International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Analysis of Monitoring and Control System in IoT-Based Filling Station

Yuliadi Erdani*[1], M. Nursyam Rizal[2], Stya Ragen[3]

Submitted:14/03/2024 **Revised**: 29/04/2024 **Accepted**: 06/05/2024

Abstract: A filling machine or commonly called a filling station is a machine used to fill materials into bottles at industrial factories. This machine will automatically fill the material into the bottle and then proceed to the next process. At this time the filling machine is still operating with technology 3.0 which causes the machine to still be operated directly through the control panel so it is less effective. One of the technologies that can make these machines effective is the Internet, also known as the Internet of Things (IoT). Therefore, in this research, an IoT system design was created which aims to streamline the monitoring and control of workstations. Utilizing the MQTT and ESP32 communication protocols as controllers and 24VDC relays as a bridge for giving and reading signals from the PLC. The results of the research conducted by the system can already monitor and control the station through the Blynk application with an input control delay of 0.0129 seconds while the output reading is 0.012575 seconds and the hopper level notification alarm feature functions when the hopper level is in low level condition, and the system will send an alarm notification to the user.

Keywords: *PLC, IoT, MQTT, Blynk, ESP32*

1. Introduction

In the bottle filling process in the industrial world, an automation system is used which actually shortens the production process and increases company profits[1]. As technology develops, industrial automation systems require PLC technology to make the production process flexible[2]. One technology that can streamline the production process is the Internet or also known as the Internet of Things (IoT)[3]. The need to monitor and control the station remotely via the internet (IoT)[4], so it is necessary to develop a system that can retrieve data, monitor and control the processes running at the station[5]. With IoT technology, monitoring and control can be done remotely and must use the internet as an access medium[6]. This IoT system has several advantages including, not bound by space and time so that it can minimize the occurrence of production failures. This is an opportunity to develop and improve production quality[1]. In this research, the monitoring system applied is to monitor the mobile bottle filling process[7]. With the urgent need for station monitoring quickly and efficiently[2], it is necessary to create a mobile android-based monitoring system, especially with an interface design that can display station process simulations and provide indicators and control of the running plant.

[1] *Associate Professor, Department of Automation Manufacturing and Mechatronics Engineering, Politeknik Manufaktur Bandung – 40135, Indonesia, Email: yul_erdani@yahoo.com*(Corresponding Author)*
ORCID ID : 0009-0007-3362-1822
[2] *Lecturer, Department of Automation Manufacturing and Mechatronics Engineering, Politeknik Manufaktur Bandung – 40135, Indonesia, Email: nursyam@ae.polman-bandung.ac.id*
[3] *Student, Department of Automation Manufacturing and Mechatronics Engineering, Politeknik Manufaktur Bandung – 40135, Indonesia, Email: styaragen.nardisty@gmail.com*

2. Literature Review

The monitoring and control process was previously carried out using a computer[1]. The communication protocol used in the system is Modbus serial RS485 and has a response time test result of 2.6 seconds with a maximum cable length of 100 meters[8]. In addition to the RS485 serial Modbus protocol, the bluetooth communication protocol can be used to connect between the controller and smartphone, using a microcontroller as the controller. From this method, a connection with a maximum distance of 15 meters is obtained. Of course, this is a weakness in terms of range because it has a very limited range[9]. In other studies using NodeMCU as the controller which is used to connect between hardware and software[10]. As technology develops, the development of IoT systems applied to PLCs is carried out by researchers. The application of the system is carried out on a garbage transporter where monitoring and control are via email[11] with the help of NodeMCU as the controller. It can be concluded, from the discussion of previous research, that IoT systems cannot be applied directly to PLCs without the intermediary of microcontrollers and the internet. To be able to monitor and control via a smartphone, a communication system design and an interface connected to the internet are needed[1].

In designing a mobile monitoring and control system at the filling workstation[12] using the MQTT (Message Queuing Telemetry Transport) protocol[13]. The PLC connection is carried out so that it can be connected to the application[12] by using relays as output readers and input providers. In this study, the HC-SR04 ultrasonic sensor[14] was added as an additional feature which will monitor the material content in the station hopper. ESP32 as an intermediary for sending data between[15] ultrasonic sensors[14], PLC and applications. For the data storage system using the Firebase platform[16]. With this system, it is expected to produce an android-based application system to monitor and control the production process effectively and efficiently and make it easier for users to find out the time to refill the material in the hopper if the hopper fill indicator reaches 10%.

3. Proposed Work

3.1. System Development Method

Number equations consecutively with equation numbers in parentheses In designing the IoT-based Filling Station Control System, the author uses the Software Development Live Cycle (SDLC) waterfall model to develop the system. This method is carried out systematically and sequentially (cascade), to be able to carry out the next step, must wait for the previous step to be completed. This method is linear, with the initial stage being planning and the final stage being maintenance. The next step will not be carried out until the previous step is completed, and it is impossible to return or repeat the previous step[17].

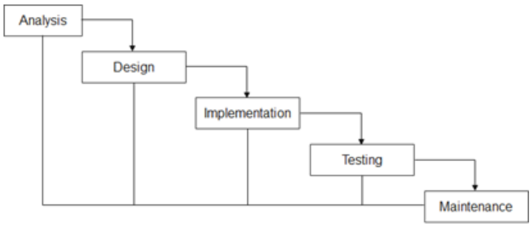


Fig 1. Stages of the Waterfall Method

a. Analysis, at this stage the researcher observes the filling station, including analyzing the needs and collecting research data.

b. Design, user interface design using Blynk software, communication system between ESP32 and PLC using the MQTT protocol. This stage includes designing programs and application systems, creating databases and designing ESP32 and PLC communications.

c. Implementation, applying the design that has been made on software and hardware.

d. Testing, this stage is carried out in order to find out whether the system is working properly or not.

e. Maintenance, this stage is maintenance when using features that need to be repaired. This step is taken to fix errors when the target or user accesses the application system.

3.2. Analysis of Design Requirements

The things that need to be done in analysing the design needs include, namely:

1. Observation at the station, carried out to find out the type of PLC used, the number of I / O addresses used. Based on the observation results, the FX3U PLC type is obtained with the number of I / O to be monitored 5 inputs and 8 outputs.
2. Finding suitable components and microcontrollers for data acquisition with a sufficient number of pinouts.
3. Creating a circuit using a simulator, so that a schematic or circuit is obtained that is definitely correct.
4. Applying the circuit that has been made in the simulator to the component.
5. Creating a microcontroller program on the arduino IDE for components that have been assembled and integrated into the Blynk application then transferred to the microcontroller, and creating an interface design on Blynk.
6. Apply the circuit that has been made to the PLC.
7. Conduct monitoring and control testing to ensure whether the data acquired by the microcontroller is delivered to the destination (Blynk and PLC) or not.

The following devices will be used in the system created:

a. Hardware

The hardware needed in the design of this system are:

1. ESP32: Is a microcontroller board that is used as a controller and connector between the sensor components used, this type of microcontroller board has been equipped with an ESP Wi-Fi chip that has been embedded in one board so that it will be easier to use for the development of Internet Of Things (IoT) based microcontrollers, has a lot of pinouts so that it is suitable for taking many outputs and inputs, besides that this version of the microcontroller board can be found on the market at a relatively affordable price. In this final project, ESP32 is used as a controller that communicates with PLC.
2. Relay Module: Is a component that is used as a link between the PLC and the microcontroller, where the relay will be controlled to provide input to the PLC and read the PLC output.
3. HC-SR04 Ultrasonic Sensor: Is a sensor tool that works based on the principle of sound wave reflection and is used to detect the presence of a particular object in front of it. This sensor is used to monitor the level in the hopper.
4. Mitsubishi FX3U PLC: This type of PLC has a serial port for communication and does not have an Ethernet module, used to control the process at the filling station which is also connected to the ESP32 microcontroller.
5. Smartphone: A device that has high capabilities like a computer, in this research the smartphone is used

as a device to monitor and control the station with the interface design that will be made.

b. Software

The software needed in designing this system is as follows

1. Arduino IDE: Is software used in writing program sketches into microcontrollers. In the process of writing the program using a combination of C / C ++ and Java languages.
2. Blynk: Is a platform used to design monitoring and control.
3. Firebase: Is a platform used for storing realtime data sent by microcontrollers.

3.3. System Overview

The overview of this study shows the relationship between one component and another. In the Fig 2, there are areas 3.0 and 4.0 where, area 3.0 shows that the system is still conventional and area 4.0 shows the development of the system by utilizing the use of the internet.

a. In area 3.0 it is illustrated that the PLC is the main control that has its own input and output. The input that is outputted is through the HMI and also the control panel to provide manual control input to the PLC. then the actuator becomes part of the PLC output itself, where the actuators used are DC motors and cylinders.

b. In area 4.0 depicted microcontroller as a control unit connected to the network. The microcontroller used is ESP32. The ultrasonic sensor is connected as a microcontroller output which is useful for detecting the material level in the hopper.

c. Mobile or smartphone connected to the network in area 4.0 is useful as a device to control and monitor station movements. As well as using Firebase as a platform for storing databases that will be accessed by smartphone devices to display data that has been acquired from the PLC.

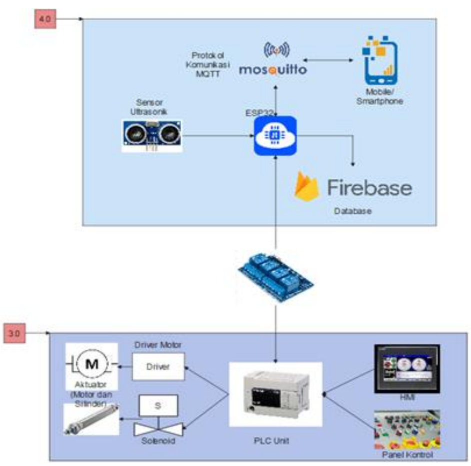


Fig 2. System Overview

3.4. System and Interface Design

In the system to be created, it has a sequence of working systems, when the system is activated, the ESP32 will capture signals from ultrasonic sensors and signals from the PLC. ESP32 will connect with Blynk, and then ESP32 will try to connect to the server on the MQTT server and firebase, after connecting, ESP32 will ask to subscribe to MQTT and then publish the signal on MQTT. The value that has been published in MQTT will be sent by ESP32 to the Blynk application, and then in the Blynk application, the output indicator will be active. When giving input, ESP32 will publish the signal to MQTT and send it to the relay connected to the PLC input.

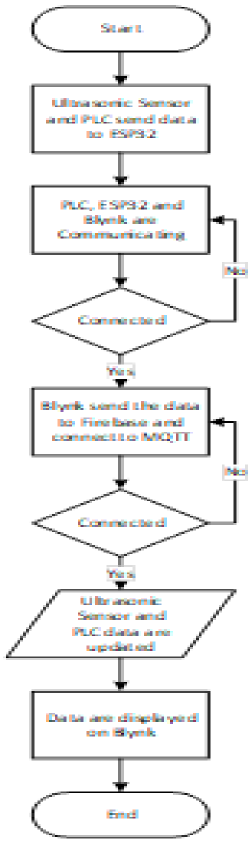


Fig 3. Flowchart System

a. Hardware Design

In this study, the data acquisition system from the PLC uses 7 channels of SPDT relays connected to the PLC output and 4 channels of SPDT relays connected to the PLC input and both are connected to ESP32. As well as ultrasonic sensors that are directly connected to ESP32.

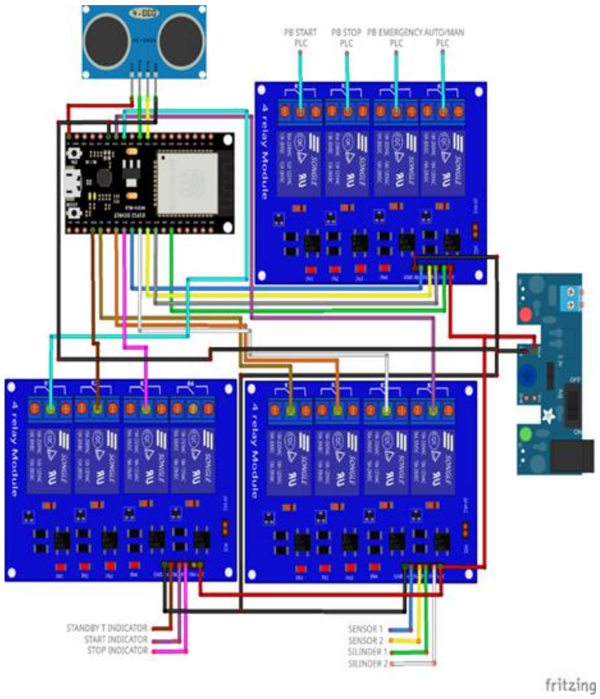


Fig 4. Schematic of ESP32 Circuit on PLC and Ultrasonic Sensor

Table 1. Cable Color of the Schematic

Pin ESP	Relay Module		Ultrasonic Sensor	Cable Color
	4 Ch	8 Ch		
GPIO 2	IN1	-	-	dark yellow
GPIO 4	IN2	-	-	Orange
GPIO 5	IN3	-	-	White
GPIO 12	IN4	-	-	Purple
GPIO14	IN5	-	-	Aqua
GPIO 15	IN6	-	-	Brown
GPIO 16	IN7	-	-	Pink
GPIO 17	-	IN1	-	Blue
GPIO 18	-	IN2	-	Yellow
GPIO 19	-	IN3	-	Gray
GPIO 21	-	IN4	-	Green
GPIO 25	-	-	ECHO	Yellow
GPIO 26	-	-	TRIG	Green
GND	GND	GND	-	Black
Vin	VCC	VCC	-	Red

b. Interface Design

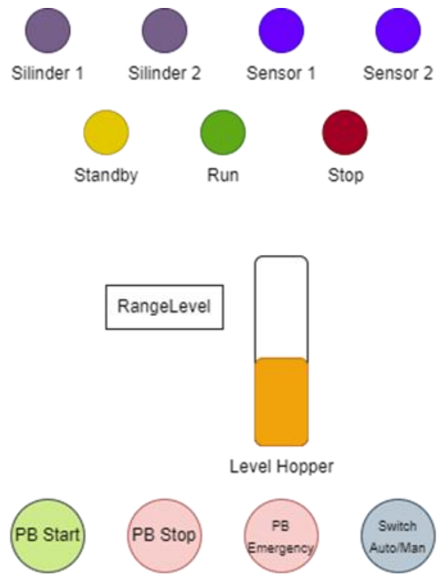


Fig 5. Interface Design on Blynk

4. Experimental Analysis

4.1. IoT System Testing Results

The test results on the IoT system include a system for monitoring stations and controlling stations, the following test results have been carried out.

4.1.1. Filling Station Control System Testing

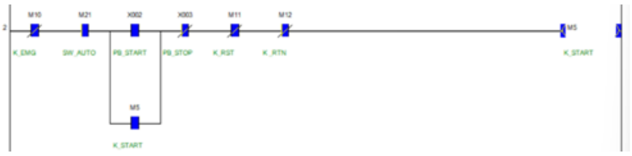
Control system testing aims to ensure the integration of the control system runs well. The control system is made in the form of push-button input from the Blynk application. The following table shows the results of testing the filling station control system on the Blynk application.

Table 2. Testing the Filling Station Control System on the Blynk Application

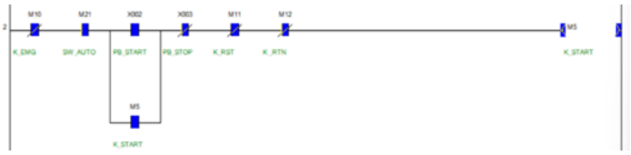
Input	Address	Testing Results
Switch Auto	X000	Works
Switch Manual	X001	Works
Start	X002	Works
Stop	X003	Works
Emergency	X004	Works

Fig 6. Control System on on Blynk Interface

Fig 6 shows the control buttons and switches to control the filling station. Station control can be done by pressing the buttons/switches available on the Blynk interface. The start button serves to activate the movement of the station according to the existing program. Stop button serves to stop the sequential movement of the station. The reset and return buttons function to return the station to its initial condition when a problem occurs in the station sequential, the reset and return buttons are used when the emergency button has been released. The emergency button serves to stop the sequential process when a problem occurs at the station.



(a)



(b)

(c)

(d)

Fig 7. Program PLC (a) Program PLC Switch A/M (b) Program PLC Tombol Start (c) Program PLC Tombol Stop (d) Program PLC Tombol Emergency

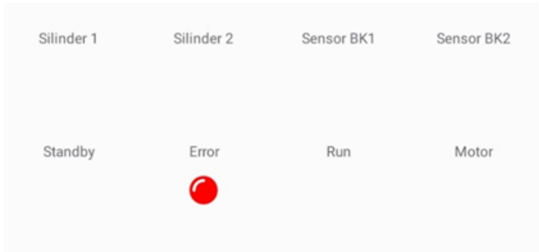
Fig 7 shows the results of testing the buttons controlled through the Blynk application by looking at the changes that occur in the PLC ladder program. Testing is done by pressing the button on the Blynk application interface. When the button on the Blynk application is pressed, the ESP32 will capture a binary signal worth 1 and publish on MQTT in the form of a string value that is "ON". After the value is published, the ESP32 will retrieve the value on MQTT and activate the relay which will then activate the intended address.

4.1.2. Testing the Filling Station Monitoring System

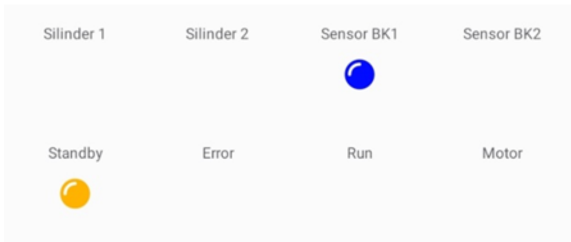
Monitoring system testing aims to ensure that the parameters that can be monitored through the application are in accordance with running properly. The following table shows the results of testing the filling station monitoring system in the Blynk application.

Table 3. Testing the Filling Station Control System on the Blynk Application

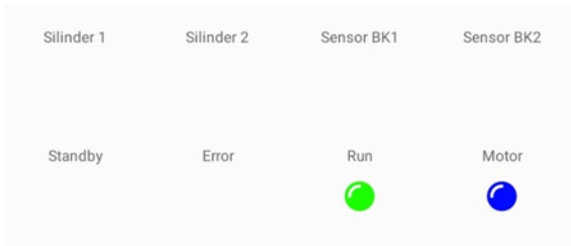
Indicator	Address	Testing Results
Red_Light	Y000	Works
Yellow_Light	Y001	Works
Green_Light	Y002	Works
Motor Conveyor	Y005	Works
Solenoid_A+	Y006	Works
Solenoid_B-	Y011	Works
Sensor_Bk1	X015	Works
Sensor_BK2	X016	Works



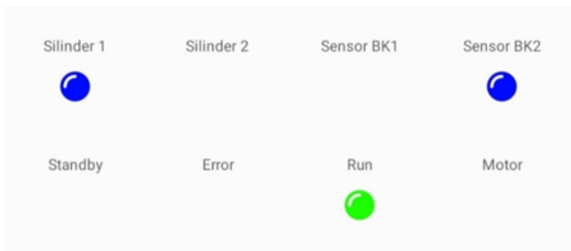
(a)



(b)



(c)



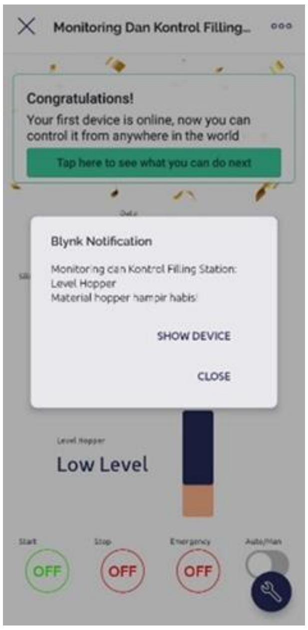
(d)

(e)

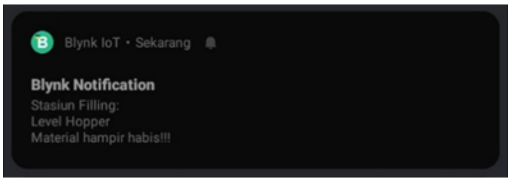
Fig 8. Status Monitoring of Sensor and Output Indicators (a) Error Status (b) Status when Workpiece is Detected (c) Status when Run and Motor is Active (d) Status when Bottle Filling (e) Status after Bottle Filling

4.2. Hopper Level Alarm System Testing Results on Blynk Application

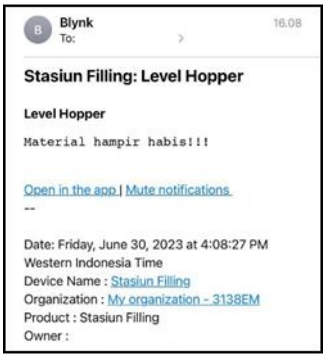
In Fig 9, we can see the notification sent by the Blynk application to the monitoring page, email and the application notification itself. The alarm is active when the ultrasonic sensor detects that the hopper level is running out, with the level distance category that has been determined in the program. If the range measured from the ultrasonic sensor to the surface of the hopper material is 0cm - 7cm, then the hopper level status is in a high level condition, if the ultrasonic sensor detects a distance in the range of 7cm - 15cm then the hopper level status is in a medium level condition, finally, if the ultrasonic sensor detects a distance from the range of 15cm - 20cm then the hopper level status is in a low-level condition. So with this distance range, the alarm notification will be active when the detected distance is 15cm - 20cm or in a low-level status.



(a)



(b)



(c)

Fig 8. (a) Alarm Notification on Page Open (b) Alarm Notification on Email (c) Alarm Notification on Notification Bar

4.3. Delay Testing Results

In previous research, using the Blynk application with the test results there is a delay of about 1-2 minutes when receiving commands from the application to the device, this is because receiving data on the device takes about 1242ms[18]. Testing the delay time when monitoring and control is done by reading the PLC output and giving commands to the PLC input. Based on Table 3, the delay calculation is by calculating the time from pressing the button in the Blynk application that sends the signal to MQTT until MQTT receives the signal and the relay connected to the PLC address is active. While in Table 4 the calculation of delay time is calculated from ESP sending a signal obtained from PLC until MQTT gets a signal from ESP and displays it in the Blynk application.

Table 4. Average input delay

Input	Rata-rata *delay* (s)
Start	0.00846
Stop	0.011035
Emergency	0.024267
Auto/Man	0.007838
Average delay (ms)	0.0129

The monitoring and control system that has been created is proven effective for monitoring and controlling stations remotely through applications connected to the internet.

There is no LAN cable port so that to send the PLC address signal requires additional components. ESP32 is added to be the controller in this system which is useful for sending and capturing signals from the PLC. This module has several advantages, besides its low cost, the ESP32 has 38 pinouts so that it can be connected to many inputs/outputs, and the ESP32 has a Wi-Fi module that is already installed on the ESP32 board. The use of the MQTT protocol in this system is very effective, the delay test results show that the signal transmission distance between the application and the relay is very small so real-time monitoring and control can be done well.

5. Conclusion

The IoT-based control and monitoring system using ESP32 as a controller and MQTT as a communication protocol between ESP and PLC can be declared successful, based on the test results above.

PLC control and monitoring through the Blynk application and the MQTT protocol obtained an average delay time from input control of 0.0129 seconds while the output reading was 0.012575 seconds. The magnitude of the delay value is influenced by the internet connection used.

Hopper level alarm notification functions properly, where when the hopper level is in a low-level condition, the system will send an alarm notification to the user.

References

[1] S. Wahyu Jadmiko, D. Nono Suharno, and S. Kurniawan Nugraha, “Aplikasi Internet of Things (IoT) untuk Pemantauan Simulator Plant Berbasis PLC-Web Server,” in Seminar Nasional Teknologi dan Riset Terapan (SEMNASTERA), 2020, pp. 28–33.

[2] G. A. Laksamana, P. Santoso, and F. Pasila, “Aplikasi untuk Memonitor PLC Pada Mesin Filling dan Capping,” Jurnal Teknik Elektro, vol. 10, no. 2, pp. 48–53, 2017.

[3] A. R. AL TAHTAWI, T. D. HENDRAWATI, A. ABDURRAHIM, and E. ANDIKA, “Perancangan dan Analisis Kinerja Sistem Kontrol dan Penjadwalan Lampu Berbasis IoT,” ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika, vol. 7, no. 3, p. 533, Sep. 2019, doi: 10.26760/elkomika.v7i3.533.

[4] A. Syarif, Harianto, and I. Puspasari, “Rancang Bangun Automatic Liquid Filling Machine Berbasis IoT (Internet of Things),” Journal of Technology and Informatics (JoTI), vol. 2, no. 2, pp. 72–82, Apr. 2021, doi: 10.37802/joti.v3i1.178.

[5] A. Binazar, H. Rachmat, and D. S. E. A. Atmaja, “PERANCANGAN SUPERVISORY CONTROL AND DATA ACQUISITION (SCADA) PADA PROSES PENGEPAKAN TEH DILENGKAPI DENGAN PELAPORAN DATA MENGGUNAKAN GENERIC DATA GRID,” e-Proceeding of Engineering, vol. 2, no. 1, pp. 817–826, 2015.

[6] A. M. Daffa, Y. Saragih, and H. R. Satrio, “UTILIZATION OF TELEMETRY SYSTEMS IN QUAIL FARMING USING THE IOT CONCEPT,” J Teknol, vol. 15, no. 2, 2023, doi: 10.24853/jurtek.15.2.217-224.

[7] H. Haryanto, E. Permata, and N. R. U. Nainggolan, “Sistem Monitoring Proses Produksi pada Mesin Bardi di PT. Tirta Investama (Danone Aqua) Sukabumi Berbasis Web,” Jurnal Ilmiah SETRUM, vol. 3, no. 1, 2014.

[8] T. Pangaribowo et al., “SISTEM MONITORING SUHU MELALUI SISTEM KOMUNIKASI PROGRAMMABLE LOGIC CONTROLLER TO PERSONAL COMPUTER,” Jurnal Teknologi Elektro, vol. 7, no. 3, pp. 175–180, Sep. 2016.

[9] A. Susanto and I. Darisman Jauhari, “RANCANG BANGUN APLIKASI ANDROID UNTUK KONTROL LAMPU GEDUNG MENGGUNAKAN MEDIA BLUETOOTH BERBASIS ARDUINO UNO,” Universitas Muhammadiyah Tangerang, vol. 8, no. 1, 2019.

[10] M. A. Ashari and L. Lidyawati, “IOT BERBASIS SISTEM SMART HOME MENGGUNAKAN NODEMCU V3,” Jurnal Kajian Teknik Elektro, vol. 3, no. 2, pp. 138–149, 2019.

[11] M. Choir Afrian, M. T. Asron, and R. Wicaksono, “PROTOTIPE PENGANGKUT SAMPAH OTOMATIS PADA PINTU ALI DENGAN SISTEM INFORMASI MENGGUNAKAN NODE MCU ESP8266 BERBASIS PLC (PROGRAMMABLE LOGIC CONTROLLER),” Jurnal Autocracy, vol. 5, no. 2, pp. 95–105, 2018, doi: 10.21009/autocracy.05.2.5.

[12] M. Dimas Nauval, H. Rachmat, and M. D. Prasetio, “PERANCANGAN SISTEM MONITORING DAN CONTROLLING MENGGUNAKAN SMARTPHONE PADA STASIUN KERJA DELAY DAN PICK AND PLACE SIMULATOR BOTTLING PLANT DENGAN METODE WATERFALL,” in e-Proceeding of Engineering, 2021, pp. 8198–8207.

[13] R. A. Atmoko, R. Riantini, and M. K. Hasin, “IoT real time data acquisition using MQTT protocol,” in Journal of Physics: Conference Series, Institute of Physics Publishing, Jun. 2017. doi: 10.1088/1742-6596/853/1/012003.

[14] F.- Puspasari, I.- Fahrurrozi, T. P. Satya, G.-

Setyawan, M. R. Al Fauzan, and E. M. D. Admoko, "Sensor Ultrasonik HCSR04 Berbasis Arduino Due Untuk Sistem Monitoring Ketinggian," Jurnal Fisika dan Aplikasinya, vol. 15, no. 2, p. 36, Jun. 2019, doi: 10.12962/j24604682.v15i2.4393.

[15] A. Ridho'i, K. Setyadjit, and B. Era Yordhan, "Sistem Monitoring Suhu Dan Kelembaban Pada Budidaya Jamur Tiram Menggunakan ESP32," Jurnal FORTECH, vol. 4, no. 1, pp. 20–26, Jan. 2023, doi: 10.56795/fortech.v4i1.4103.

[16] H. P. Ramadhan, C. Kartiko, and A. Prasetiadi, "Monitoring Kualitas Air Tambak Udang Menggunakan NodeMCU, Firebase, dan Flutter," Jurnal Teknik Informatika dan Sistem Informasi, vol. 6, no. 1, Apr. 2020, doi: 10.28932/jutisi.v6i1.2365.

[17] A. A. Wahid, "Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi," Jurnal Ilmu-ilmu Informatika dan Manajemen STMIK, 2020, [Online]. Available: https://www.researchgate.net/publication/346397070

[18] W. A. Prayitno, A. Muttaqin, and D. Syauqy, "Sistem Monitoring Suhu, Kelembaban, dan Pengendali Penyiraman Tanaman Hidroponik menggunakan Blynk Android," Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, vol. 1, no. 4, pp. 292–297, 2017.