CHAPTER I

INTRODUCTION

1.1. Background

The use of renewable energy sources is a solution in responding to the challenges of the occurring energy crisis. One of the renewable energy is the use of solar energy. Solar panels are devices that can convert solar energy into electrical energy. The use of solar panels in Indonesia is beneficial because Indonesia locates in the equator where almost the entire area gets 10 to 12 hours of sunlight every day and Indonesia is a tropical country that has the potential for solar energy with average daily insolation of 4.5-4.8 kWh/m^2/day [1].

To be able to maximize the use of solar panels, it is necessary to monitor the parameters of the solar panels so that, from the results of the daily *monitoring*, the average produced energy can be seen, how large the loading capacity is, thus ensuring the continuity of the electricity supply. The data can also be usable to analyze whether there is a decrease or disturbance in tool/panel. Therefore, maintenance planning or corrective actions can be implementable on solar panels.

Many studies have been carried out to simplify the process of monitoring solar panel parameters, including *website*-based *online monitoring (*in network). However, from the done studies, there are still spaces for development, namely:

1. Some *monitoring* systems already built are still *offline*.

2. The existing *online monitoring* system requires additional financing due to the large number of components required and the complexity of system design.
3. Time to send data from the designed tool to the web application still takes 15-60 seconds.

Based on the above problems, it is necessary to design an *online monitoring* system of solar panel parameters that is practical in its design, with minimal financing and fast data sending.

1.2. Formulation of Question

Devices to design and build:

a) A hardware device capable of measuring the value of the voltage and current flowing from the solar panel;

b) A hardware device capable of sending current, voltage, and power data over the internet, with low *latency*;

c) A simple system with minimal financing.

1.3. Limitation of question

In this study, the limitations of the questions are as follows:

a) In this study, the sensor used to read the current and voltage flowing from the solar panel was the INA219, because the sensor could provide current and voltage measurements at one time;

b) The obtained current and voltage reading data were sent via an internet network connection using the NodeMCU module;

c) The sent data, namely current, voltage and power, could be monitored through the *website firebase* that could be opened using a computer or smartphone;

d) The data displayed on the website consisted of voltage, current, and power. Power was obtainable from the product of the voltage and current.

1.4. Objectives & Benefits

1.4.1. Objectives

a) To design a system that can measure solar panel current and voltage data, utilizing the INA219 sensor using an I2C type communication line so that it allowed the sensor to measure 2 parameters at one time.

b) To implement the *Internet of Things* concept utilizing the ESP 8266 contained in the NodeMCU so that the device could connect to the internet, and the Arduino IDE for programming the tool.

c) To design a *monitoring online* system for current, voltage and power data from solar panels that were simple and easy to access with minimum finance utilizing the free *internet of things* platform.

1.4.2. Benefits

The benefits of this study are:

a) To apply the knowledge gained during education in lectures;

b) To contribute thought in design and manufacture of *monitoring online* systems (in the network) of solar panel parameters;

c) To facilitate the process of *monitoring* the solar panel parameters be accessible anywhere using an internet connection.

1.5. Systematics

To facilitate this study, the author made a systematic study in five chapters. The first chapter discusses the basics of making a final project, such as, background, formulation of question, limitation of question, objectives and benefits and systematics. CHAPTER II contains a literature review to support this study. CHAPTER III contains description of the devices to design. The description is in the designs of flow scheme system, hardware and software. Furthermore, Chapter IV describes the results and discussion, including the results of making and testing the Designs of an IoT-based *monitoring online* system for solar panel parameters. Finally, CHAPTER V contains conclusions.

REFERENCES

This section attaches sources of writing and information that the author used in writing this thesis. The sources used could be in the form of scientific writings, books, theses, or information from trusted or official media.

APPENDIX

The appendix contains additional information related to thesis writing, such as program data and photos at the time of data collection.

CHAPTER II

LITERATURE REVIEW

2.1. Solar Panel

2.1.1. Definition

A solar panel is a device consisting of solar cell technology that can convert light into electricity. Solar panel utilizes the sun that is the most substantial light source. Solar panel relies on photovoltaic effect to absorb solar energy, causing a strong current to flow between two layers of different charges. A solar module (photovoltaic) is a number of solar cells arranged in series and parallel, to increase the generated voltage and current so that it is sufficient for use of systems of load power supply. To get maximum output of electrical energy, surface of solar module must always point to the sun. The solar cell material itself consists of protective glass and transparent adhesive material. These protect the solar cell material from environmental conditions, anti-reflection material to absorb more light and reduce the amount of reflected light, P-type and N-type semi-conductors (made of a mixture of silicon) to generate an electric field, initial and final channels (made of thin metal) to send electrons to electrical appliances [2].

Figure 2.1. Solar Panel

2.1.2. Working Principle of Solar Panel

The working principle of silicon solar cells based on the concept of a pn *junction* semiconductor. The cell consists of an n-doped and p-doped semiconductor layer forming a p-n *junction*, an antireflection layer, and a metal substrate as a place for current to flow from the n-type (*electron*) and p-type (*hole*) layers. The n-type semiconductor was obtainable by doping silicon with elements from group V so that there were more valence electrons than the surrounding atoms. On the other hand, p-type semiconductors were available by doping through group III so that the valence electrons were one deficit compared to the surrounding atoms. When the two types of material were in contact, the excessive electrons of the n-type diffused into the p-type. Therefore, the n-doping area would have positive charge, while the p-doping area would have negative charge. The electric field occurring in both pushed electrons back into the n-region and *hole* into the p-region. In this process, a p-n *junction* had formed. By adding metal contacts in the p and n areas, a diode had formed [3]. In simple terms, the process of forming the electromotive force (EMF) in a solar cell is as follows:

1) Photons of sunlight hitting the solar panel are then absorbed by a semiconductor material such as silicon;
2) Electrons (negative charge) throw out of their atoms, so they flow through the semiconductor material to generate electricity. Positive charges called *holes* flow in the opposite direction to the electrons in silicon solar panels;
3) The combination/arrangement of several solar panels converts solar energy into a DC current source;

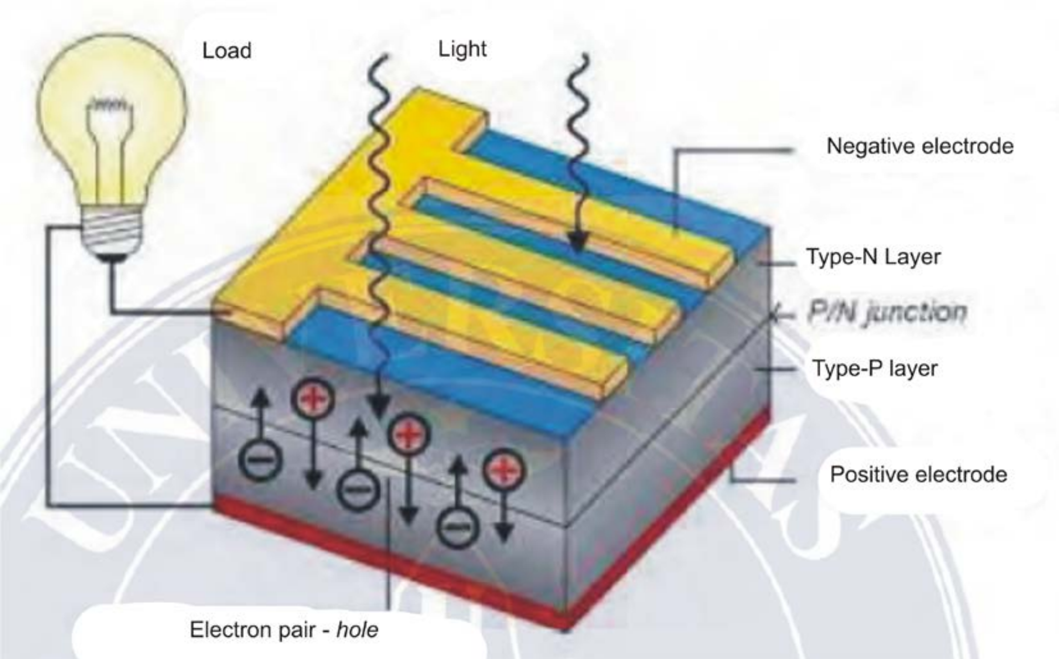


Figure 2.2. Working Principle of Solar Panel

2.1.3. Basic Theory of Semiconductors

Solar radiation energy can be convertible into direct current using thin layers of pure silicon (Si) or other semiconductor materials. For use as a semiconductor, silicon must be pure to less than one impurity atom per 1010 silicon atoms. This crystallization will occur when liquid silicon becomes solid because each atom has a valence electron, thus the occurrence of a crystal form where each silicon atom with tension has one valence electron. Semiconductor is a material functioning as a conductor and acting as an insulator depending on the place and condition of the material. There are two types of semiconductors, namely intrinsic semiconductors and extrinsic semiconductors. These semiconductors consist of two types, namely type P and type N. Pure silicon crystals have no free electrons, so it is a poor conductor of electricity. It takes a lot of energy to remove an electron from its bond. To form a P-type semiconductor, a semiconductor with a valence of 4 is addable with a material having a valence of 3, usually known as an impurity material. These types of materials include boron, aluminum, calcium, and indium. The addition of this impurity material will reduce one bond so that it takes the form of a *hole*. This *hole* can move from one place to another in the crystal. What happens forever is that the electrons of the crystal fill the vacant holes and create new ones. The holes move because there are electrons that fill them, so each hole will have a positive charge equal to and opposite to the negative charge of the electrons. Likewise, to form an

N-type silicon semiconductor, which is addable with 5-valence materials commonly used, such as phosphorus, it is called an N-type silicon semiconductor.

A *junction* semiconductor is a combination of P-type and N-type semiconductors causing a potential difference called the barrier voltage and the limitation or boundary between the two junctions is called a *junction* [4].

2.1.4. *Internet Of Things* (IoT)

Internet of Things is a concept aiming at expanding the benefits of continuously connected internet connectivity. Capabilities such as data sharing, remote control, and so on, include objects in the real world. For example, food, electronics, collectibles, any equipment, including living things, are connected to local and global networks through embedded sensors and are always active [5]. This system can monitor or manage connected objects and equipment. The connected sensors can also monitor the nature, humans and animals. The goal is to make humans interact with objects more easily, so that objects can also communicate to other objects. *Internet of things* is a technological revolution representing the future of computing and communications [6].

2.2.1. IoT Techniques Perspective

Physical things (physical objects) can be representable in the information world through one or more *virtual things* (virtual objects). However, virtual objects can stand alone without any *mapping* with physical objects. A *device* is a tool that can communicate and some additional capabilities (*sensing, actuation, data capture, data storage* and *data processing*). A *device* will take the required information, present it as information and send it for further processing. However, there are also *devices* that can directly process is based on the information and communication received. *Communication between devices*: devices can communicate over a communication network through a gateway (case A), communicate without passing through a gateway (case B) or *direct* or ad-hoc (case C) or a combination of the two. [7].

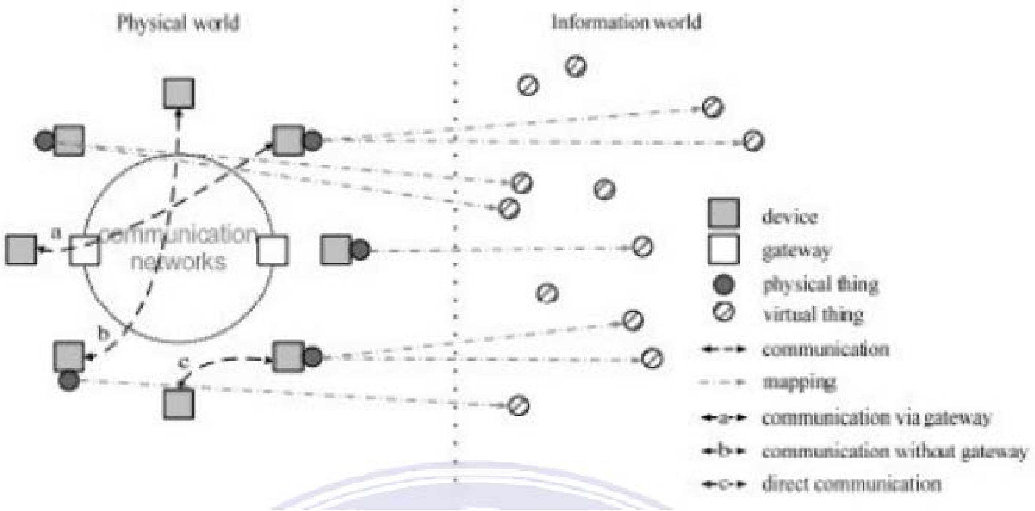


Figure 2.3 Technical Perspective of the *Internet of Things*

2.3. Sensor INA219

The INA219 is a shunt current and power monitor via an I2C-connection or an SMBUS compatible interface. The sensor monitors shunt voltage drop and supply voltage (bus supply voltage), with programmable multiplication conversion and filtering. Programmable calibration values, combined with an internal multiplier, allow direct reading of current in amperes. Additional multiplier registers can produce power in watts. This module is capable of measuring currents up to 3.2A and voltages of 26VDC with VCC 3V-5.5V [8].

INA219 Sensor Specifications:

1) Measuring voltage 0-26 V
2) Measuring currents up to 3.2 A with a resolution of 0.8 mA
3) Equipped with a 0.1 ohm 1% 2 W resistor
4) Having 16 programmable addresses
5) Having high accuracy, maximum 0.5%
6) Options for interference filtering
7) Having registered for calibration
8) Small size 2 x 2.2 cm
9) Equipped with I2C communication

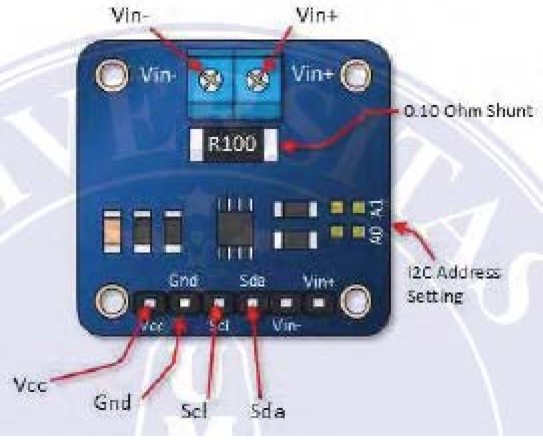


Figure 2.4. INA219 Sensor

2.4. *Solar Charge Controller* (SCC)

Solar Charge Controller is an electronic device used in solar power generation systems. This tool regulates the direct current charged to the battery and regulates the current coming out of the battery to the load. This tool also functions as controller so that *overcharging* does not occur (*overcharging* because the battery is full), full discharging and overvoltage of solar panels. Overvoltage, charging and overvoltage can reduce battery life. SCC (*Solar Charge Controller*) applies the principle of PWM (*pulse with modulation*) to regulate the function of charging the battery and releasing current from the battery to the load [9]. This tool usually has two terminals (positive and negative) for input from the solar panel, two terminals for output to the battery, and two terminals for output to the load. 12-V solar panels generally have an output voltage of 16 - 21 V, so without this tool, the battery will damage due to overcharging and voltage instability of the solar panel.

Here are the complete functions of the *Solar Charge Controller*:

1) To regulate charging current to battery, avoid overcharging and overvoltage;
2) To regulate the current released from the battery to the load, so that the battery is not overloaded or fully discharged; and
3) To monitor temperature from battery.

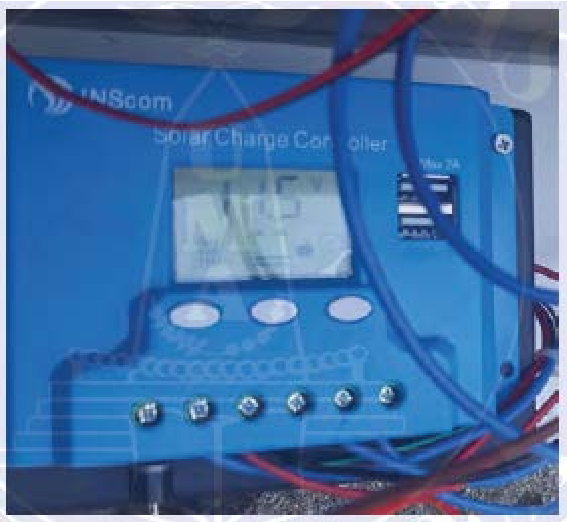


Figure 2.5. *Solar Charge Controller*

2.5. ESP8266 NodeMKU Vz

NodeMCU is an IoT *platform* that is open-source, meaning that many manufacturers are free to produce and develop it. The NodeMCU contains hardware in the form of a *system on chip* ESP8266. NodeMCU is an extension of the ESP8266 with e-Lua based firmware [10]. It is equipped with a micro USB port serving as an interface for programming and power supply. In addition, the NodeMCU has a button (*push button*), namely the reset and flash buttons. NodeMCU uses the Lua programming language, a package of ESP8266. The Lua language has the same logic and programming structure as the C language, and only the syntax is different. If you want to use the Lua language, you must use the Lua *loader* and Lua *uploader tools*.

NodeMCU can also be programmable using the Arduino IDE *software*, making a few changes in the *board manager* in the Arduino IDE *software*. Before using this board, you must flash the *firmware* first, so that it can support the *tool* you want to use. When using Arduino IDE, the *firmware* used is AT-Command output from *AI-Thinker*. To use the *tool loader*, the *firmware* used is the NodeMCU firmware.

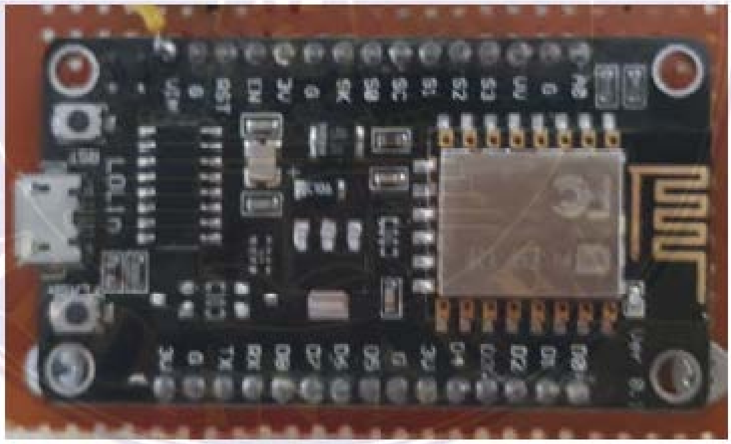


Figure 2.6. ESP8266 NodeMCU V3

Table 2.1 Specifications of ESP8266 NodeMCU V3:

SPECIFICATION	**NODEMCU V3**
Manufacturer Vendor	LoLin
Type ESP8266	ESP-12E
USB Port	Micro USB
GPIO Pin	13
ADC	1 pin (10 bit)
USB to Serial Converter	CH340G
Power Input	3.3 - 5V DC
Module Size	57 x 30 mm
Frequency	2.4 GHz – 22.5 GHz
WiFi	IEEE 802.11 b/g/n

Clock Speed	40/26/24 MHz
Flash Memory	4 MB
PWM Canal	10 Canals

2.5.1. *Wifi Shield* ESP8266 ESP-12E

ESP8266 is a wifi module functioning as an additional device for microcontrollers such as Arduino so that it can connect directly to wifi and make TCP/IP connections. There are several types of ESP8266 in circulation, some of which are easy to find, namely, the ESP-01, ESP-07, and ESP-12 types. The function of each type of ESP is almost the same, the difference is the GPIO (*General Purpose Input Output*) pin on each type. NodeMCU already has ESP8266 (Especially the ESP-12 series, including ESP-12E), so the features of NodeMCU will be more or less the same as ESP-12 (also ESP-12E for NodeMCU V2 and V3) [11]. Some of these features include:

1) 10 GPIO ports of D0-D10
2) PWM functionality
3) I2C and SPI interface
4) 1-wire interface
5) ADC

The following figure shows more clearly the position of the feet/pins of the ESP-12E.

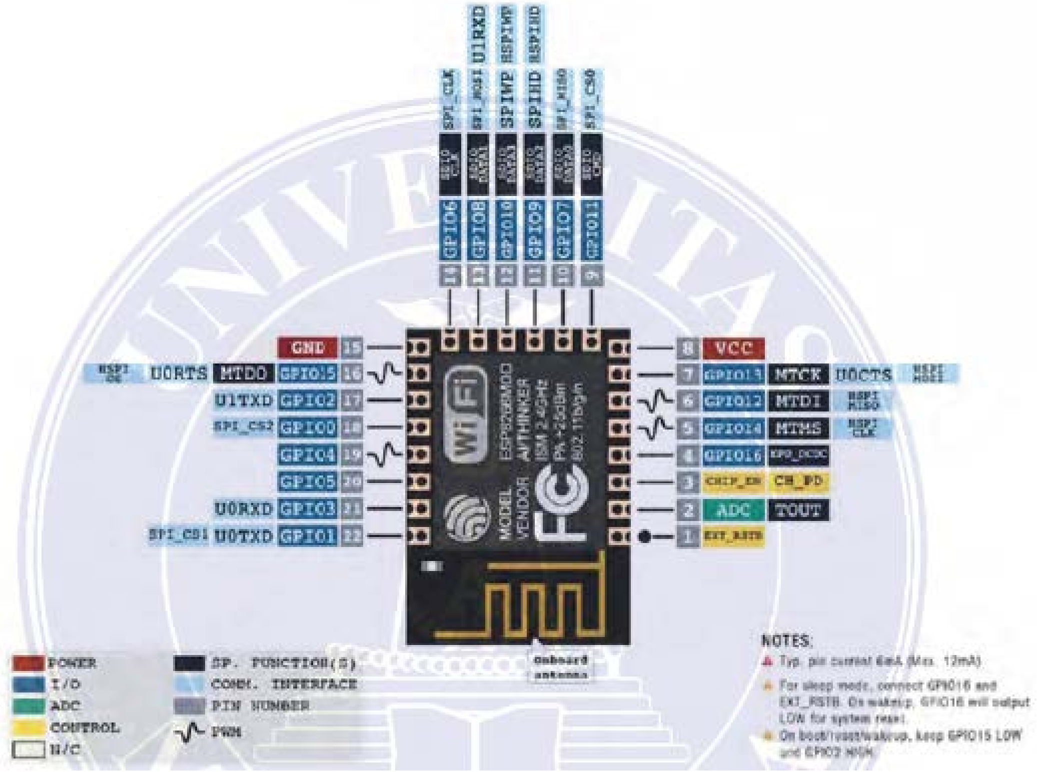


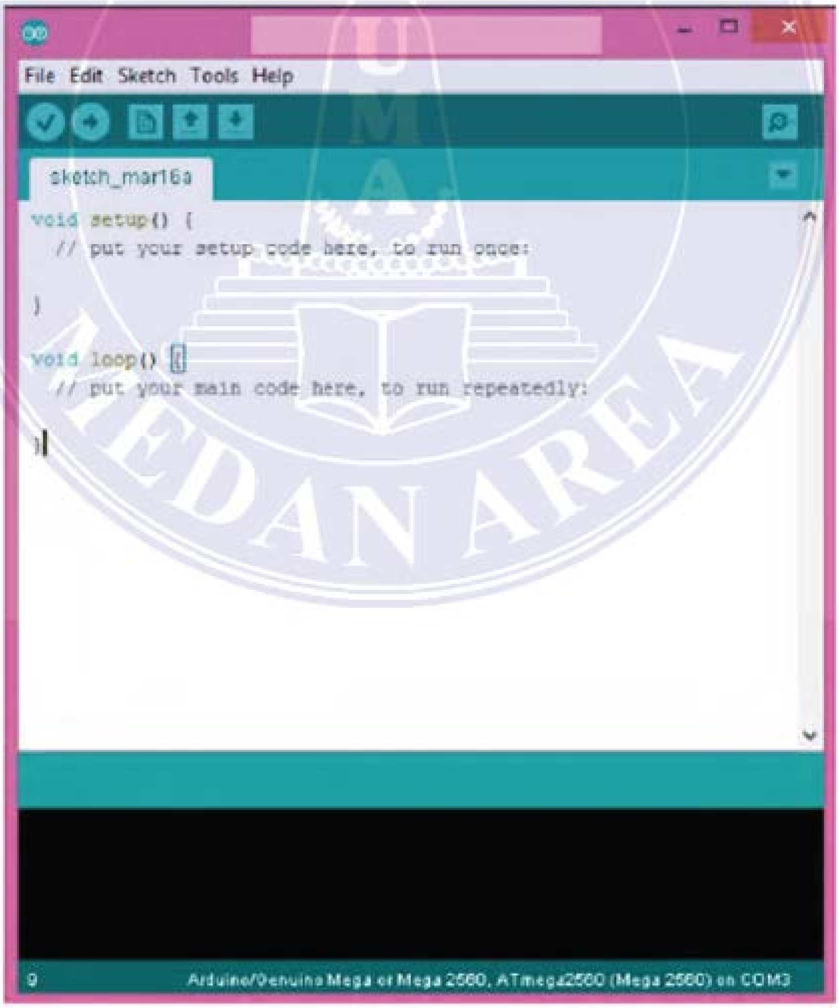
Figure 2.7 ESP8266 ESP-12E

The working voltage of the ESP8266 uses JEDEC (*Joint Electron Device Engineering Council*) voltage standard, namely 3.3 VDC to function. It is unlike the AVR microcontroller and most Arduino boards having a standard TTL voltage of 5 VDC. Even so, the NodeMCU can still be connected to a 5-VDC voltage via the available Micro USB port or Vin pin.

2.5.2. Arduino IDE *Software*

To enter the program into the NodeMCU to run as desired, the Arduino IDE *software* is required. IDE stands for *Integrated Development Environment*. Programs written using the Arduino IDE *software* are called sketches. The text editor writes the sketches and saves them in a file using the .ino extension [12].

This *software* contains a black *message box* functioning to display status, such as *error* messages, *compile*, and *upload* programs. The bottom of this *software* shows information about the configured *board* and the *COM ports* used.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

9 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM3

Figure 2.8 Arduino IDE *Software* Display

The following is information on the Arduino IDE parts based on Figure 2.6.

1) *Verify/compile*

It functions to check whether the sketch made is in accordance with programming rules.

2) *Upload*

It functions to send programs already made/verified into the Microcontroller board, in this study, NodeMCU.

3) *New*

It functions to make a new *sketch*.

4) *Open*

It functions to sketch ever made to change/edit and re-*upload* on microcontroller *board*.

5) *Save*

It functions to save the made sketch

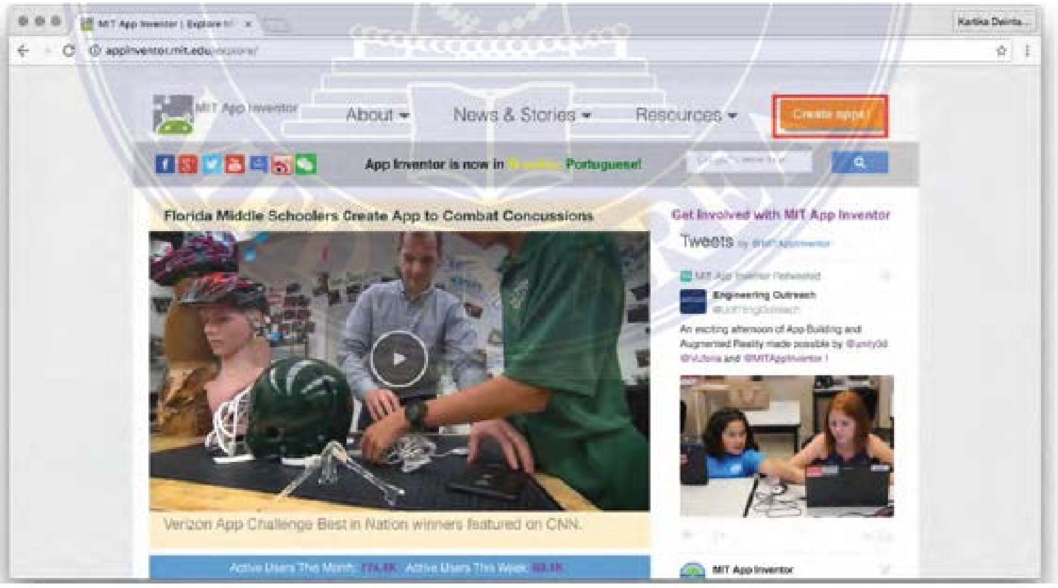
6) *Serial Monitor*

It functions to open *serial monitor* that is window displaying any data sent and exchanged between the microcontroller *board* and *sketch* in the serial *port*. *Serial monitor* can be usable to display process values, reading values, and even error messages.

2.6. MIT App Inventor

MIT App Inventor is an innovative application developed by Google and MIT to introduce and develop android programming by transforming complex text-based programming languages into visual-based (*drag and drop*) blocks in the form of blocks [13].

This *visual programming framework* is related to the Scratch programming language of MIT, which is specifically an implementation of the open block distributed by the MIT *Scheller Teacher Education Program* taken from the study by Ricarose Roque. The app inventor uses the *Kawa Language Framework* and *Kawa's dialect* developed by Per Bothner and distributed as part of the GNU *Operating System* by the *Free Software Foundation* as a *compiler* that translates *visual block programming* to be implemented on the Android platform.


MIT App Inventor | Explore MIT
appinventor.mit.edu/explore/
Kartika Dwinta...
MIT App Inventor
About
News & Stories
Resources
Create apps!
Tweets by @MITAppInventor
MIT App Inventor Retweeted
Engineering Outreach
MIT App Inventor

App Inventor is now in Brazilian Portuguese!

Florida Middle Schoolers Create App to Combat Concussions

Verizon App Challenge Best in Nation winners featured on CNN.

Active Users This Month: [illegible] Active Users This Week: [illegible]

Get Involved with MIT App Inventor

An exciting afternoon of App Building and Augmented Reality made possible by @unity3d @Vuforia and @MITAppInventor !

Figure 2.9. MIT App Inventor web page display

The following are the installation/Export stages of the Application.

A. Testing

There are three connection lines to test the application that has been made, namely:

1. AI Companion

Required devices:

- Computer
- Android Device
- Wifi Connection

Processing stage:

- Install “MIT AI2 Companion” app on PlayStore
- Connect android device and computer on the same wifi
- In the app inventor application, select the Connect – AI Companion menu
- Scan QRCode using Android device

2. Emulator

Required devices:

- Computer

Processing stage:

- Install the App Inventor desktop app
- Run the App Inventor desktop application
- In the App Inventor web application, select the Connect – Emulator menu

3. USB

Required devices:

- Computer
- Android Device
- USB Cable

Processing Stages:

- Install the App Inventor desktop app
- Run the App Inventor desktop application
- Set USB settings on Android device “USB Debugging On”
- In the App Inventor web application, select the *Connect – USB* menu

B. Building

After finishing design of the desired application, the application can be installed on Android by downloading the APK file that can be done in 2 ways, namely:

1. Scan QR Code

   The APK can be downloaded by scanning the QRCode provided when selecting the *Build – Provide QRCode for APK* menu.

2. Save to Computer

   APKs can be saved to your computer, by selecting the *Build - Save .apk to My Computer* menu.

2.7. Firebase

Firebase is a service from Google used to make it easier for application developers to develop applications. With Firebase, app developers can focus on developing apps without having to put in a lot of effort. Two interesting features of Firebase are *Firebase Remote Config* and *Firebase Realtime Database*. In addition, there are supporting features for applications requiring notifications, namely *Firebase Notification*.

The following are the features of Firebase:

A. Firebase Cloud Messaging

Firebase Cloud Messaging (FCM) used is known as Google Cloud Messaging (GCM), where Firebase Cloud Messaging (FCM) can send and receive messages across platforms such as Android, iOS, and web apps. Firebase Cloud Messaging (FCM) is a service provided by Firebase to replace Google Cloud Messaging. Using Firebase Cloud Messaging, we can provide notifications and establish two-way communication between devices. The technology used is classifiable into two, namely: XMPP (Extensible Messaging and Presence Protocol) and HTTP (Hypertext Transfer Protocol).

B. Firebase Authentication

Firebase Authentication is an authentication system service implementing a client-side code, so that users can register and login to Facebook, GitHub, Twitter and Google (Google Play Games) applications.

C. Firebase Remote Config

Firebase Remote Config is a feature allowing you to make configuration changes within Android/iOS apps, without having to update apps on the Play Store/App Store. The way Remote Config works is that the application first saves an XML file containing parameters whose values can be changeable through the Firebase console. Then the firebase object of the application will request data from the server, then process the data.

D. Firebase Real-time Database

Firebase provides a realtime database and backend as a service. This service provides developers with an application-programming interface allowing application data to synchronize across clients and store in the Firebase cloud. Developers using real-time databases can secure data using security rules enforced by enterprise servers. The *Firebase Realtime Database* is a database hosted on the cloud. Data is stored as JSON and synchronized in real-time to every connected client.

E. Firebase Storage

Firebase Storage is designable for app developers who need to store and display user-generated content, such as photos or videos and add Google security to file uploads and downloads for Firebase apps, regardless of network quality. Developers can use it to store images, audio, video, or other user-generated content directly from Client's Firebase SDK. Google Cloud Storage powers the Firebase Storage.

F. Firebase Hosting

Firebase Hosting provides fast and secure hosting for web applications and static and dynamic content. This service delivers files via the Content Delivery Network (CDN) or via HTTP Secure (HTTPS) and Secure Sockets Layer (SSL) encryption.

CHAPTER III

SYSTEM DESIGN

3.1. System Planning Flow Scheme

Figure 3.1 shows the flow diagram of system design.

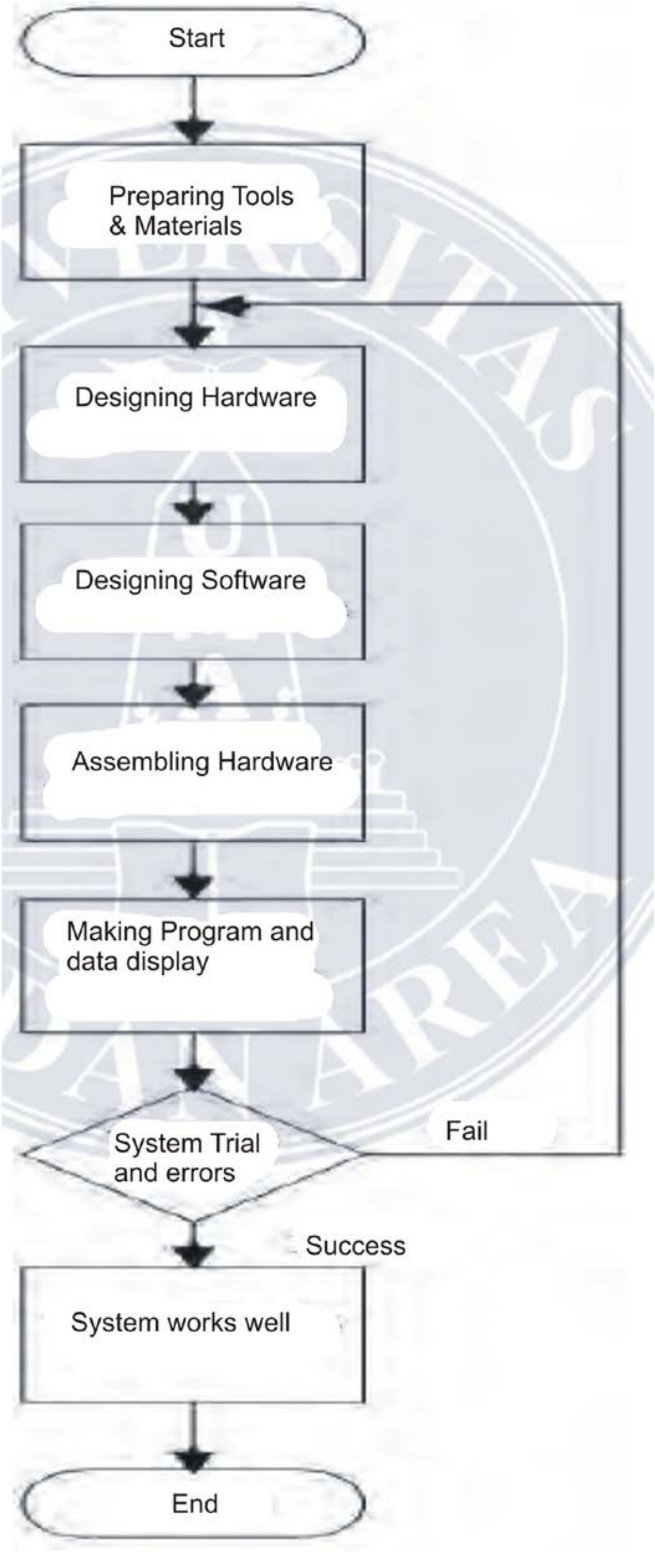


Figure 3.1. System Design Flowchart

3.2. Tools and Materials

The following is a description of the materials and tools needed in making an IoT-based online monitoring system for solar panel parameters:

1) 30-WP solar panel
2) INA219 Sensor
3) NodeMCU
4) ESP8266 Wifi Module
5) LM2596S
6) Solar Charge Control (SCC)
7) 12-V 7-Ah Battery
8) Digital Voltmeter
9) Digital Multimeter
10) 12-V/3-W LED, 3 Block
11) Laptops, Android phones
12) Wires, tin, solder, hot glue
13) Pipe (panel mount), wood (assembly mount)
14) Other complementary tools needed in assembling electronic devices.

3.3. System Design

The block diagram below shows the hardware design of the designed system.

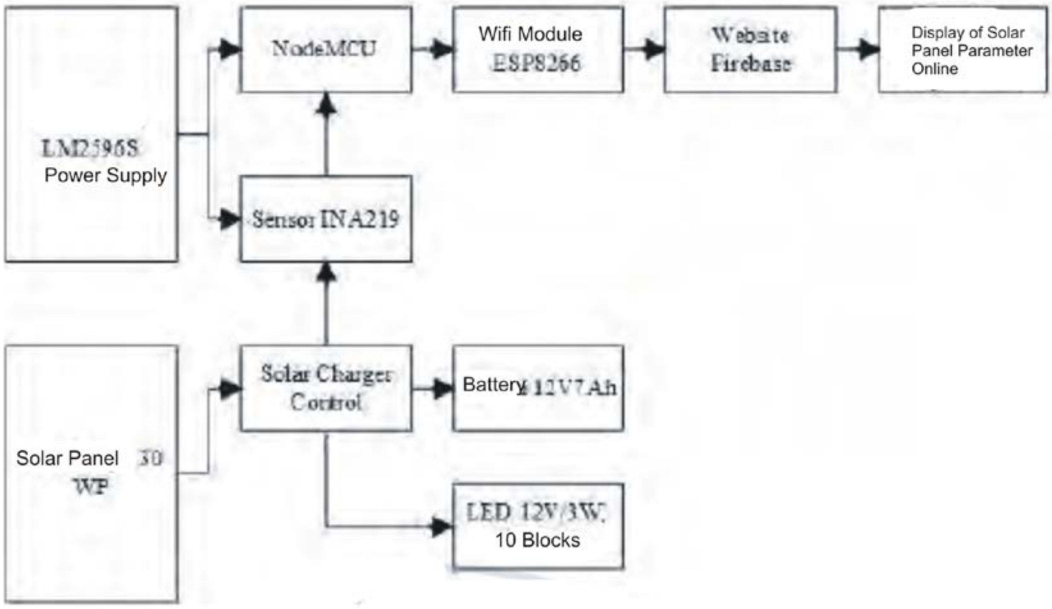


Figure 3.2. System Diagram Block

The description on the block diagram above is as follows:

1) NodeMCU is the control center of this tool, used to control the whole starting from the sensor input to the output of the monitoring system of solar panel.
2) ESP8266 Wifi module is a communication medium between the microcontroller and the internet network.
3) The Firebase website is an IoT platform used to display solar panel parameter data sent by ESP8266.

3.3.1. Hardware Design

The hardware design of this tool starts from the placement and installation of small components such as sensors, power supplies and NodeMCU in the box, and the installation of larger components such as digital voltmeters, batteries, LEDs, *Solar Charge Controls* on the wood board. The solar panels are placed on a stand made of pipes.

3.3.2. INA219 Sensor Circuit

The INA219 sensor is a sensor used to measure current and voltage of solar panels. The I2C communication type is possible to connect more sensors to just two cable lines. Figure 3.3 and table 3.1 below show the installation of the sensor circuit and the description of the sensor pins.

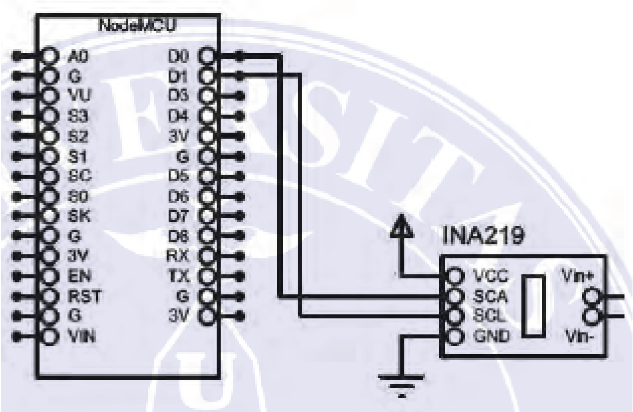
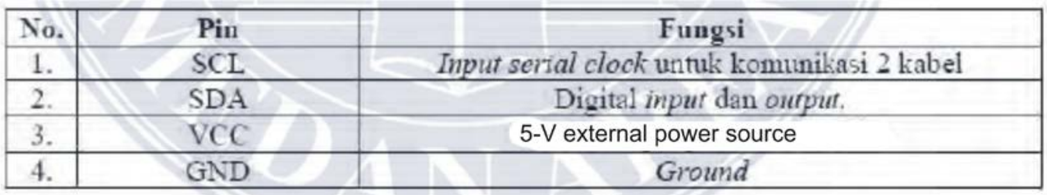


Figure 3.3. INA219 Sensor Circuit

Table 3.1. Use of INA219 sensor feet/pins sensor



No.	Pin	Fungsi
1.	SCL	*Input serial clock* untuk komunikasi 2 kabel
2.	SDA	Digital *input* dan *output*.
3.	VCC	5-V external power source
4.	GND	*Ground*

3.3.1.2. Power Supply Circuit

This circuit serves to provide power to the entire existing circuit. This circuit is inputted with 12-V DC voltage from the battery or adapter, then the voltage will be leveled by a 680μF capacitor. A voltage regulator

(LM2596S) is used so that the resulting output remains 5-V DC even though there is a change in the input voltage. Figure 3.4 below shows the power supply circuit.

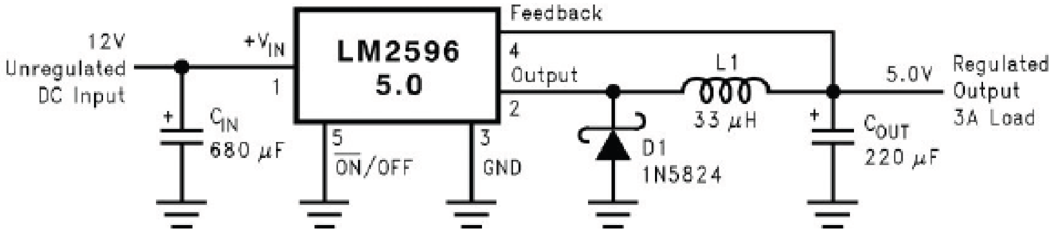


Figure 3.4. Power Supply Circuit

3.3.1.3. Entire Circuit

This circuit is the entire circuit used in the design and development of this IoT-based solar panel parameter online monitoring system. Figure 3.5 below shows the whole set.

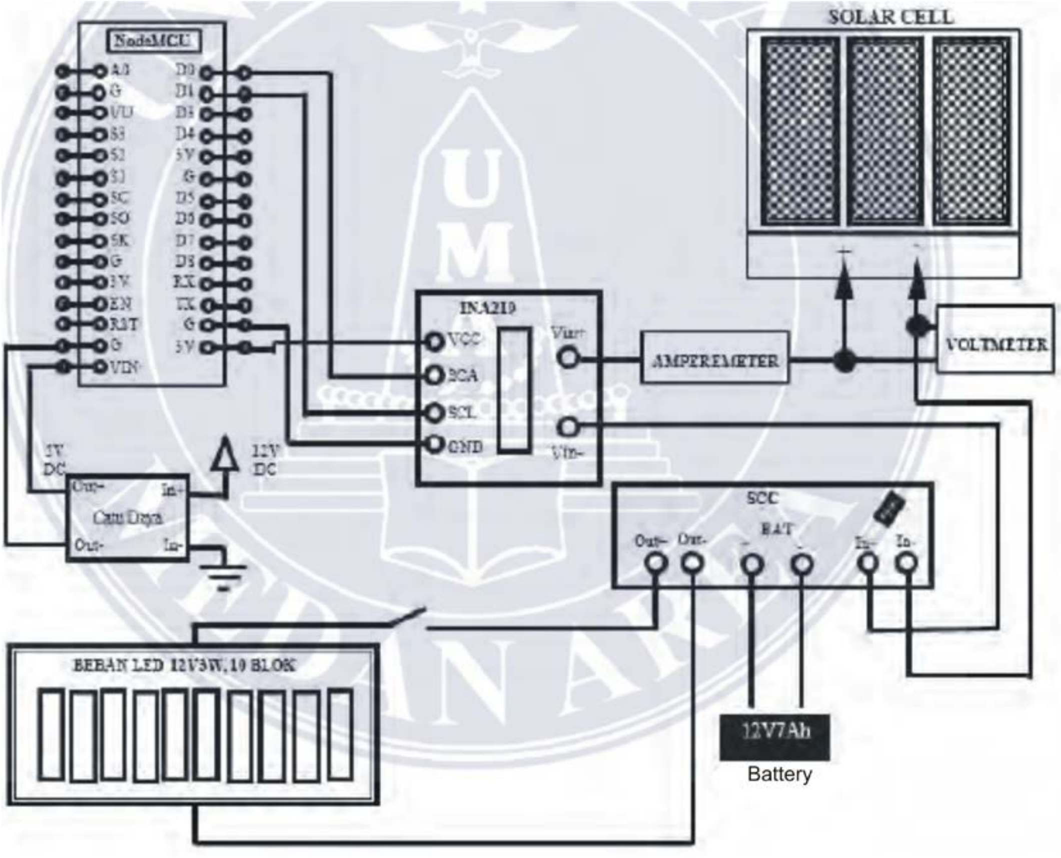


Figure 3.5. Entire Gadget Circuit

3.3.2. Software Design

Software design is made after the hardware design completes. The design of this software includes the design of the program used to translate the data taken from the sensor into the value of voltage and current and send it via the internet to a website page or cellular phone. In this study, the software used in the design of an IoT-based online monitoring system for solar panel parameters includes:

1) Microsoft Visio

This software is used to draw circuit schemas.

2) Arduino IDE 1.8.1

This software is used for program writing and interface with NodeMCU.

3) Website Firebase

This web application is used to display the online solar panel parameter readings, namely current, voltage and power.

4) App Inventor

This *software* is used in designing android applications *online monitoring* system.

3.3.3. Flow Chart

To be able to display the INA219 sensor reading data *online*, the program is designed in such a way that the data can be displayed on *website* pages and on smart cellular phones (Android). The NodeMCU will calibrate and process the data obtained from the INA219 sensor. Then Wifi Shield ESP8266 will send the data to the server to display and store in the database. Figure 3.6 below shows the work system flow chart of the tool.

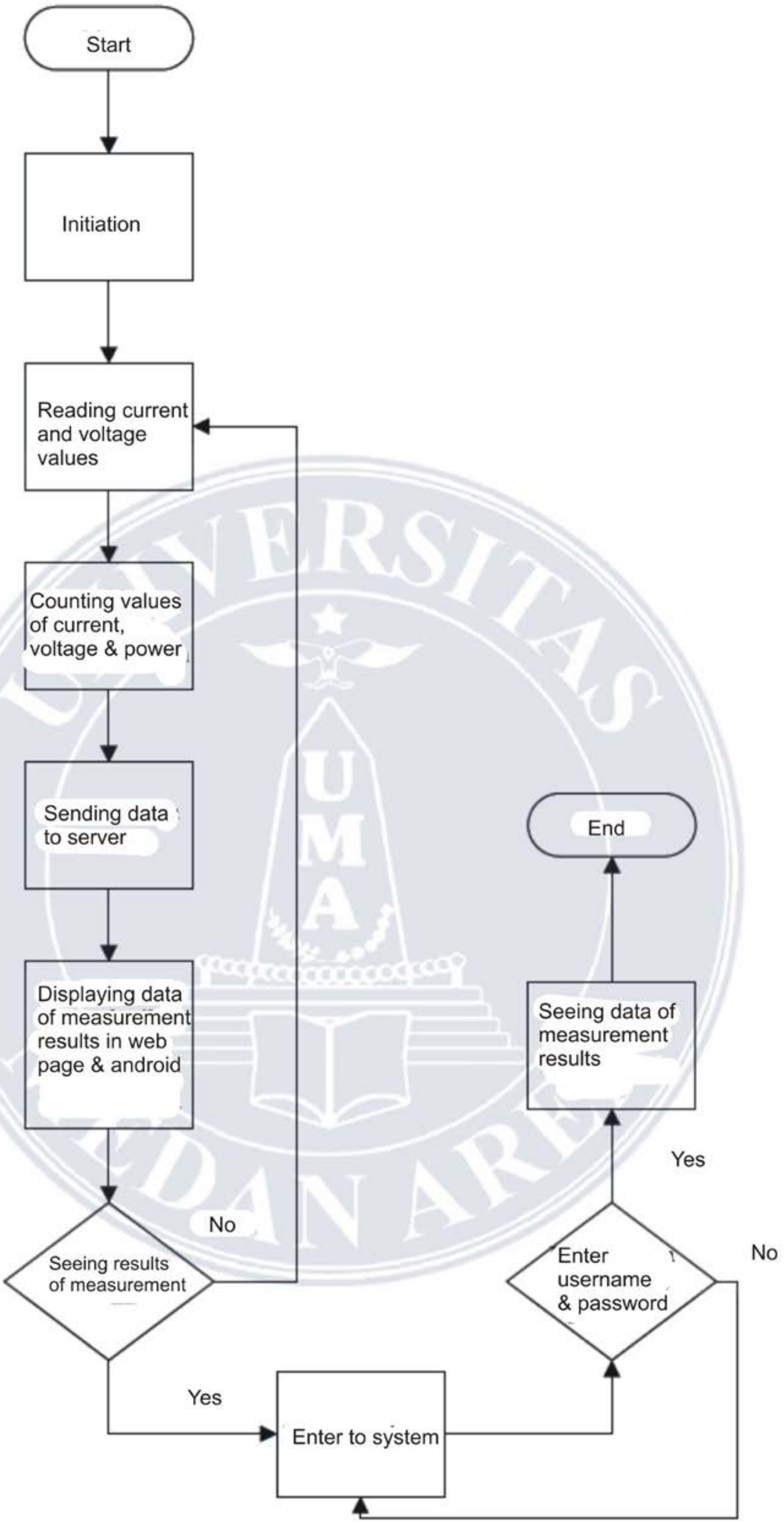

Start
Initiation
Reading current and voltage values
Counting values of current, voltage & power
Sending data to server
Displaying data of measurement results in web page & android
Seeing results of measurement
No
Yes
Enter to system
Enter username & password
No
Yes
Seeing data of measurement results
End

Figure 3.6. Tool work system flow chart

CHAPTER V

CONCLUSION DAN RECOMMENDATION

5.1. Conclusions

The results of the study in chapter IV can be usable to conclude as follows:

1) Designing and building of an IoT-Based Solar Panel Parameter Online Monitoring System used the ESP8266 NodeMCU V3 as a control center for devices connected to the internet to send data to websites and Android.
2) The error percentages of reading the voltage and current of the INA219 sensor with a digital multi-tester were 0.28% and 2.29%, respectively. The limitations of the multi-tester in reading small measurement values affected it.
3) The solar charge controller functioned to maintain the incoming voltage from the solar panel in accordance with the working voltage of the battery ± 12 Volts.
4) The displayed measurement data had an update time of 1 second.

5.2. Recommendations

Based on the study results and conclusions, the following are the authors' recommendations for further study to develop, namely:

1) The solar panel holder is designed to be able to follow the movement of sunlight, so that the panel can rotate automatically.

2) The results of monitoring solar panels can be usable as an input for the microcontroller to control the movement of the solar panels.
3) The concept of the *Internet of Things* (IoT) can be usable as a reference to the development.
4) The results of monitoring solar panels can be usable as an input for the microcontroller to control the movement of the solar panels.

PROOFREADING

1.	occurring	:	The occurring
2.	very useful	:	beneficial
3.	Transparent- adhesive	:	transparent adhesive
4.	it	:	is
5.	registers	:	registered
6.	Open source	:	open-source,
7.	only	:	, and only
8.	in	:	them in
9.	Edit,	:	edit
10.	study	:	The study
11.	Thus	:	Therefore
12.	extra	:	additional
13.	strongest	:	most substantial
14.	obtainable	:	available
15.	entire	:	whole
16.	To simplify the process of *monitoring*	:	Many studies have been carried out to simplify
17.	use	:	The use
18.	An average	:	average
19.	as well as	:	and
20.	limitations	:	The limitations
21.	was	:	were
22.	process	:	The process
23.	first	:	The first
24.	contains a discussion of	:	discusses
25.	was	:	were
26.	having the ability to	:	That can
27.	Are able to	:	can
28.	This tool is used	:	This tool regulates
29.	text	:	The text
30.	An Information	:	Information
31.	microcontroller	:	the microcontroller