DATA LOGGER FOR IOT SYSTEM

AMIRUL HAQIM BIN ZAMRI

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

DATA LOGGER FOR IOT SYSTEM

AMIRUL HAQIM BIN ZAMRI

This report is submitted in partial fulfilment of the requirement for the

Bachelor of Computer Science (Computer Networking) with Honors.

FACULTY OF INFORMATION AND COMMUNICATION TECHONOLOGY UNIVRRSITI TEKNIKAL MALAYSIA MELAKA

DECLARATION

I hereby declare that this project report entitled

DATA LOGGER FOR IOT SYSTEM

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

STUDENT : ________________________ Date : 6/9/2021

(AMIRUL HAQIM BIN ZAMRI)

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 : 10/9/2021

(Ts Dr. Norharyati Binti Harum)

DEDICATION

First of all, this dedication is for my creator, Allah S.W.T that give me strength to complete this project. This thesis itself is dedicated to my supervisor, Ts Dr. Norharyati Binti Harum that always assist me during the process of completing this project. To my beloved parents En. Zamri Bin Mohd Noor and Puan Surina Binti Kassim that always support me especially in term of emotion and moral. Last but not least, I would like to dedicate this project to the entire lecturer of Faculty of Information and Communication Technology and my classmate. Without them, this project cannot be completed in the time scheduled.

ACKNOWLEDGEMENT

First and foremost, I would like to thank you to my supervisor, Ts Dr. Norharyati Binti Harum for the valuable guidance and advice to lead me until the end of this final year project session 2020/2021. Her willingness to motivate me contributed tremendously to my project. In this opportunity, I would like to thank you my evaluator Ts. Dr Nurul Azma Zakaria for taking some of her time to evaluate me. This evaluation gave me a deeper understanding of my weakness and what I can improve to make it better.

I would like also to thank the authority of Universiti Teknikal Malaysia Melaka (UTeM) for providing me with a good environment and facilities to complete this project. Finally, a honorable mention goes to my families and friends for their understandings and supports me in completing this project. With the help of everyone that was mentioned above, I was able to overcome many problems and completed my project successfully on time.

ABSTRACT

Data Logger for IoT System is a product that will programmed using a specific sensor which will be implemented to company environment and will not use a lot of cost in term of installing the technology. Data Logger for IoT System allow multiple sensors to be connected for this project, this system used a sensor that can detect the current temperature and humidity around the area. It also equipped with gas sensor to detect any gas leakage on the surrounding. The problem statement states that data logger available on the market needs high installation and maintenance fees. Which caused a difficulty for small company to install a data logger on their company to monitor the condition of their premises. Based on the problem statement, Data Logger for IoT System is proposed to develop with the objectives to implement a lower cost data logger. Other than that, to integrate the system with the database for it to save the data. Furthermore, it is also designed to notify the user through phone if the temperature exceeds the set limit. To ensure the project can be successfully developed, the Prototyping Model has been used. The reason is, prototyping model are used for software development model that required of building the prototype, tested, and modified until it become an acceptable prototype for the end user to use. Result obtained from this project is, the system can collect data by using a specific sensor and saved the data into a database. Other than that, user will be notify if the data collected from the sensor exceeds the limit set. All the data that has been saved inside the database can be shown to the user by using a simple user interface.

ABSTRAK

Data Logger for IoT System adalah sebuah produk yang akan diprogramkan menggunakan sensor tertentu dan akan dilaksanakan untuk persekitaran syarikat dan tidak akan menggunakan kos yang tinggi bagi pemasangan teknologi tersebut. Data Logger for IoT System membolehkan beberapa sensor disambungkan ke peralatan tersebut, tetapi untuk projek ini, sistem ini hanya akan menggunakan sensor yang dapat mengesan suhu dan kelembapan bagi persekitaran kawasan tersebut. Selain daripada itu, sensor gas juga akan di sambungkan bagi mengesan sebarang kebocoran gas pada keadaan sekeliling. Pernyataan masalah menyatakan bahawa kebanyakan *data logger* yang tersedia ada di pasaran memerlukan kos yang tinggi bagi menampung kos pemasangan dan penyelenggaraan yang tinggi. Hal ini menyebabkan kesukaran bagi syarikat kecil untuk memasang *data logger* di syarikat mereka bagi tujuan memantau keaadan premis mereka. Berdasarkan penyataan masalah tersebut, Data Logger for IoT System diusulkan untuk membangunkan dengan objektif untuk melaksanakan *data logger* dengan kos yang rendah. Selain daripada itu, untuk mengintegrasikan sistem dengan pangkalan data bagi tujuan penyimpanan data. Di samping itu, ia juga dirancang untuk memberi tahu pengguna melalui tefon sekiranya suhu melebihi had yang telah ditetapkan. Bagi memastikan projek ini dapat dikembangkan dengan jayanya, Model Prototaip telah digunakan. Hal ini kerana, model prototaip merupakan model pengembangan perisian yang diperlukan untuk membangunkan prototaip, diuji, dan diubah sehingga menjadi prototaip yang dapat diterima oleh pengguna akhir. Hasil yang diperoleh daripada projek ini adalah, sistem ini dapat mengumpulkan data dengan menggunakan sensor tertentu dan menyimpan data tersebut ke dalam pangkalan data. Selain itu, pengguna akan diberitahu sekiranya data yang dikesan oleh sensor melebihi had yang telah ditetapkan. Semua data yang telah disimpan di dalam pangkalan data akan dipaparkan kepada pengguna dengan menggunakan antara muka pengguna yang sederhana.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Introduction

Room temperature and humidity is a normal term to indicate a temperature level within a closed space at which humans can live with a more comfortable. Most of the time, room temperature can be used to indicated by general human comfort, though climate may acclimatize people to higher or lower temperatures. Other than that, multiple equipment used for the company are sensitive to the temperature and humidity on the surroundings. It will affect the performance and the life span of the equipment. Other than that, a save working space for the worker to work on also important thing to consider inside every building. Because of that, Data Logger for IoT System is needed to solve the problem of unexpected turn of event to the equipment and worker inside the company.

Most of the Data logger used by the company nowadays have a lot of problem which still need to be improved. For example, the data logger itself only record the data by collecting the data from the sensor and store it inside the database. Other than that, the cost needed to build one data logger used now quite expensive such as HOBO U20L-04. The price for the whole system is up to RM2667.50. While for the cheap system, the number of sensors that can be connected to the system is limited such as LogTag data logger. Due to that, it causes a lot of difficulty for a small company to install a data logger on their premises.

Data Logger for IoT System is a product that will programmed using a specific sensor to monitor and control room temperature and humidity together with the gas presence around the equipment. This program will detect the surrounding temperature and humidity reading and also gas presence from the surroundings and saved it inside the database. All of the recorded data can be shown to the user by using a simple type of interface so the user can read the data and alert about the condition of their company. Other than that, the system will alert the user if there any abnormalities happen to the temperature and humidity around the equipment. The system also will send an alert notification to the user if the presence of the is high and not save to the human. By doing that, an early measure can be done by the user and the record can be analyze back for the future use.

1.2 Problem Statement

Some of the industries have rising concerns for the demand to store specific production materials within a particular temperature range. Nowadays, some of the industries do not have incomprehensive data logger to monitor the surrounding for their equipment. The main reason behind it is due to the high cost to develop and installation fee for the data logger.

Table 1.1: Problem Statement

PS	**Problem Statement**
PS1	Existing data logger is expensive with very limited features.

1.3 Project Question

Project questions is used to identify the question of the existing temperature and electric current usage control system. Based on the research, we can conclude that there are few weaknesses of the current temperature and electric current usage control system.

Table 1.2: Summary of Project Question

PQ	**Project Question**
PQ1	What type of data logger exist in factory?
PQ2	How to save the collected data, notify the user, and display the data to user?
PQ3	How will the data logger perform?

1.4 Objective

Objectives provide structure and clarity of expectation for the project. Thus, the project aim had been stated below:

Table 1.3: Summary of Project Objective

PO	Project Objective
PO1	To identify characteristic of existing data logger in factory
PO2	To develop a data logger that integrated with database, smart notification, and dashboard for IoT System
PO3	To validate the performance of the developed data logger

1.5 Project Scope

1. The project would detect the temperature and humidity changes within the premise.
2. The system will save collected data from the sensor into a database.
3. Data Logger for IoT System will notify the user using Telegram if the temperature and humidity exceed the limit set.

1.6 Project Contribution

Project contribution defines the expected output from this project. This part can be referred to the objectives of this project. Therefore, the project contribution had been stated below.

Table.2.4: Summary of Project Contribution

PC	Project Contribution
PC1	Use Node Red as a software system to program the data logger.
PC2	Hardware system using a cheap microprocessor that enable to connect with multiple sensors.
PC2	Data logger that can save data from sensor into a database and have interface for user to view the data.

1.7 Conclusion

For the conclusion, this chapter helps to understand the project background and also the objective that should be accomplish and problems happened before begun the project. Based on the related topic and subtopic in this chapter, which is the problem statement, project objective, and expected output that this study wants to purpose a new approach of manufacturing monitoring system.

Additionally, this project will record all the reading from the sensor, and it will be uploaded into the database. This study is expected to be able to ease the user in order to monitor the temperature and humidity inside their premise in the best way.

This project also provides a contribution on how the tools can be used as easy as a plug-and-play in the real environment perhaps for the big network inside the industries. In the next chapter, the discussion will be focusing on the literature review that will cover about the model approached and related work about the framework, enhancing the framework and system integration.

CHAPTER II

LITERATURE REVIEW

2.1 Introduction

This chapter will discuss about the problem and solution regarding Data Logger for IoT System, to have a better understanding about the concept and technique needed to be implemented in this project. Other than that, this chapter will contain previous data logger monitoring system design by many types of engineers in the past that would be studied together with their thoughts on the project would be properly recorded. Their multiple type of design and implementation methodologies and techniques will also be documented in attempt to reach a comparison between the past and the current project timeline.

In this modern era, there were a lot of IoT device available for a company owner to improve their production and management. North Star BlueScope Steel that keeps track health of the worker and environmental temperature to avoid any accident to occur during work (O'Connor, 2016). Unfortunately, very little manufacturing company implement data logger system to monitor their company. Although the company used data logger as a monitoring system, the system itself is expensive and can result in a loss of funds. Therefore, the main reason for this data logger for manufacturing system invented is to create a low-cost data logger while at the same time it can work successfully.

2.2 Related Work / Previous Work

2.2.1 A Model for Working Environment Monitoring in Smart Manufacturing

Due to the new development in Industry 4.0 which is the expansion of smart and easy manufacturing systems that will allow mass production. According to Kadir et al., Industry 4.0 research is new and mainly focused on technical and technological aspects, while ignoring the human factor. Reason behind that is most of the current systems are run by human workers that may cause disturbance and cannot running smoothly. Latest trends in Industry 4.0 put a lot of requests on the workforce because they need to build up their organization, communication, cognitive, and physical skills in to order to work efficiently and safely in the new digitalized industrial era. The paper presented a model to monitor the working environment in in smart manufacturing and designed with idea to include in developing technologies and contemporary methods in complex building.

2.2.2 Improved Internet of Things (IoT) Monitoring System for Growth Optimization of Brassica Chinensis

Marimoto and Hashimoto stated that there are multiple climatic parameters need to be considered such as humidity, temperature, and light parameters in order for the plant to grow with consistent quality inside the plant factory. The WSN nodes which consist of multiple sensor such as temperature, humidity, light sensor, and actuator. The WSN nodes is connected to Zigbee or WIFI module that can send data over the air to a local gateway or router for it to transfer to the cloud server. The purpose of the WSN nodes was to sample the data and perform environmental control mechanism in real time.

2.2.3 A New Data Logger based on Raspberry-pi for Arctic Notostraca Locomotion Investigations

Based on the study by V. Pasqualli et al., an infrared sensor device is used and controlled by a Raspberry-Pi which act as a controller of the measurements and data logger. Infrared barriers are mounted in the aquarium for it to detect the locomotor activity of the animal. The electric pulsed will be generated when the animal passes through the barrier then go through a conditioning circuit that adapt it to the input range of the Raspberry-Pi. The signals are stored and will be processed based on the requirement needed.

2.2.4 Design and Implementation of a Low-cost Sensor Network to Monitor Environmental and Agronomic Variables in a Plant Factory

An Arduino Mega 2560 Rev3 was used in this low-cost system that act as a datalogger and also multiple sensors is used such as temperature, humidity, canopy surface temperature, air velocity, photosynthetic photon flux density, PH, electric conductivity, dissolved oxygen and nutrient solution temperature sensor. Additional hardware was used to measure the real time and recorded inside the system. All of the collected data was sent by using Bluetooth device that connected to the Arduino serial port.

2.2.5 Design and Validation of a Low-cost Indoor Environment Quality Data Logger

The data logger design in this research is to support on the research seeking for a better understand the links between resident satisfaction and indoor environment quality inside an apartment. Frontczak et al. describe that the heat, visual, sound and air- quality environments contribute to a satisfying and safe indoor environment for the residents. The device itself consist of Arduino Mega 2560 micro-controller board and multiple sensor that connected to the controller.

2.2.6 IoT Based Real Time Energy Monitoring System Raspberry Pi

The proposed IoT based energy monitoring system consists of existing energy meters that commonly used in the industry, Raspberry Pi, cloud, and visualization systems. The cloud is consisting of database, data web service, control application, and monitoring application. Energy meter that has been used are Schneider and Elmeasure. The meter itself capable of reading different electrical parameters such as power, energy, and real power reactive power. The microcontroller used is Raspberry Pi that has been installed with Node.js and other various libraries.

2.2.7 Intelligent Manufacturing Production Line Data Monitoring System for Industrial Internet of Things

Based on this paper written by Wei Chen (2020), this paper suggests a reference architecture for smart factories by referring existing product of IoT application in industrial workshops. The studies itself said that the company management and real manufacture process need a monitoring system that anti-interferences, can functioning in a real-time and has a rapid deployment capability. The client or server mode needs to be implemented in order to pass the data. The server will process data that received from the client and send the result back to the client.

2.2.8 Deep Learning Techniques for Energy Forecasting and Condition Monitoring in the Manufacturing Sector

V.J Mawson and B.R Hughes (2020) stated that two deep learning based neural network models is used to monitor and make a prediction about the hourly energy and workshop environmental condition. All of the data from the building obtained from pre-installed energy data collection system located at the manufacturing facility in order to validate the model. Other sub-metered data used for individual components such as energy consumption of lighting and HVAC (heating, ventilation, and air condition) was not available because it only collects energy data on a building level.

2.2.9 A Cloud-Monitoring Service for Manufacturing Environments

The proposed architecture by Ricardo Toro et al. is gather relevant data from one or multiple equipment and show it to the user by using cloud-based service application. It used two-layer-network architecture named as sensing layer and a cloud layer. The sensing layer consist of nodes gathering sensor data which connected to the manufacturing system. While the cloud layer consists of a network of data consumers together with other manufacturing application. There were two types of nodes used which is slave node that collect data while the other node is the master node that received data from the slave node.

2.2.10 Data Logger-based Measurement of Household Water Consumption and Micro-component Analysis of an Intermittent Water supply system

Method that has been proposed by B. Guragai et al. in this research is a water level data logger used to monitor the water level in the rooftop tanks. The data logger able to monitor the water level of rooftop tanks with a height minimum 1.5m. The accuracy of the sensor is depending on the water level measurement plus or minus 0.004m. The data loggers manually set at a logging interval of 30 seconds, which is the minimum resolution for the weeklong monitoring of the data logger model used. The raw data obtained from the data logger was used to determine the household water consumption.

2.3 Critical Review of Current Problem and Solution for the Studies

From the related work that has been listed, there are many types of development and methods being used. Multiple weakness and flaws have been discovered from the previous related work. In order to create a success and reliable IoT device, all of these flaws and weakness need to be avoided. The summary from the study has been done below in table 2.1.

Table 2.1: Problem and solution proposed based on the previous work

Data Logger Monitoring System			
Research Title / Product	**Purpose**	**Problem**	**Solution**
A Model for Working Environment Monitoring Smart Manufacturing By: Dalibor Dobrilovic, Vladimir Brtka, Zeljko Stojanov, Gordana Jotanovic, Dragan Perakovic, and Goran Jausevac (2021)	This paper presented a model for monitoring a working environment in smart manufacturing and designed with idea to include an emerging technologies and modern methods. The sensor nodes divided into two which is fixed wireless sensor and portable wireless sensor.	The two divided sensor nodes can cause lot of error, especially the portable wireless sensor. The battery used may wear out due to a long-time usage and may affect the sensor board. The data recorded by the sensor may not be accurate in the future.	Only used one type of sensor nodes only which is fixed wireless sensor. It can be connected directly to the power supply without relying on battery. The data can be captured all the time by the sensor.
Improved Internet of Things (IoT) Monitoring System for Growth Optimization of Brassica Chinensis	Monitoring system is used to meet the requirement for the best growth. Microcontroller	Microcontroller ESP-07 only featured with 16 MHz maximum speed 32-bit processor and 36Kbyte RAM	Raspberry Pi 4 is used for the microcontroller. The product itself come with 1.5 GHz Quad core Cortex-A72 (ARM v8) 64-bit

By: Ahmad Nizar Harun, Norliza Mohamed, Robiah Ahmad, Abd Rahman Abdul Rahim, and Nurul Najwa Ani (2019)	that has been used is ESP-07 made by AI Thinker.	build in it. Three AA battery is used to power up the microcontroller which is not efficient. The device itself can only connected to 3 sensors at one time.	processor and come with minimum 2GB ram. The high specifications help to process data faster. It can be power up by using USB type c cable. This microcontroller can be connected to 5 sensors at one time. It also has a temporary storage which is an essential feature for a microcontroller.
A New Data Logger based on Raspberry-Pi for Arctic Notostraca Locomotion Investigations By: V. Pasquali, G. D'Alessandro, R. Gualtieri, and F. Lecesse (2017)	Based on this paper, Raspberry Pi is used as a data logger instead of computer-based data logger. Data collected by the sensor are shown directly to the user by using Raspberry Pi hardware interface.	The data collected from the sensor is view directly to the monitor, but the data is not stored inside the system or database. All of the data are shown by using the Raspberry Pi hardware interface. Users need to record the data fetch by the sensor manually.	A system that can saved the recorded data from the sensor directly to the database can be used. The recorded data can be shown by using Node Red dashboard interface which is an open-source software.
Design and Implementation of a Low-cost Sensor Network to Monitor	The monitoring device used is Arduino as a microcontroller	A lot of sensors used may consume all port available at	Use a reliable microcontroller to process a lot of data from multiple

Environmental and Agronomic variables in a Plant Factory. By: A.P. Montoya, F.A. Obando, J.A. Osorio, J.G. Morales, and M. Kacira (2020)	which is Arduino Mega 2560 Rev23. Multiple difference sensor is used to monitor the plant factory condition. HC-06 is used as a Bluetooth communication device in order to access the data.	the Arduino Mega 2560 Rev23. HC-06 that used as communication device is not very reliable because it used Bluetooth connection.	sensors in a short time. A quick connection also needed for the user able to connect to the microcontroller such as 5.0 Bluetooth and 5.0 GHz wireless connection.
Design and Validation of a Low-Cost Indoor Environment Quality Data Logger By: Andrew Carre and Terence Williamson (2018)	It used Arduino Mega 2560 micro-controller board as a logger controller. 2.4 GHz radio is used to communicate between the microcontroller and sensor nodes. All of the recorded data is saved inside the Secure Digital (SD) card in text file format.	The highest speed for Arduino Mega 2560 to processing data only up to16 MHz. The 2.4 GHz radio communication is not effective to communicate between microcontroller and sensor nodes. Limited recorded data can be saved inside the Secure Digital (SD) card.	Use a microcontroller that can process data with the speed up to 1.5 GHz. The recorded data can be saved on a database for it to store a lot of data for future used.
IoT Based Real Time Energy Monitoring System Using Raspberry Pi By: Mani Dheeraj Mudalir and N. Sivakumar (2020)	Based on the paper, the data logger will record all of the data that has been collected by the power sensor. User can view the data captured from the sensor by using Node.js	The system will not give any alert to the user if any abnormalities occur around the sensor. This might cause huge lost to the company if the	A notification will be sent to the user smartphone from the system by using a suitable application to alert the user. So they can make an immediate step to control the abnormalities.

	applications on the local device.	equipment broken due to the abnormalities.	
Intelligent Manufacturing Production Line Data Monitoring System for Industrial Internet of Things By: Wei Chen (2020)	In this paper, the author discusses about the use of Internet of Things technology that can be used in order to create a smart factory.	User cannot access the microcontroller used for monitoring the system remotely by using another computer. This might cause a difficulty to check the condition of the sensor connected to the microcontroller.	Established a connection between microcontroller and other computer by using wireless Internet connection. By doing this, user can connect to the microcontroller remotely in order to check the condition of the sensor or conducting update to the system.
Deep Learning Techniques for Energy Forecasting and Condition Monitoring in the Manufacturing Sector By: Victoria Jayne Mawson and Ben Richard Hughes (2020)	An artificial neural network is used to forecast the energy consumption in the manufacturing facility. The data captured from the energy data collection system will be processed by the artificial neural network with multiple of simulation for it to forecast the energy consumption for the upcoming time.	Complicated artificial neural network needs a lot of time to develop the simulation for it to generate a precise result. A high error was obtained for prediction of energy consumption at the manufacturing facility.	A simple data logger and sensor nodes can be used to collect the data from the manufacturing facility. Less time consuming to be developed and can be used instantly. The real time data can be shown to the user by using an open-source platform.

A Cloud-Monitoring Service for Manufacturing Environments By: Ricardo Toro, Jorge E. Correa, and Placid M. Ferreira (2018)	A battery powered sensor node is used to collect the data from the facility. It used 915MHz radio band to communicate between the sensor and the microcontroller.	The battery used for the sensor node cannot last long especially when the data collection frequencies is high. The connection between the sensor and the microcontroller is not stable and need to troubleshoot frequently.	The sensor can used direct power supply by using micro-USB wired and it will let the sensor to capture much more data without need to worry about the battery live. Used 5GHz wireless connection or ethernet wire to connect the microcontroller and the sensor.
Data Logger-based Measurement of Household Water Consumption and Micro-component Analysis of an Intermittent Water Supply System By: B. Guragai, T. Hashimoto, K. Oguma and S. Takizawa (2018)	In this paper, datalogger that has been used will show directly to the user the captured data from the sensor with 30 seconds intervals. All of the data is recorded manually by the researcher and align with other data obtained by using a questionnaire.	The captured data from the sensor is not stored directly into the system will may cause multiple work need to be done and time consuming. A non-friendly user interface will create a difficulty for the user to read the data.	A datalogger that will record all the data from the sensor into the system autonomously can be implemented. User interface can be design using Node Red dashboard that will show all the recorded data to the user.

Table 2.2: Comparison of Functionality with Previous Project and Proposed Solution

Author/ Year	**Microcontroller**	**Communication Technology**	**Data Storage**	**Type of Sensor**	**Cost**
Dalibor Dobrilovic, Vladimir Brtka, Zeljko Stojanov, Gordana Jotanovic, Dragan Perakovic, and Goran Jausevac (2021)	Arduino Uno NodeMCU	Wi-Fi	Database	• Gas Sensor	RM 788.60
Ahmad Nizar Harun, Norliza Mohamed, Robiah Ahmad, Abd Rahman Abdul Rahim, and Nurul Najwa Ani (2019)	ATMega	Wi-Fi	Database	• Light Sensor	RM 400.60
V. Pasquali, G. D'Alessandro, R. Gualtieri, and F. Lecesse (2017)	Raspberry-Pi	Wired Connection	-	• Infrared Sensor	RM 407.26
A.P. Montoya, F.A. Obando, J.A. Osorio, J.G. Morales, and M. Kacira (2020)	Arduino Mega	Bluetooth	MicroSD memory card	• Temperature and Humidity Sensor	RM 556.90
Andrew Carre and Terence Williamson (2018)	Arduino Mega	Wired Connection	MicroSD memory card	• Light Sensor • Air velocity Sensor	RM 2479.96

Mani Dheeraj Mudalir and N. Sivakumar (2020)	Raspberry Pi	Wi-Fi	MicroSD memory card	• Energy Meter	RM 1083.68
Wei Chen (2020)	Not Stated	Wi-Fi	Database	• RFID	RM 625.20
Victoria Jayne Mawson and Ben Richard Hughes (2020)	Not Stated	Wired Connection	-	• Energy Meter • Temperature Sensor	RM 2000.00
Ricardo Toro, Jorge E. Correa, and Placid M. Ferreira (2018)	Adafruit Feather M0	Wi-Fi	Database	• Temperature Sensor	RM 600.00
B. Guragai, T. Hashimoto, K. Oguma and S. Takizawa (2018)	HOBO U20L-04	Wired Connection	-	• Pressure Sensor	RM 2667.50
Proposed Solution	Raspberry Pi	Wi-Fi	Database	• Temperature and Humidity Sensor • Gas Sensor	RM 265.00

2.4 Proposed Solution / Further Project

Multiple problems have been occurred on the previous project such as based on (A.P. Montoya, F.A. Obando, J.A. Osorio, J.G. Morales, and M. Kacira, 2020), (Andrew Carre and Terence Williamson, 2018), and (Mani Dheeraj Mudalir and N. Sivakumar, 2020), most of the data collected by the sensor is stored inside an internal SD card that most likely will run out of space. Other than that, in research done by (V. Pasquali, G. D'Alessandro, R. Gualtieri, and F. Lecesse , 2017), (A.P. Montoya, F.A. Obando, J.A. Osorio, J.G. Morales, and M. Kacira , 2020), (Victoria Jayne Mawson and Ben Richard Hughes, 2020), and (B. Guragai, T. Hashimoto, K. Oguma and S. Takizawa, 2018), the system have a non-reliable connection for the system to be communicated by the user. Next problem is the cost needed to build the data logger from the previous project are not suitable with the functionality offered by the system. Based on (Andrew Carre and Terence Williamson, 2018), (Mani Dheeraj Mudalir and N. Sivakumar, 2020), (Victoria Jayne Mawson and Ben Richard Hughes, 2020), and (B. Guragai, T. Hashimoto, K. Oguma and S. Takizawa, 2018), the system is so expensive which is above RM 1000.

From the study that have been done based on the previous work, a lot of problem and weakness that has been occurred for the implementation of data logger. To overcome the weakness and problem from the existing project, Data Logger for IoT System is suggested to be implement. In this project, a Data Logger has been created by connecting a sensor directly to the Raspberry Pi 4 microprocessor. The sensor will capture the data from the facility environment based on the type of sensor that been used. Data that have been received from the sensor will be uploaded and recorded by the microcontroller into the database automatically. The Microcontroller can be connected to the wireless Internet connection so it can upload the data directly to the system without causing any delay. A user interface will be created by an open-source software which is Node Red dashboard. The data recorded inside the database will be shown to the user by using the Node Red interface. By using Node Red interface, it will create a user-friendly interface is needed to help the user to view and make an analysis from the data for future use.

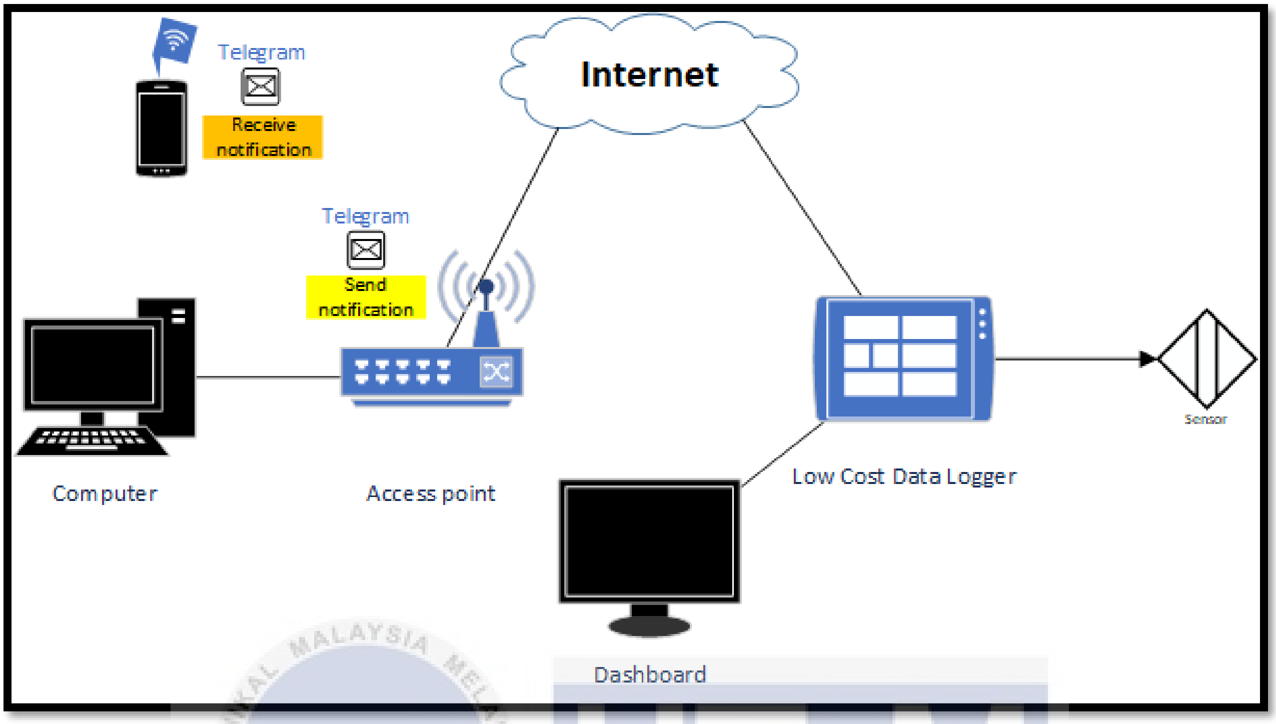


Figure 2.1: Proposed solution

2.5 Conclusion

From all of the review that has been done, we can summarize the purpose, weakness and solution of the previous project. Other than that, some of the technology used in this project has been elaborate. Multiple problem and weakness have been occurred such as high-cost data logger and slow processing data logger can be avoided by implementing the proposed solution. The development of Data Logger for IoT System seems to be the best solution for a simple and reliable data logger. In conclusion, literature review is a necessary chapter, and it is very important part to build the project concept, it helps to understand the existing features of the system and to get a clear picture to implement the system. The research and study will make the progression on doing this project smoothly and more understanding.

CHAPTER III

METHODOLOGY

3.1 Introduction

In this chapter, it will be focus on the methodology used in the project. Methodology refers to systematic, theoretical analysis of the methods applied to the field of study. The first topic is the project methodology which it will discuss about the software development life cycle used. The stages of the model will be detailed inside this chapter.

3.2 Project Methodology

The purpose of project methodology is to allow for managing the entire management process through the best decision making and problem solving, while make sure the accomplishment of specific processes, approaches, techniques, methods, and technologies. Basically, a methodology gives a backbone for explaining every step much more specific so it will help on what need to be done to deliver and implement the work accordingly. There are multiple type of software development methodology model such as waterfall model, spiral model, fountain model, and prototyping model. In this project, software development methodologies will be used as a project methodology.

In this project, prototyping model has been used to complete the project. Prototyping model is a software development model that required of building the prototype, tested, and modified until it become an acceptable prototype to be developed as a product for the end user. There are six sequence that need to be followed such as requirements, quick design, building prototype, user evaluation, refining the prototype, and implement and maintain. All of the listed stages need to be followed in a proper sequence so that the project can be success and finish on time. The flow of the prototyping can be referred to the figure below:

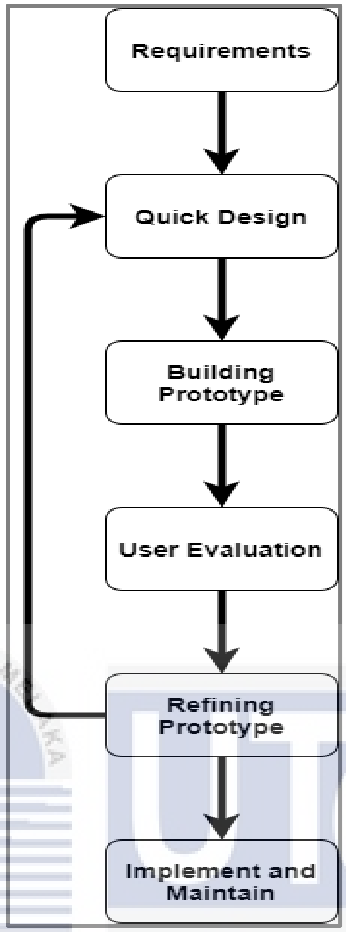


Figure 3.1: Prototyping Model

3.2.1 Requirements

In this phase, every requirement needed for the system are defined in detail manner. To complete the Data Logger for IoT System project, some requirement has been listed down.

1. Hardware
    - Raspberry Pi 4 Model B
    - Micro SD card (32GB)
    - USB C wire
    - HDMI to Micro HDMI connecter
    - HDMI Cable
    - Monitor

- Mouse
- Keyboard
- Breadboard
- Jumper Wire
- DHT11 Temperature and humidity sensor
- Access point

2. Software
   - Node Red
   - Raspbian OS
   - InfluxDB
   - Grafana

3.2.2 Quick Design

The first design of Data Logger for IoT system is created in this phase. To a achieve a brief idea about the system that would be developed, a simple design of the system is designed. However, the produce design is not a complete design. To visualize the connection of the project, physical design is developed for the used device. The simple physical design for this project is shown below:

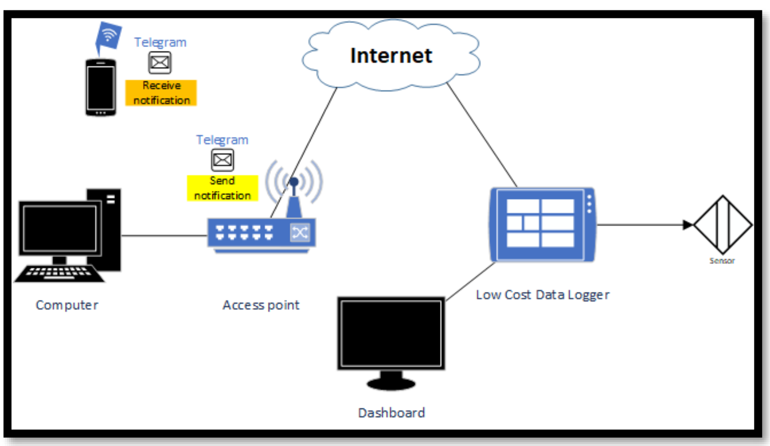


Figure 3.2: Physical Design

3.2.3 Building Prototype

Based on the information that has been gathered from the quick design phase, an actual prototype is built. The prototype consists of the hardware and software listed before. All of the hardware will be connected with each other so the installation and basic setup of Raspberry Pi OS can be done. After the Raspberry Pi OS has been setup, which is Raspbian, other software package listed before can be installed. The Raspberry Pi will be connected to the Node Red so the data from the sensor can be read and stored inside the database.

3.2.4 User Evaluation

After the prototype complete or finished, the client or evaluator will evaluate the prototypes. It helps a lot in order to find and point out to the inventor where the weakness and strength that can be found from the proposed prototype. All of the comment and suggestion received from the evaluation will be collected and analyzed. For this project to achieve success result, a usability testing will be conduct in terms of performance and functionality testing. For the performance testing aspects, the Data Logger for IoT Systems will be evaluated thoroughly whether it can meet the requirement of the project such as success rate and user satisfaction. While from functionality testing aspects, it should verify that the functions of Data Logger for IoT System are working as intended and it also measure the ease of use for the product to be used by the user.

3.3 Project Milestones

Project Milestones as a reference point that will be used to monitor the project's progress and marks the major activity in a project. To make sure the flow of the project runs smoothly, the project milestones will be created and well planned to ensure all of the activities in the project are able to complete within the timeframe. The milestones and Gantt chart used are shown below:

Table.3.1: Milestones for the project

Activity	Responsibility	Date Start	Date End
Gathering the requirements	Student	Week 1	Week 2
Analyze the requirement	Student	Week 2	Week 2
Designing the project	Student	Week 3	Week 4
Submit design to the supervisor	Student	Week 4	Week 4
Hardware gathering	Student	Week 5	Week 10
Installing required OS	Student	Week 5	Week 10
Building the prototype	Student	Week 6	Week 10
Progress evaluation	Student and supervisor	Week 7	Week 10
Testing the prototype	Student	Week 8	Week 10
User evaluation	Student, supervisor, and evaluator	Week 13	Week 13
Drafting new design based on the evaluation	Student	Week 1	Week 3
Refining the prototype	Student	Week 4	Week 4
Progress evaluation	Student and supervisor	Week 4	Week 4
Building the full project	Student	Week 5	Week 5
Testing the full project	Student, supervisor, and evaluator	Week 6	Week 6

Refining the full project	Student and supervisor	Week 7	Week 7
User evaluation	Student, supervisor, and evaluator	Week 8	Week 8

Table 3.2: Gantt Chart for PSM1 and PSM2

Week / Method	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Requirements gathering and analysis																							
Quick design																							
Building prototype																							
User evaluation																							
Refining prototype																							
Implement and maintain																							

3.4 Conclusions

As a conclusion, this chapter explains the methodology that been used in the project. The Data Logger for IoT System development consists of different phase that help to develop the system faster and systematically. The milestone set the time to finish the project so that the progression of the project will always keep in track. This is a very important aspect to make sure the project can be achieved within the timeframe.

CHAPTER IV

ANALYSIS AND DESIGN

4.1 Introduction

This chapter will describe the results of the analysis and the preliminary design of the project. It will focus on the problem analysis, requirement analysis, high-level design, and conclusion of the requirements of the project. Problem analysis will cover the analysis of current data logger problem while in requirements analysis, certain requirement such as data requirements, functional requirements, and hardware together with the software requirements will be discussed. Hardware and software used to build this project will be determined. The block diagram architecture and proper analysis in detail for this project also will be stated to ensure the project can be completed and well designed.

4.2 Problem Analysis

The function of the existing data logger system in the market is not comprehensive as it expensive and does not provide alert or notification to users if the room temperature is exceeding the desired temperature. This situation might lead to a massive loss to company because of the late notification to the user. Hence, there is and application that will implement in this project to notify the user if any abnormalities occur.

4.3 Requirement Analysis

4.3.1 Data Requirements

Data that have been collected by the sensor will act as input for the device. From there, the system will process the data by using Node Red APIs. Data will be divided into two which is temperature data and humidity data. The data also will be filtered based on the limit that has been set up in the system. One of the outputs from the device is alert message to notify the. If the data exceed the limit value, it will alert and notify the user so they can alert about the abnormalities that occur. Other than that, all of the data collected from the sensor will be stored inside a database. User interface also included for the user to view all of the recorded data saved inside the database.

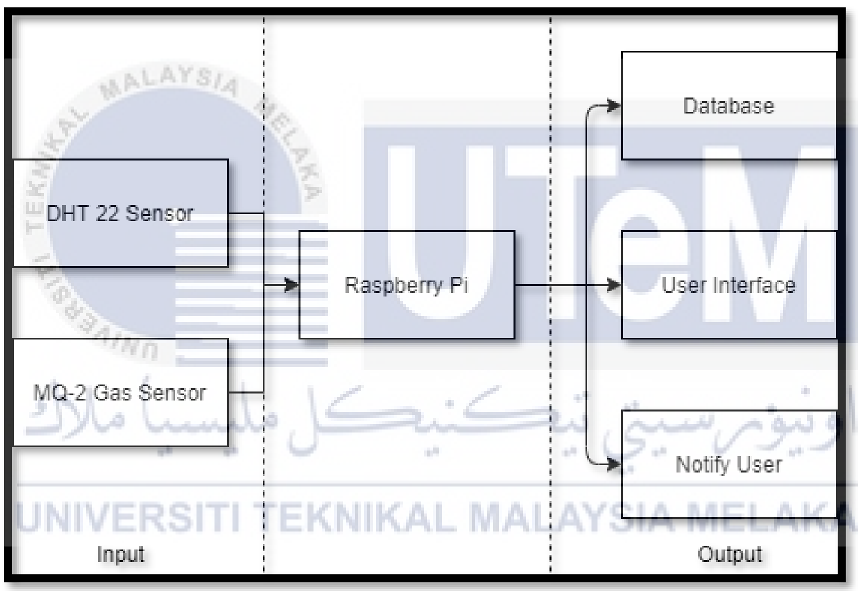


Figure 4.1: Block Diagram for Data Logger for IoT System

4.3.2 Functional Requirements

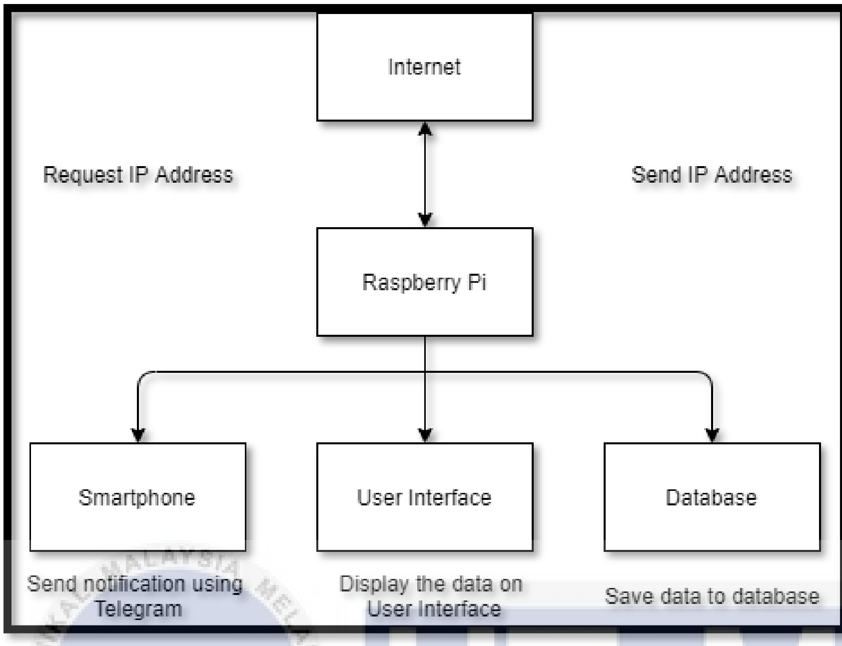


Figure 4.2: Context Diagram for Data Logger for IoT System

Based on the diagram, Raspberry Pi need to connect to the Internet so the device can run its function. When the sensor collect data based on the surrounding, the data will be saved into a database, and it will show the data to the user by using a user interface. If the reading of the data exceeds the limit value that has been set up, a notification will be sent to the user using Telegram application.

4.3.3 Hardware and Software Requirements

4.3.3.1 Hardware Requirements

In order to this project, the requirement is defined. There are multiple requirement need to be used such as the hardware and software. Proper hardware must be used to detect and stored data from the sensor accurately.

i. Raspberry Pi 4 Model B

Raspberry Pi is a single-board computer or the small computer. This single-board computer is widely used in teaching, learning, robotics used and many more because of its good performance in programming, processing and running series of program. Furthermore, the capability of Raspberry Pi to connect to various type of sensor make it more efficient to develop a project. The information collected by the sensor will be send to Raspberry Pi in order to be processed.

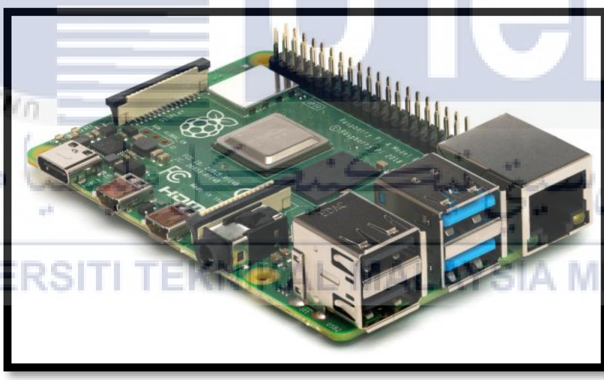


Figure 4.3: Raspberry Pi Model B

ii. Micro SD Card (32 GB)

All of the operating system needed to operate the Raspberry Pi and other software needed in this project is stored inside the Micro SD card. This Micro SD card will be installed into specific slot available on the Raspberry Pi.

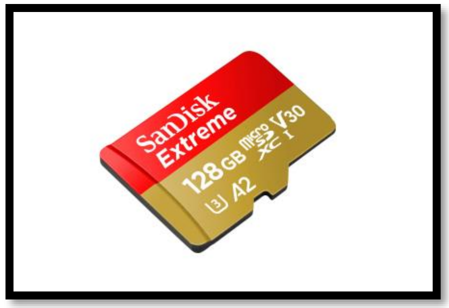


Figure 4.4: Micro SD Card

iii. USB C Wire

In order to run the Raspberry Pi, USB C wire is used to connect the Raspberry Pi to a power supply. Power voltage required for Raspberry Pi to run is 5V. There are multiple choice for the power supply such as directly connect to a plug-in electricity or just by using portable power supply such as power bank.

Figure 4.5: USB C Wire

iv. HDMI to Micro HDMI connector

HDMI to Micro HDMI connector is used in order to connect the Raspberry Pi to a monitor. This is because, most of the monitor in the market used HDMI type of wire but on Raspberry pi, only Micro HDMI slot available.

Figure 4.6: HDMI to Micro HDMI connector

v. HDMI Cable

HDMI cable is used to display an interface from the Raspberry Pi to a monitor.

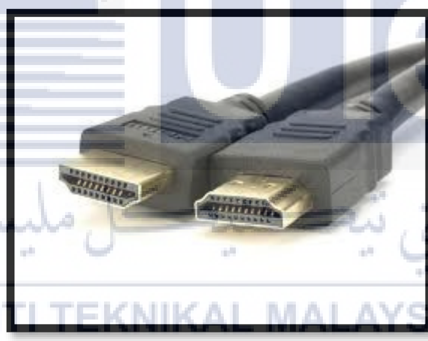


Figure 4.7: HDMI Cable

vi. Monitor

To display the user interface and open programs, allowing the user to interact with the Raspberry Pi.

Figure 4.8: Monitor

vii. Mouse

A hand-held device to transmits user commands to the Raspberry Pi by controlling the movement of the cursor/pointer on the computer screen.

Figure 4.9: Mouse

viii. Keyboard

Act as user input device to input any command and perform a variety of other tasks.

Figure 4.10: Keyboard

ix. Breadboards

Breadboards are used to help connect components to complete the circuit between Raspberry Pi and the sensor.

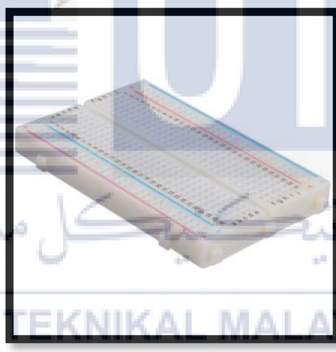


Figure 4.11: Breadboards

x. Jumper Wire

An electrical wire which is used to interconnect the components of a breadboard to other equipment or components without soldering which in this project is sensor to Raspberry Pi.

Figure 4.12: Jumper Wire

xi. DHT11 Temperature and humidity sensor

This sensor has a temperature and humidity sensor complex that can be calibrated by using a digital signal output. By using this sensor, it can detect temperature and humidity in the surrounding precisely.

Figure 4.13: DHT11

xii. MQ-2 Gas Sensor

One of the gas sensors available from the MQ sensor series. It can detect any gas concentration in the air.

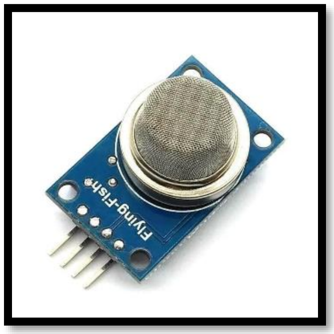


Figure 4.14: MQ-2 Gas Sensor

xiii. MCP3008 AC to DC Converter

A low cost 8 channel and 10-bit analog to digital converter. Used to convert analog signal from the sensor into digital reading so the data can be read by the PI.

Figure 4.15: MCP3008

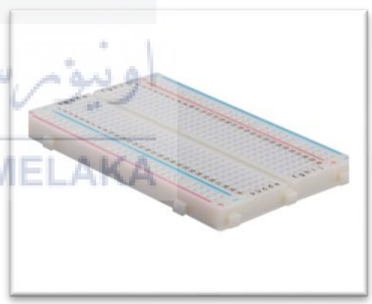
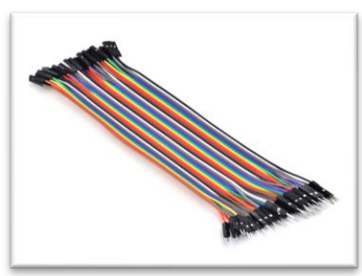
xiv. Access point

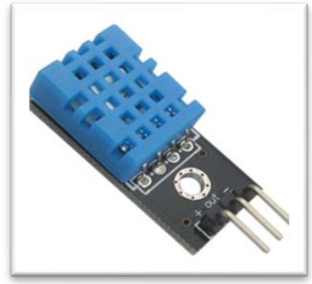
An access point is needed for the project to request and IP Address for the Raspberry Pi to be able to be control from other computer by using SSH connection.

Figure 4.16: Access point

Table 4.1: Summary of Hardware Equipment

Hardware	Functionality	Picture
Raspberry Pi 4 Model B	Used as microcontroller to embedded electronics and sensor	
Micro SD Card (32 GB)	Store all of the operating system needed for this project	
USB C Wire	Connecting the microcontroller to power supply	
HDMI to Micro HDMI connector	Used to connect the microcontroller to the HDMI cable	
HDMI Cable	Used to display an interface from the microcontroller to a monitor	

Monitor	Display the user interface	
Mouse	Controlling the cursor on the user interface	
Keyboard	Used to input any command into the system	
Breadboards	Act as platform to connect all the electronics components	
Jumper Wire	Used to interconnect the components of a breadboard to other equipment	

DHT11 Temperature and humidity sensor	To detect temperature and humidity in the surrounding	
MQ-2 Gas Sensor	To detect any gas concentration in the air	
MCP3008 AC to DC Converter	Converting analog data into digital data	
Access point	Used to transmit data from the microcontroller into user device by using Wi-Fi	

4.3.3.2 Software Requirements

i. Node Red

To connect all the hardware devices used in this project, Node-Red APIs is used for the programming. In this project, Node-Red used to create a configuration, develop, and create multiple extensions to be connected with the data collected from the sensor.

Figure 4.17: Node Red

ii. Raspbian OS

Raspbian Pi OS is a Debian-based operating system for the Raspberry Pi to be functioning.

Figure 4.18: Raspbian OS

iii. InfluxDB

InfluxDB is an open-source time series database used to save data detected by the sensor. It is one of the best and reliable databases that can be used to save data.

Figure 4.18: Influxdb

iv. Grafana

In this project, Grafana is used to display data collected from the sensor to the user. It is a multi-platform open-source analytics and have interactive web application interface. It comes with charts and graphs for the web when connected to the data sources.

Figure 4.20: Grafana

v. Telegram

Telegram is a communication platform between smartphones. By using Telegram, people can communicate with each other by sharing image, text, video, and others. In this project, telegram is used to notify the user if the sensor collected data that exceeds the limit set in the system.

Figure 4.21: Telegram

vi. Python

Python language is used in this project to detect gas changing in the surroundings. This python script will be run in the Raspberry Pi which it will compile and execute the code.

Figure 4.22: Python

4.4 High-Level Design

Connection between the computer and smartphone and Raspberry Pi is defined. In order for the device to interact with the smartphone and other computer, wireless communication is used. Notification will be sent to the user through a Telegram application if the data value exceeds the limit set. User also can access and control the device remotely by using another computer. Proper design must be determined so that the data sent to the user is accurate without any loss of data and the data can be stored inside a database.

4.4.1 System Architecture, Physical Design

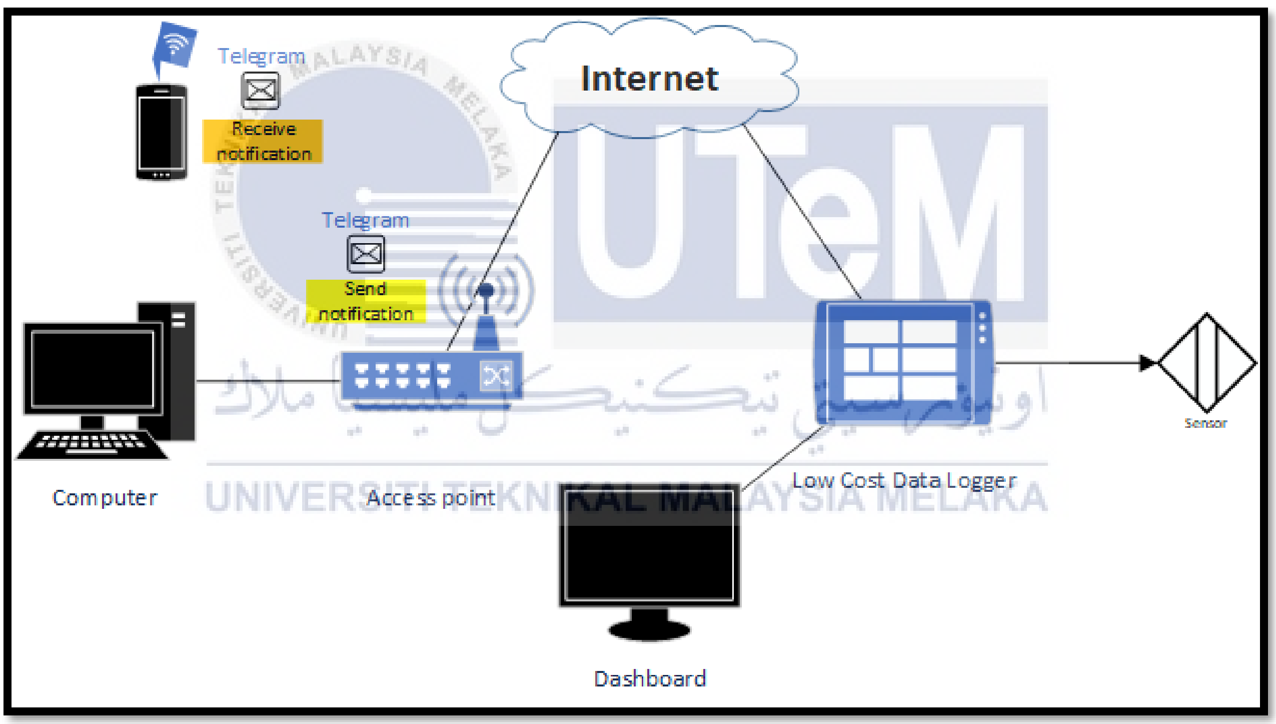


Figure 4.23: Physical Design of Data Logger for IoT System

4.4.2 User Interface Design

i. Flowchart

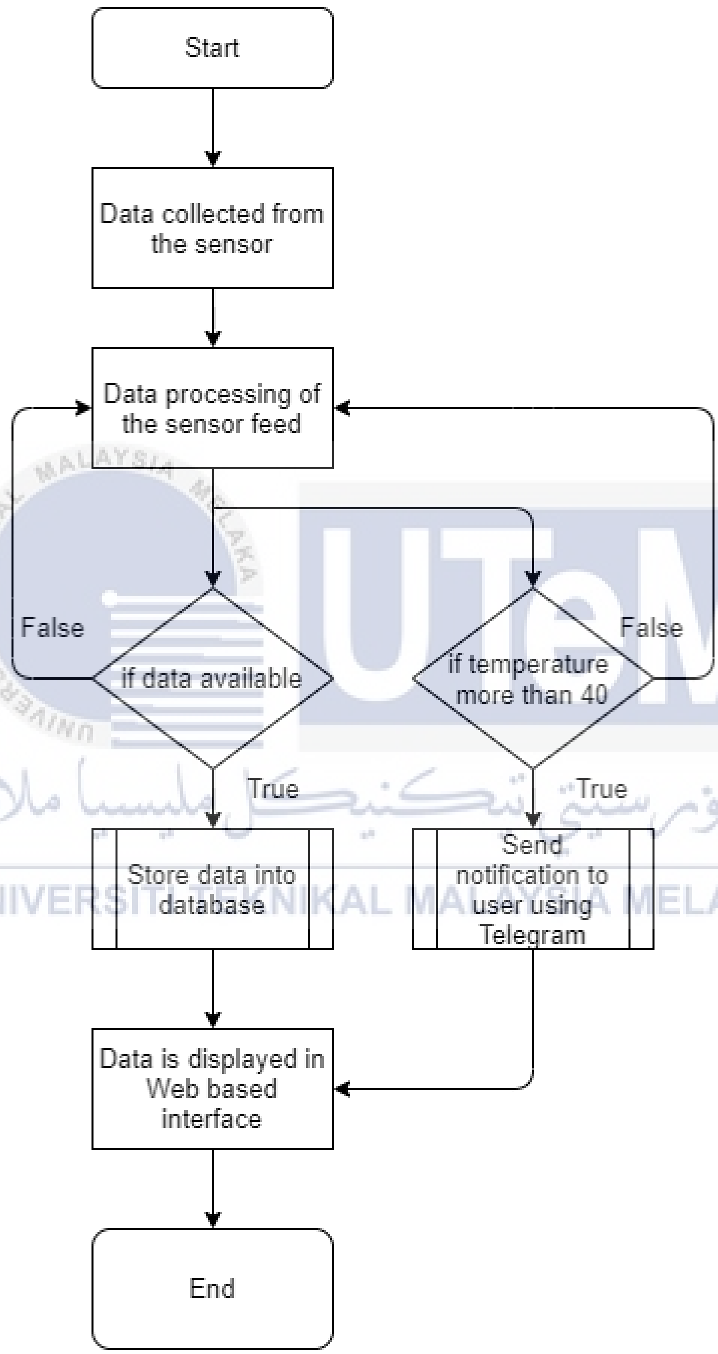


Figure 4.24: Flowchart of Data Logger for IoT System

Figure below shows on how the Data Logger for IoT System works. The Data Logger will receive data from the sensor. From there, it will be sent to the system in order to be process. All of the collected data will be saved inside a database together with date and time. Although the data is higher than the exceed limit, it still will be saved inside the database. If the reading of the data is high than the maximum limit set in the system, it will send alert notification to the user by using Telegram application. The notification will be included with the reading from the sensor which has been exceed the set limit. After the data is saved inside the database, it will be extracted by the system and displayed it to the user using a web-based interface which is Grafana. On the interface, user can see all of the detail on when the data has been collected and also the reading from the sensor.

ii. Output Design

Input that are processed by the Raspberry Pi will produce the output. The collected data from the sensor that have been stored inside a database will be shown to the user by using a web-based interface. Other than that, the system would send an alert notification to the user by using Telegram application if the data detected by sensor is above the limit of the set data that has been set. The system also allows the user to view the dashboard remotely by using another computer and it can be connected by using Internet connection.

a) Grafana Dashboard

- Diagram below show the value of temperature and humidity percentage data on the surrounding. There was also a line graph to show the up and down of the temperature data.

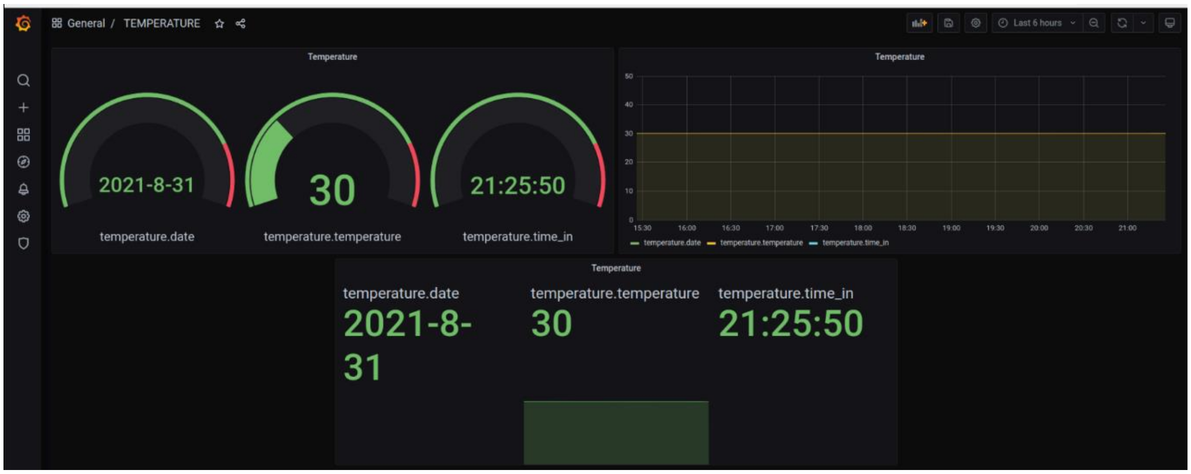


Figure 4.25: User Interface Output (Temperature)

Figure 4.26: User Interface Output (Humidity)

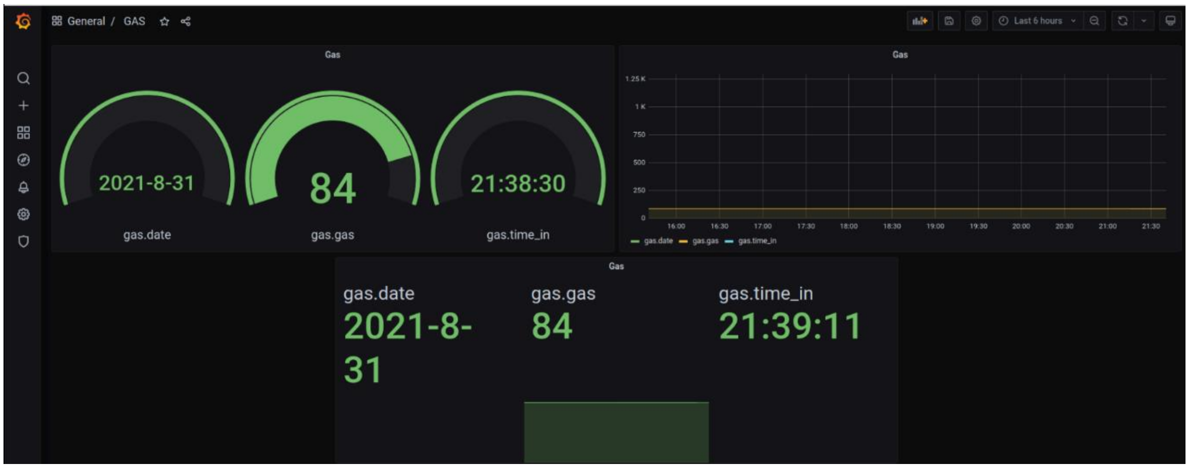


Figure 4.27: User Interface Output (Gas)

b) Telegram Notification Alert

- From the data collected by the sensor, the system will process and detect if the data exceeds the data set limit or not. When the system detects the data exceeds the set limit, it will give notification alert to the user. Diagram below shows the notification alert received by the user due to the abnormalities. The notification will be received by using telegram application.

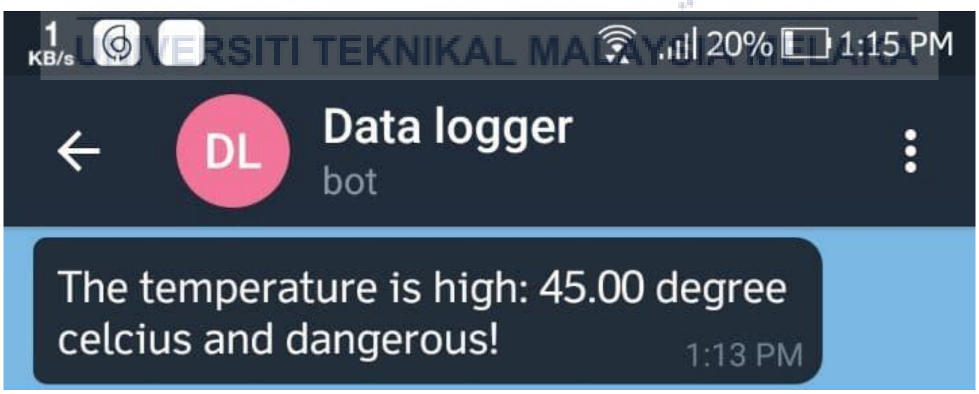


Figure 4.28: Alert received by the user (Temperature)

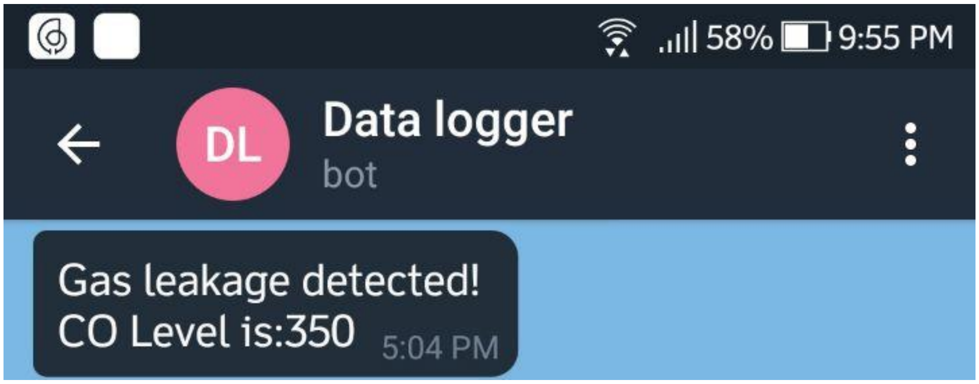


Figure 4.29: Alert received by the user (Gas)

4.4.3 Database Design

4.4.3.1 Conceptual and Logical Database Design

i. ERD Diagram for Data Logger

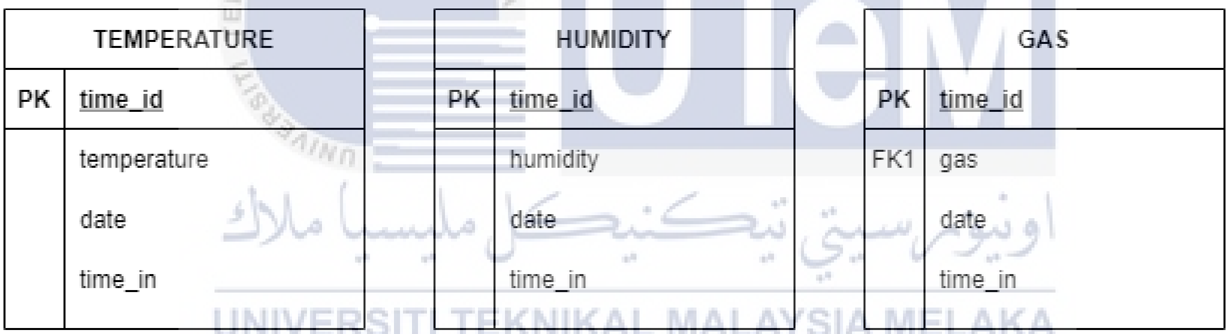


Figure 4.30: ERD Diagram

ii. Business Rule

1. Many data can be recorded in TEMPERATURE, HUMIDITY and GAS

iii. Data Dictionary

Table 4.2: Table Temperature

TEMPERATURE									
ATTRIBUTE NAME	**CONTENTS**	**DATA TYPE AND SIZE**	**FORMAT**	**RANGE**	**REQUIRED**	**DEFAULT VALUES**	**UNIQUE**	**PK or FK**	**FK REFERENCED TABLE**
time_id	time_id	INT	99999		Yes			PK	
temperature	temperature	INT	99999						
date	date	DATE	0000-00-00						
time_in	time_in	TIME	00:00:00						

Table 4.3: Table Humidity

HUMIDITY									
ATTRIBUTE NAME	**CONTENTS**	**DATA TYPE AND SIZE**	**FORMAT**	**RANGE**	**REQUIRED**	**DEFAULT VALUES**	**UNIQUE**	**PK or FK**	**FK REFERENCED TABLE**
time_id	time_id	INT	99999		Yes			PK	
humidity	humidity	INT	99999						
date	date	DATE	0000-00-00						
time_in	time_in	TIME	00:00:00						

Table 4.4: Table Gas

GAS									
ATTRIBUTE NAME	**CONTENTS**	**DATA TYPE AND SIZE**	**FORMAT**	**RANGE**	**REQUIRED**	**DEFAULT VALUES**	**UNIQUE**	**PK or FK**	**FK REFERENCED TABLE**
time_id	time_id	INT	99999		Yes			PK	
gas	gas	INT	99999						
date	date	DATE	0000-00-00						
time_in	time_in	TIME	00:00:00						

4.5 Conclusion

Analysis and design is one of the important part to implement a project. All of the software and hardware requirements need to be identified and studied before carrying out a project. This chapter is the pre-preparation stage for the implementation and include the flow of the overall system in order to have a better understanding before implementing it. In the next chapter, further discussion about the project implementation and the output expected from this project will be discussed.

CHAPTER V

IMPLEMENTATION

5.1 Introduction

This chapter will focus on how to implement the Data Logger for IoT System in both software and hardware development. The testing process will also be carrying out to ensure the system can functioning well and achieve the satisfaction of the users. The system will be tested by the developer and the end user by following the proper procedure to identify the system's performance.

5.2 Hardware and Software Development Environment setup

The development environment setup for Data Logger for IoT System will involve hardware and software requirements. All of the setups will be stated step by step and clearly shown. The hardware and software requirements are stated in Chapter 4 and will be explain further for the connection in further section.

5.2.1 Hardware Development Environment Setup

For this project, the hardware used are stated in the Chapter 4. There are two sensor which is DHT11 sensor and MQ-2 Gas sensor that are connected to the Raspberry Pi microprocessor by using jumper wires. The Raspberry Pi will process data collected from the sensor by using Node Red command line. Figure 5.1 shows the idea of prototype to demonstrate on how the project will be implemented. This prototype is set up by all of the hardware that has been listed before.

Table.5.1: Details of each Pin Numbers

Hardware	Wire	Pins
DHT11 Sensor	+VE	5V
	-VE	GND
	OUT	GPIO 4
MQ-2 GAS Sensor	AO	CH0
	DO	GPIO 26
	GND	GND
	VCC	5V
MCP8002 AC to DC Converter	CH0	AO
	VDD	5V
	VREF	5V
	AGND	5V
	CLK	5V
	DOUT	GPIO 09 MISO (SPI0)
	DIN	GPIO 10 MOSI (SPI0)
	CS/SHDN	GPIO 25
	DGND	GND

Next, we are going to install the hardware. Figure below shows the complete hardware setup after following the details provided by the table on how it is connected.

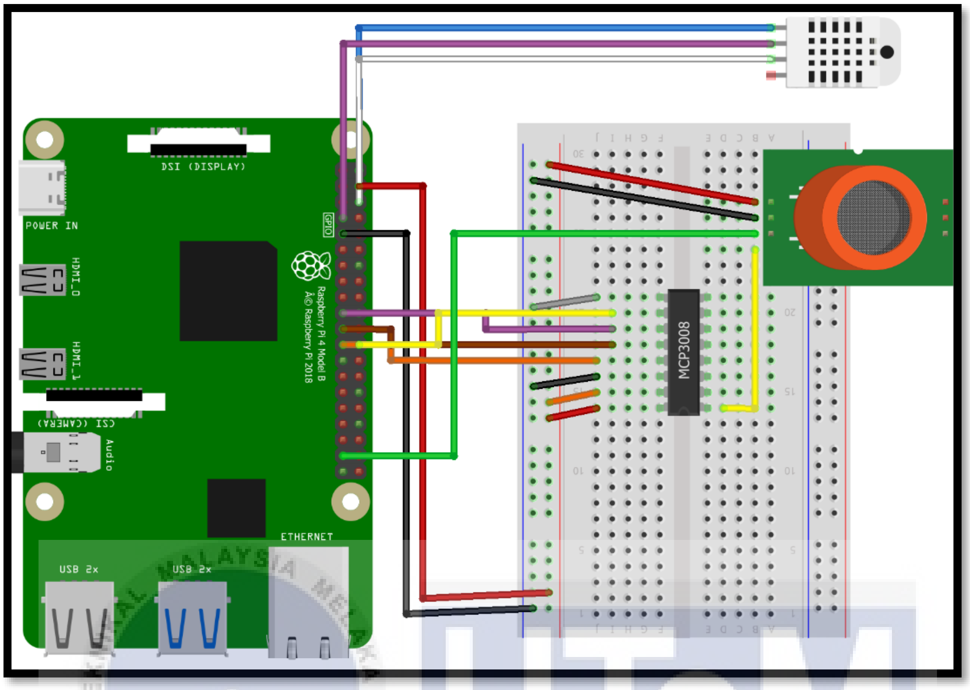


Figure 5.1: Schematic Diagram

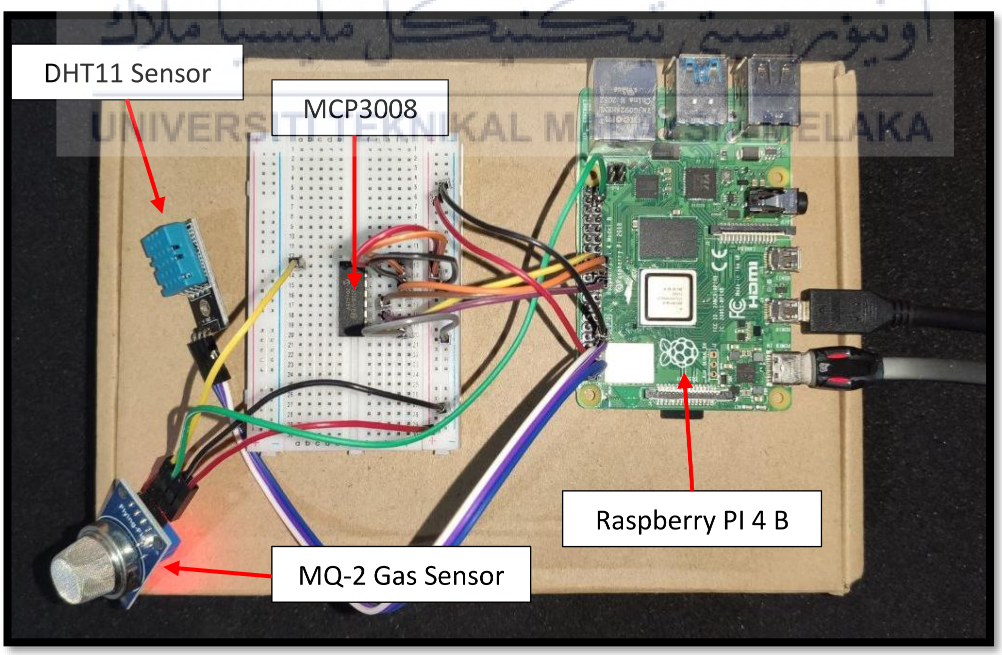


Figure 5.2: Hardware Installation

5.2.2 Software Setup

a) Software Installation

There are several software that need to be installed for this project such as Node-Red, Raspbian OS, InfluxDB, and Grafana. All of this software will be used to ensure the project is fully functional. The installation method used to install all of the software are by using terminal.

i. Raspbian OS need to be upgraded to the latest version. In order to ensure the system is up to date, this command is used.

```
- sudo apt-get update
- sudo apt-get upgrade
```

ii. By using the terminal, install or upgrade Node-Red on the Raspbian OS.

```
- bash <(curl -sL https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/resources/update-nodejs-and-nodered)
```

iii. Influxdb is installed into the Raspberry Pi in order to create a database. This database will save data collected from the sensor.

```
- wget -qO- https://repos.influxdata.com/influxdb.key | sudo apt-key add –
- source /etc/os-release
- echo "deb https://repos.influxdata.com/debian $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/influxdb.list
```

iv. Grafana will be installed into tha Raspberry Pi in order to create a web-based user interface to show data collected from the sensor to the user.

```
-   wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add –
-   echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee
    -a /etc/apt/sources.list.d/grafana.list
-   sudo apt-get update
-   sudo apt-get install -y grafana
```

5.3 Software Configuration Setup

5.3.1 Gas detection configuration

In order to detect any high presence of gas on the surrounding collected by the MQ-2 gas sensor, we must configure it by using Python IDE which is Thonny that installed inside the Raspberry PI. The reason why the python language is selected to program the MQ-2 gas sensor is, because the reading collected by the sensor itself is an analog output. Raspberry Pi cannot read any analog data unless it is converted into digital output. After that, every single data collected by the sensor will be saved in a text file.

```
import RPi.GPIO as GPIO
import time

# change these as desired - they're the pins connected from the
# SPI port on the ADC to the Cobbler
SPICLK = 11
SPIMISO = 9
SPIMOSI = 10
SPICS = 8
mq2_dpin = 26
mq2_apin = 0

#port init
def init():
        GPIO.setwarnings(False)
        GPIO.cleanup()          #clean up at the end of your script
        GPIO.setmode(GPIO.BCM)      #to specify whilch pin numbering system
        # set up the SPI interface pins
        GPIO.setup(SPIMOSI, GPIO.OUT)
        GPIO.setup(SPIMISO, GPIO.IN)
        GPIO.setup(SPICLK, GPIO.OUT)
        GPIO.setup(SPICS, GPIO.OUT)
        GPIO.setup(mq2_dpin,GPIO.IN,pull_up_down=GPIO.PUD_DOWN)

#read SPI data from MCP3008(or MCP3204) chip,8 possible adc's (0 thru 7)
def readadc(adcnum, clockpin, mosipin, misopin, cspin):


        if ((adcnum > 7) or (adcnum < 0)):
                return -1
        GPIO.output(cspin, True)

        GPIO.output(clockpin, False)  # start clock low
        GPIO.output(cspin, False)     # bring CS low

        commandout = adcnum
        commandout |= 0x18  # start bit + single-ended bit
        commandout <<= 3    # we only need to send 5 bits here
        for i in range(5):
                if (commandout & 0x80):
                        GPIO.output(mosipin, True)
                else:
                        GPIO.output(mosipin, False)
                commandout <<= 1
                GPIO.output(clockpin, True)
                GPIO.output(clockpin, False)

        adcout = 0
        # read in one empty bit, one null bit and 10 ADC bits
        for i in range(12):
                GPIO.output(clockpin, True)
                GPIO.output(clockpin, False)
                adcout <<= 1
                if (GPIO.input(misopin)):
                        adcout |= 0x1
```


Figure 5.3: Code for Gas Detection

```
        GPIO.output(cspin, True)

        adcout >>= 1       # first bit is 'null' so drop it
        return adcout
#main ioop
def main():
        init()
        print("please wait...")
        time.sleep(20)
        while True:
                COlevel=readadc(mq2_apin, SPICLK, SPIMOSI, SPIMISO, SPICS)
                file = open("demofile.txt", "w") #jgn ubah - dh boleh
                #if GPIO.input(mq2_dpin):
                print(COlevel)
                file.write(str(COlevel))
                file.close()
                time.sleep(0.5)



if __name__ =='__main__':
        try:
                main()
                pass
        except KeyboardInterrupt:
                pass

GPIO.cleanup()
```

Figure 5.4: Code for Gas Detection:

5.3.2 Telegram configuration

In order to send alert message from the Node Red to the user by using Telegram, a Telegram bot must be created first by using the Telegram BotFather in order to get the token that will be used to connect the Raspberry Pi to Telegram. The Raspberry Pi will send notification to the user by using the bot which the token is the unique ID for the bot.

Use this token to access the HTTP API:
1820083145:AAFMOWZAFzRE-GRMsrZEYVsJtmWNUfQFFTg
Keep your token **secure** and **store it safely**, it can be used by anyone to control your bot.

Figure 5.5: Telegram Token

The token must be inserted into the Node Red node that specified for Telegram. The following chat ID is for specific user or group that the Raspberry Pi will send the notification to.

Figure 5.6: Telegram Token Implemented in the Node

5.3.3 Node Red configuration

Node Red application mostly has all the configuration needed for the implementation of the Datalogger System. It receives data through the sensor that connected to the pin available on the Raspberry Pi. It also makes it easier to configure and program the system.

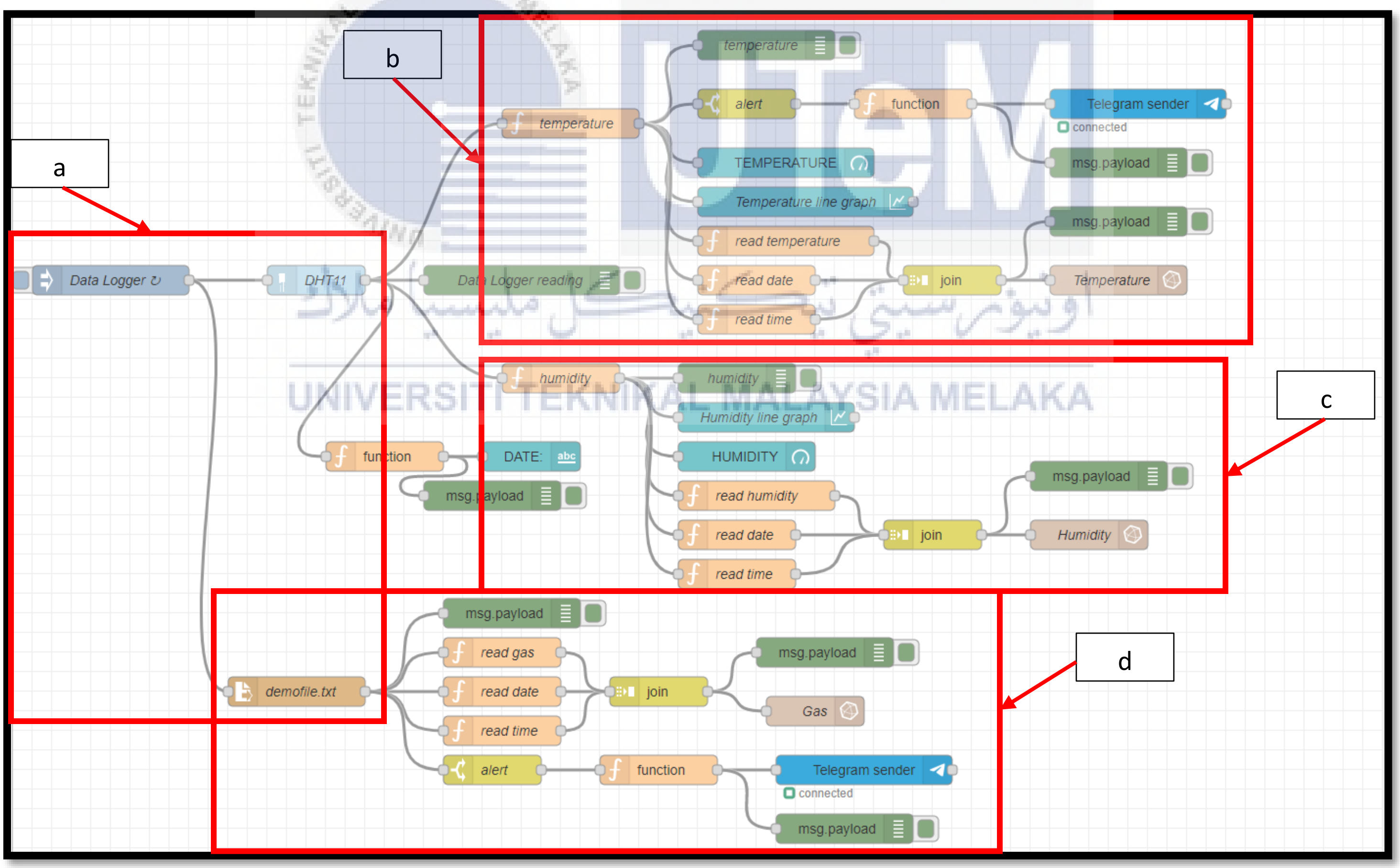


Figure 5.7: Node Red Flow Configuration

a) Input data from the sensor

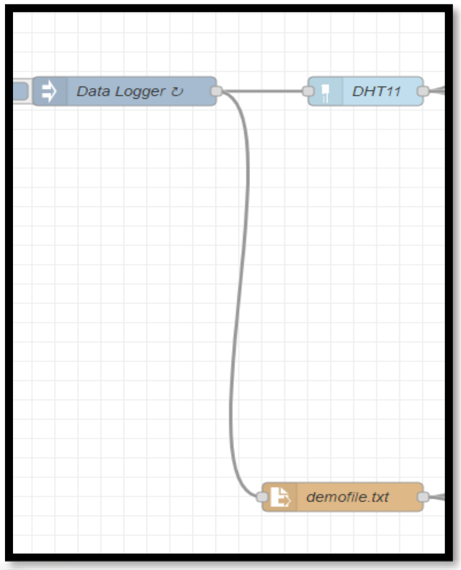


Figure 5.8: Input Data from The Sensor

The code snippet shown above is the configuration of receiving temperature and humidity data from the DHT11 sensor. While for gas data, it is read from the text file created by running the python language code by using the python IDE.

b) Input processing of temperature data

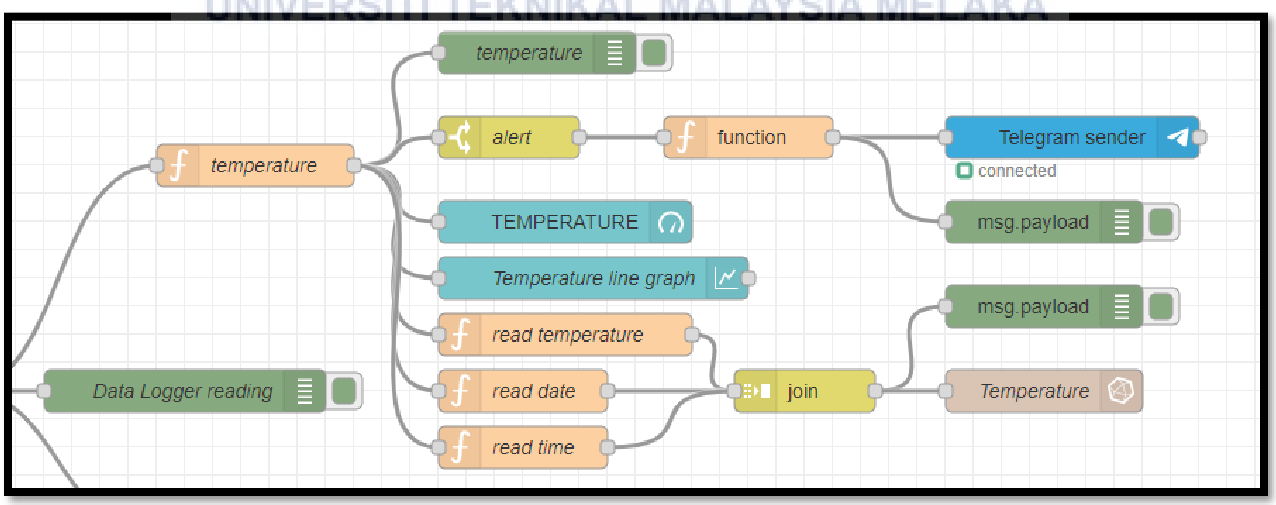


Figure 5.9: Input Processing of Temperature Data

Figure above shown the temperature collected from the sensor will be saved inside a database table named as Temperature. All of the data will be saved inside the database together with date and time. Other than that, the system will send a notification to the user by using Telegram if the temperature reading exceeds the set limit.

c) Input processing of humidity data

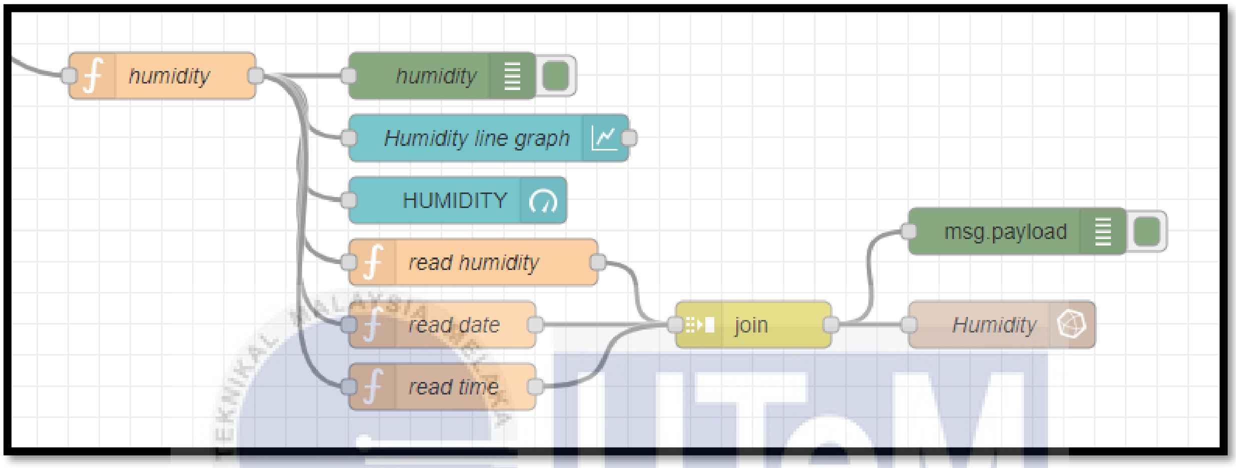


Figure 5.10: Input Processing of Humidity Data

The code snippet shown above is the configuration of inserting the humidity data into the database. Data collected by the sensor will be saved in a database together with the date and time when the data received by the sensor.

d) **Input processing of gas data**

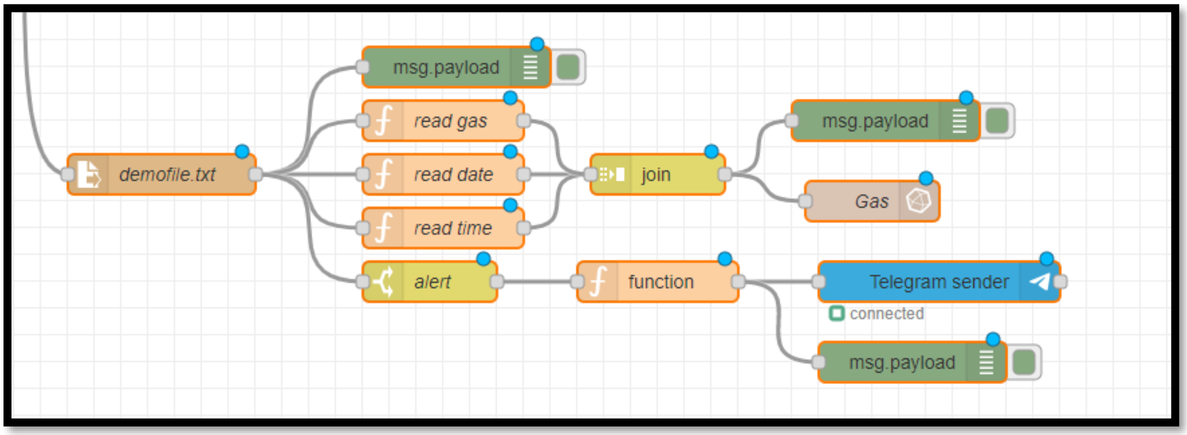


Figure 5.11: Input Processing of Gas Data

The code snippet shown above is the configuration for the Gas data that has been collected by the MQ-2 gas sensor. For this data, it is different from other data collected by the sensor. In this configuration, the Node Red will read data inside a text file that has been created by running the python-based coding executed before by using Thonny. All of the data such as gas reading, date and time will be saved inside a database. Other than that, an alert message will be send to the user via Telegram if the gas reading is exceed the set limit.

5.3.4 Grafana Configuration

For this project, the user interface used is Grafana. The user can view all of the data that has been saved inside the database such as temperature value, humidity value, gas value and also date and time. On the interface, the data will be shown in a multiple type such as line graph and gauge.

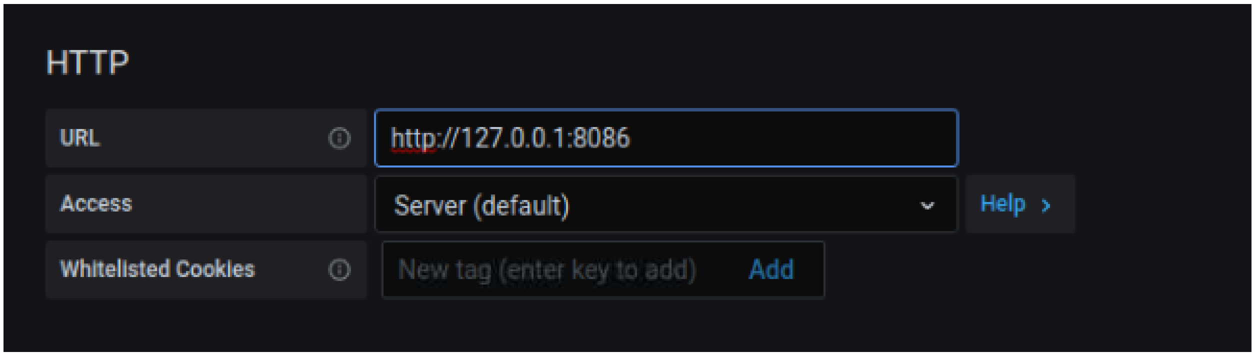


Figure 5.12: Grafana HTTP Setup

Figure above shown on how to setup the data source for the Grafana. As the Grafana are running on the same Pi, the URL set to localhost is a default influx port which is 8086 ports.

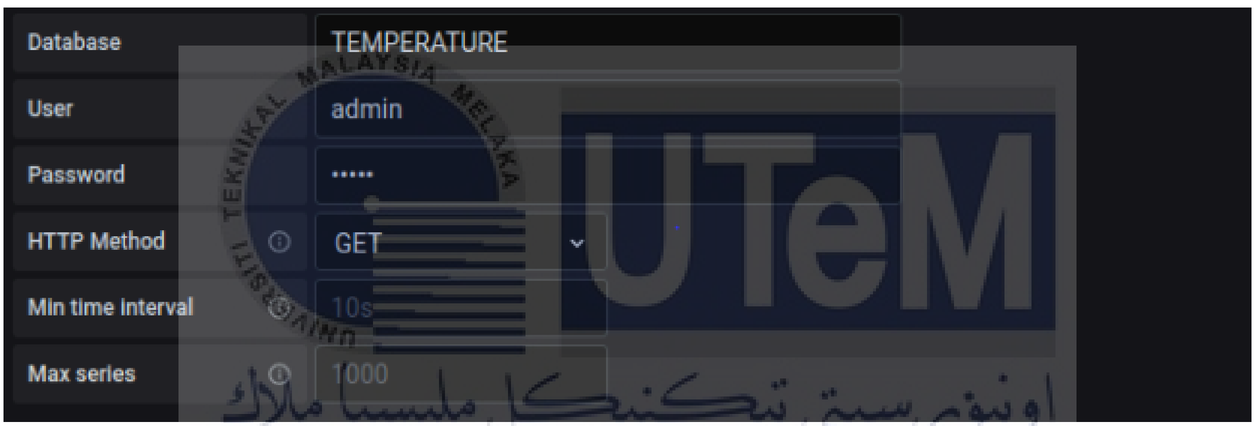


Figure 5.13: Grafana Source File Setup

On this part, the database name for the source file is inserted together with the username and password to the InfluxDB database.

5.4 Implementation Status

In this part, the duration for each component's development will be discuss and explain in detail.

Table 5.2: Implementation Status

No.	Component	Description	Duration for completed
1.	Assembling Hardware	Process to connect the jumper wires and sensor to the microcontroller. Connecting the microprocessor to the monitor and other peripheral for the setup. Established an SSH connection for the microprocessor to be accessed by another computer remotely.	13 days
2.	Building Prototype	This process is to setup the prototype of the project.	15 days
3.	Upload Raspberry Pi Coding	This process is to ensure the Raspberry Pi microprocessor can control the hardware attached to it.	10 days
4.	Developing System	All the functions of the system will be created using Node-Red APIs.	60 days

5.5 Conclusion

In Conclusion, the implementation chapter is the most important part for implement a system as it includes the process of hardware installation and software setup also together with the development. It provides a clear picture and idea to develop the project in order to achieve the objective of this project. The configuration is tested once it is implemented in order to avoid any error which will cause project malfunction. The configuration implemented in this chapter will ensure the Data Logger for IoT System is working properly.

CHAPTER VI

TESTING

6.1 Introduction

This chapter will more focus on discussion about testing the project. Purpose of this testing is to verify that the system will work without any problem. The first part of this chapter will more be focusing on test plan which will explain about the testing activities. Next, test strategy will be discussed follows by the test design. The test description will be focusing on a specific test in detail. It will cover the temperature and humidity detection test, gas detection, database connection, user interface test, and user alert notification. After that, the result and analysis will be discussed which covers the functionality test, performance test and system usability scale (SUS). Final part of this chapter is the conclusion which summarize all of the chapter.

6.2 Test Plan

Testing plan describes the activities, scope, and basic testing for the system. It is important to ensure all of the objective achieved and overall, of the system is running smoothly without any problem. Any bugs or errors can be detected and fixed before any further process.

6.2.1 Test Organization

In this section, the test will be done by the system developer who has develop the system. The project will be test according to its functionality. The developer will be in charge for this test organization because they know a lot better on how the device work.

6.2.2 Test Environment

For the test environment section testing, the structure of the testing will be determined. It is crucial for the test requirement, and it will be decided in the test environment. When the project is test, certain problem needed to be face if the test environment did not manage appropriately.

6.2.3 System Usability Scale (SUS)

System Usability Scale is a set of questionnaires that has been made to collect feedback from the end user. All of the participants will be brief by the developer about the system. After that, the participant will answer the questionnaires based on what they feel about the project created by the developer. The demo process can be done by using various type of online platform available on the Internet such as Google Meet, Microsoft Teams, and Discord. Hence, the System Usability Scale (SUS) will provide data to the developer on how the end user feel about the project whether it meets the user satisfaction or need to do some improvement.

6.3 Test Strategy

In this section, the flow of the testing will be determined. The test strategy for this project is how the test will be conducted in order to collect the temperature and humidity reading together with the gas reading on the surrounding. First, the sensor will collect data from the surrounding and save it into the database. From there, the saved data will be shown to the user by using user interface provided by the system. Other than that, the system will send an alert notification to the user by using Telegram if the data reading collected by the user exceed the set limit that has been set in the system. Hence the test of the project is successful.

6.4 Test Design

Test design discussed about the test case identification, test cases and expected result for each scenario which are designed and documented. The test description discusses integration test and functionality test. Table 6.1-6.4 shows the result of testing.

6.4.1 Test Description

Specific test such as temperature and humidity detection test, gas detection test, Telegram notification test, and database test will be conducted in detail. The test purpose, test environment, test setup, and expected result will be tabulated for each test.

Table 6.1: Temperature and Humidity Detection Test

Test	Temperature and Humidity Detection Test
Test Purpose/ Test Functionality	To be able to detect temperature reading from the surrounding.
Test Environment	To test this project, the sensor will be in one closed spaced such as room.
Hardware Needed	i. DHT11 Sensor
Test Setup/ Execution Test	1. Connect the Raspberry Pi to Internet connection 2. Run Node Red 3. Edit Node Red configuration 4. Click Deploy 5. Put any hot element closed to the sensor such as hot water.
Expected Result	When the program is running, the sensor will collect all of the data from the surrounding. The data will be saved in a database. If the sensor reading exceed the set limit, the system will notify user by using Telegram message.

Table 6.2: Gas Detection Test

Test	Gas Detection Test
Test Purpose/ Test Functionality	To be able to detect any high gas reading from the surrounding.
Test Environment	To test this project, the sensor will be in one closed spaced such as room.
Hardware Needed	i. MQ-2 Gas Sensor
Test Setup/ Execution Test	1. Run Python IDE 2. Write coding to upload data from the sensor into a text file. 3. Click Execute. 4. Connect the Raspberry Pi to Internet connection. 5. Run Node Red 6. Edit Node Red configuration 7. Click Deploy 8. Put any gas closed to the sensor.
Expected Result	When the program is running, the sensor will collect gas reading from the surrounding. The data will be saved in a database. If the sensor reading exceed the set limit, the system will notify user by using Telegram message.

Table 6.3: Telegram Notification Test

Test	Telegram Notification Test
Test Purpose/ Test Functionality	To ensure that notification about the high temperature or gas leakage is received by the user through Telegram.
Test Environment	To test this project, user mobile phone must be connected to the Internet.

Hardware Needed	i. Mobile Phone
Test Setup/ Execution Test	1. Connect the mobile phone to Internet 2. Ensure that the Telegram account is logged in and activated.
Expected Result	Once there any high temperature or high gas reading detected at the area, user will receive the notification through Telegram application along with the reading collected by the sensor.

Table 6.4: InfluxDB Test

Test	InfluxDB Test
Test Purpose/ Test Functionality	To be able to save all of the data collected from the sensor into a database.
Test Environment	To test this project, user need to check the database by typing SQL command into the database by using Raspberry Pi terminal.
Hardware Needed	-
Test Setup/ Execution Test	1. Run the InfluxDB 2. Select database table 3. Enter the table measurement 4. Enter SQL command to show data from the table.
Expected Result	When the command is inserted on the Raspberry Pi terminal, all of the data collected by the sensor will be shown to the user. The sensor reading is kept within the specific id, date, and time.

6.5 Result and Analysis

In this section, the result from the testing will be discussed in detail. Analysis of the result also will be covered in this part.

6.5.1 Functionality Test

Functionality test will be handled to check the functionality of the temperature and humidity detector and also gas detector. Test that will be perform is temperature and humidity detection, gas detection, Telegram notification test, and InfluxDB test.

6.5.1.1 Temperature and Humidity Test

Temperature and humidity detection test is done by referring to the figure below. The result of the test is shown below in figure 6.1 until 6.4. It shows the temperature and humidity detection on hardware and data displayed on Node Red debug console.

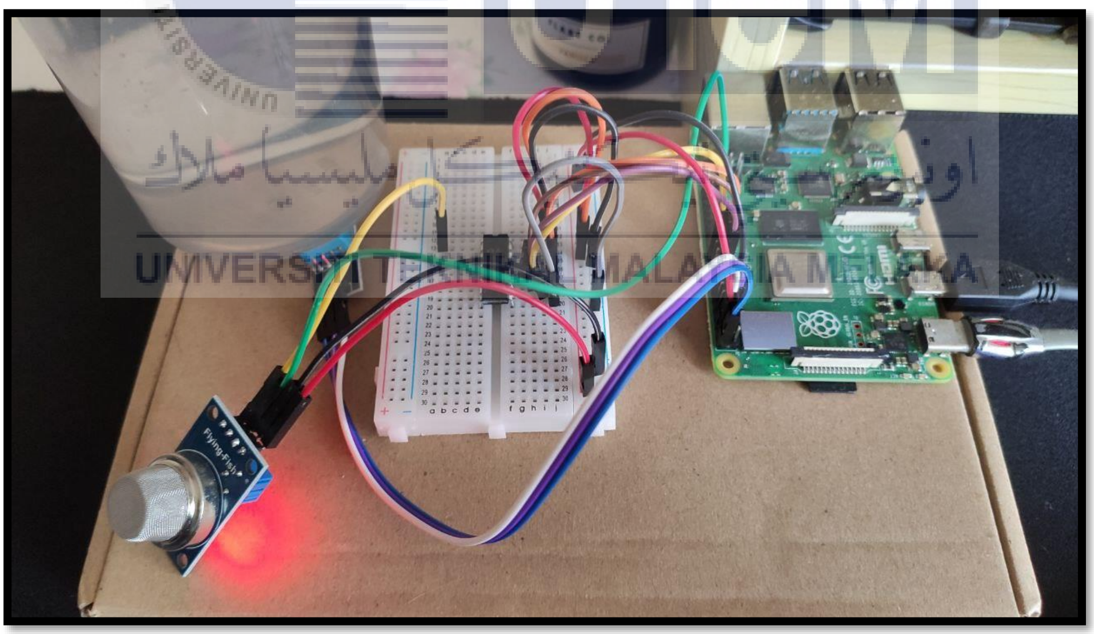


Figure 6.1: Detecting Temperature and Humidity

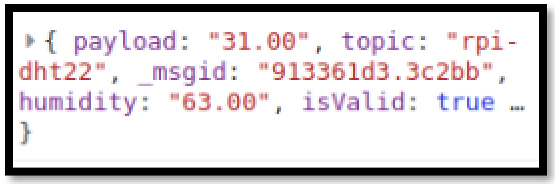


Figure 6.2: Successfully Detecting Temperature and Humidity

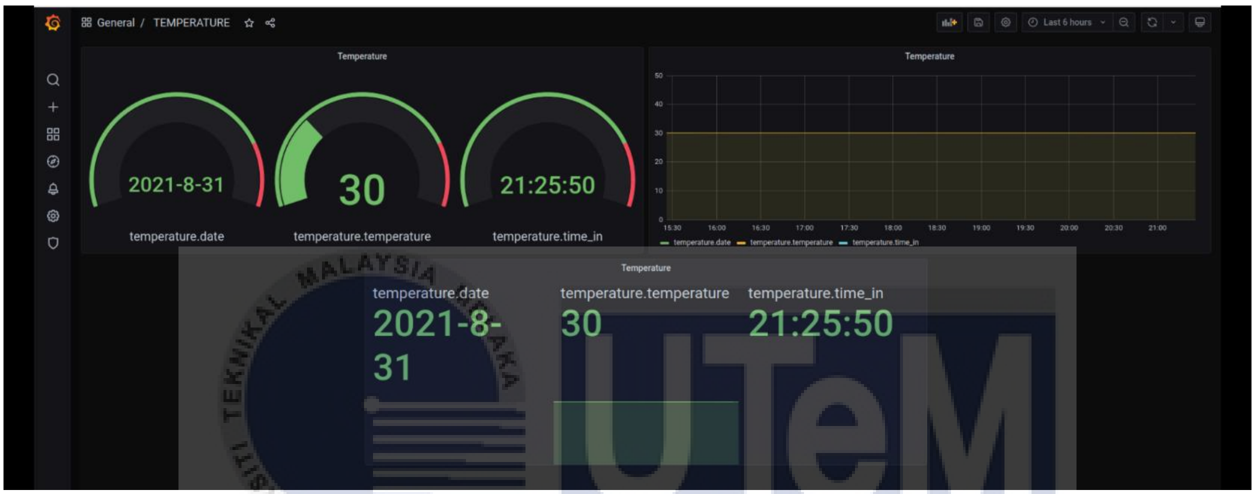


Figure 6.3: Temperature Grafana Dashboard

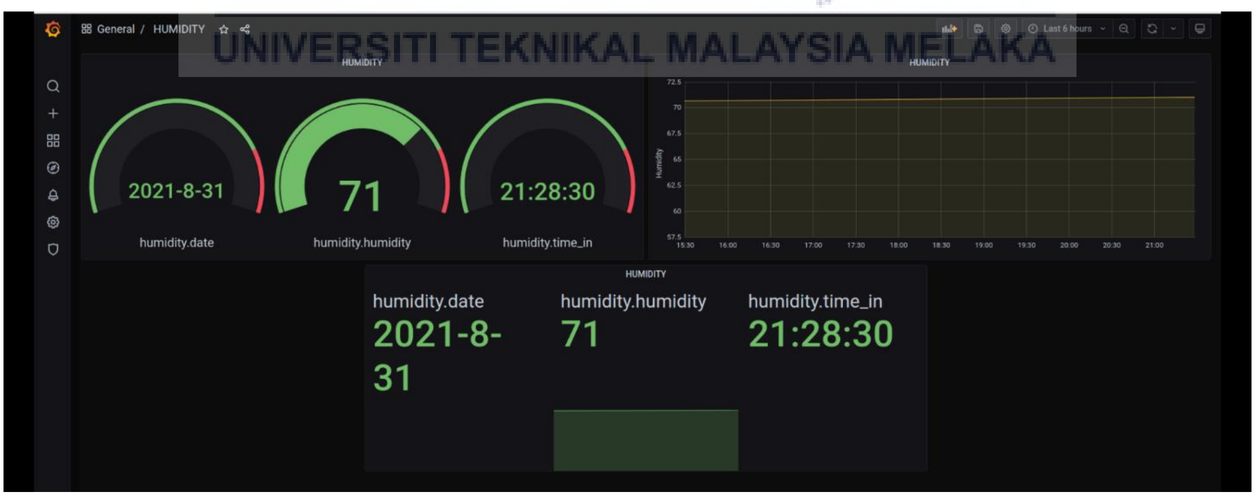


Figure 6.4: Humidity Grafana Dashboard

6.5.1.2 Gas Detection Test

Gas detection test is done by referring to the figure below. The result from conducting the test is shown below. In figure 6.5 until 6.7, it shows the gas detection sensor can read gas reading from the surrounding and the data is shown on the Node Red console.

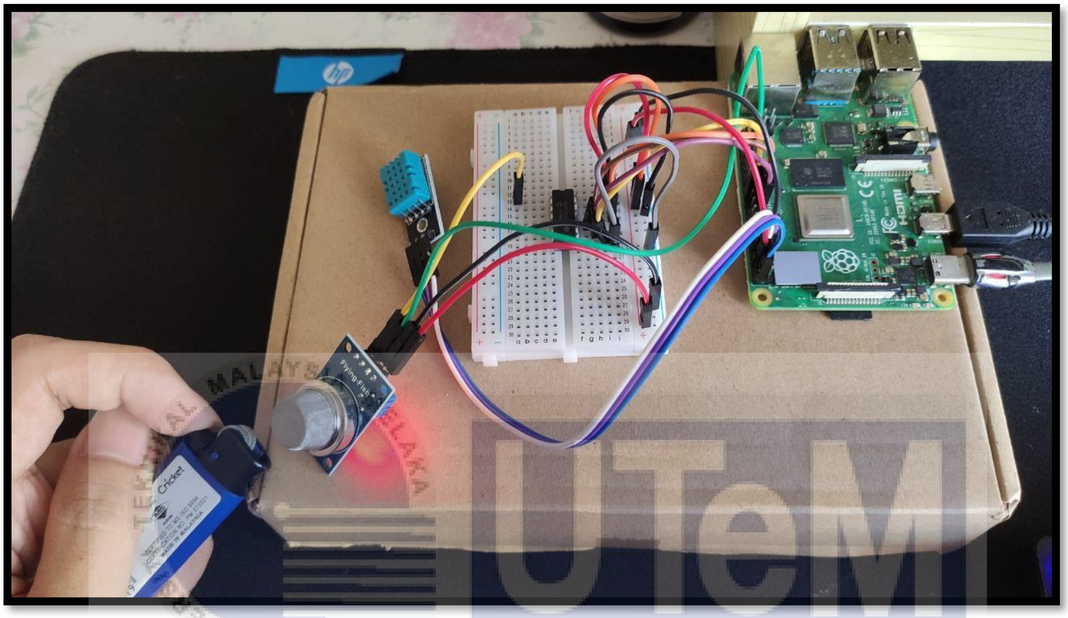


Figure 6.5: Detecting Gas

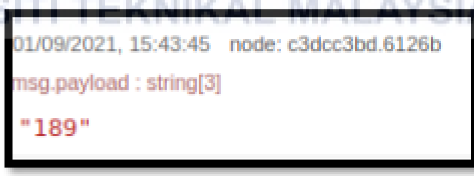


Figure 6.6: Successfully Detecting Gas

Figure 6.7: Gas Grafana Dashboard

6.5.1.3 Telegram Test

Once the system detects any high temperature reading or high gas reading, a notification message will be sent to the user through Telegram along with the reading.

Figure 6.8: Notification Received by the User when High Temperature Reading Detected

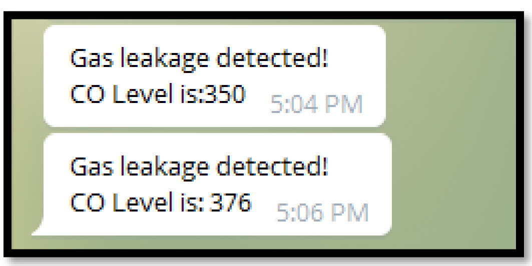
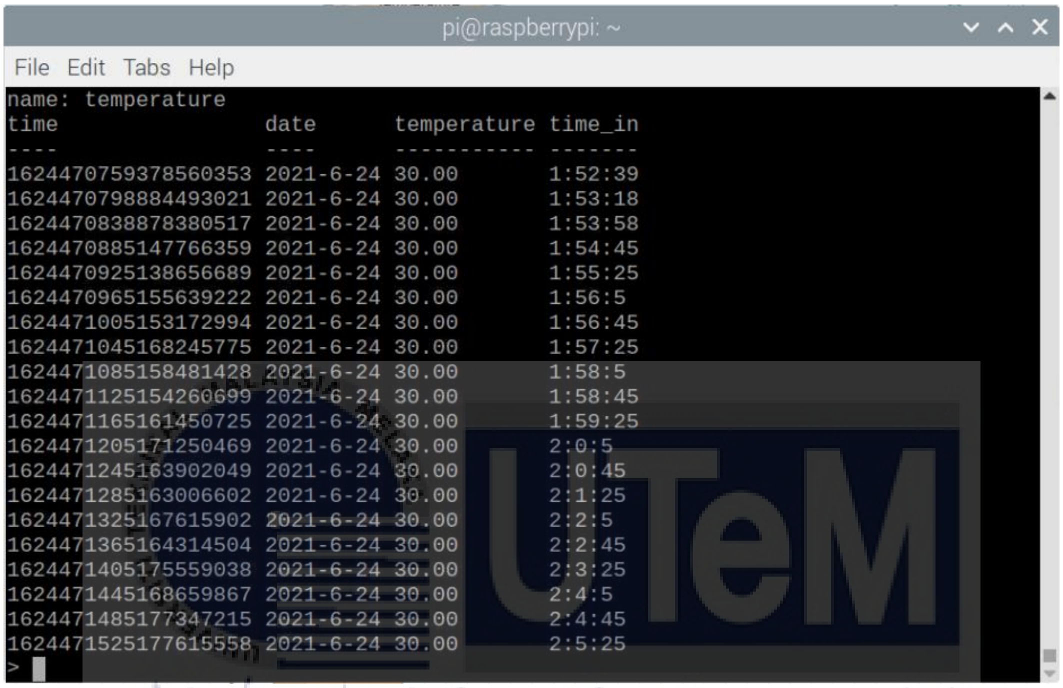


Figure 6.9: Notification Received by the User when High Gas Reading Detected

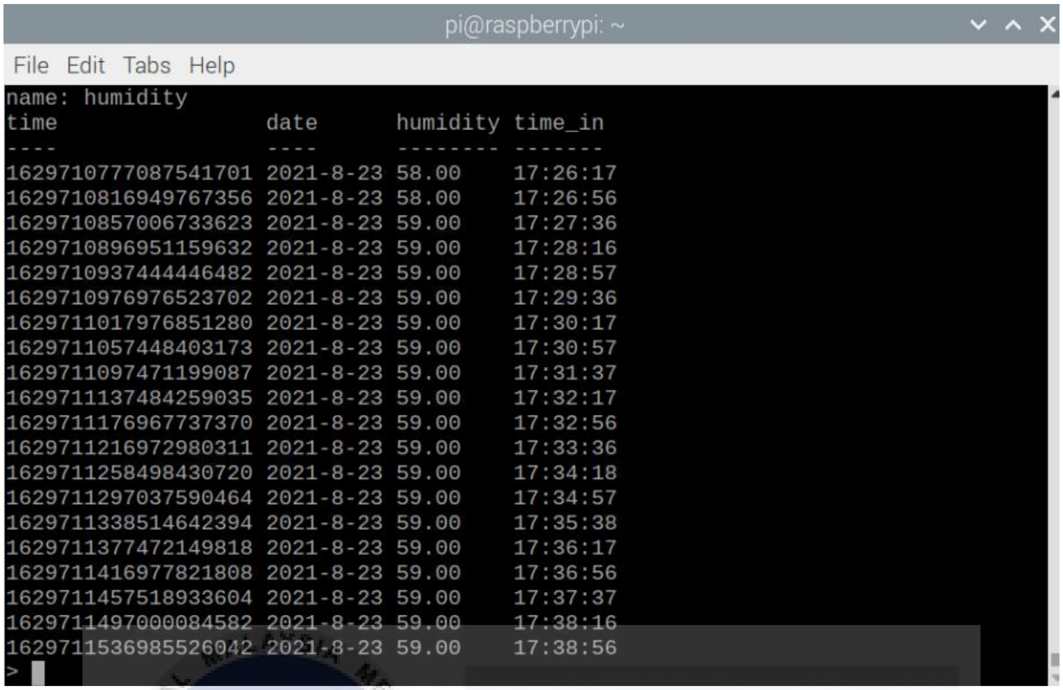
6.5.1.4 InfluxDB Test

When the sensor collected reading from the surroundings, the system will process the data and store it in InfluxDB database. The reading will be included with time_id, date, and time of the reading from the sensor.



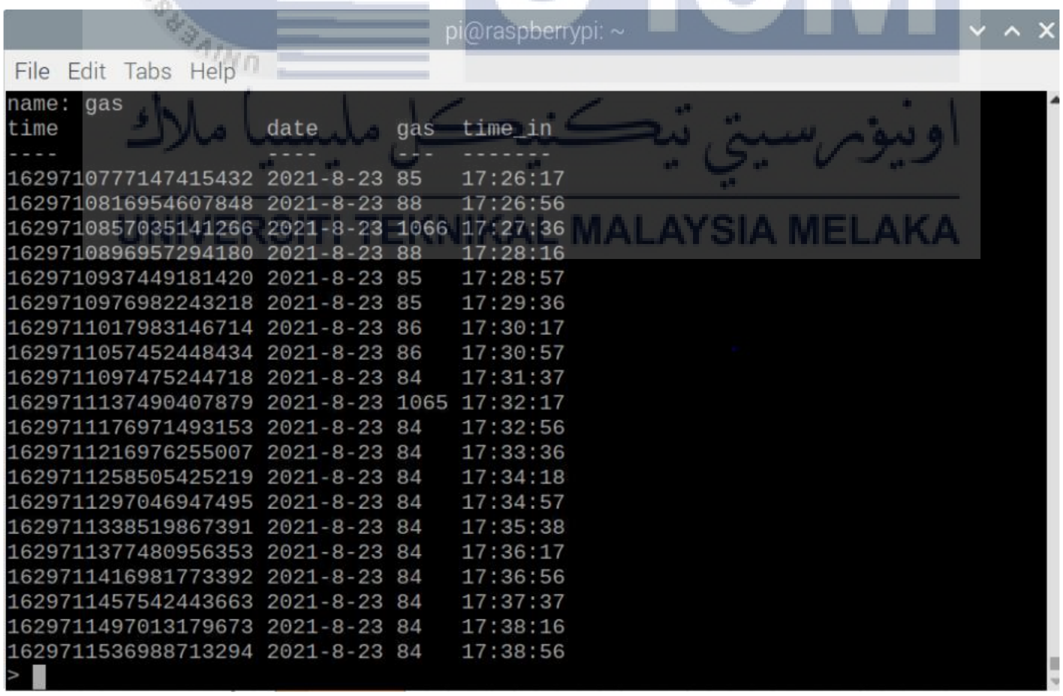

```
name: temperature
time                date      temperature time_in
----                ----      ----------- -------
1624470759378560353 2021-6-24 30.00       1:52:39
1624470798884493021 2021-6-24 30.00       1:53:18
1624470838878380517 2021-6-24 30.00       1:53:58
1624470885147766359 2021-6-24 30.00       1:54:45
1624470925138656689 2021-6-24 30.00       1:55:25
1624470965155639222 2021-6-24 30.00       1:56:5
1624471005153172994 2021-6-24 30.00       1:56:45
1624471045168245775 2021-6-24 30.00       1:57:25
1624471085158481428 2021-6-24 30.00       1:58:5
1624471125154260699 2021-6-24 30.00       1:58:45
1624471165161450725 2021-6-24 30.00       1:59:25
1624471205171250469 2021-6-24 30.00       2:0:5
1624471245163902049 2021-6-24 30.00       2:0:45
1624471285163006602 2021-6-24 30.00       2:1:25
1624471325167615902 2021-6-24 30.00       2:2:5
1624471365164314504 2021-6-24 30.00       2:2:45
1624471405175559038 2021-6-24 30.00       2:3:25
1624471445168659867 2021-6-24 30.00       2:4:5
1624471485177347215 2021-6-24 30.00       2:4:45
1624471525177615558 2021-6-24 30.00       2:5:25
>
```

Figure 6.10: Temperature Reading Stored in Temperature Table



```
pi@raspberrypi: ~
File Edit Tabs Help
name: humidity
time                date      humidity time_in
----                ----      -------- -------
1629710777087541701 2021-8-23 58.00    17:26:17
1629710816949767356 2021-8-23 58.00    17:26:56
1629710857006733623 2021-8-23 59.00    17:27:36
1629710896951159632 2021-8-23 59.00    17:28:16
1629710937444446482 2021-8-23 59.00    17:28:57
1629710976976523702 2021-8-23 59.00    17:29:36
1629711017976851280 2021-8-23 59.00    17:30:17
1629711057448403173 2021-8-23 59.00    17:30:57
1629711097471199087 2021-8-23 59.00    17:31:37
1629711137484259035 2021-8-23 59.00    17:32:17
1629711176967737370 2021-8-23 59.00    17:32:56
1629711216972980311 2021-8-23 59.00    17:33:36
1629711258498430720 2021-8-23 59.00    17:34:18
1629711297037590464 2021-8-23 59.00    17:34:57
1629711338514642394 2021-8-23 59.00    17:35:38
1629711377472149818 2021-8-23 59.00    17:36:17
1629711416977821808 2021-8-23 59.00    17:36:56
1629711457518933604 2021-8-23 59.00    17:37:37
1629711497000084582 2021-8-23 59.00    17:38:16
1629711536985526042 2021-8-23 59.00    17:38:56
>
```

Figure 6.11 Humidity Reading Stored in Humidity Table



```
pi@raspberrypi: ~
File Edit Tabs Help
name: gas
time                date      gas  time_in
----                ----      ---  -------
1629710777147415432 2021-8-23 85   17:26:17
1629710816954607848 2021-8-23 88   17:26:56
1629710857035141266 2021-8-23 1066 17:27:36
1629710896957294180 2021-8-23 88   17:28:16
1629710937449181420 2021-8-23 85   17:28:57
1629710976982243218 2021-8-23 85   17:29:36
1629711017983146714 2021-8-23 86   17:30:17
1629711057452448434 2021-8-23 86   17:30:57
1629711097475244718 2021-8-23 84   17:31:37
1629711137490407879 2021-8-23 1065 17:32:17
1629711176971493153 2021-8-23 84   17:32:56
1629711216976255007 2021-8-23 84   17:33:36
1629711258505425219 2021-8-23 84   17:34:18
1629711297046947495 2021-8-23 84   17:34:57
1629711338519867391 2021-8-23 84   17:35:38
1629711377480956353 2021-8-23 84   17:36:17
1629711416981773392 2021-8-23 84   17:36:56
1629711457542443663 2021-8-23 84   17:37:37
1629711497013179673 2021-8-23 84   17:38:16
1629711536988713294 2021-8-23 84   17:38:56
>
```

Figure 6.12: Gas Reading Stored in Gas Table

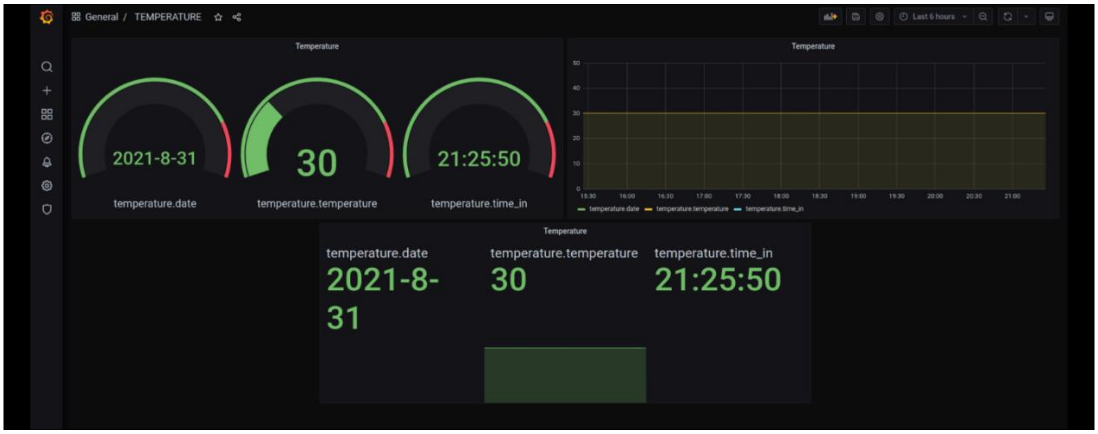


Figure 6.13: Temperature Data Grafana Dashboard

Figure 6.14: Humidity Data Grafana Dashboard

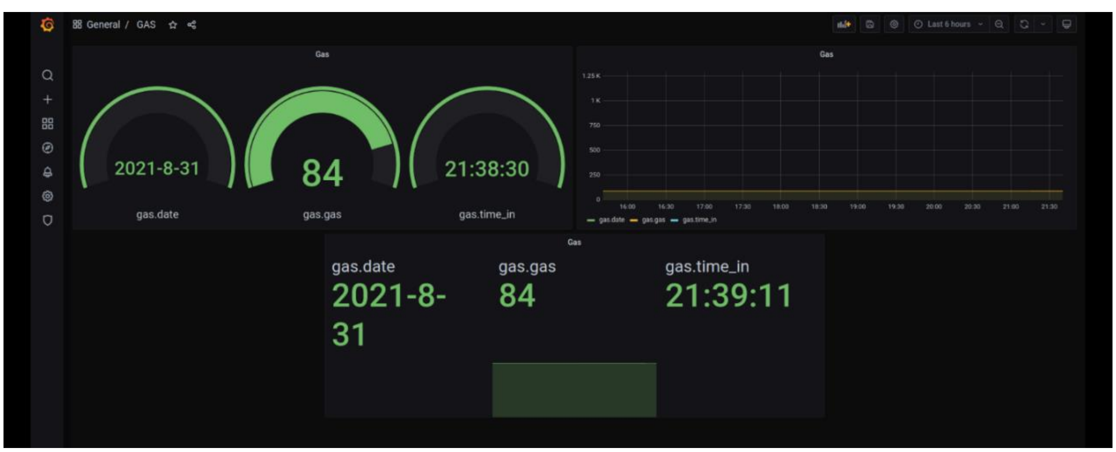


Figure 6.15: Gas Data Grafana Dashboard

6.5.2 System Usability Scale (SUS)

For this testing, 10 respondents are chosen to give their feedback and recorded regarding Data Logger for IoT System. The feedback only consists of one type of testing which is user interface testing that has been provided by the system. The test is conducted from user aspect which they will navigate through the user interface which is Grafana on how the data is shown. Due to the pandemics time, the testing is done by using online meeting platform such as Google Meet, Zoom, Microsoft Team, and Discord. The tester will be brief shortly about the purpose of the system before they can test the user interface. The tester will be given permission to control the system through the developer's computer by using option provided by the online meeting platform. After completing the testing, the tester will be given a google form to be answered which containing SUS question. Figure 6.16 and Figure 6.17 shows the information about the tester.

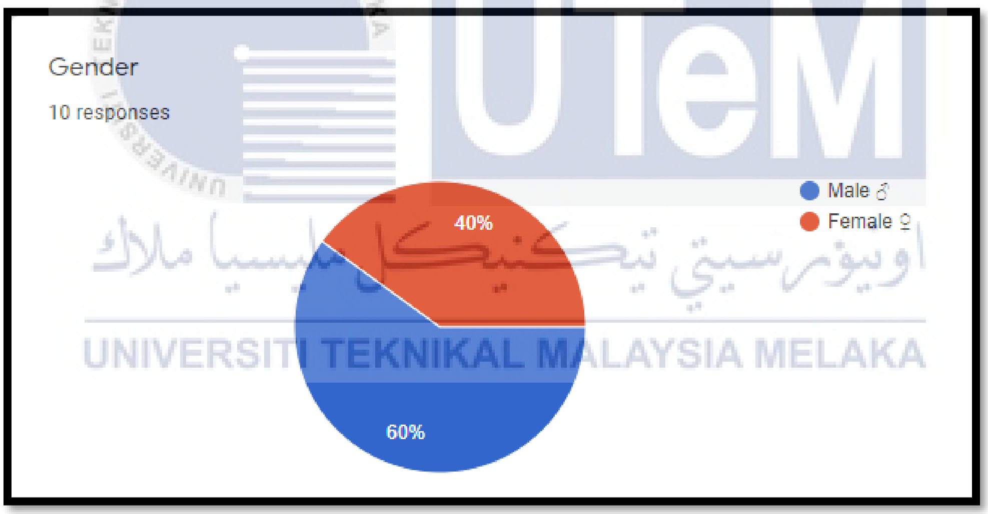


Figure 6.16: Gender Information

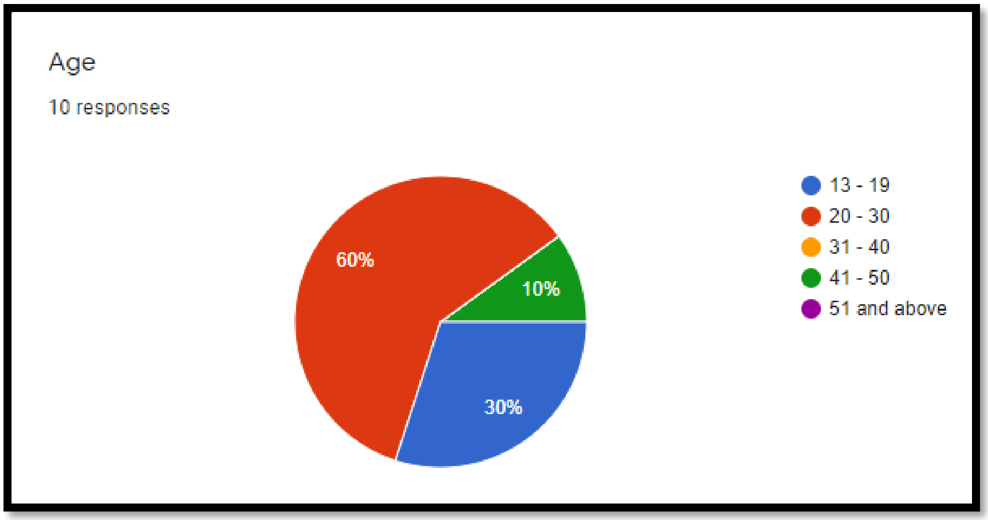


Figure 6.17: Age Information

For the question provided inside the Google Form, a set of ten question is asked (Thomas, n.d.), to the tester which is listed in Figure 6.18 below.

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

Figure 6.18: Set of Questions

The feedback question is answered based on scale from strongly disagree, disagree, average, agree, and strongly agree. All of the answer is tabulated into histogram chart as shown in Figure 6.19.

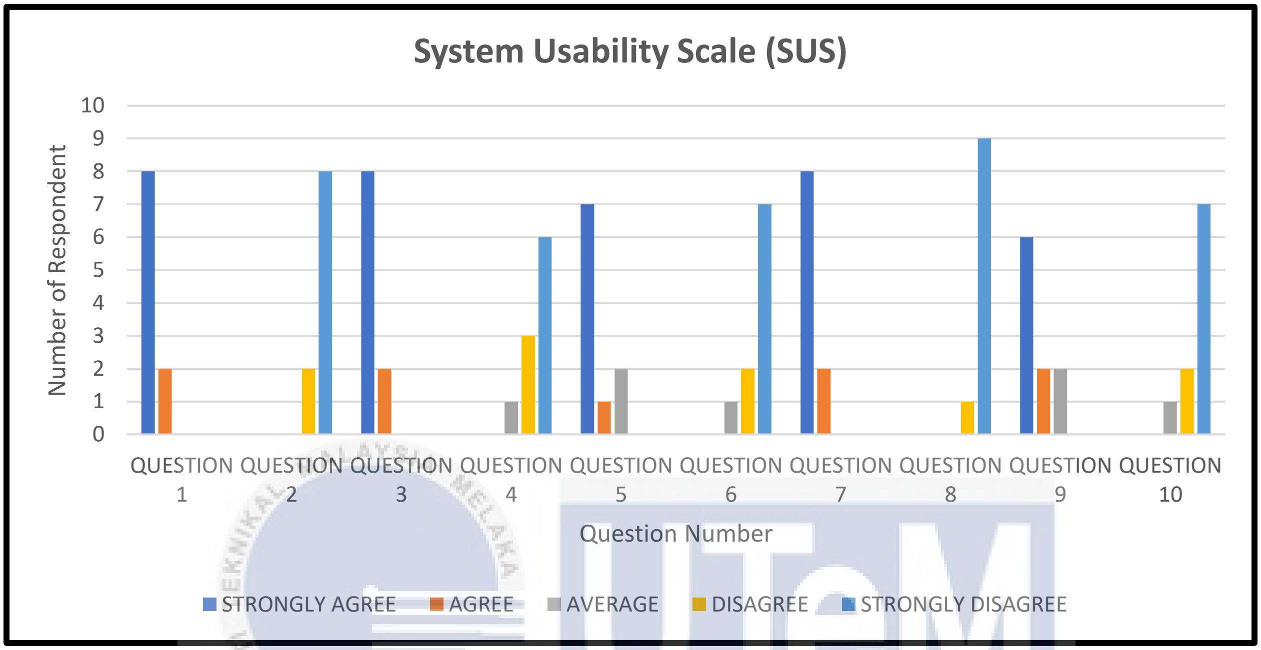


Figure 6.19: Data of System Usability Scale

Based on the collected data shown on previous figure, the calculation score for System Usability Scale (SUS) is counted. Formula used to calculate the data is shown as follows:

1. For each of the odd numbered questions, subtract 1 from the score.
2. For each of the even numbered questions, subtract their value from 5.
3. Take these new values which you have found and add up the total score. Then multiply it by 2.5.

Table 6.5 shows a tabulated data calculation of the System Usability Scale (SUS). The grade for the Data Logger for IoT System is A because the final System Usability Score is 93.25.

Table 6.5: Calculation of System Usability Scale

Number of respondents	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	SUS Raw Score	SUS Final Score
1	4	4	4	4	4	4	4	4	4	4	40	100
2	4	4	4	4	4	4	4	4	4	4	40	100
3	4	4	4	4	4	4	4	4	4	4	40	100
4	3	3	4	2	4	3	3	3	4	2	37	92.5
5	4	4	4	4	4	4	4	4	4	4	40	100
6	4	4	3	3	4	3	4	4	3	4	36	90
7	3	3	3	3	2	4	3	4	2	3	30	75
8	4	4	4	3	3	4	4	4	2	3	35	87.5
9	4	4	4	4	2	2	4	4	3	4	35	87.5
10	4	4	4	4	4	4	4	4	4	4	40	100
Average:												**93.25/100**

6.6 Conclusion

For the conclusion, the testing process is a crucial part in order to verify whether the project is successful or not. It helps the developer to improve the performance of the system by minimizing any error that occur in the system.it is also help in improving the user experiences from the feedback given by the end users. As for the result and analysis, we can conclude that the device can detect temperature and humidity from the surrounding and the presence of gas.

CHAPTER VII:

PROJECT CONCLUSION

7.1 Introduction

In this chapter, there will be a conclusion and summarization for the project from the beginning until the completion of the system. The limitation and future work of the project also will be stated for any further improvement in the future. This will effectively improve the efficiency of the system and make it much more efficient and comprehensive.

7.2 Project Summarization

7.2.1 Project Objective

There are three project objective that is used for the completion of the project which is:

i. To identify characteristic of existing data logger in factory
ii. To develop a data logger that integrated with database, smart notification, and dashboard for IoT System.
iii. To validate the performance of the developed data logger

7.2.2 Project Weakness and Strength

7.2.2.1 Project Weakness

In this project, there are multiple weakness identified during the implementation of the project such as the high power consumes. This project needs to standby at any time in order to verify the temperature and humidity change and also gas presence from the surroundings. It needs to operate 24/7 a week without any downtime. Because of that, the system cannot be operated when there no electricity supplied. Other than that, the sensor uses to get reading from the surrounding might show a wrong reading because of the device overheating. MQ-2 gas sensor is sensitive to overheating. It might show a high gas reading although the gas presence at the surrounding is low due to the sensor overheating.

7.2.2.2 Project Strength

In this project, multiple strengths are found. One of the strengths is the system provides a real-time notification to the users. The system will send alert notification to the user through Telegram application if the reading get by the sensor is exceed the set limit. The alert notification will be included with alert message together with the exceed data collected by the sensor. Other than that, the system has its own user interface that can be used by the user to view all the collected data. The user only needs to login into the Grafana interface by using the correct username and password. After that, user can choose on how to view the data simply by choosing available option on the Grafana dashboard. This project also easy to install and maintain because it is developed with minimum cost which suitable for every small or medium industries.

7.3 Project Contribution

Data Logger for IoT System will help any organization or small industries to create a data logger simply by programming it using Node Red as software system. This project is developed in a low cost and comprehensive function because hardware use for this system is using a cheap microprocessor that are able to connect with multiple sensors. Other than that, this data logger will be integrated with a database and a web-based interface. Data collected from the sensors will be saved in a database at it can be shown to the user by using a web-based user interface.

7.4 Project Limitation

There are several limitations about the project such as this project requires a lot of storage in order to save a lot of data collected by the sensor. Other than that, the Raspberry Pi need to connect to the Internet to ensure the system can send a notification message to the user. The Internet connection signal and speed are important to prevent any delay from happening.

7.5 Future Work

One of the future works that can be done to the system to make it more efficient and comprehensive is provide it with a cloud database so it can reduce storage space used on the microprocessor storage. Other than that, a microprocessor with high ram capacity can be used in order to increase the system processing time. Hence, it can ease the user to manage and handle the data.

7.6 Conclusion

In conclusion, this project is successful since it meets all of the objectives that has been stated before. All of the implementation and testing part is clearly shown in the previous chapter. Based on the project weakness and project strength that has been discussed before, it can be concluded that there are several weaknesses that need to be fixed in the future to ensure the system can be operated with a better performance. The project strength also determined that there were multiple advantages that user can get by using this system. Other than that, project contribution and project limitation are also identified and discussed in detail. Last but not least, multiple enhancement can be done to this project so it can be operated with a better performance in the future.

REFERENCES

[1] Dalibor Dobrilovic, Vladimir Brtka, Zeljko Stojanov, Gordana Jotanovic, Dragan Perakovic and Goran Jausevac. (2021). A Model for Working Environment Monitoring in Smart Manufacturing

[2] Ahmad Nizar Harun, Norliza Mohamed, Robiah Ahmad, Abd Rahman Abdul Rahim, Nurul Najwa Ani. (2019). Improved Internet of Things (IoT) monitoring system for growth optimization of Brassica chinensis

[3] V. Pasquali, G. D'Alessandro, R. Gualtieri, F. Leccese. (2017). A new data logger based on Raspberry-Pi for Arctic Notostraca locomotion investigations

[4] A.P. Montoya, F.A. Obando, J.A. Osorio, J.G. Morales, M. Kacira. (2020). Design and implementation of a low-cost sensor network to monitor environmental and agronomic variables in a plant factory

[5] Andrew Carre, Terence Williamson. (2018). Design and validation of a low cost indoor environment quality data logger

[6] Mani Dheeraj Mudaliar, N. Sivakumar. (2020). IoT based real time energy monitoring system using Raspberry Pi

[7] Ricardo Toro, Jorge E. Correa and Placid M. Ferreira. (2018). A Cloud-Monitoring Service for Manufacturing Environments

[8] B. Guragai, T. Hashimoto, K. Oguma, S. Takizawa. (2018). Data logger-based measurement of household water consumption and micro-component analysis of an intermittent water supply system

[9] Victoria Jayne Mawson, Ben Richard Hughes. (2020). Deep learning techniques for energy forecasting and condition monitoring in the manufacturing sector

[10] Wei Chen. (2020). Intelligent manufacturing production line data monitoring system for industrial internet of things

APPENDICES

CHAPTER 6 TESTING

Implementation of System Usability Scale (SUS)