Indonesian Journal 1

by Pranowo Deradjad

Smart monitoring system using NodeMCU for maintenance of production machines

Ignatius Deradjad Pranowo[1], Dian Artanto[2]
[1,2] Mekatronika, Fakultas Vokasi, Universitas Sanata Dharma, Yogyakarta 55282, Indonesia

Article Info

Article history:

Received
Revised
Accepted

Keywords:

Monitoring system
IoT
NodeMCU
ESP8266
Maintenance

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,
Mekatronika, Fakultas Vokasi,
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, products, etc. 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 follows: 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. The related works

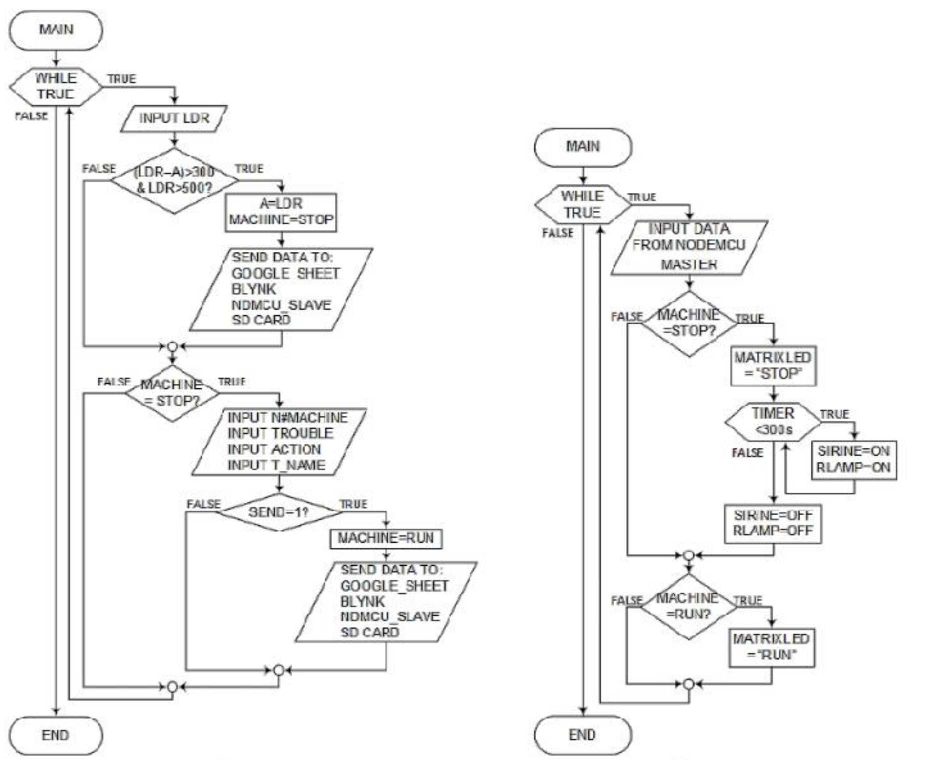
related works	controller	application field
Das A, et al. [4]	Arduino UNO with an ESP8266 Wi-Fi module	Real time monitoring of various environmental conditions
Srivastava P, Bajaj M, Rana AS. [7]	Arduino UNO with an ESP8266 Wi-Fi module	Monitor and control, smart irrigation systems
Astutik Y, et al. [8]	NodeMCU ESP8266 microcontroller and Android	Remote monitoring systems in greenhouse
Pakalapati SS, et al. [11]	ESP8266 wireless module and Android	The prosthetic hand control interface to choose a gesture design
Faisal F, Hossain SA. [12]	NodeMCU ESP8266 microcontroller and Android	Remote monitoring medical condition (heartbeat and temperature)
Ali AH, et al. [22]	NodeMCU ESP8266 Wi-Fi module and MATLAB®	Remote monitoring system of a pressure regulator
Binti Sarnin SS, Ametefe DS, Naim NF, 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

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.

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 has described as the flowchart shown in Figure 1. The NodeMCU Master monitors by reading the LDR sensor. When the Red Andon lamp turns on (machine failure), the LDR sensor output changes, NodeMCU Master sends machine stop condition data to Google Sheet, Blynk, Memory card, and NodeMCU Slave. Once the data is received, NodeMCU Slave sounds Buzzer and turns on the Rotary lamp in the technician room for 5 minutes, then displays the S (stop) code on the LED matrix display. 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 LED matrix display.



a. b.

Figure 1. The flowchart of: (a) NodeMCU Master; and (b) NodeMCU Slave

The monitoring system described above can be illustrated in a block diagram as shown in Figure 2 and Figure 3. The main parts of this system are two ESP8266 NodeMCUs, the first as a master and the second as a slave. The master block diagram shown in Figure 2 serves as a monitor for the four machines, positioned close to each machine's Andon lamp. The slave block diagram shown in Figure 3 is positioned in the technician's room. The following is a brief explanation of how the Master and Slave units work with this monitoring system: The Master NodeMCU will monitor the machine condition by reading signals from the LDR sensor mounted on the Red Andon lamp. When the Red Andon lamp turns on, indicating machine failure, the LDR sensor output changes causing the NodeMCU Master to send machine stops condition data to Google Sheet, Blynk, memory card, and NodeMCU Slave. 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 LED matrix. 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 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.

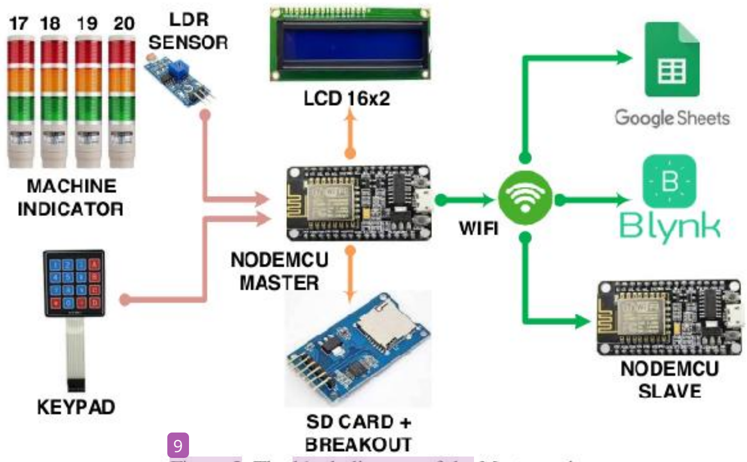


Figure 2. The block diagram of the Master unit

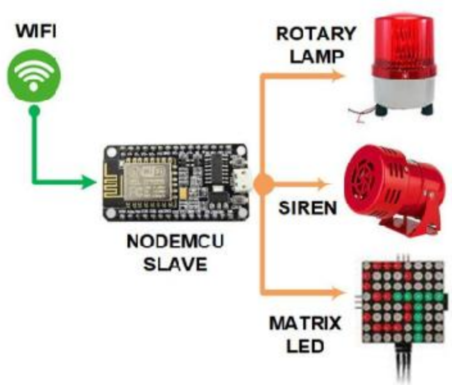
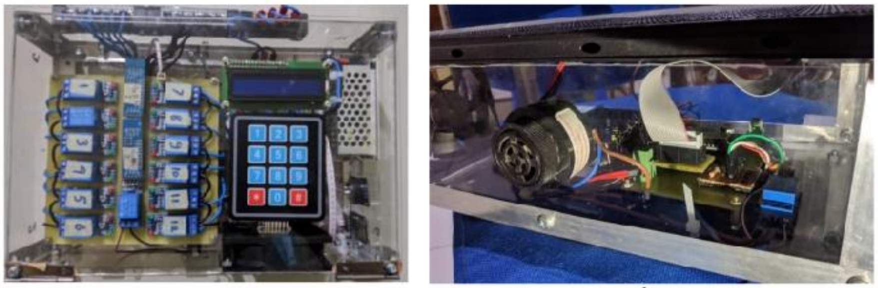


Figure 3. The block diagram of the Slave unit

4. RESULTS AND DISCUSSION

Figure 4 shows NodeMCU Master and NodeMCU Slave created based on the block diagrams in Figure 2 and Figure 3 above. The Master NodeMCU connects to the LDR light sensor, Keypad, LCD16x2, and Memory Card Breakout. While the Slave NodeMCU connects to the Buzzer, Rotary lamp, and LED Matrix. NodeMCU Master sends data to Google Sheet, Blynk, and NodeMCU Slave via Wi-Fi connection.



a. b.

Figure 4. The Master unit (a) and Slave unit (b)

Table 2 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 5 shows the data display on Blynk when the machine is in Stop and Run condition. Figure 6 shows the contents of the data stored on the memory card.

Table 2. The Google Sheet data display

Maintenance Performance

1	Date	Time	Idle Time	Condition	Machine ID	Technician	Trouble	Action
46	3/22/2021	10:20:46		Stop	19			
47	3/22/2021	12:11:15	1:50:29	Run	19	AD	model	changed
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	DM	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

a. b. c.

Figure 5. The Blynk's display: (a) notification when machine stop, (b) data entry by technician, (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 6. The data stored on the memory card

From Table 2, 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. Table 2 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.

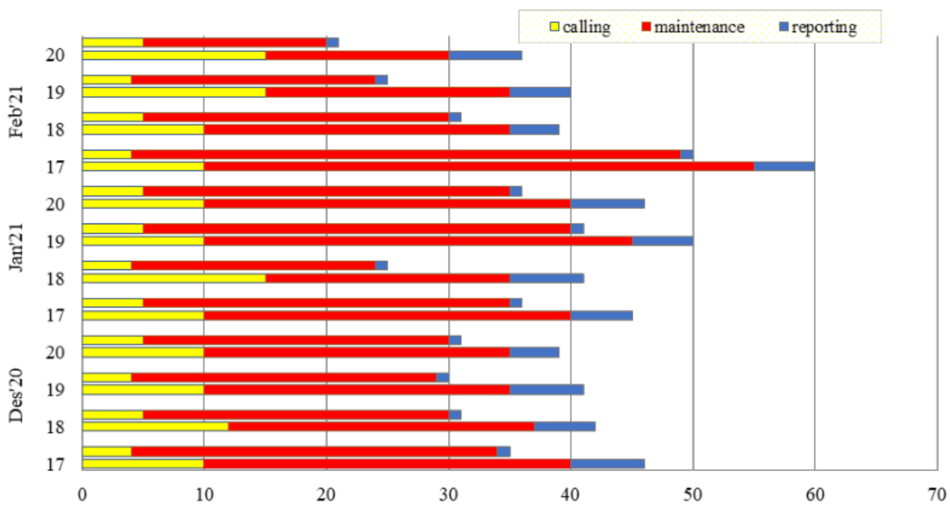


Figure 7. The comparation data between before and after applying the monitoring system

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 7 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.

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]. Sarah A, Ghozali T, Giano G, et al. Learning IoT: Basic Experiments of Home Automation using ESP8266, Arduino and XBee. *Proc - 2020 IEEE Int Conf Smart Internet Things, SmartIoT 2020* 2020; 290–294.

[2]. Taryudi, Adriano DB, Ciptoning Budi WA. Iot-based Integrated Home Security and Monitoring System. *J Phys Conf Ser*; 1140. Epub ahead of print 2018. DOI: 10.1088/1742-6596/1140/1/012006.

[3]. Makhanya SP, Dogo EM, Nwulu NI, et al. A Smart Switch Control System Using ESP8266 Wi-Fi Module Integrated with an Android Application. *Proc 2019 7th Int Conf Smart Energy Grid Eng SEGE 2019* 2019; 125–128.

[4]. Das A, Sarma MP, Sarma KK, et al. Design of an IoT based Real Time Environment Monitoring System using Legacy Sensors. *MATEC Web Conf* 2018; 210: 2–5.

[5]. Kompella S, Vikas B, Reddy YS. A Novel Environmental Monitoring System for Real Time using Arduino and Node JS. *Int J Recent Technol Eng* 2020; 8: 5520–5525.

[6]. Mahmood SN, Alani S, Hasan FF, et al. Esp 8266 node mcu based weather monitoring system. *PervasiveHealth Pervasive Comput Technol Healthc* 2020; 2: 150–163.

[7]. Srivastava P, Bajaj M, Rana AS. Irrigation System using IOT. *2018 Fourth Int Conf Adv Electr Electron Information, Commun Bio-Informatics* 2018; 1–5.

[8]. Astutik Y, Murad, Putra GMD, et al. Remote monitoring systems in greenhouse based on NodeMCU ESP8266 microcontroller and Android. *AIP Conf Proc*; 2199. Epub ahead of print 2019. DOI: 10.1063/1.5141286.

[9]. Jian-Min AJZ. IoT Based Patient Health Monitoring System Using LabVIEW and Wireless Sensor Network. *Int J Sci Res* 2017; 6: 894–900.

[10]. Jayaysingh R, David J, Joel Morris Raaj M, et al. IoT Based Patient Monitoring System Using NodeMCU. *ICDCS 2020 - 2020 5th Int Conf Devices, Circuits Syst* 2020; 240–243.

[11]. Pakalapati SS, Govardhana Chary G, Yadaw AK, et al. A prosthetic hand control interface using ESP8266 Wi-Fi module and Android application. *Proc 2017 Int Conf Innov Information, Embed Commun Syst ICIIECS 2017* 2018; 2018-Janua: 1–3.

[12]. Faisal F, Hossain SA. IoT based remote medical diagnosis system using NodeMCU. *2019 13th Int Conf Software, Knowledge, Inf Manag Appl Ski 2019*. Epub ahead of print 2019. DOI: 10.1109/SKIMA47702.2019.8982509.

[13]. Khin Thet Wai | Nyan Phyo Aung | Lwin Lwin 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* 2019; 3: 755–759.

[14]. Dharmoji S, Anigolkar A, M PS. IoT based Patient Health Monitoring using ESP8266. 2020; 3619–3624.

[15]. Mukhopadhyay S, Urooj S, IEEE Power & Energy Society, et al. 2018 IEEMA Engineer Infinite Conference (eTechNxT) : March 13th-14th, 2018 : India Expo Mart, Greater Noida, Gautam Bugh Nagar, Uttar Pradesh 201306, India. *2018 IEEMA Eng Infin Conf* 2018; 1–5.

[16]. Jain A, K. Roy B. Design and Implementation of an IoT Ready Smart Sensor for Speed Sensing of a DC Motor using IEEE 802.15.1 and ESP8266. *Int J Eng Technol* 2018; 7: 974.

[17]. Ibnu Hajar MH, Dani AW, Miharno S. Monitoring of Electrical System Using Internet of Things With Smart Current Electric Sensors. *Sinergi* 2018; 22: 211.

[18]. Sisinni E, Saifullah A, Han S, et al. Industrial internet of things: Challenges, opportunities, and directions. *IEEE Trans Ind Informatics* 2018; 14: 4724–4734.

[19]. Long NB, Tran-Dang H, Kim DS. Energy-Aware Real-Time Routing for Large-Scale Industrial Internet of Things. *IEEE Internet Things J* 2018; 5: 2190–2199.

[20]. Dhanalaxmi B, Naidu GA. A survey on design and analysis of robust IoT architecture. *IEEE Int Conf Innov Mech Ind Appl ICIMIA 2017 - Proc* 2017; 375–378.

[21]. Kusriyanto M, Utama H, Rais R. Prototype of Microhydro Generator with Complement Load and Telemonitoring Using Labview Software with ESP8266 Module Interface. *J Phys Conf Ser*; 1201. Epub ahead

of print 2019. DOI: 10.1088/1742-6596/1201/1/012014.
[22]. Ali AH, Duhis AH, Lafta Alzurfi NA, et al. Smart monitoring system for pressure regulator based on IOT. *Int J Electr Comput Eng* 2019, 9. 3450–3456.
[23]. Magare NG, Pal P. IoT Based Industrial Level Sensor Data Acquisition & Monitoring. 2019; 1985–1987.
[24]. Pasha S. Thingspeak Based Sensing and Monitoring System for IoT with Matlab Analysis. *Int J New Technol Res* 2016; 2: 19–23.
[25]. Binti Sarnin SS, Ametefe DS, Naim NF, et al. Liquefied petroleum gas monitoring and leakage detection system using nodemcu ESP8266 and wi-fi technology. *Indones J Electr Eng Comput Sci* 2019; 17: 166–174.
[26]. Koyanangi F. NodeMCU ESP8266: Details and Pinout. *WwwInstructablesCom* 2018; 1–7.
[27]. Laskowski L. Google Forms and Sheets for library gate counts. *J Access Serv* 2016; 13: 151–158.

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 systems, SCADA.

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.

Indonesian Journal 1

ORIGINALITY REPORT

9%	5%	8%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Joao Mesquita, Diana Guimaraes, Carlos Pereira, Frederico Santos, Luis Almeida. "Assessing the ESP8266 WiFi module for the Internet of Things", 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), 2018 Publication	2%
2	ijece.iaescore.com Internet Source	1%
3	Michel A. Kinsy, Ivan Celanovic, Omer Khan, Srinivas Devadas. "MARTHA: Architecture for Control and Emulation of Power Electronics and Smart Grid Systems", 'EDAA', 2013 Internet Source	1%
4	Riccardo Manzini, Alberto Regattieri, Hoang Pham, Emilio Ferrari. "Maintenance for Industrial Systems", Springer Science and Business Media LLC, 2010 Publication	1%
5	rymayadi.com	

Internet Source

1%

6 Submitted to Universiti Teknologi MARA
Student Paper
<1%

7 Aparajita Das, Manash Pratim Sarma, Kandarpa Kumar Sarma, Nikos Mastorakis. "Design of an IoT based Real Time Environment Monitoring System using Legacy Sensors", MATEC Web of Conferences, 2018
Publication
<1%

8 Ivan Vaccari, Maurizio Aiello, Federico Pastorino, Enrico Cambiaso. "Protecting the ESP8266 Module from Replay Attacks", 2020 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI), 2020
Publication
<1%

9 WWW.MDPI.COM
Internet Source
<1%

10 "Advances in Artificial Intelligence and Data Engineering", Springer Science and Business Media LLC, 2021
Publication
<1%

11 eudl.eu
Internet Source
<1%

12 ijeecs.iaescore.com
Internet Source
<1%

13	Sameer Alani, Sarmad Nozad Mahmood, Sarah Zaeead Attaallah, Haneen Sameer Mhmood et al. "IoT based implemented comparison analysis of two well-known network platforms for smart home automation", International Journal of Electrical and Computer Engineering (IJECE), 2021 Publication	<1 %
14	"Computing, Communication and Signal Processing", Springer Science and Business Media LLC, 2019 Publication	<1 %

Exclude quotes	On	Exclude matches	< 5 words
Exclude bibliography	On		