International Journal of Academic Information Systems Research (IJAISR)
ISSN: 2643-9026
Vol. 5 Issue 12, December - 2021, Pages:13-20

Design of Human Machine Interface (HMI) IoT-Based Coffee Bean Packing System

Sumardi*, Prasetyo, Sudjadi

Diponegoro University, Department of Electrical Engineering
Semarang, Indonesia
*corresponding email: sumardi.undip@gmail.com

Abstract: *The Internet of Things, also known as the abbreviation IoT, is a concept that aims to expand the benefits of continuously connected internet connectivity. As for capabilities such as data sharing, remote control, and so on, including objects in the real world. One of the uses of IoT in the real world is the application of IoT in industrial automation processes. The production costs needed to procure industrial automation equipment from a company may not seem big to an industry, but for MSMEs these tools require large costs. The purpose of designing an IoT-based seed packing system aims to reduce the output costs incurred by MSMEs. This can be achieved by creating your own IoT system, one of which is using a website. In this final project, a Supervisory Control and Data Acquisition (SCADA) system and a Human Machine Interface (HMI) system for packing IoT-based coffee beans are designed using a website. It is hoped that from this design, the system can control and monitor properly using the website platform so that it can be further developed and can be applied directly by the MSME industry*

Keywords— Internet of Things, SCADA, HMI, Website.

1. INTRODUCTION

Humans are constantly trying to improve the quality and effectiveness in their lives. Likewise with supporting technology that continues to develop and is increasingly modern. IoT is opening tremendous opportunities for a large number of novel applications that promise to improve the quality of our lives [1]. The International Telecommunication Union (ITU) defines the Internet of Things (IoT) as a global infrastructure for information technology circles that allows advanced services with interconnection (physical and virtual) of things based on the existence and development of technology, and can be operated in information technology and communication [2]. IoT can be used for remote control of electronic devices and supervision of the production process. This can be utilized and used as well as possible to optimize the production process in an industry, especially the MSME industry.

One of the MSMEs that has a very important role in the Indonesian economy is culinary MSMEs [3]. Various types of food are traded in the community such as snacks (snacks), coffee grounds, kitchen ingredients and so on. Many MSMEs in the production process such as in the food packaging process are still manually [4]. Manually packing food is considered to be able to reduce production costs because it uses human labor, but if production is needed in large quantities it will make the packaging process very inefficient. Automatic packing tools sold in the market are also considered very expensive, so MSMEs still prefer manual production.

An example of the use of automation processes in industry is the Supervisory Control and Data Acquisition (SCADA) system technology. SCADA is not a unified control system, but rather focuses on the level of supervision. A SCADA system is a centrally controlled master system that commands terminal RTUs. Master terminal units (MTUs) are higher level units, including supporting applications, human machine interfaces (HMIs), data storage, and acquisition systems. PLCs are used as control sensory devices and RTUs [5]. Basic service of SCADA system include telemetering, remote signaling, telecontrol and remote regulating[6]. This SCADA system can be used to automate a production process so that it will increase the amount of production. This SCADA system can be designed according to the desired production process, making it easier for an industry to make production equipment specifically for its industrial processes. This is what underlies the research to design a coffee bean packaging tool automatically at a cost that can still be reached by MSMEs.

Human Machine Interface (HMI) is a system that connects humans and machine technology. HMI can be both controller and visualization status either manually or through computer visualization that is real time. HMI systems usually work online and in real time by reading data sent via the I/O port used by the system controller. In general, HMI has the following functions: settings, monitoring, take action, data logging and storage, alarm history and summary and trending [7].

This study aims to provide an alternative in the production process, especially in food SMEs, namely the use of HMI in the design of an IoT-based automatic coffee bean packaging tool. HMI is used to simplify the production process and monitoring of coffee bean packaging equipment. The HMI that will be designed is based on the Out-seal Mega V1.1 PLC and Arduino Uno which will later be integrated with sensors and actuators on coffee bean packaging equipment.

2. METHOD

The design of this IoT-based coffee bean packing system HMI uses a reference to the previous final project research, namely the design of a web-based human machine interface

(HMI) as a monitoring and controlling medium for smart home prototypes [8] with website development and an easier-to-operate HMI display. There are another reference used that is Web Technologies in the HMI industry design industrial HMI systems that can work well and efficiently and can produce solutions for commercial HMI designs. This research uses web and cloud technology for HMI creation and uses javascript, C# and Microsoft Azure frameworks for HMI visuals[9] The HMI design method for coffee bean packing equipment is divided into 2 parts, namely the design of the HMI workflow, and the design of the HMI software.

2.1 HMI Workflow Design

The design of a web-based control and monitoring system on the coffee bean packaging prototype allows the user to control and monitor the coffee bean packaging system by utilizing a web interface that can be accessed anytime and anywhere that provides an internet network. The whole designed system is shown in Figure 1.

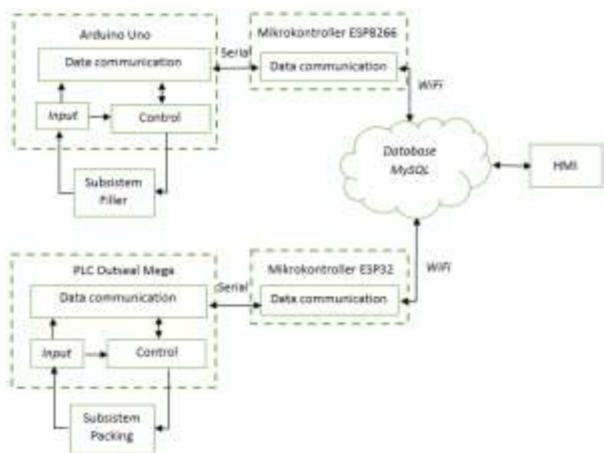

Fig 1. Overall diagram of coffee bean packing system with web-based HMI

Figure 1 is the overall design of a coffee bean packing system that can be controlled online via a web-based HMI using an internet connection between the ESP32 microcontroller and the database. Plant data received by ESP32 will be sent to the cloud database via an internet connection connected by the wifi module on the ESP32. The data that is successfully entered into the cloud database, is then called using jQuery and PHP lists to be displayed on the HMI. Plant control from HMI is done by giving on command, off command, selecting several available modes and setting PID parameters which are entered from the web page, then entered into the database which will then be read by ESP32. The data that is read is used as system input.

2.2 HMI Software Design using the website

1. HMI Software Design

In this system there are 2 modes for controlling coffee bean packing production. The modes provided include automatic, semi-automatic, and manual. The two modes provided for controlling and monitoring via HMI have slight differences. In automatic mode, the coffee bean packing system will run according to the program and setpoints that have been determined according to the user's production standards so that the HMI page will only display the state of the equipment as well as monitoring the weight and quantity produced. In manual mode, the tool user can select which components are turned on such as power, conveyor, fan, and set the setpoint for the PID parameter, weight and quantity of product. This mode can be used when the user only wants to use certain components. Login as admin has algorithm and flow chart as shown in Figure 2.

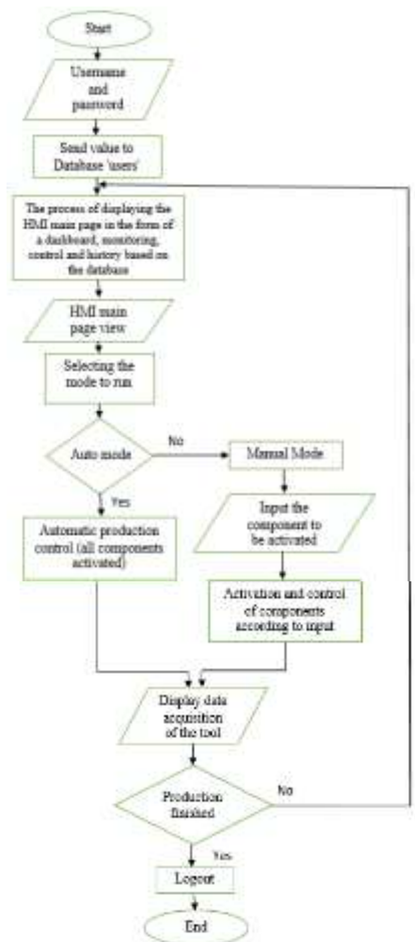

Fig 2. HMI workflow flowchart

The tool system can communicate with databases and websites through an Application Programming Interface (API). It is this API that functions for data retrieval and reception from the database as shown in Figure 3.

International Journal of Academic Engineering Research (IJAER)
ISSN: 2643-9085
Vol. 3 Issue 7, July – 2019, Pages: 1-1

Fig 3. API flowchart for communication system

2. HMI Display Design

The web-based HMI display is designed using Visual Studio Code software. Display design includes:

a. HMI Main Page Display

The main page on the HMI which contains information on the status of the house conditions such as time, the status of components on the tool, as well as the status of the alarm, namely the security alarm. This command will relate to the home security system. The menubar on the left is used to move to another page. Figure 4 shows the HMI main page view.

Fig 4. HMI main page isplay

b. Monitoring Page View

The monitoring page displays data which is a plant data acquisition obtained by the database. The data displayed by HMI is in the form of packaging weight measured using a load cell sensor and the number of packages produced. Figure 5 shows the monitoring page display

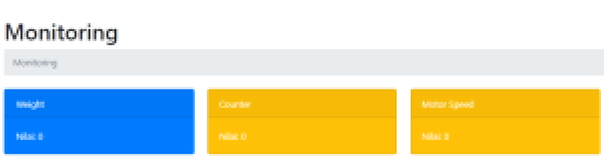

Fig 5. Monitoring Page Display

c. Control Page View

The control page is used to set or control the components used. Users can operate this page to select the mode they want to use, namely automatic mode or manual mode. There are three choices of buttons that can be pressed to turn on the filler, conveyor and fan on the tool. Each mode option on this page will send the set point value to the database and then forward it to the microcontroller. Figure 6 shows the control page display.

Fig 6. Control page display

d. Data Record Page Display

• Table

The table on the data record page displays the acquisition data read by the microcontroller and sent to the database. The data displayed is a history of production data consisting of name, time, PID value, weight, amount and description. The data in the table can be displayed sequentially from the top or from the bottom. There is a search feature to find data as desired. Figure 7 shows the table on the data process page.

Fig 7. Table page display

• Graphics

The graph on the data process page that displays the acquisition data read by the microcontroller and sent to the database. The data displayed is the number of packages produced each time. The graph will display the last 100 data sent by the microcontroller to the database. Figure 8 shows a graph on the process data page.

International Journal of Academic Engineering Research (IJAER)
ISSN: 2643-9085
Vol. 3 Issue 7, July – 2019, Pages: 1-1

Fig 8. Graphics page display

2.3 Database System Design

Database is a collection of interrelated files indicated by the key of each existing file. One database shows a collection of data that is used in a single scope of information[10]. The design of the cloud database system consists of designing tables and designing PHP programs. The database system built is used as a bridge to connect the microcontroller with the HMI web display. Database Management System (DBMS) used is MySQL Database[11]. In this study, web hosting services and domain purchases were used. The leased domain is .site with the address hmitasakhir.site.

The design of the database system on this system uses one database with six tables in it. six tables are used for data exchange to the microcontroller and one table for the login system. Data Flow Diagram was used to to create an information system model in the form of a network of processes that are interconnected with each other by data flows[12]. Figure 9 shows the context diagram of the data flow in the database system

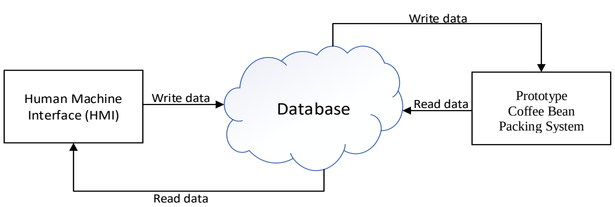

Fig 9. Data flow context diagram in database system

Based on Figure 9 the database system designed consists of HMI blocks, databases, and a prototype of the coffee bean packaging system. It can be seen that there are 4 data traffic lanes to connect the plant with HMI through the database.

First, data from the plant is sent to the database and then retrieved to be displayed in HMI as a monitoring system. Furthermore, the HMI can be controlled by providing input control parameters for the plant or commands for certain modes. The data entered from the HMI will fill the database and will then be read by the microcontroller. The data that is read is used as input for controlling the coffee bean packaging prototype

2.4 Communication System Design

PHP is a very popular scripting language commonly used for web applications. PHP is perfect for fast growing applications both large and small[13]. The process of reading and writing data by the microcontroller from or to the database is carried out using a PHP script as a communication link between the two. In this study, two PHP scripts were created to assist the communication between the two, namely one PHP script for the process of writing microcontroller data to the database and the process of reading data from the database by the microcontroller.

a. POST Method

Write data PHP Script helps the process of entering plant data sent by the microcontroller to the database. The data sent and entered into the acquisition table is used for the acquisition process so that remote monitoring can be carried out.

b. GET Method

Read data PHP Script helps the process of reading control parameter data from the cloud database input from the HMI page. The data is read by the microcontroller through the web display generated by the parsing.php PHP code. The URL that can be used to check the display of data to be read by the microcontroller is kemikroivan.php. The read data is then parsed to take only the control parameter values and used as system control input.

c. Parse data

After getting the reading data from the database, then parsing the data is carried out to retrieve and separate the data needed as system input. The reading results obtained data that undergo the parsing process. Data that has been parsed or parsed has a string data type. Where means that these data types must be converted into integers. After going through a series of programs to change the data type, the data read from the database has been changed to data with an integer data type. The data from the database can already be run as a command in the coffee bean packing system.

2.5 ESP32 and ESP8266 Hardware Design

The microcontroller used in this study is the center of the system control. There are a total of 3 types of microcontrollers, namely Arduino uno, PLC Outseal and ESP module. There are 2 types of ESP modules that used that is ESP8266 to Arduino uno for filler sub system controller, and ESP32 to PLC Outseal for packing sub system controller for 2 way communication control between HMI and the whole system. The ESP32 and ESP8266 microcontroller were chosen as the communication control center for the system because it has good performance to run the control and data acquisition processes simultaneously.

International Journal of Academic Engineering Research (IJAER)
ISSN: 2643-9085
Vol. 3 Issue 7, July – 2019, Pages: 1-1

The wiring for connecting PLC Out-seal to ESP32 and Arduino Uno to ESP8266 is using Rx and Tx pin from microcontroller that can be connected to Tx and Rx from IoT modules[14][15]. The design block diagram is shown in Figure 10 and figure 11.

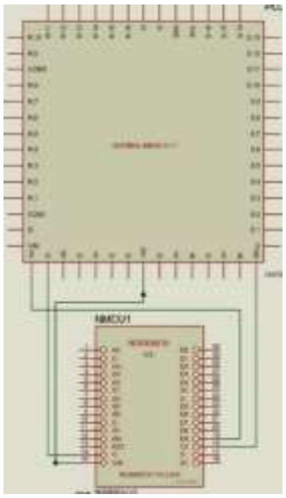

Fig 10. ESP32 hardware design block diagram

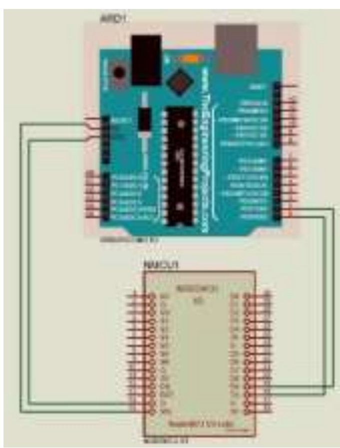

Fig 11. ESP8266 hardware design block diagram

To support data exchange communication, ESP32 has a fairly complete interface, namely UART, I2C, SPI, WiFi and Bluetooth is available. The design of the coffee bean packing control and monitoring system requires data exchange communication that will utilize the ESP32 interface facility, namely WiFi.

3. RESULT AND DISCUSSION

This section is divided into 4 parts, namely communication system testing, control testing, monitoring testing, and Data retention testing.

3.1 Communication System Test

1. Testing of sending data to the database

Testing of sending data to the database is done by sending input data on the number of packages and the results of the weight sensor readings contained in the coffee bean packing system plant. The data sent from the microcontroller can then be seen in the monitorings table display as shown in Figure 12

id	id_alat	counter	berat	kecepatan_motor	kp	ki	kd	created_at	updated_at
214	7	2	16.75	NULL	NULL	NULL	NULL	2021-11-05 09:32:16	2021-11-05 09:32:16
215	7	2	16.75	NULL	NULL	NULL	NULL	2021-11-05 09:32:28	2021-11-05 09:32:28
216	7	2	16.75	NULL	NULL	NULL	NULL	2021-11-05 09:32:33	2021-11-05 09:32:33
217	7	2	16.75	NULL	NULL	NULL	NULL	2021-11-05 09:32:51	2021-11-05 09:32:51
218	7	2	16.75	NULL	NULL	NULL	NULL	2021-11-05 09:32:56	2021-11-05 09:32:56

Fig 12. Incoming data from the microcontroller to the monitorings database table

From Figure 12 it can be seen that there are as many as eight data sent within minutes, seconds or seconds. Meanwhile, by calculation, the data that can be sent within 4 minutes is as much as:

• Total delivery time = 40 seconds
• Total data = 5
• Average data transmission time 40/5 = 8 seconds

In the program list, the data transmission interval has been determined to be 0.5 seconds. While the calculation results obtained the average time of sending data to the database with the reconnecting process that occurs is 0.816 seconds. The length of the reconnecting process is also determined by the strength of the internet network being used and the condition of data traffic on the web server. The reconnecting process is used to reconnect to the web server to avoid loss of connection that can break the sending process to the database.

2. Testing the reading of data from the database

Testing of reading data from the database is done by sending eight data from the database to be read by the microcontroller and then used as system input. The response to receiving data can be seen in Figure 13.

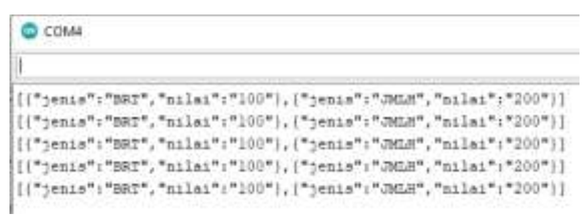

Fig 13. Response to Receipt of Data by the Microcontroller

Figure 13 shows the response to reading data from the database was successful, it was indicated by the entry of five data that was read on the serial monitor within 26 seconds. The data that is read can be checked in the "settings" and "controls" tables, which are tables for storing input data from the HMI. By calculation, the data that can be sent within 26 seconds is as much as:

- Total delivery time = 26 seconds
- Total data = 5
- Average data transmission time = 26/5 = 5.2 seconds.

Figure 13 shows the difference in the time of receiving data from the database with a time interval that is not much different. All five data are received by the ESP32 in about 6 to 7 seconds. In the program list, the data transmission interval has been determined to be 2 seconds. While the calculation results obtained the average time of sending data to the database with the reconnecting process that occurs is 5.2 seconds. This happens because there is a queue in the get and post database communication process or because there is internet interference. The length of the reconnecting process is also determined by the strength of the internet network being used and the condition of data traffic on the web server. The reconnecting process is used to reconnect back to the web server to avoid loss of connection that can break the sending process to the database

From the overall test of the communication system, it can be concluded that the communication system can run well. Receiving and sending can be carried out smoothly so that system operation requirements can be met. The success of two-way communication using the internet is also determined by the strength of the internet network and servers that are not busy. The reconnecting process causes the sending and reading process to be delayed. This is due to the use of delay in the program

3.2 Control Test

The test on this page is to test the available option buttons for plant settings, modes and components in the coffee bean packing system. The buttons consist of the POWER button to turn the plant on or off, the AUTO, and MANUAL buttons for setting the production mode, the CONVEYOR, FILLER, SEALER, and FAN buttons to turn on or off the components manually and the SETTINGS button for setting the values of the KP, KI, and KD controllers. PID for conveyor motor control. The actions to be performed by each button are presented in Table 1.

Table 1: Button action on HMI

	Button	Action
1.	POWER	Provides a choice of modes for the process of turning on or off the tool, namely automatic mode or manual mode
2.	AUTO	Entering data in the form of the value "1" for all components on the tool from the web to the controls table in the database with the update method on the row with id
3.	MANUAL	Provides a choice of buttons to turn on or off the conveyor, sealer and fan components
4.	KONVEYOR	Entering data in the form of the value "1" from the web to the controls table on the conveyor component in the database with the update method on the row with id
5.	SEALER	Entering data in the form of the value "1" from the web to the controls table on the sealer component in the database with the update method
6.	FAN	Entering data in the form of the value "1" from the web to the controls table on the fan component in the database with the update method
7.	SETTINGS	Entering data in the form of parameter values KP, KI, KD, weight and number of packages according to the numbers entered from the web to the settings table in the database with the update method

Controlling Delay Time Testing is carried out because it is controlled wirelessly using IoT to find out how much delay time is needed for the system to execute commands. The test results can be seen in Table 2

Table 2: Data of component control delay time.

Button	Attempt 1	Attempt 2	Attemp 3
Power	340 ms	300 ms	332 ms
Konveyor	434 ms	389 ms	409 ms
Sealer	392 ms	321 ms	265 ms
Servo	365 ms	342 ms	358 ms
Stop	392 ms	381 ms	398 ms

In Table 2, it was tested for 5 buttons 3 times. It can be seen that the delay time of components in executing time is in the range of 3 to 4 because ESP32 only requires state data whose value is 1 or 0. In contrast to input which requires a value that can reach hundreds so that it can increase the processing time of sending data. From the table, the average time is 361 ms, with the fastest time being 265 ms and the longest time being 434 ms.

3.3 Monitoring System Test

The monitoring page test is carried out by reading the acquisition data sent by the microcontroller and then taken to be displayed in an update on the HMI monitoring page. The

International Journal of Academic Engineering Research (IJAER)
ISSN: 2643-9085
Vol. 3 Issue 7, July – 2019, Pages: 1-1

monitoring table as a table for storing weight data and data on the number of packages that will be displayed on the monitoring page can be seen in Figure 14.

id	id_alat	counter	berat	kecepatan_motor	kp	ki	kd	created_at	updated_at
1	7	5	388		NULL	NULL	NULL	2021-11-03 08:38:13	2021-11-03 08:38:13
177	7	5	388		NULL	NULL	NULL	2021-11-03 13:58:37	2021-11-03 13:58:37
166	7	25	120		NULL	NULL	NULL	2021-11-03 13:59:38	2021-11-03 13:59:38
167	7	25	120		NULL	NULL	NULL	2021-11-03 13:59:[illegible]	2021-11-03 13:59:[illegible]
168	7	25	120		NULL	NULL	NULL	2021-11-03 13:59:41	2021-11-03 13:59:41
169	7	25	120		NULL	NULL	NULL	2021-11-03 13:59:46	2021-11-03 13:59:46
170	7	25	120		NULL	NULL	NULL	2021-11-03 13:59:51	2021-11-03 13:59:51
171	7	25	120		NULL	NULL	NULL	2021-11-03 13:59:58	2021-11-03 13:59:58
172	7	25	120		NULL	NULL	NULL	2021-11-03 14:00:04	2021-11-03 14:00:04
173	7	25	120		NULL	NULL	NULL	2021-11-03 14:00:09	2021-11-03 14:00:09
174	7	25	120		NULL	NULL	NULL	2021-11-03 14:00:16	2021-11-03 14:00:16

Fig. 14 Tabel monitorings

Figure 14 shows the data on the weight and number of packages that have been successfully received by the database, which are then taken to be displayed on the HMI monitoring page as shown in Figure 15.

Figure 15 Display of weight and number of packages with data from monitorings table

Figure 15 shows that the monitoring performance of the weight and number of packages produced by the coffee bean packing system is in accordance with the data in the table. The data listed on the monitoring page is in accordance with the updated weight data and the last number of packages obtained so that it is accurate enough to determine the current working condition of the plant, so that the weight appearance test is said to be successful. The length of time HMI responds to a data update is obtained from the calculation of the time when the data is successfully input into the database, with the data then displayed on the screen.

3.4 Data Retention Test

Testing of the data acquisition display in the form of graphs and tables is carried out by taking data from the monitoring table and then displaying it in the form of tables and graphs as shown in Figure 16.

id	id_alat	counter	berat	kecepatan_motor	kp	ki	kd	created_at	updated_at
1	7	5	388		NULL	NULL	NULL	2021-11-03 08:38:13	2021-11-03 08:38:13
177	7	5	388		NULL	NULL	NULL	2021-11-03 13:58:37	2021-11-03 13:58:37
166	7	25	120		NULL	NULL	NULL	2021-11-03 13:59:38	2021-11-03 13:59:38
167	7	25	120		NULL	NULL	NULL	2021-11-03 13:59:[illegible]	2021-11-03 13:59:[illegible]
168	7	25	120		NULL	NULL	NULL	2021-11-03 13:59:41	2021-11-03 13:59:41
169	7	25	120		NULL	NULL	NULL	2021-11-03 13:59:46	2021-11-03 13:59:46
170	7	25	120		NULL	NULL	NULL	2021-11-03 13:59:51	2021-11-03 13:59:51
171	7	25	120		NULL	NULL	NULL	2021-11-03 13:59:58	2021-11-03 13:59:58
172	7	25	120		NULL	NULL	NULL	2021-11-03 14:00:04	2021-11-03 14:00:04
173	7	25	120		NULL	NULL	NULL	2021-11-03 14:00:09	2021-11-03 14:00:09

Fig. 16 Data monitorings table

Figure 16 shows the last 10 data sent by the microcontroller. These data will be displayed on the Data Record page of the HMI. The data will be displayed in the form of tables and graphs. Data in tabular form can be seen in Figure 17.

Fig. 17 Data viewer in table form

Figure 17 shows the data presented in tabular form for the modes. Tables can be displayed properly, and can be sorted from the top or bottom. There is a search feature to find data at certain desired times. The table displays data from the database completely and well so that the test can be said to be successful.

The viewer of plant data acquisition in graphical form is the same. The graph displays the data on the weight and number of packages acquired against the time of receipt of data by the database, as shown in Figure 18.

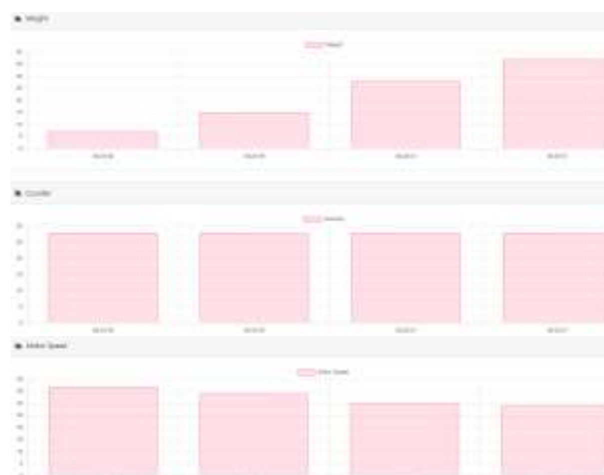

Figure 18 Data viewer in graphical form

Figure 18 shows the data on the weight and number of packages presented from the monitoring table. The graph shown already provides sufficient information to determine the state of the weight and number of packages in the coffee bean packaging system from the beginning of production to the end of production with a more concise and clear appearance.

From all HMI system tests carried out, all of them can work well. Minor problems that occur are caused by the internet connection and connection to the web server. When the connection is lost, the HMI page cannot be accessed so that the monitoring and controlling process is disrupted

COUNCLUSION

Control and monitoring system HMI has been made for a prototype coffee bean packaging system with an IoT-based HMI using a website. ESP32 and ESP8266 microcontrollers are used as processing units and data communication between the HMI and the microcontroller. The average time of sending data to the database with the reconnecting process that occurs is 8 seconds, while the average time of receiving data from the database is 5.2 seconds. The accuracy of execution time also depends on the state of the internet connection and the state of the connection with the server. State data sent by the user from the HMI to the database can be sent and executed properly by the tool, namely state 0 for the off state and state 1 for the on state. industry so as to minimize errors in tool operation

REFERENCES

[1] Xia, F., Yang, L. T., Wang, L., & Vinel, A. (2012). Internet of things. *International journal of communication systems*, *25*(9), 1101.

[2] S M. Gunturi, H. D. Kotha, and M. Srinivasa Reddy, “An overview of internet of things,” *J. Adv. Res. Dyn. Control Syst.*, vol. 10, no. 9, pp. 659–665, 2018

[3] D. Krisnawati, “PERAN PERKEMBANGAN TEKNOLOGI DIGITAL PADA STRATEGI PEMASARAN DAN JALUR DISTRIBUSI UMKM DI INDONESIA (Studi Kasus: UMKM Kuliner Tanpa Restaurant ’Kepiting Nyinyir’),” vol. 6, no. 1, pp. 2338–4794, 2018

[4] H. O. Zulkarnaen, “Analisis Strategi Pemasaran Pada Usaha Kecil Menengah (UKM) Makanan Ringan,” *Fak. Ekon. Dan Bisnis*, p. 2015, 2012

[5] Gao, J., Liu, J., Rajan, B., Nori, R., Fu, B., Xiao, Y., ... & Philip Chen, C. L. (2014). SCADA communication and security issues. *Security and Communication Networks*, *7*(1), 175-194.

[6] A. Hou, C. Hu, K. Ma, Z. Cai, C. Huang and T. Pan, "Research on modeling and simulation of communication in power SCADA system," *2015 5th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT)*, 2015, pp. 226-230, doi: 10.1109/DRPT.2015.7432232.

[7] H. Haryanto and S. Hidayat, “Perancangan HMI (Human Machine Interface) Untuk Pengendalian Kecepatan Motor DC,” vol. 1, no. 2, 2012

[8] I. I. Hafidh, “PERANCANGAN HUMAN MACHINE INTERFACE (HMI) BERBASIS WEB SEBAGAI MEDIA PEMANTAUAN DAN PENGONTROLAN PROTOTIPE SMART HOME,” Universitas Diponegoro, 2019

[9] T. Lojka, P. Šatala, J. Mocnej, and I. Zolotová, “Web technologies in industry HMI,” *INES 2015 - IEEE 19th Int. Conf. Intell. Eng. Syst. Proc.*, pp. 103–106, 2015, doi: 10.1109/INES.2015.7329647

[10] P. Soepomo, “PERANCANGAN SISTEM INFORMASI INVENTORY SPARE PART ELEKTRONIK BERBASIS WEB PHP,” vol. 2, 2014

[11] C. Győrödi, R. Győrödi, G. Pecherle and A. Olah, "A comparative study: MongoDB vs. MySQL," *2015 13th International Conference on Engineering of Modern Electric Systems (EMES)*, 2015, pp. 1-6, doi: 10.1109/EMES.2015.7158433.

[12] M. P. Prosedural, S. Kasus, and B. Kabupaten, “Prosiding SNST ke-4 Tahun 2013 Fakultas Teknik Universitas Wahid Hasyim Semarang 95,” pp. 95–103, 2013

[13] V. Barriere *et al.*, “The Underground PHP and Oracle Manual,” *Environment*, 2012

[14] Espressif Systems, “Esp32-Wroom-32,” p. 27, 2021, [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf.

[15] D.F.Arranda, “KONTROL LAMPU RUANGAN BERBASIS WEB MENGGUNAKAN NODEMCU ESP8266,” no. 1, pp. 1–5, 2017