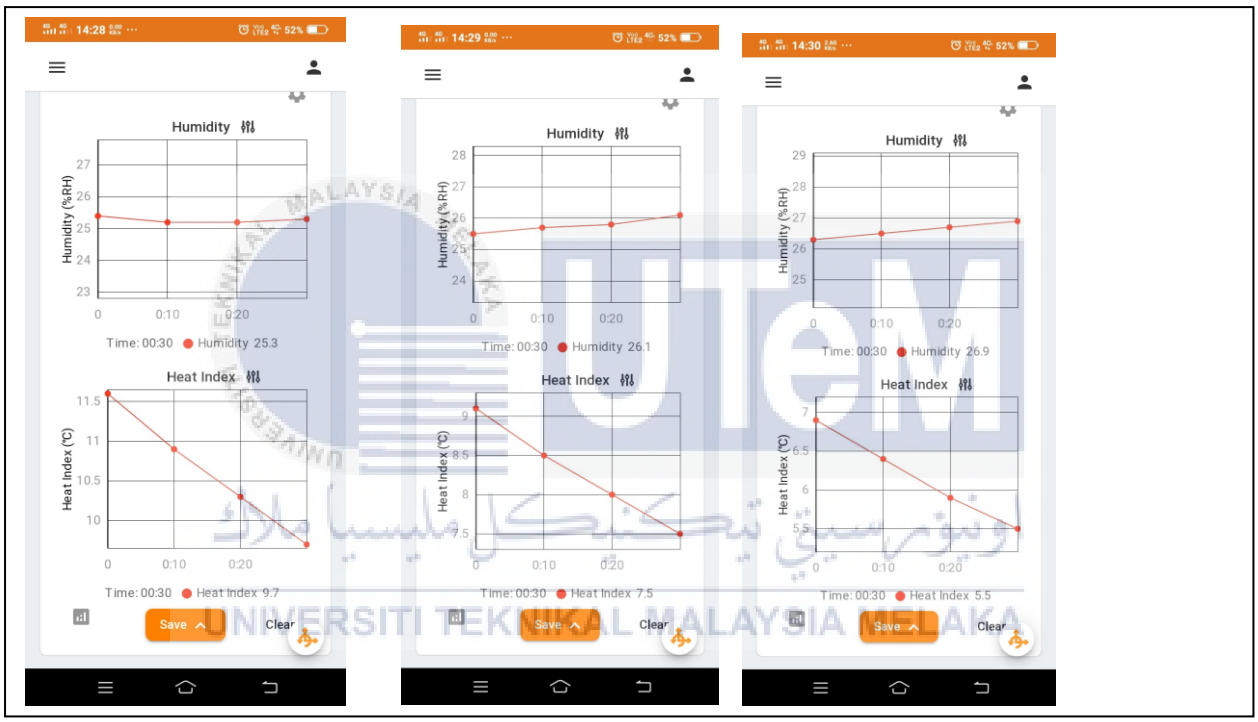

Figure 6. 11, Figure 6. 12, Figure 6. 13: Visualize data from the Pocket Lab Application

Figure 6.11, Figure 6.12 and Figure 6.13 shows the data that is received from the Pocket Lab and integrated into a graph to visualize in order for the user to understand easily what the data is received from the sensor. Figure 6.11 is from the scenario 3 testing 1. As we can see for the humidity graph, the graph is showing constant value and shows the data is between 25% and 26%. This means during this time, the humidity is in low humidity as it is not normal for a normal room condition. Meanwhile, the temperature is shown falling between 11.5 C to 9 C. With this graph, we can know that the temperature is slowly falling as it is telling that the room is in a low temperature. Figure 6.12 is from scenario 3 testing 2. As we can see for the humidity graph, the graph is gradually increasing

and shows the data is between 25% and 26%. This means during this time, the humidity is still in low humidity as it is not normal for a normal room condition. Meanwhile, the temperature is shown is still falling between 9 C to 7.5 C. With this graph, we can know that the temperature is slowly falling as it is telling that the room is in a low temperature. And lastly Figure 6.13 is from scenario 3 testing 3. As we can see for the humidity graph, the graph is showing gradually increasing and shows the data is between 25% and 26%. This means during this time, the humidity is in low humidity as it is not normal for a normal room condition as it has low humidity. Meanwhile, the temperature is shown falling between 7 C to 5.5 C. With this graph, we can know that the temperature is slowly falling as it is telling that the room is in a low temperature. The data given is separated into 3 trials in order to get the average number of humidity and temperature from the room.

6.2 TEST STRATEGY

Generally, a test strategy is an outline that describes the testing approach of the software development cycle. For this project, in order to undergo the testing phase, both of the input-output of the Raspberry Pi will be monitored. In order to show the data amount captured by the sensors, the data will be shown on the Raspberry Pi terminal, allowing the developer to monitor both prototype and the configuration code at the same time.

Therefore, White-Box testing has proven to be the most suitable testing technique as it allows testing of the internal structure, design and configuration code of the prototype to verify the input and output data flow. Through this testing strategy, the usability and usability of the testing phase can also be improved. Testing can be performed at the software development unit, integration, and system level. One of the basic goals of White-Box testing is to verify the application workflow. It involves testing a series of predefined inputs with the expected or desired output so that when a particular input does not produce the expected output, it encounters an error.

For this project, the testing started by starting the microcontroller, which will detect the humidity and temperature from the sensors as the prototype started. After 1 seconds, the microcontroller will start sending data to the table form. The time for the data is captured is taken, also the room humidity and the room temperature are also taken and put into the table form.

6.2.1 Classes of Test

i. Understand the source code

- To start the testing phase, the developer must understand the configuration code in the Raspberry Pi and ensure that every code in the IDE works according to the programming purpose of the program.

- The developer must ensure that the software configuration is correctly configured before testing the prototype to the user, so that the testing phase runs smoothly without any internal problems with the program and the prototype.

ii. Create Test Cases and Execute

- This class is considered as a class where the testing phase involves the application proper flow and structure.

- The prototype is ready to be tested by user in this class, allowing the developer to create a test case beforehand, making it as a guideline for the testing phase to achieve the expected results.

6.3 TEST DESIGN

6.3.1 Test Description

In this part, the test cases of the testing phase for this project will be constructed here. All of the test cases will be started by the functionality of the prototypes, the condition of the hardware, the execution steps, the expected results and the actual results. All of the data will be tabulated and will be analysed in details.

Table 6. 5: Test Case I – Microcontroller Connectivity

Test Functionality	Testing the connectivity of microcontroller (Raspberry Pi) to a monitor
Precondition	- USB adapter from Raspberry Pi will be connected to a power supply. - Allowing the Raspberry Pi to connect to the monitor through a HDMI cable.
Execution steps	i. The monitor detects the Raspberry Pi that is connected. ii. Raspberry Pi can be successfully be used and can be seen putting code and configurations into it.

Table 6. 6: Test Case II – DHT11 Sensor Connectivity

Test Functionality	Testing the DHT11 Sensor connectivity with the pre-compiled configuration code.
Pre-condition	- Configuration code has been uploaded to Raspberry Pi. - The sensors are connected to the Raspberry Pi according to the schematic diagram. - The sensor is connected to their designated pin on the Raspberry Pi.

Execution Steps	- Once the PIN is connected successfully between the DHT11 and Raspberry Pi using a jumper cable, we can start with the configurations for the sensor. - Install the correct library in order to use the sensor. - Once the library is installed, insert configurations in the Raspberry Pi terminal using a Python Language. - Once the configuration is complete, we can start and use the DHT11 sensor to get the humidity and temperature data. - Type in the correct command in order to use the DHT11 sensor.
Expected Results	- The terminal will show the humidity and temperature that is taken from the sensor. - The sensor will capture the data from the room that is placed in
Error Message	None
Results	The Raspberry Pi terminal updates the data that is taken from the sensor.

Table 6. 7: Test Case III – Pocket Lab SensoConnectivity

Test Functionality	Testing the connectivity of the Pocket Lab to the user's phone

Pre-condition	- The Pocket Lab has battery life in order to use it. - Download the Pocket Lab Application on the user's phone - The user phone is connected to the Pocket Lab sensor via Bluetooth.
Execution Steps	- Open Pocket Lab application on the user's phone. - Once it is connected, it will collect the data from the sensor - The sensor will collect data based on the interval time set by the user
Expected results	The Pocket Lab sensor will show the data trough graph visual.
Error Message	None
Result	Pocket Lab is successfully connected with the user's phone.

Table 6. 8: Test Case IV – Alert Notification Connectivity

Test Functionality	Testing the connectivity of the Telegram Alert Notification to the user's phone from Raspberry Pi.

Pre-condition	- The Telegram Application is installed on the user's phone - The Telegram Bot is already configured on the user's phone - The Telegram Bot is connected with the Raspberry Pi using the terminal configuration.
Execution Steps	- Install Telegram on the user's phone. - Get Telegram bot token for Raspberry Pi configurations - Type in the configurations in the Raspberry Pi terminal and put in the configurations - Connect the Raspberry Pi with the Telegram token
Expected results	The Telegram Application will send notification alert to the user's phone
Error Message	None
Result	Alert Notification is successfully connected with the user's phone.

6.4 TEST RESULTS AND ANALYSIS TESTING

After all of the hardware and software for this project has been setup according to all of the implementation phase from previous chapter, this section will demonstrate the testing phase for all of the test case listed from previous subsections.

6.4.1 Test Case I – Microcontroller Connectivity

For the initial setup, the fully-configured prototype will be connected to the monitor through HDMI cable. Each Raspberry Pi are accompanied with a power plug, allowing the Raspberry Pi to be plugged in to power up the sensors. The Raspberry Pi can be powered up by two methods, which are through USB cable or External power supply, such as Power Supply Adaptor, batteries, power banks, etc.

For this project, the Raspberry Pi will be connected to the power supply adaptor, which will power up the microcontroller and allowing the it to be configured. Once the Raspberry Pi is connected to the power supply and the monitor, The Raspberry Pi will automatically install itself with a new Raspberry Pi Operating System and allowing us to prompt configurations in it.

For this test case, the execution steps begin by opening the Raspberry Pi terminal once the Raspberry Pi is plugged to power supply. Before power it on, make sure to connect the monitor using a HDMI cable. Also, insert a micro-SD card to the Raspberry Pi. Once it turns on, let the Operating System installed the latest version of the OS in order to use the microcontroller.

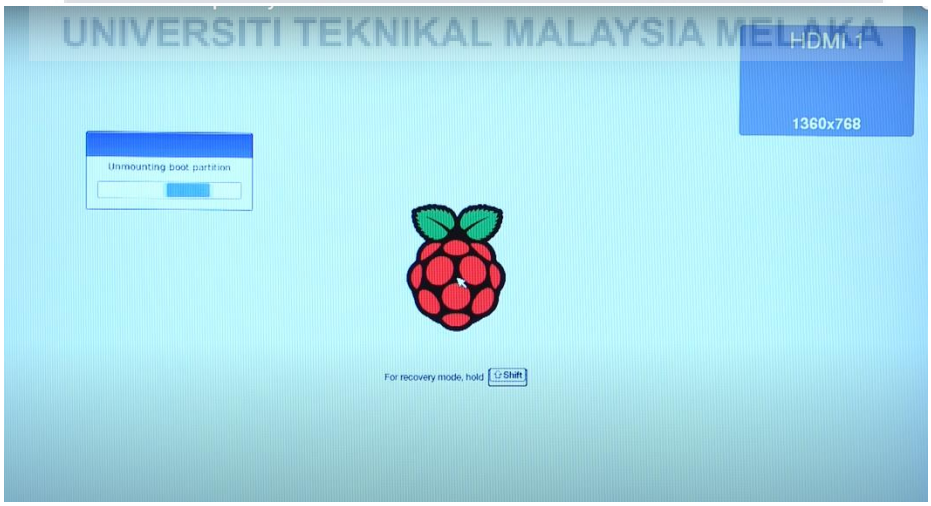

Figure 6. 14: Raspberry Pi Powered on

6.4.2 Test Case II – DHT11 Sensor Connectivity

For this test case, it can only be done after Test Case I meets the expected outcome, allowing the sensor to integrate with both Raspberry Pi and Raspberry Pi IDE. In general, there are two variants of the DHT11 that are likely to come across. One is a three pin PCB mounted module and the other is a four pin stand-alone module. In this project, we are using a three pin DHT11 as it the three pin DHT11 and want to output the humidity and temperature to an SSH terminal, wire it like this:

Figure 6. 15: DHT 11 is connected to the Raspberry Pi according to the schematic

Figure 6.15 shows the DHT11 sensor is connected. Once it is connected, we can start with the configurations of the DHT11 sensor. We will be using Python Language for the Raspberry Pi terminal. We'll be using the Adafruit DHT11 Python library. We can download the library using Git by enter this at the terminal:

i) gitclone https://github.com/adafruit/Adafruit_Python_DHT.git

ii) cd Adafruit_Python_DHT

iii) sudo apt-get install build-essential python-dev

iv) `sudo python setup.py install`

Once the library is installed, put this python command to get an output on the terminal:

```
#!/usr/bin/python
import sys
import Adafruit_DHT

while True:
```

```
    humidity, temperature = Adafruit_DHT.read_retry(11, 4)

    print 'Temp: {0:0.1f} C  Humidity: {1:0.1f} %'.format(temperature, humid
ity)
```

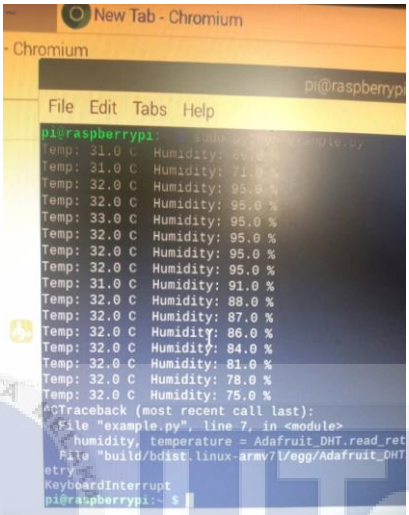

Figure 6. 16: Testing DHT11 sensors

Figure 6.16 shows the terminal where the sensors send the data to the Raspberry Pi. Once the data is shown, the data will automatically be inserted into a csv file where the data is converted into a table where the user can view it. Once the sensors are ready to received the data, we can now proceed with the script for the Raspberry Pi in which we can set the threshold humidity and temperature for the sensor in order to get the output of “The Room Condition is Normal”. We also inserted the script where if the humidity and temperature is below the threshold, it will create an output “The Room Humidity is low” and if the humidity and temperature is above the threshold, the output will be “The Room Temperature is High”.

```
if humidity > 55:
        print("Room Humidity is High")
            print("Humidity: %d %%" % humidity)
            bot.sendMessage(chat_id, 'Room Humidity is High')
bot.sendMessage(chat_id,"Humidity: %d %%" % humidity)
    elif humidity < 40:
        print("Room Humidity is Low")
            print("Humidity: %d %%" % humidity)
            bot.sendMessage(chat_id,"Humidity: %d %%" % humidity)
bot.sendMessage(chat_id, 'Room Humidity is Low')
    if temperature < 18:
        print("Room Temperature is Low")
            bot.sendMessage(chat_id, 'Room Temperature is Low')
        bot.sendMessage(chat_id,"Temperature: %d C" % temperature)
        print("Temperature: %d C" % temperature)
    elif temperature > 27:
        print("Room Temperature is High")
            bot.sendMessage(chat_id, 'Room Temperature is High')
            bot.sendMessage(chat_id,"Temperature: %d C" % temperature)
            print("Temperature: %d C" % temperature)
```


Figure 6. 17: Script for DHT11 sensors in Raspberry Pi terminal

```
if temperature < 27 and temperature > 18 and humidity > 40 and humidity <55:
        print("Room Condition is Normal")
        bot.sendMessage(chat_id, 'Room Condition is Normal')
        bot.sendMessage(chat_id,"Temperature: %d C" % temperature)
            bot.sendMessage(chat_id,"Humidity: %d %%" % humidity)
        print("Temperature: %d C" % temperature)
            print("Humidity: %d %%" % humidity)


    bot.sendMessage(chat_id, str(datetime.datetime.now()))
    time.sleep(10)

while 1:
            time.sleep(10)
while 0:
            time.sleep(10)
```

Figure 6. 18: Script for DHT11 sensors in Raspberry Pi terminal (cont)

Figure 6.17 and Figure 6.18 shows the script of how the coding of setting the threshold for the DHT11 sensors in the Raspberry Pi terminal. The file is saved in a python file. Once that is configured, we can also insert the telegram coding into the Raspberry Pi for the user to get notifications from the user Telegram application.

```
def handle(msg):
    global telegramText
    global chat_id
    chat_id = msg['chat']['id']
    telegramText = msg['text']
    print('Message received from ' + str(chat_id))
    if telegramText == '/start':
        bot.sendMessage(chat_id, 'Welcome to IoT- Based Device for Real Time
Server Room Environment Monitoring System Notification')
    while True:
        main()
bot = telepot.Bot('1808061189:AAEWrqkun1myu0qtj53FbnfPLsUXta1ur3I')
MessageLoop(bot, handle).run_as_thread()
```

Figure 6. 19: Script for Telegram Application

Figure 6.19 shows the snippet of the Raspberry Pi to send notification to the user's phone. Once it is configured, the user can received the alert notification once the python file is running and the user can start receiving data by typing the "/start" command in the Telegram Application.

6.4.3 Test Case III – Pocket Lab Sensor Connectivity

In order to produce a statistical graph of data, Pocket Lab application can generate a graph based on the user's preference. It only requires the data sent to the sensor, onto the user's phone. For this project, Pocket Lab will be used to generate a statistical graph, enable the monitoring ability regarding the humidity and temperature of the room. The Pocket Lab sensor can also record the data and will send the data to the user's phone **every FIVE (5)** seconds. The graph inside the Pocket Lab application will be updated every 5 seconds.

In this test case, a series of data will be sent to user's phone, sending a set of data to the user' phone in order to allow the Pocket Lab application to generate the statistical graph for the project. Initially, this project will generate TWO (2) graph, which are;

- The humidity of the room.
- The Heat Index or temperature of the room.

Pocket Lab sensor is used in this project to get the visualize data of the room that is conducted in. In order to establish a connection between the sensor and the user, the user must first install the Pocket Lab application that is available on the app store. Once they have installed it, they can connect the sensor via Bluetooth connection.

Below is the list of executing techniques for this test case in details;

i. Open the PocketLab app.

ii. Click "Connect a PocketLab" - a connection window should appear.

iii. Turn on the PocketLab then the name of the PocketLab should appear in the connection window.

iv. Click on the name of the PocketLab device to connect

v. The PocketLab should connect automatically and we should now see data streaming directly to the graph

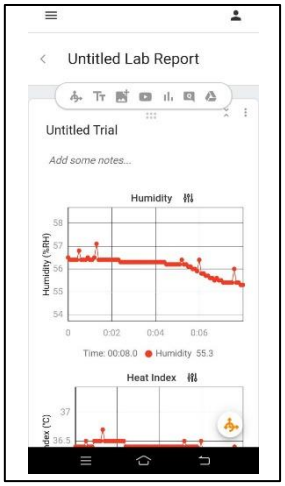

Figure 6. 20: Pocket Lab Application for connecting with user's phone

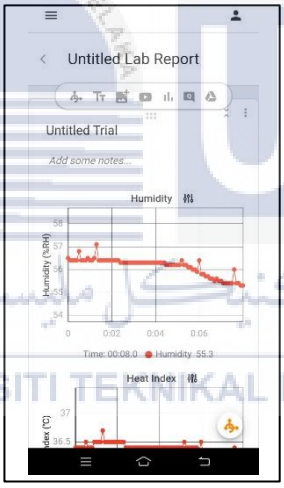

Figure 6. 21: Data Graph in Pocket Lab Application

Based on the Figure 6.8, the Pocket Lab sensor has connected to the user's phone. The enables to use it to be monitored. This also indicates that this test case is successfully returns the expected results from it, which are testing the connectivity of the prototype with Pocket Lab application and automatically generate a statistic data presentation about the usage of the prototype. Thus, the expected results from this test case are achieved successfully.

6.4.4 Test Case IV – Alert Notification Connectivity

For this test case, it can only be done after Test Case II meets the expected outcome, allowing the sensor to integrate with the Raspberry Pi and the DHT11 sensor. Telegram bot is an application where it can receive notification from the Raspberry Pi to the user's phone. This can notify the user whenever the room condition is above or below the threshold that is set by the developer.

Below are the list of executing techniques for this test case in details;

i. Install Telegram on the user's Phone. Go to App Store (iPhone) or Play Store (Android), download and install Telegram on the user's phone. Telegram reserves a special kind of accounts for machines, called bot accounts.

ii. Text /newbot to BotFatherTurn on the Telegram Application. At the end of process, we will be given a token. This token represents the bot account. We are going to put this token on the Raspberry Pi.

iii. Install Telepot on Raspberry Pi

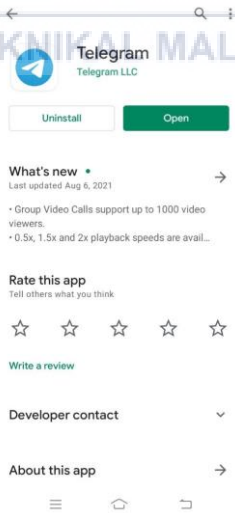

Figure 6. 22: Install Telegram Application on the user's phone

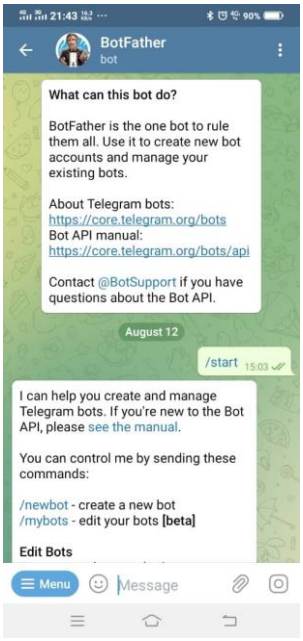

Figure 6. 23: Search for Telegram "BotFather" on the Telegram Application

iv. After installing the Telepot into the Raspberry Pi, we can proceed with the configurations of the sensors where it will detect if the room is in the optimum condition or not. We set the threshold in order to get the notification from the Raspbery Pi.

Figure 6. 24: Telegram Bot was created successfully

v. Once we have named the file we have just saved, type python and the save file name to run the bot. Open Telegram on the user's phone, search for the bot using its name or username. Text it /start and it will start sending notifications to the user

In conclusion, this test case returns the expected result, which indicating that the test case is successfully bring the expected outcome.

ANALYSIS TESTING

Based on the testing that was taken place from the scenario previously, we can now make an analysis table on how accurate the given data from both the DHT11 sensors and the Pocket Lab sensors. As stated before, we are going to use the average of data from 3 testings' that was taken from the sensors and use it as our analytical data to compare the accuracy and consistency of both sensors.

Firstly, for scenario 1 where the condition of the room is temperature is within the range for normal condition and also for the humidity. For scenario 2, the prototype and the pocket Lab device is place under a cold temperature and low humidity room to test the prototype alert notification the precision of data that both devices received. Scenario 3 is place under a condition where the humidity and temperature are under hot weather and high humidity.

Table 6. 9: Alert analysis of three scenario

True =✓; False =×

Scenario	Number of testings'	Alert Notification from Telegram Application
Scenario 1	1	✓
	2	✓
	3	✓
Scenario 2	1	✓
	2	✓
	3	✓
Scenario 3	1	✓
	2	✓
	3	✓

Based on this analysis, we can see that all of the testings' from all 3 scenarios came up with true (✓). This shows that the prototype is working as intended and the notification alert is producing results as intended by the developer. This also shows that the sensors are working as it should be.

6.5 DISCUSSION

For this chapter, testing phase is specifically performed and explained in details within this chapter. All of the testing case, hardware used, data involved are explained in this chapter. All of the results from the test cases will be discussed below.

Initially, all of the test cases listed from previous subsection are connected with each other, which are crucial that if one of the test cases are failed to achieve the expected results, then the other test cases after the failed one cannot be performed. Thus, if there is any error or misconception that contribute to the failure of the test case must be troubleshoot and correcting it in order to complete the testing phase. Final results for the overall project can only be obtained once all of the test cases listed achieved their expected results.

For the first test case, the connection between the Raspberry Pi and the Monitor are the most important thing in building the prototype. Once we connected the monitor and the Raspberry Pi, we can start to use the Raspberry Pi OS as it will install itself into it. Without Raspberry Pi OS, it is impossible to program and fuse the configuration code into the terminal. Thus, connecting both of these components are the crucial steps in order to allow more functionalities to be added into the prototype. And based on this test case, the Raspberry Pi OS has installed successfully, allowing the IDE to verify and upload the configuration code to the Raspberry Pi.

Table 6. 10: Test Case I – Microcontroller Connectivity

Test Functionality	Testing the connectivity of microcontroller (Raspberry Pi) to a monitor
Precondition	- USB adapter from Raspberry Pi will be connected to a power supply. - Allowing the Raspberry Pi to connect to the monitor through a HDMI cable.

Execution steps	iii. The monitor detects the Raspberry Pi that is connected. iv. Raspberry Pi can be successfully be used and can be seen putting code and configurations into it.

Second, for the second test case, the DHT11 sensors plays an important role in this project. The sensors are the key of this project, enabling the automation of the prototype to run the overall system. Thus, the configuration of the sensors is important in order to make sure that the sensors work properly. In this test case, the sensors successfully detect and record all of the required data for the prototypes, indicate that the sensors are configured correctly.

Table 6. 11: Test Case II – DHT11 Sensor Connectivity

Test Functionality	Testing the DHT11 Sensor connectivity with the pre-compiled configuration code.
Pre-condition	- Configuration code has been uploaded to Raspberry Pi. - The sensors are connected to the Raspberry Pi according to the schematic diagram. - The sensor is connected to their designated pin on the Raspberry Pi.

Execution Steps	- Once the PIN is connected successfully between the DHT11 and Raspberry Pi using a jumper cable, we can start with the configurations for the sensor. - Install the correct library in order to use the sensor. - Once the library is installed, insert configurations in the Raspberry Pi terminal using a Python Language. - Once the configuration is complete, we can start and use the DHT11 sensor to get the humidity and temperature data. - Type in the correct command in order to use the DHT11 sensor.
Expected Results	- The terminal will show the humidity and temperature that is taken from the sensor. - The sensor will capture the data from the room that is placed in
Error Message	None
Results	The Raspberry Pi terminal updates the data that is taken from the sensor.

For the third test case, in order to generate a statistical graph for the data, the Pocket Lab sensors needs to establish a connection with the user's phone as it will send the data to the Pocket Lab Application, which will be tabulated into pole of graph, indicating the humidity and temperature in the room. In order to ensure that the connection between the user's phone and the Pocket Lab established properly, this test case was performed to monitor the status and the availability of both software. As a result, the prototype successfully sent data to the Pocket Lab application without any complication, and it successfully tabulate the data in a graph, visualizing the data in a line graph.

Table 6. 12: Test Case III – Pocket Lab Sensor Connectivity

Test Functionality	Testing the connectivity of the Pocket Lab to the user's phone
Pre-condition	- The Pocket Lab has battery life in order to use it. - Download the Pocket Lab Application on the user's phone - The user phone is connected to the Pocket Lab sensor via Bluetooth.
Execution Steps	- Open Pocket Lab application on the user's phone. - Once it is connected, it will collect the data from the sensor - The sensor will collect data based on the interval time set by the user
Expected results	The Pocket Lab sensor will show the data trough graph visual.
Error Message	None
Result	Pocket Lab is successfully connected with the user's phone.

For the last test case, connection of Alert Notification from Telegram Application will allow user to get notified by the system if there is something wrong with the room temperature or humidity. In order to connect the

microcontroller to the Telegram Application, a token from the telegram was created in order to establish a connection with the Telegram application. Once the Raspberry Pi is configured and the user's phone has been successfully connected, all the user has to was just typed in the command that was programmed in and let the sensors on the microcontroller does it work.

Table 6. 13: Test Case IV – Alert Notification Connectivity

Test Functionality	Testing the connectivity of the Telegram Alert Notification to the user's phone from Raspberry Pi.
Pre-condition	- The Telegram Application is installed on the user's phone - The Telegram Bot is already configured on the user's phone - The Telegram Bot is connected with the Raspberry Pi using the terminal configuration.
Execution Steps	- Install Telegram on the user's phone. - Get Telegram bot token for Raspberry Pi configurations - Type in the configurations in the Raspberry Pi terminal and put in the configurations - Connect the Raspberry Pi with the Telegram token
Expected results	The Telegram Application will sends notification alert to the user's phone

Error Message	None
Result	Alert Notification is successfully connected with the user's phone.

6.6 CONCLUSION

In conclusion, this chapter shows how to conduct prototype testing based on the system's expected capabilities. This chapter introduces the test users, environment, and timetable, as well as the test plan for the test phase. This chapter also goes through the testing phase strategy and implementation in greater depth. According to user requirements. And while it's not possible to solve all the failures that we might find during the testing phase, it is possible to use the results from this phase to reduce the number of errors within the software program. The next chapter will discuss the overall summary of the report, including observations, improvements, and contributions to strengths and weaknesses.

CHAPTER 7

PROJECT CONCLUSION

7.0 INTRODUCTION

In this chapter, we would like to address the overall project and explain it from the beginning until the conclusion of the method. For any more development, the limitations and potential position of the project will also be identified. This would eventually boost machine efficiency and make it more reliable and effective. This strategy still has drawbacks, but there are several benefits as it can be modified for long term use.

7.1 PROJECT SUMMARIZATION

7.1.1 Project Objective

The objectives of this project that had been identified throughout the development of the project are listed below:

i. To study humidity and temperature in a server room. In order to design the monitoring system, research was done to gain literature review, information and all related details about the IoT-Based Device for Real Time Server Room Environment Monitoring System development in the literature review phase. The hardware and software components of the monitoring system were identified to be developed for this project. Therefore, the monitoring system composed of Raspberry Pi, Pocket Lab, DHT11 and software like Python Library, Telegram.

ii. The last objective of this project is to develop a system that can monitor the humidity and temperature of a server room by alert notification and statistical information. This project is successfully able to demonstrate the notification process when the humidity and temperature is above or below the threshold, user's phone is notified via Telegram.

7.2 PROPOSITION OF IMPROVEMENT

With the information to the vulnerability and strength listed above, there can be room for improvements to the project that can be considered for future reference. The system can be improved by adding several other sensors to the project in order to get a better result and can be tested with different scenarios. Not only that, the alert notification can also be improved by attaching the data received with files such as in excel file. With this, user can access see the files in a table form to make it easier for them to monitor. In conclusion, once all the flaws in this system have been solved so that this system will be more effective in the near future.

7.3 PROJECT LIMITATION

i. Power Usage

To check the humidity and temperature change, the device must be turn on the whole time. As a microcontroller, the device must always be supplied with electricity. Pocket Lab sensor also works on a battery life. It must be recharge again in order to use it once it runs out of battery.

ii. Inaccessible of Server Room

Due to pandemic, we are unable to test and use this device in an actual server room. Due to that, we can only use it in an air-condition room to imitate the optimum condition of the server room.

iii. Sensor limitation

Due to DHT11 sensor being a sensor that can detect both humidity and temperature at the same time, it can cause limitation to testing phase as we cannot get independent data from either humidity or temperature. Other than that, we are unable to create scenario where we can test the humidity sensor and temperature sensor independently.

7.4 FUTURE WORKS

In the future, this project can enhance its features by adding new implementation or improve the current features performances. The future implementation or improvement that can be considered included:

i. Firstly, this project can add the database system to make the temperature and humidity monitoring system become more effective. The database system can help user to have login interface to enhance security and give authority to use the system.

ii. Lastly, the monitoring system can be improved it effectiveness by having a dedicated humidity and temperature sensors installed in order to received a very detailed data from this project.

7.5 PROJECT CONTRIBUTION

This project was developed to help people to have affordable and user-friendly server room monitoring system. During this research of this project, the range of humidity and temperature for device and server room was identified and applied in the monitoring system. With the methods to capture the data of the server room monitoring system in mind, this project was based on the idea of monitoring the server room without entering the server room. Therefore, an alert notification was developed during this project. Using the alert notification, the user can always be notified when the server room is not in an optimum condition that is when the temperature is between 18 C to 27 C and the humidity is not between 40% to 55%. Using that threshold, we are able to create an accurate server room monitoring system with alert notification and statical info.

7.6 CONCLUSION

The results of system design have been able to send temperature and humidity information in real time to the Raspberry Pi. Data is sent via a wireless network to the Raspberry Pi. The results will be displayed on the Raspberry Pi terminal. The data of

temperature and humidity is relatively stored in the excel form that is stored in the Raspberry Pi and can be displayed as a table form. Data consists of time, temperature and humidity. The monitoring system in the form of notification to the user when the humidity and temperature is above or below the threshold that is set. For a normal server room condition, the optimum humidity is between 40% to 55% and the temperature is between 18 C to 27 C. In a high temperature condition such as between 27 C to 65 C, the humidity will be shown between 35% to 60%. In a low temperature room such as between 0 C to 18 C, the humidity will be around 10% to 35%. Alert Notification occurs every 10 seconds to notify if the room is in a normal condition, the room humidity and temperature is low or the room humidity and temperature is high. Notifications are sent via the Telegram application on the user's phone. The system can be performing once the user type in the correct command on the Raspberry Pi terminal and command on the Telegram application.

REFERENCES

Chung, W.-Y., & Oh, S.-J (2006)
Remote monitoring system with wireless sensors module for room environment. *Sensors and Actuators B: Chemical,*, 113(1), 64–70.

C. Chen, L. Sun, Y. Shao, Z. Hu and Q. Shi, (2012)
IEMS: An Intelligent Environment Monitoring System of Server Room. *Fifth International Conference on Intelligent Computation Technology and Automation*, (pp. 189-192).

Purwanto, F. H., Utami, E., & Pramono, E. (2018).
Design of server room temperature and humidity control system using fuzzy logic based on microcontroller. *2018 International Conference on Information and Communications Technology (ICOIACT)*, (pp. 66-70).

T H Nasution, M A Muchtar, S Seniman, I Siregar (2018).
Monitoring temperature and humidity of server room using Lattepanda and ThingSpeak. *IOP Conf. Series: Journal of Physics: Conf. Series 1235 (2019).* IOP Publishing.

Dwi Ely Kurniawan, Mohd Iqbal, John Friadi, Rohmat Indra Borman, Rio Rinaldi (2019).
Smart Monitoring Temperature and Humidity of the Room Server Using Raspberry Pi and Whatsapp Notifications. *Journal of Physics: Conference Series* . IOP Publishing.

Moechammad Alvan Prastoyo Utomo, Abdul Aziz, Winarno, Bambang Harjito (2019).
Server Room Temperature & Humidity Monitoring Based on. 2019 Journal of Physics: Conference Series. *Journal of Physics: Conference Series.* IOP Publishing.

Mohd Hakimi Zohari, Visvani Bala, Aimi Syamimi Abd Ghafar (2019).
Server monitoring based on IoT using ThingSpeak, Journal of Electrical Power and Electronics Systems. *Journal of Electrical Power and Electronics Systems Vol. 1 No. 2(2019)*, 1-4.

Rafizah Ab Rahman, Ummi Raba'ah Hashim, Sabrina Ahmad (2019).
IoT based temperature and humidity monitoring framework. Bulletin of Electrical Engineering and Informatics. *Bulletin of Electrical Engineering and Informatics Vol.9, No.1*, 229-237.

Onibonoje, M. O., Bokoro, P. N., Nwulu, N. I., & Gbadamosi, S. L (2019).
An IoT-Based Approach to Real-Time Conditioning and Control in a Server Room. 2019 International Artificial Intelligence and Data Processing Symposium (IDAP). . *2019 International Artificial Intelligence and Data Processing Symposium (IDAP)*, (pp. 26-32).

Roy, A. (2017).
Temperature and humidity monitoring system for storage rooms of industries. . *2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*, (pp. 25-30).