IoT Monitoring System for Solar Power Plant Based on MQTT Publisher / Subscriber Protocol

Januar Muhamad Ramadhan
Department of Electrical Engineering
UIN Sunan Gunung Djati Bandung
Bandung, Indonesia
januarmuhamad@gmail.com

Rina Mardiati
Department of Electrical Engineering
UIN Sunan Gunung Djati Bandung
Bandung, Indonesia
r_mardiati@uinsgd.ac.id

Irsyad Nashirul Haq
Department of Physic Engineering
Bandung Institute of Technology
Bandung, Indonesia
inhprof@gmail.com

***Abstract*—One of the renewable sources of electrical energy that can be used as an alternative to fossil fuels is solar energy. Basically, solar energy can be converted into electrical energy by solar cells. In the operation itself, a reliable system is needed to monitor the performance of energy production in the PLTS system used so that it can be monitored and electricity supply and disturbances are maintained in real time. In this study, we proposed a monitoring system for IoT-based solar power plants using the publisher and subscriber communication method with ESP32 as a microcontroller and google cloud platform as a cloud server which can be used in real-time PV mini-grid monitoring systems. This proposed system aims to monitor the performance of PV mini-grid which displayed in the database which placed in a virtual computer that acts as an MQTT broker. The results of this study showed that the proposed system performs well and can be accessed from the existing database. Furthermore, this system has good accuracy since the measurement error value is very small and data transmission system is able to work in real time because the broker is able to receive data with a minimum response time.**

Keywords—IoT, Monitoring System, Solar Panel, MQTT,

I. INTRODUCTION

Fossil fuels that are used as a resource of electrical energy are running low because fossil fuels cannot be renewed. Renewable energy sources are now the answer to overcome these problems [1]. Basically, solar energy can be converted into electrical energy by solar cells with a Solar Power Generator System [1].

In the operation process, a reliable system is needed to monitor the performance of energy production from solar panels and monitor electrical loads so that electricity supply and electricity disruptions are maintained. With Internet of Things (IoT), solar power plants can be remotely and periodically monitoring so that electrical conditions remain stable.

Based on existing literatures, several solar power monitoring system could be implemented using web-based monitoring and control techniques, such as using the microcontroller that integrated with UART client server data communication and the http protocol [2] [3]. While other techniques used a series of Arduino sensors, a GUI-based communication and programming system to monitor weather conditions, and the flow of solar panel electricity from or to the PLN network [4]. Since BMS monitoring system in PLTS using IED communication, acquisition algorithms, cloud databases, and HMI with data communication in the form of TCP / IP with JSON format to retrieve battery parameters [5].

In this research, we proposed the communication protocol using MQTT broker Google Cloud Platform as a server service with computation engines and databases as data storage. MQTT Broker has been worked the advantage that it can simultaneously reduce network device resources, minimum respon time and minimize storage space [6] [7]. Furthermore, we used Google Cloud Platform as a cloud server that has been integrated to the computed engine and mysql database as well as providing MQTT broker features [8]. Our proposed system is built from hardware to back end in the cloud server. Hardware system that has been developed in this study is a prototype off-grid solar power plant which monitored by sensors. The information from the sensors will be forwarded by a microcontroller (publisher) integrated with the cloud server (subcriber) using the internet network (*IoT*).

Prototype off-grid solar power plant works independently and does not depend on other power grid [9] [10]. Solar power plant works when the particles in the solar panels absorb sunlight and then generate electricity. System can work with reliable parameters and performance by source operating voltage, solar irradiation, temperature, spectral distribution [1] [11] [12]. In addition to using solar panels, solar power plants use batteries as energy storage and inverters to convert DC voltage into AC so that it can be consumed by the load [10] [13] [14]. Prototype off-grid solar power plant will be monitored using the ESP32 which supports the IoT-based data transmission process [15] [16] and uses the INA219 and PZEMM004t sensors to monitor energy production and power consumption [17] [18].

The paper is organized as follows. Section 1 gives a general introduction, while Section 2 describes the modeling of solar power plants monitoring with MQTT Broker. In Section 3 describes the testing of functional and perfomance of solar power plants monitoring. Finally, Section 4 presents the conclusion of this research.

II. DESIGN AND IMPLEMENTATION

A. Block Diagram System

The problem posed in this study is how to design a monitoring system for solar power plants using the MQTT Broker. The initial stage of designing a tool requires an initial explanation of how the system works from the tool. We used INA 219 and PZEM004t as the sensors. The data from sensors will sent to ESP32 as the publisher, while virtual computers as subcriber will receive sensor data and then stored in the mysql database. Figure 1 shows the block diagram of this research. As we seen in Fig. 1, our prototype system will send the data

978-1-6654-4402-6/21/$31.00 ©2021 IEEE

to ESP32 as a publisher and at the same time the data will stored to the database.

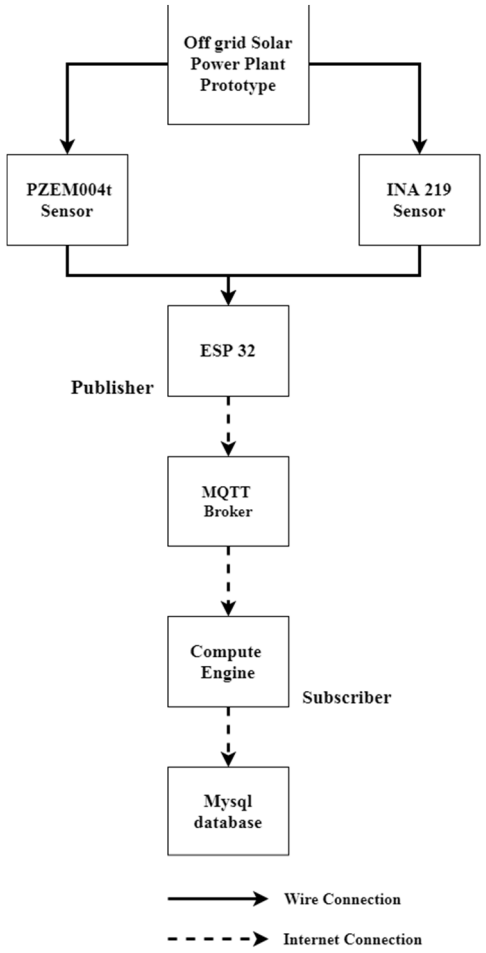

Fig. 1. Research Block Diagram.

B. *Design System*

Solar power plant system consists of three main subsystems, which are solar panels as an energy source, solar charger controller as a link between components, and an inverter as a DC to AC voltage converter. The sensor INA219 and PZEM004t will operate when the sensor gets 5V input from the LM2596 module, then sends data to the ESP32. The data that has been obtained will be published by ESP32 and subcribed by virtual on the Google Cloud Platform using the MQTT communication protocol. The design of hardware can be seen in Fig. 2.

Designing a monitoring software system requires 2 ways with different applications, namely Arduino IDE and GCP virtual computer configuration. The use of the Arduino IDE software compiles files of type ".ino", namely .ino files which contain the source code for the monitoring system that will be used on this system. For configuration on a Google Cloud Platform virtual computer, we have to compile a file of type ".js" containing the monitoring system source code, then specify the publisher and subscriber. The next step, the microcontroller (device) will send the data from the sensor to the virtual computer and stored in a database using MQTT communication protocol. Software design can be seen by Fig. 3.

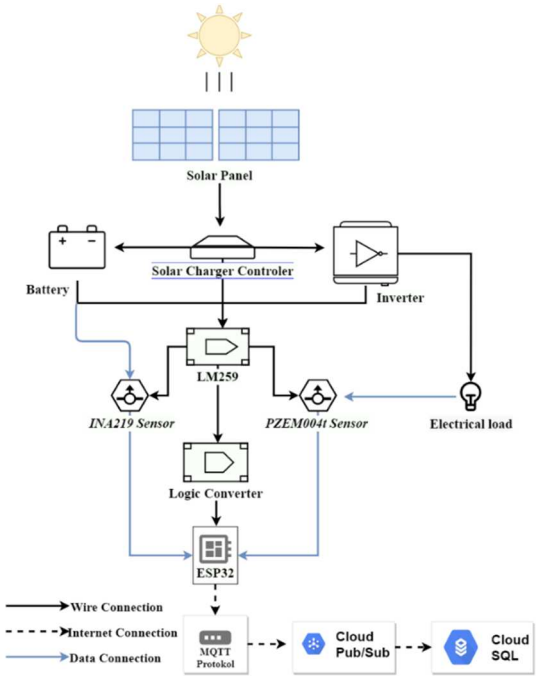

Fig. 2. Hardware Design.

.

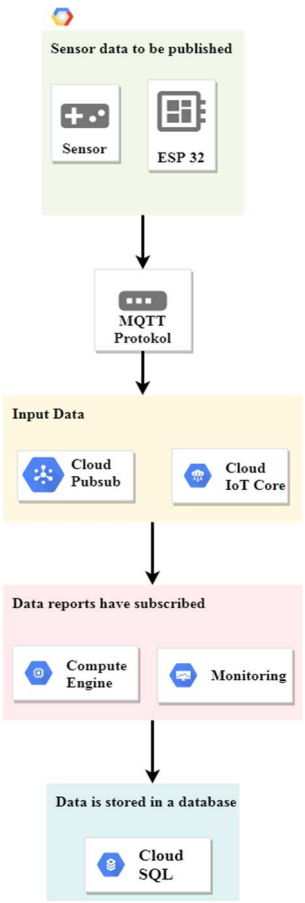

Fig. 3. Software Design.

C. *Implementation*

Hardware implementation was developed by assembling a pre-designed circuit. This circuit will be connected to the monitoring system as shown in Fig. 4 and 5.

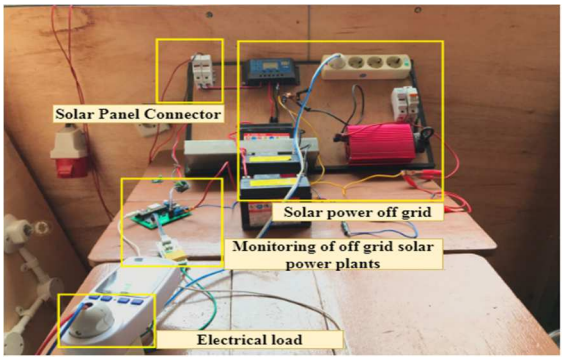

Fig. 4. Prototype Solar Power Plant Off Grid.

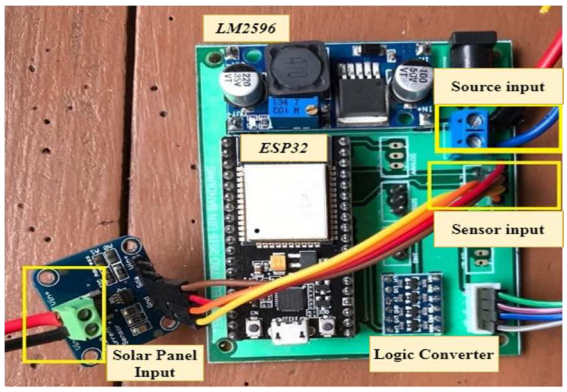

Fig. 5. Monitoring Power Generated by Solar Panels.

Implementation of load monitoring using the PZEM004t sensor with input data in the form of voltage, current and power generated with 2 lamp by the load used can be seen by Fig. 6.

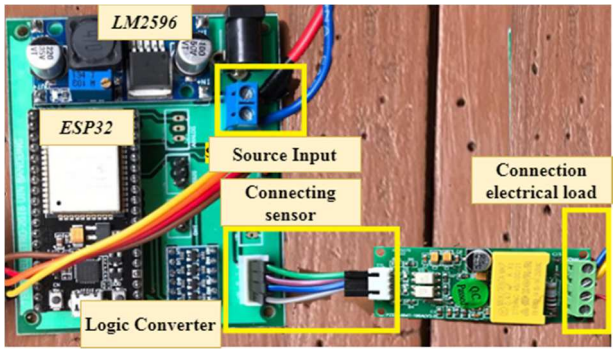

Fig. 6 Monitoring Power Generated by Load.

Software implementation process using ESP32 with Arduino IDE is carried out by executing a modified program which then send from the publisher to a virtual computer using the internet network with the MQTT communication protocol. The program contains a library for creating JWT (web token json) and security declarations on the MQTT Google Cloud Platform. Software implementation can be seen by Fig. 7.

Fig. 7 Implementation Software using ESP32 with Arduino IDE.

For software implementation process using a virtual computer is done by executing a modified program which later receives the publisher's data to a compute engine (subscribe) with the MQTT communication protocol. Software implementation can be seen by Fig. 8.

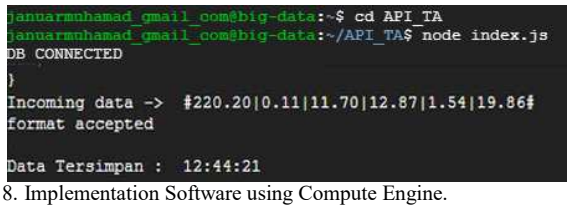

Fig. 8. Implementation Software using Compute Engine.

After successfully entering the virtual computer, the data will automatically be stored in the mysql database that is on the virtual computer. database view can be seen by Fig. 9.

	id	v_in	i_in	p_in	v_out	i_out	p_out	timestamp
Edit Copy Delete	1	12.18	-0.4	4	233.3	0.03	1	2021-03-10 12:44:05
Edit Copy Delete	2	12.17	-0.5	4	233.6	0.03	1	2021-03-10 12:44:05
Edit Copy Delete	3	12.18	-0.3	4	232.2	0.03	1.1	2021-03-10 12:44:08
Edit Copy Delete	4	12.17	0.1	4	232.1	0.03	1.1	2021-03-10 12:44:08
Edit Copy Delete	5	12.18	-0.6	4	232.9	0.03	1	2021-03-10 12:44:08
Edit Copy Delete	6	9.49	-0.4	4	232.2	0.03	1	2021-03-10 12:44:08
Edit Copy Delete	7	0.82	-0.4	0	232.7	0.03	1.1	2021-03-10 12:44:08
Edit Copy Delete	8	0.81	-0.6	0	232.8	0.03	1.1	2021-03-10 12:44:08
Edit Copy Delete	9	12.18	-0.4	4	232.9	0.03	1.1	2021-03-10 12:44:08
Edit Copy Delete	10	0	0	0	233	0.03	1.1	2021-03-10 12:44:08

Fig. 9. Mysql Database Display.

III. Result and Analysis

A. *Functional testing and analysis*

The test was done by comparing the result from the sensor with a measuring instrument and then looking for the sensor error value. This measurement scenario is carried out when the monitoring system works for 30 minutes using two light loads which varies. Measurements are made at one minute intervals either when measuring with a digital multimeter or on the ArduinoIDE serial monitor. This test can be seen Fig. 10.

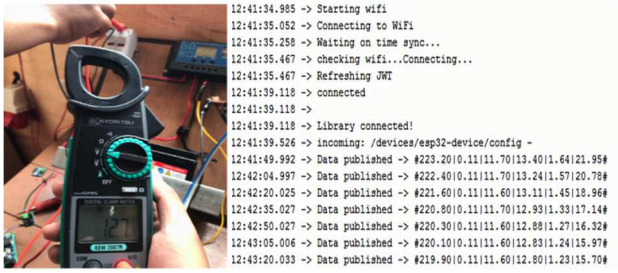

Fig. 10. Functional Testing.

After getting the value of energy production and electrical load both with sensors and measurements with a multimeter, then calculate the accuracy value using the formula :

$$\frac{(multimeter\ measurement - sensor\ measurement)}{multimeter\ measurement} \times 100\%$$

Parameters that are measured for the energy production process, namely the voltage, current and power generated by the solar panel which can be seen in graphic form on Fig. 11.

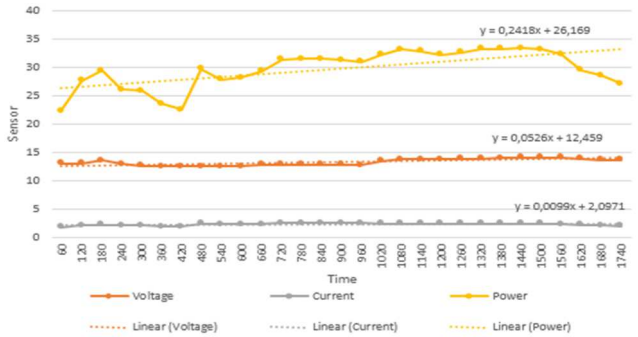

Fig. 11. Power Production Graph.

Based on existing data, the INA219 sensor value obtained has increased and decreased depending on the intensity of sunlight. In accordance with the specifications for the maximum input voltage that works on the solar panel, which is 18 volts DC. The sensor itself has a good linearity value because the value of the x variable which is less than 1 is considered to have a good linearity value because the sensor error value is ± 1%. The factors that cause these differences are due to differences in error values in the specifications of the measuring instruments and sensors used. The digital multimeter has an error value of 1.3% and the sensor is 1%.

For the parameters measured for the load, namely the voltage, current and power generated by the load of 2 lamps, it can be seen in graphical form Fig. 12.

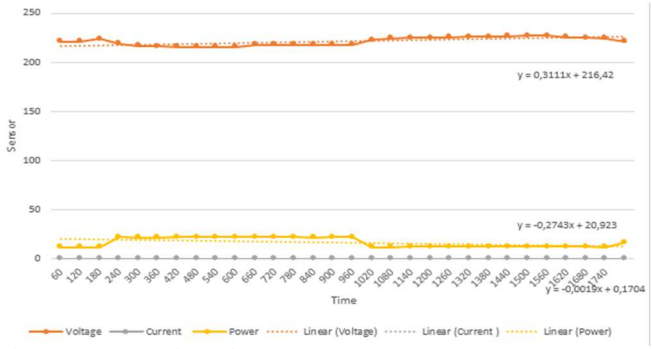

Fig. 12. Power Graph Used.

Based on existing data, the value of the PZEM004t sensor shows a change from the DC power grid which is converted into an AC power grid by the inverter so that the source voltage can be consumed by the load. The value of the load output voltage used is 220 volts AC. From these results also obtained an error value of ± 1% except for the power value of 9.29% due to limitations of measuring instruments and sensors. The factors that cause these differences are caused by differences in the specifications of the measuring instrument and the sensor used. The digital multimeter has an error value of 1.3% and a sensor of 0.5%.

Next, database capacity testing that has been configured is carried out by measuring the database capacity in 30 minute intervals can be seen Fig. 13. Based on the database data it takes about 16 seconds to start up the system. proven in the first 36 data, the results are not in accordance with the measured quantity.

	id	v_in	i_in	p_in	v_out	i_out	p_out	timestamp
Edit Copy Delete	1	12.18	-0.4	4	233.3	0.03	1	2021-03-10 12:44:05
Edit Copy Delete	2	12.17	-0.5	4	233.6	0.03	1	2021-03-10 12:44:05
Edit Copy Delete	3	12.18	-0.3	4	232.2	0.03	1.1	2021-03-10 12:44:08
Edit Copy Delete	4	12.17	0.1	4	232.1	0.03	1.1	2021-03-10 12:44:08
Edit Copy Delete	5	12.18	-0.6	4	232.9	0.03	1	2021-03-10 12:44:08
Edit Copy Delete	6	9.49	-0.4	4	232.2	0.03	1	2021-03-10 12:44:08
Edit Copy Delete	7	0.82	-0.4	0	232.7	0.03	1.1	2021-03-10 12:44:08
Edit Copy Delete	8	0.81	-0.6	0	232.8	0.03	1.1	2021-03-10 12:44:08
Edit Copy Delete	9	12.18	-0.4	4	232.9	0.03	1.1	2021-03-10 12:44:08
Edit Copy Delete	10	0	0	0	233	0.03	1.1	2021-03-10 12:44:08

Fig. 13. Display Database on Testing.

B. *Perfomance testing and analysis*

This test is done by measuring the response time value on the ESP32, virtual computer and the response time value for the entire system starting from the sensor reading until the value is stored in the database. The test lasts 5 minutes and uses an internet network. Testing can be seen Fig. 14.

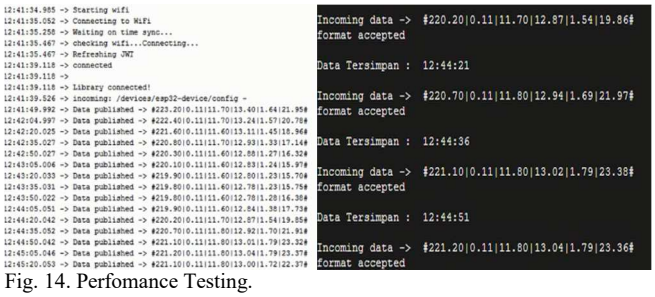

Fig. 14. Perfomance Testing.

Response time testing and analysis is carried out to determine how long it takes the sensor execution until the sensor data is published. Testing can be seen on graph in Fig. 15.

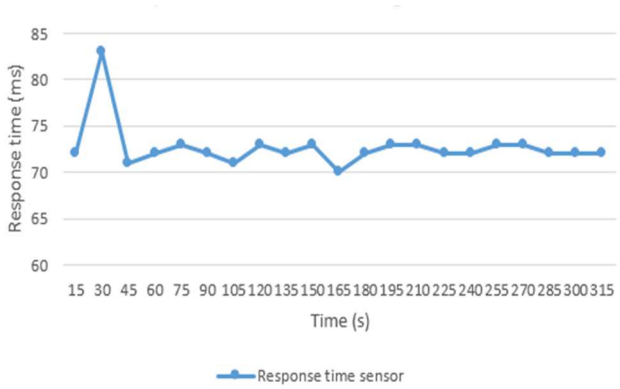

Fig. 15. Response Time Sensor.

Test results of the response time sensor with ESP32 are in the form of an average response time value of 72.67 ms. This response time value is obtained from the length of time the sensor reads the value until the value is published. For response time the broker is obtained by looking at the speed

of the server in receiving data. Testing can be seen on graph in Fig. 16.

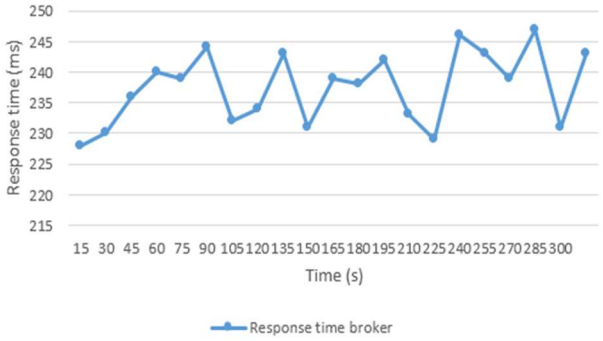

Fig. 16. Response Time Broker.

Results of the broker's response time test are in the form of an average response time value of 237.48 ms as well. This response time value is obtained from the length of the subscriber process by the server until the data is stored in the database and data transmission system is able to work in real time because the broker is able to receive data with a minimum response time.

Testing and analysis of the overall performance is carried out with calculates the difference in execution time when the sensor publishes data until the sensor data is stored in the database. This test is done to find out how fast data communication is using MQTT Broker. Testing can be seen on graph in Fig. 17.

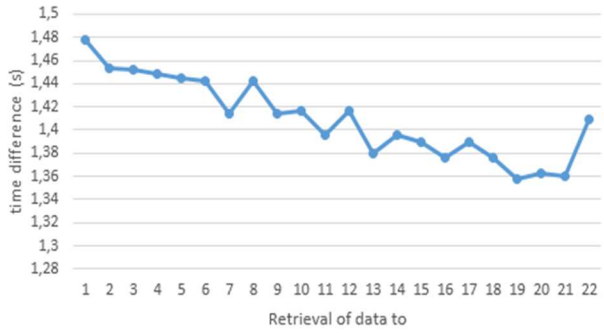

Fig. 17. Overall Response Time.

Overall performance difference in execution time when the sensor publishes data using the microcontroller n until the sensor data is stored in the database get an average value of 1.410 second. This response time value is obtained from the length of the publish process to the data storage process in the database. There is a difference between the response time on the ESP32 and the response time in the database because mikrocontroler takes time to process sensor data that will be published and subscriber on the MQTT communication protocol in the compute engine. Another factor is the speed of data transmission from the internet network operator used resulting in a delay of 1,410 s from the publication process until the data is stored in the database.

IV. Conclusion

Based on the results of tests and analyzes that have been carried out, the prototype of the Solar Power Plant performance monitoring system using the INA219 and PZEM004t sensors and the ESP32 microcontroller can measure and display the value of energy production and electricity load on solar power plants contained in a virtual server integrated database on Google Cloud Platform. Error value on the sensor is ±1%, except output power values due to limitations of sensors and measuring instruments. While the data transmission system is able to work in real time because the broker is able to receive data with a minimum response time and average response time total value is 1,410s. Consistency value of response time is influenced by the data transmission speed of the internet network used by the operator.

V. References

[1] M. F. Hakim, "Perancangan Rooftop Off Grid Solar Panel Pada Rumah," *Dinamika Dotcom,,* vol. 8, p. 1, 2017.

[2] R. R. A. Siregar, N. Wardana, and L. Luqman, "Sistem Monitoring Kinerja Panel Listrik Tenaga Surya Menggunakan Arduino Uno,," *Jetri: Jurnal Ilmiah,* vol. 14, pp. 81-100, 2017.

[3] P. Srivastava, M. Bajaj, and A. S. Rana, "IoT Based Controlling of Hybrid Energy System using ESP8266," *IEEMA Engineer Infinite Conference (eTechNxT),* p. 1–5., 2018.

[4] H. Satria and S. Syafii, "Sistem Monitoring Online dan Analisa Performansi PLTS Rooftop Terhubung ke Grid PLN," *Jurnal Rekayasa Elektrika,* vol. 14, p. 2, 2018.

[5] K. Friansa, I. N. Haq, B. M. Santi, D. Kurniadi, E. Leksono, and B. Yuliarto, "Development Of Battery Monitoring System In Smart Microgrid Based On Internet Of Things (IoT)," *Procedia Engineering,* vol. 170, p. 482–487, 2017.

[6] Z. B. Abilovani, W. Yahya, and F. A. Bakhtiar, "Implementasi Protokol MQTT Untuk Sistem Monitoring Perangkat IoT," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* p. 7521–7527, 2018..

[7] Z. B. Abilovani, W. Yahya, and F. A. Bakhtiar , "IoT Monitoring System Based on MQTT Publisher/Subscriber Protocol," *Iraqi Journal Of Computers, Communication, Control & Systems Engineering,,* vol. 20, p. 3, 2020.

[8] S. Krishnan and J. L. U. Gonzalez, Building Your Next Big Thing with Google Cloud Platform: A Guide For Developers and Enterprise Architects. Springer, 2015.

[9] M. R. Indrawan, "Pengaruh Intermittent Cost Pembangkit Listrik Tenaga Surya (PLTS) on Grid Photovoltaic Farm pada Sistem Kelistrikan menggunakan model IEEE 7 Bus," 2018.

[10] P. K. Sutawan, I. N. S. Kumara, and W. Ariastina,, "Simulasi Sistem Kontrol Operasi On Grid serta Islanding Pembangkit Listrik Tenaga Surya Di Jurusan Teknik Elektro Universitas Udayana," *Majalah Ilmiah Teknologi Elektro,,* vol. 14, no. 2, pp. 57-63, 2015.

[11] B. Setiawan, G. Hidayat, and A. Y. Candra, "Rancang Bangun Dc Pompa Submersible Sistem Baterai Fotovoltaik Digabungkan Dengan Panel Surya Tipe Polycrystalline Skala Laboratorium," *Prosiding Semnastek,* 2017.

[12] I. N. Haq, "“Sistem On-Line Condition Monitoring Pembangkit Listrik Tenaga Surya Berbasiskan Web Menggunakan Sensor Nirkabe," *Tesis. Institut Teknologi Bandung,* 2010.

[13] I. A. Setiawan, I. S. Kumara, and I. W. Sukerayasa, "Analisis Unjuk Kerja Pembangkit Listrik Tenaga Surya (PLTS) Satu MWP Terinterkoneksi Jaringan di Kayubihi, Bangli," *Majalah Ilmiah Teknologi Elektro,* vol. 13, p. 1, 2014.

[14] H. Nazif and M. I. Hamid, “Pemodelan Dan Simulasi PV-Inverter Terintegrasi Ke Grid Dengan Kontrol Arus Ramp

Comparison Of Current Control," *Jurnal Nasional Teknik Elektro,* vol. 4, pp. 129-139, 2015.

[15] Putra and I. K. Darminta, "Monitoring Penggunaan Daya Listrik Sebagai Implementasi Internet of Things Berbasis ESP8266,," vol. 3, no. 1, p. TE313–TE327, 2017.

[16] I. Espressif, ESP32 Datasheet IoT Based Microcontroller, 2017.

[17] W. Indrasari, R. Fahdiran et al, "Karakteristik Panel Surya Hybrid Berbasis Sensor INA219," *Prosiding Seminar Nasional Fisika (E-JOURNAL),* vol. . 8, p. SNF2019–PA, 2019.

[18] R. Pandu, W. Putra, M. Mukhsim, and F. Rofii, “Sistem Pemantauan dan Pengendalian Modul Automatic Transfer Switch (ATS) Melalui Android Berbasis Arduino,," vol. 5, pp. 43-54, 2016.