Smart monitoring system using NodeMCU for maintenance of production machines

Ignatius Deradjad Pranowo, Dian Artanto
Faculty of Vocational, Universitas Sanata Dharma, Yogyakarta, Indonesia

Article Info

Article history:

Received Jul 25, 2021
Revised Nov 11, 2021
Accepted Dec 9, 2021

Keywords:

ESP8266
IoT
Maintenance
Monitoring system
NodeMCU

ABSTRACT

Maintenance is an activity that helps to reduce risk, increase productivity, improve quality, and minimize production costs. The necessity for maintenance actions will increase efficiency and enhance the safety and quality of products and processes. On getting these conditions, it is necessary to implement a monitoring system used to observe machines' conditions from time to time, especially the machine parts that often experience problems. This paper presents a low-cost intelligent monitoring system using NodeMCU to continuously monitor machine conditions and provide warnings in the case of machine failure. Not only does it provide alerts, but this monitoring system also generates historical data on machine conditions to the Google Cloud (Google Sheet), includes which machines were down, downtime, issues occurred, repairs made, and technician handling. The results obtained are machine operators do not need to lose a relatively long time to call the technician. Likewise, the technicians assisted in carrying out machine maintenance activities and online reports so that errors that often occur due to human error do not happen again. The system succeeded in reducing the technician-calling time and maintenance work-reporting time up to 50%. The availability of online and real-time maintenance historical data will support further maintenance strategy.

Corresponding Author:

Ignatius Deradjad Pranowo
Faculty of Vocational, Universitas Sanata Dharma
Yogyakarta 55282, Indonesia
Email: dradjad@usd.ac.id

1. INTRODUCTION

The number of devices with capabilities sensing and wireless communication has been growing as a cost reduction, communication interfaces, and data analysis. This rapid development of the internet of things (IoT) across multiple domains starts from home automation to the environment, agriculture, health, and manufacturing systems. The application of IoT in home automation as a module trainer in [1], as home security and monitoring system in [2], like a remote control for home devices in [3]. The application of IoT in the environment as a real-time environment monitoring in [4], [5], as a weather monitoring system in [6]. The application of IoT in agriculture, we can find as a monitoring system in an irrigation system in [7], as a remote monitoring system for a greenhouse in [8]. The application of IoT in health as a patient monitoring system in [9], [10], as a prosthetic hand control in [11], as a remote medical diagnosis system in [12], as a healthcare monitoring system in [13], [14]. The application of IoT in manufacturing, we can find as a remote for power and energy in [15], as a speed sensing of a DC motor in [16], as monitoring electrical system in [17]. IoT uses Internet technology to allow devices to share information and communicate in sensing and actuating, i.e., the so-called Machine-to-Machine paradigm. We find the expression industrial IoT where Things refer to intelligent components, machines, and products. in the industrial environment [18], [19].

The development of this paper focuses on the new monitoring system for the performance of maintenance works through IoT monitored from the technician room. This work aimed to implement monitoring with low-cost and easy to build for maintenance service of production machines based on IoT. The NodeMCU ESP8266, as a cheap open-source software and has data processing capabilities, is used here. The remainder of this paper is structured as: section 2 presents the literature review, section 3 presents the method and proposed system, section 4 results and discussion, and section 5 provides the conclusion of the implemented system.

2. LITERATURE REVIEW

Currently, IoT is an emerging technology that has many potential benefits in many domains, including technology. IoT forms a heterogeneous network using various devices, protocols, technology, networks, middleware, applications, and systems data [20]. Many applications can be developed using IoT. Many research papers in this domain propose sensors and transducers embedded in microcontrollers integrated wireless transmission with local or cloud databases that translate input data into valuable and efficient information. To mention some of these works are: in [6] proposed a low-cost, simple weather monitoring and controlling system using ESP8266 NodeMCU, in [8] proposed monitoring system for a greenhouse based on NodeMCU ESP8266 and Blynk, in [11] proposed a prosthetic hand control interface using an ESP8266 Wi-Fi module, in [12] proposed a remote medical diagnosis system using NodeMCU ESP8266, in [21] proposed a prototype of micro-hydro monitored remotely using ESP8266 and LabVIEW, in [22] proposed smart monitoring for a pressure regulator based on IoT, in [23] discussed implementation of an industrial level sensor data acquisition and monitoring system using IoT, in [24] provided a sensing and monitoring system based on Thingspeak with Mathlab analysis, and in [25] proposed monitoring and leakage detection system using NodeMCU ESP8266 and Wi-Fi technology.

Table 1 presents some previous related works focusing on the controllers in use and their application areas. Several previous papers have discussed the use of the NodeMCU ESP8266. Most of the literature uses the NodeMCU ESP8266 to perform monitoring systems applied to several fields. Some of those areas are for applications in health systems, the environment, and home security. In this work, we focus on applications in industrial automation using the NodeMCU ESP8266 for monitoring and reporting systems for maintenance works. The NodeMCU ESP8266 is not used only as a monitoring tool, but by adding facilities from Google Sheets and Blynk can be used as an intelligent system for online reporting of maintenance work. This data is stored in the database. Furthermore, the data can be used as information to determine the maintenance strategies.

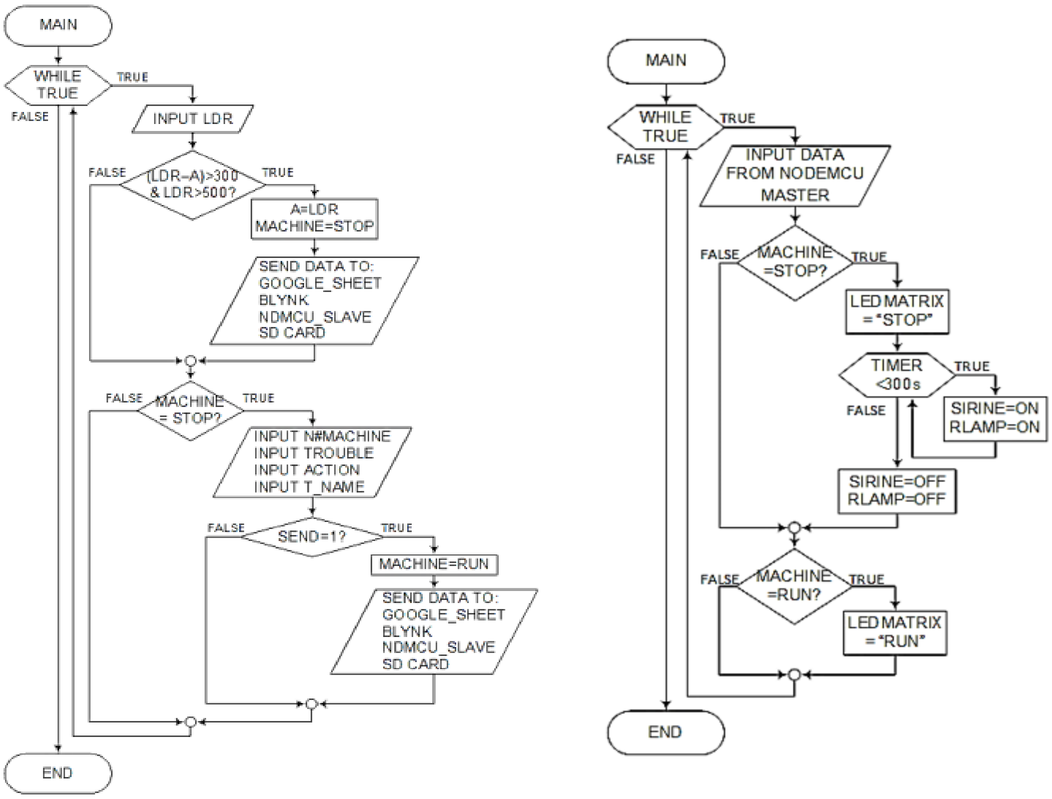
Table 1. The related works

Related Works	Controller	Application field
Das *et al.* [4]	Arduino UNO with an ESP8266 Wi-Fi module	Real time monitoring of various environmental conditions
Srivastava P. *et al.* [7]	Arduino UNO with an ESP8266 Wi-Fi module	Monitor and control, smart irrigation systems
Astutik *et al.* [8]	NodeMCU ESP8266 microcontroller and Android	Remote monitoring systems in greenhouse
Pakalapati *et al.* [11]	ESP8266 wireless module and Android	The prosthetic hand control interface to choose a gesture design
Fahim and Hossain [12]	NodeMCU ESP8266 microcontroller and Android	Remote monitoring medical condition (heartbeat and temperature)
Hasan *et al.* [22]	NodeMCU ESP8266 Wi-Fi module and MATLAB®	Remote monitoring system of a pressure regulator
Sarnin *et al.* [25]	NodeMCU ESP8266 Wi-Fi technology	Liquefied petroleum gas monitoring and leakage detection system
This work	NodeMCU ESP8266 Wi-Fi module, GoogleSheet, and Blynk	Monitoring and reporting system: the additional use of Google Sheets and Blynk provides the benefit of an intelligent online reporting system for maintenance work

3. METHOD AND PROPOSED SYSTEM

To achieve the goal of this work, namely monitoring the maintenance performance of production machines, the part of this work is the NodeMCU with ESP8266 embedded wifi module [26]. This paper presents a monitoring system using NodeMCU (ESP8266) that can monitor machine conditions every time and provide alerts to technicians in the event of a machine failure. The warning occurs in two ways, via alarms and indicator lights installed in the technician room or via a notification on a smartphone that is sent with the Blynk application so that the technician can immediately know the warning wherever he is, assuming that all technicians always carry a smartphone. Not only does it provide alerts, but this monitoring system will also send machine condition data to the Google Sheet whenever a machine failure occurs. The Google Sheet is used in this monitoring system because it has many features, some of which are: freely available, data stored in the cloud so that it is easily accessed anywhere and anytime, generate timestamps automatic every time data is entered, and can use to process and analyze data, and allows the addition of features by providing many add-on applications [27].

The monitoring system proposed in this paper consists of two parts. The first part is called NodeMCU master, and the second part is called NodeMCU slave. The working principle of NodeMCU master and NodeMCU slave is depicted in a flowchart as shown in Figure 1. The master NodeMCU will monitor the machine condition by reading signals from the light dependent resistor (LDR) sensor mounted on the red Andon lamp. When the red Andon lamp turns on, indicating machine failure, the light dependent resistor (LDR) sensor output changes causing the NodeMCU master to send machine stops condition data to Google Sheet, Blynk, memory card, and NodeMCU slave (Figure 1(a)). When the data is received, the slave NodeMCU will sound the buzzer, turn on the rotary light in the technician room for 5 minutes, and display an S (stop) code on the light emitting diode (LED) matrix (Figure 1(b)). Next, the technician makes repairs to the machine. After finished, the technician inputs data on the damage, the repairs carried out, and the name. Then the technician presses the Send button, and the NodeMCU master sends machine running condition data to Google Sheet, Blynk, memory card, and NodeMCU slave. Once the data is received, NodeMCU slave will display the R (run) code on the light emitting diode (LED) matrix display.



(a) (b)

Figure 1. The flowchart of (a) NodeMCU master and (b) NodeMCU slave

The monitoring system described in the flowchart above, is arranged in a wiring diagram as shown in Figures 2 and 3. The main parts of this system are two ESP8266 NodeMCUs, the first as a master and the second as a slave. The master unit shown in Figure 2 serves as a monitor for the four machines, positioned close to each machine's Andon lamp. The slave unit shown in Figure 3 is positioned in the technician's room. Next, the technician who received the information came and made repairs to the machine. After finished, the technician will enter data via the keypad on the master NodeMCU, namely the damage that occurred, the repairs performed, and the technician's name. As an alternative, data entry can also do through the Blynk application. After finished data entry, the technician presses the Send button, causing the NodeMCU master to send the machine running condition data to Google Sheet, Blynk, Memory Card, and NodeMCU slave. After the data is received, the NodeMCU slave will display the R (*run*) code on the light emitting diode (LED) matrix display. This failure detection system can automatically notify technicians, recording data from time to time online and in real-time, thus making monitoring better and more effective.

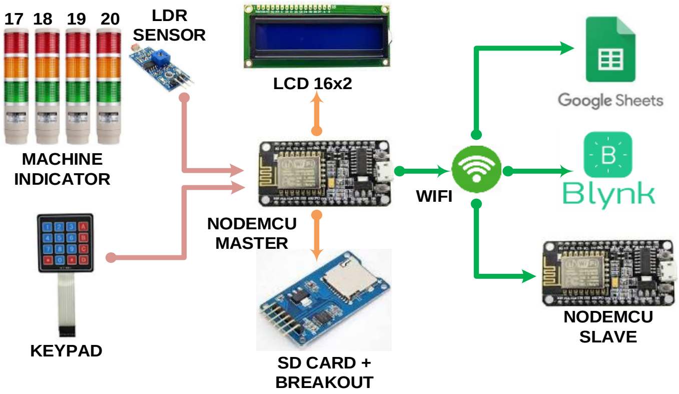


Figure 2. The wiring diagram of the master unit

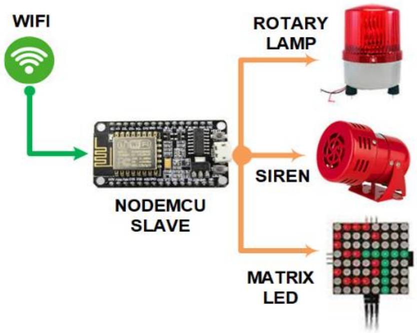
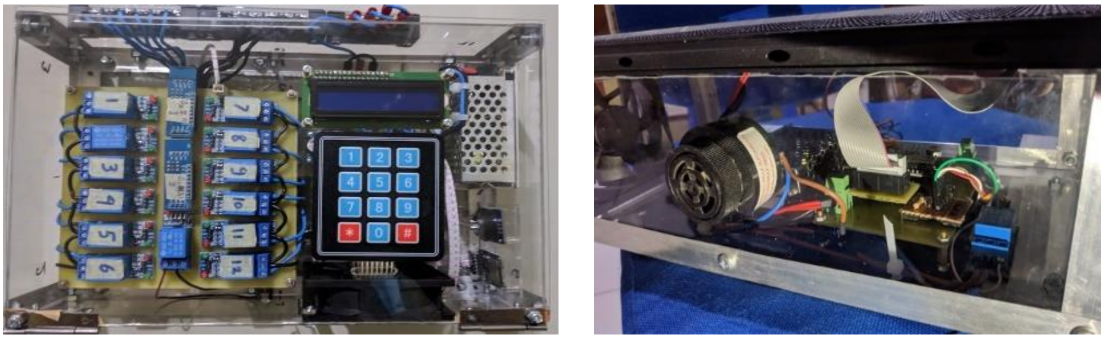


Figure 3. The wiring diagram of the slave unit

4. RESULTS AND DISCUSSION

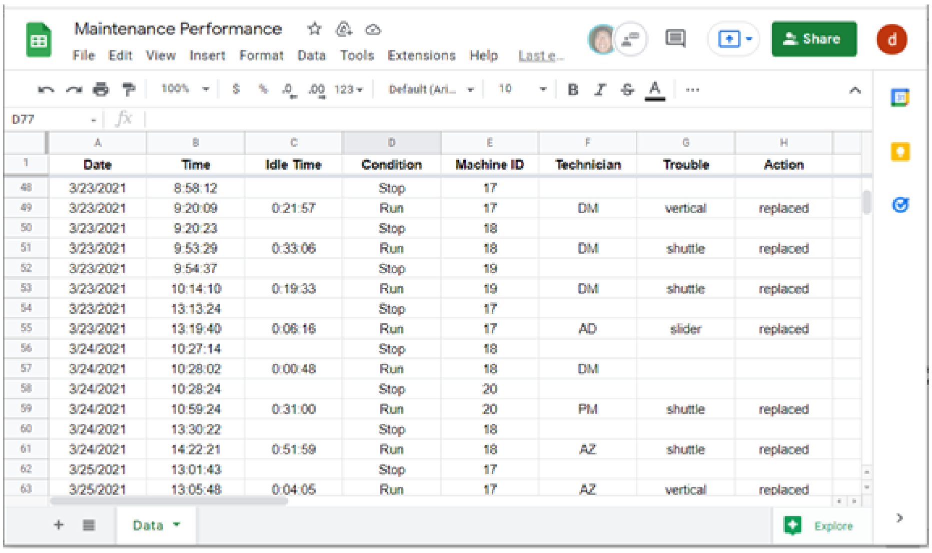
Figure 4 shows NodeMCU master and NodeMCU slave created based on the wiring diagrams in Figure 2 and Figure 3. The master NodeMCU connects to the light dependent resistor (LDR) sensor, Keypad, LCD 16x2, and Memory Card Breakout (Figure 4(a)). While the slave NodeMCU connects to the buzzer, rotary lamp, and light emitting diode (LED) Matrix (Figure 4(b)). NodeMCU master sends data to Google Sheet, Blynk, and NodeMCU slave via Wi-Fi connection.

Figure 5 shows the display of data sent by the NodeMCU master to the Google Sheet, with the contents of the data from columns 1-8 in a row containing the date, time, the machine stopped time, machine condition, machine ID, technician's name, trouble, and action did by the technician. The interesting here is those columns 1-3 are generated automatically by Google Sheet. Columns 1 and 2 are the timestamps that will be displayed every time entered data. Column 3, which shows the idle time (the duration of breakdown), is generated by calculating the difference between the Stop time and the Runtime, is made by the formula available on the Google Sheet. The Google Sheet feature makes using Google Sheets very useful because it can perform easy data processing [27]. Figure 6 shows the notification on Blynk when the machine stop (Figure 6(a)), the display of data entry by technician (Figure 6(b)), and the notification when the machine Run again (Figure 6(c)). Figure 7 shows the contents of the data stored on the memory card.



(a) (b)

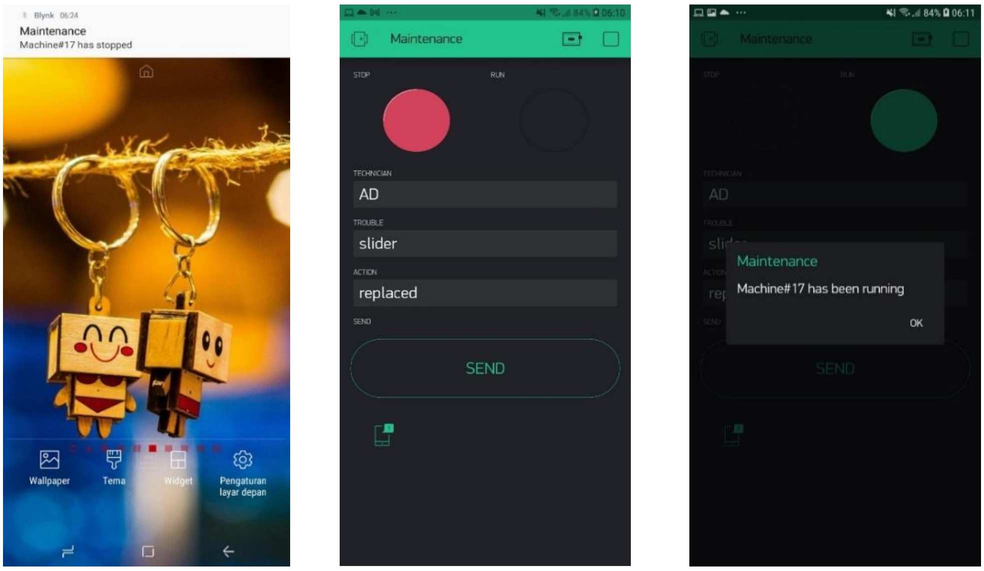
Figure 4. The controller box of (a) master unit and (b) slave unit



Maintenance Performance
File Edit View Insert Format Data Tools Extensions Help Last e...
Share
D77
Data
Explore

	Date	Time	Idle Time	Condition	Machine ID	Technician	Trouble	Action
48	3/23/2021	8:58:12		Stop	17			
49	3/23/2021	9:20:09	0:21:57	Run	17	DM	vertical	replaced
50	3/23/2021	9:20:23		Stop	18			
51	3/23/2021	9:53:29	0:33:06	Run	18	DM	shuttle	replaced
52	3/23/2021	9:54:37		Stop	19			
53	3/23/2021	10:14:10	0:19:33	Run	19	DM	shuttle	replaced
54	3/23/2021	13:13:24		Stop	17			
55	3/23/2021	13:19:40	0:06:16	Run	17	AD	slider	replaced
56	3/24/2021	10:27:14		Stop	18			
57	3/24/2021	10:28:02	0:00:48	Run	18	DM		
58	3/24/2021	10:28:24		Stop	20			
59	3/24/2021	10:59:24	0:31:00	Run	20	PM	shuttle	replaced
60	3/24/2021	13:30:22		Stop	18			
61	3/24/2021	14:22:21	0:51:59	Run	18	AZ	shuttle	replaced
62	3/25/2021	13:01:43		Stop	17			
63	3/25/2021	13:05:48	0:04:05	Run	17	AZ	vertical	replaced

Figure 5. The Google sheet data display



Maintenance
Machine#17 has stopped

TECHNICIAN
AD

TROUBLE
slider

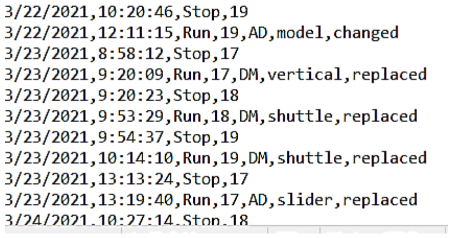
ACTION
replaced

SEND

Maintenance
Machine#17 has been running
OK

(a) (b) (c)

Figure 6. The Blynk's display: (a) notification when machine stop, (b) data entry by technician, and (c) notification when machine has run again



```
3/22/2021,10:20:46,Stop,19
3/22/2021,12:11:15,Run,19,AD,model,changed
3/23/2021,8:58:12,Stop,17
3/23/2021,9:20:09,Run,17,DM,vertical,replaced
3/23/2021,9:20:23,Stop,18
3/23/2021,9:53:29,Run,18,DM,shuttle,replaced
3/23/2021,9:54:37,Stop,19
3/23/2021,10:14:10,Run,19,DM,shuttle,replaced
3/23/2021,13:13:24,Stop,17
3/23/2021,13:19:40,Run,17,AD,slider,replaced
3/24/2021,10:27:14,Stop,18
```

Figure 7. The data stored on the memory card

From Figure 5, we can see that maintenance data is recorded automatically within a particular time. The maintenance data were previously manually recorded. Implementing the monitoring system using NodeMCU, maintenance data can be generated automatically and displayed online on the Google Sheet and be accessed anywhere and anytime. The results of the implementation of the monitoring system for maintenance work at the BTex Company were carried out by experiments on four machines with data collection allocations on March 22-25, 2021. BTex is a company that has requested to build a maintenance performance monitoring system on some of their production machines, in this case, machine numbers 17, 18, 19, and 20. Figure 5 displays the monitoring performance reporting data. The information covered in this system includes the date, time, machine condition, remark, machine number, technician's name, breakdown, and repair action.

Maintenance data show that the average failure on each machine is between 5-10 times per month. Then look for the average failure time, the time for calling a technician by the operator, and the maintenance recording time manually. From the average of the three, then a bar chart is made for each machine. Furthermore, experiments were carried out by applying a monitoring system and comparing the results. Figure 8 shows the data obtained from comparing the performance conditions of maintenance work before and after using the monitoring system. The maintenance data used are from December 2020, January 2021, and February 2021 for machines 17, 18, 19, and 20. The data are the time of operator calls a technician for repair (calling), during machine repair (maintenance), and when technicians perform reporting work (reporting). If we look at the bar chart of each machine, we can see that there are two data or bar charts. In December 2020, for machine 17, before using the monitoring system, the time for calling a technician was 10 minutes, the machine repair time was 30 minutes, and the time for a technician to do manual reporting was 6 minutes. After using the monitoring system, the time to call a technician is only 4 minutes, and the reporting time is only 1 minute, the repair time is still 30 minutes. The conditions before and after are indicated by two rods mounted on machine 17 (bottom=before; top=after), so it is clear that there is a significant reduction in overall maintenance work time. Likewise with repair failure incidents in January and February 2021 that occurred on other machines.

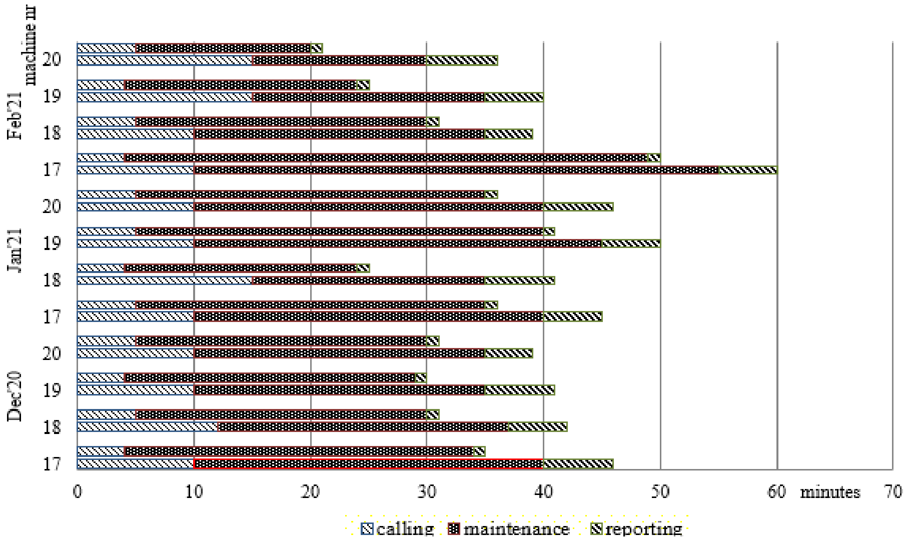


Figure 8. The comparation data between before and after applying the monitoring system

5. CONCLUSION

This paper has succeeded in developing an intelligent monitoring system for the maintenance performance of production machines. The novelty offered here is not only for monitoring but also for carrying out reporting actions. This monitoring system has supported maintenance work to be more effective and facilitated better record keeping of machine conditions from time to time. It is because the data is recorded automatically and can be accessed online anywhere and anytime. In addition, the human error in reporting machine maintenance history does not occur again. The speed and accuracy in identifying failure are better. We get a reduction time of up to 50% in terms of calling and reporting after implementing the system and acquiring maintenance data for a certain period. For further work can be done in terms of detecting machine conditions via sensor input from Andon lamps. The information will not only be in the form of a stopped machine but will also find out which part of the machine is damaged based on the color and the active Andon light indicator.

REFERENCES

[1] A. Sarah, T. Ghozali, G. Giano, M. Mulyadi, S. Octaviani, and A. Hikmaturokhman, "Learning IoT: basic experiments of home automation using ESP8266, Arduino and XBee," *2020 IEEE International Conference on Smart Internet of Things (SmartIoT),* 2020, pp. 290-294, doi: 10.1109/SmartIoT49966.2020.00051.

[2] Taryudi, D. B. Adriano, and W. A. C. Budi, "Iot-based integrated home security and monitoring system," *International Conference on Electrical, Electronic, Informatics and Vocational Education (ICE-ELINVO 2018),* 2018, pp.1-7, doi: 10.1088/1742-6596/1140/1/012006.

[3] S. P. Makhanya, E. M. Dogo, N. I. Nwulu, and U. Damisa, "A smart switch control system using ESP8266 Wi-Fi module integrated with an android application," *2019 IEEE 7th International Conference on Smart Energy Grid Engineering (SEGE)*, 2019, pp. 125-128, doi: 10.1109/SEGE.2019.8859904.

[4] A. Das, M. P. Sarma, K. K. Sarma, and N. E. Mastorakis, "Design of an IoT based real time environment monitoring system using legacy sensors," *MATEC Web Conf* 2018; 210: pp.2–5, doi: 10.1051/matecconf/201821003008.

[5] S. Kompella, B. Vikas, and Reddy, Y. S. Reddy, "A Novel environmental monitoring system for real time using Arduino and Node JS," *International Journal of Recent Technology and Engineering (IJRTE),* vol. 8, no. 5, pp. 5520–5525, 2020, doi: 10.35940/ijrte.e6310.018520.

[6] S. Mahmood, S. Alani, F. Hasan, and M. Mustafa, "Esp 8266 node mcu based weather monitoring system," *Proceedings of the 1st International Multi-Disciplinary Conference Theme: Sustainable Development and Smart Planning, IMDC-SDSP 2020, Cyperspace,* 28-30 June 2020, pp. 150-163, doi: 10.4108/eai.28-6-2020.2298609.

[7] P. Srivastava, M. Bajaj, and A. S. Rana, "Irrigation system using IOT," *2018 Fourth Int Conf Adv Electr Electron Information, Commun Bio-Informatics,* 2018, pp. 1–5.

[8] Y. Astutik, Murad, G. M. D. Putra, and D. A. Setiawati, "Remote monitoring systems in greenhouse based on NodeMCU ESP8266 microcontroller and Android," *AIP Conf Proc*; 2199, 2019, pp. 030003.1-14, doi: 10.1063/1.5141286.

[9] A. Julius and Z.-Min. "IoT Based patient health monitoring system using LabVIEW and wireless sensor network," *International Journal of Science and Research (IJSR),* vol. 6, no. 3, pp.894–900, 2017, doi: 10.21275/ART20171643.

[10] R. Jayaysingh, J. David, M. Joel Morris Raaj, D. Daniel, and D. BlessyTelagathoti, "IoT based patient monitoring system using NodeMCU," 2020 5th International Conference on Devices, Circuits and Systems (ICDCS), 2020, pp. 240-243, doi: 10.1109/ICDCS48716.2020.243588.

[11] S. S. Pakalapati, G. G. Chary, A. K. Yadaw, S. Kumar, H. K. Phulawariya, and R. Kumar, "A prosthetic hand control interface using ESP8266 Wi-Fi module and Android application," 2017 I*nternational Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS),* 2017, pp. 1-3, doi: 10.1109/ICIIECS.2017.8275978.

[12] F. Fahim and S. A. Hossain, "IoT based remote medical diagnosis system using NodeMCU," *2019 13th Int Conf Software, Knowledge, Inf Manag Appl Ski 2019,* 2019, pp.1-7, doi: 10.1109/SKIMA47702.2019.8982509.

[13] K. T. Wai, N. P. Aung, and L. L. Htay, "Internet of things (IoT) based healthcare monitoring system using NodeMCU and Arduino UNO," *Int J Trend Sci Res Dev Int J Trend Sci Res Dev*, vol. 3, no. 5. pp.755–759, 2019, doi: 10.31142/ijtsrd26482.

[14] S. Dharmoji, A. Anigolkar, and M. Shraddha, "IoT based patient health monitoring using ESP8266," *International Research Journal of Engineering and Technology (IRJET),* vol. 7, no. 3, pp. 3619–3624, 2020, [Online]. Available: https://www.irjet.net/archives/V7/i3/IRJET-V7I3728.pdf

[15] P. Srivastava, M. Bajaj and A. S. Rana, "IOT based controlling of hybrid energy system using ESP8266," *2018 IEEMA Engineer Infinite Conference (eTechNxT),* 2018, pp. 1-5, doi: 10.1109/ETECHNXT.2018.8385294.

[16] A. Jain and B. K. Roy, "Design and implementation of an IoT ready smart sensor for speed sensing of a DC motor using IEEE 802.15.1 and ESP8266," *International Journal of Engineering & Technology,* vol. 7, pp. 974-978, 2018 doi: 10.14419/ijet.v7i3.12.17615.

[17] I. Hajar, M. Hafizd, A. W. Dani, and S. Miharno, "Monitoring of electrical system using internet of things with smart current electric sensors," *Sinergi,* vol. 22, pp. 211-218, 2018, doi: 10.22441/sinergi.2018.3.010.

[18] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: challenges, opportunities, and directions," in *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724-4734, Nov. 2018, doi: 10.1109/TII.2018.2852491.

[19] N. B. Long, H. Tran-Dang, and D. Kim, "Energy-aware real-time routing for large-scale industrial internet of things," in *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2190-2199, June 2018, doi: 10.1109/JIOT.2018.2827050.

[20] B. Dhanalaxmi and G. A. Naidu, "A survey on design and analysis of robust IoT architecture," *2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 2017, pp. 375-378, doi: 10.1109/ICIMIA.2017.7975639.

[21] M. Kusriyanto, H. Utama, and R. Rais, "Prototype of microhydro generator with complement load and telemonitoring using labview software with ESP8266 module interface," *International Conference on Electronics Representation and Algorithm (ICERA 2019),* 2019, pp.1-6, doi: 10.1088/1742-6596/1201/1/012014.

[22] A. Hasan, A. H. Duhis, L. Alzurfi, N. Aad, and M. J. Mnati, "Smart monitoring system for pressure regulator based on IOT," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 5, pp. 3450–3456, 2019, doi: 10.11591/ijece.v9i5.pp3450-3456.

[23] N. G. Magere and P. Pal, "IoT based industrial level sensor data acquisition & monitoring," *International Research Journal of Engineering and Technology (IRJET)*, vol. 6, no. 7, pp. 1985–1987, 2019, [Online]. Available: https://www.irjet.net/archives/V6/i7/IRJET-V6I7279.pdf

[24] S. Pasha, "Thingspeak based sensing and monitoring system for IoT with Matlab analysis," *International Journal of New Technology and Research (IJNTR)*, vol. 2, no. 6, pp. 19–23, 2016.

[25] S. S. B. Sarnin *et al.*, "Liquefied petroleum gas monitoring and leakage detection system using nodemcu ESP8266 and wi-fi technology," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 17, no. 1, pp.166–174, 2019, doi: 10.11591/ijeecs.v17.i1.pp166-174.

[26] F. Koyanangi, "NodeMCU ESP8266: Details and Pinout," WwwInstructablesCom, pp. 1–7, 2018.

[27] L. Laskowski, "Google Forms and Sheets for library gate counts," *Journal of Access Services*, vol. 13, no. 3, pp. 151–158, 2016, doi: 10.1080/15367967.2016.1184577.

BIOGRAPHIES OF AUTHORS

Ignatius Deradjad Pranowo received his master's degree in the Department of Mechatronics from the Asian Institute of Technology in Thailand, (2002). Since 1999, he became a lecturer in Sanata Dharma University, Indonesia. His areas of interest are PLC, Mechatronic, SCADA. He can be contacted at email: dradjad@usd.ac.id; dradjad@pmsd.ac.id.

Dian Artanto received his bachelor's degree in the Department of Electrical Engineering from the Gadjah Mada University in Yogyakarta-Indonesia, (1998). He obtained his Master of Engineering (M.Eng.) degree in 2007 in the Department of Mechatronics from the Asian Institute of Technology in Thailand. Since 2000, he became a lecturer in Sanata Dharma University, Indonesia. His areas of interest are LabVIEW, Programming algorithm, Interface, SCADA. He can be contacted at email: dian.artanto@gmail.com.