# Design of Human Heartbeat Monitoring System Based on Wireless Sensor Networks

**Qisthi Alhazmi Hidayaturrohman[1], M. Arifudin Lukmana[2], Akhmad Nidhomuz Zaman[3]**

[1]Department of Electrical Engineering,
Universitas Pembangunan Nasional Veteran
Jakarta, Indonesia
qisthialhazmi@upnvj.ac.id

[2]Department of Mechanical Engineering,
Universitas Pembangunan Nasional Veteran,
Jakarta, Indonesia

[3]Department of Industrial Engineering,
Universitas Pembangunan Nasional Veteran,
Jakarta, Indonesia

**Abstrak**

Teknologi *internet of things* (IoT) memiliki peran yang penting dalam revolusi Industri 4.0. Teknologi IoT memiliki potensi untuk diimplementasikan ke dalam industri kesehatan, khususnya untuk pengembangan sistem telemedis. IoT memungkinkan pengiriman data sensor secara nirkabel ke fasilitas kesehatan terdekat, seperti rumah sakit. Pada *paper* ini, sistem pemantauan detak jantung dirancang dengan menggunakan protokol komunikasi 802.11 dan antarmuka web sederhana. Sensor pulsa yang digunakan mampu membaca detak pulsa dari denyut jantung manusia dan mengonversinya ke dalam satuan *beat per minute* (BPM). Sensor menghasilkan 98,89 persen akurasi dan 1,11 persen *error* pembacaan jika dibandingkan dengan hasil yang ada pada jam cerdas. Sistem ini juga mampu mengirimkan data sensor secara nirkabel dari *node* sensor ke *node* koordinator. *Node* koordinator juga mampu untuk menerima data sensor dan menyimpannya ke dalam basis data dengan menggunakan metode POST dan GET, serta memvisualisasikannya secara sederhana ke antarmuka *web*, sehingga pengguna lain dapat melihat hasil visual data sensor yang diterima.

**Kata kunci:** *wireless sensor networks, internet of things, heartbeat sensor*

**Abstract**

The internet of things (IoT) technology plays an important role in Industry 4.0 revolution. The IoT technology has potential to be implemented in the medical industry, especially for the development of telemedicine system. IoT able to send the medical sensor data wirelessly to the nearest medical facility like hospital. In this paper, the heart beat monitoring system was designed by using 802.11 communication protocol and simple web interface. The pulse sensor was able to read the pulse rate of the human and convert it to beat per minute (BPM). It has 98.89 percent accuracy and 1.11 percent error compared to the smartwatch result. In the other hand, ESP-32 also implemented as the microcontroller as well as the sensor node of the system. It was able to send the data wirelessly from sensor node to the coordinator

Received: 29 Desember 2022 Revised: 1 Maret 2023 Accepted: 20 Maret 2023

node. The coordinator node was also able to fetch the sensor data into the database using POST and GET method and then visualize the sensor data over web interface so that the other users are able to see the visualization of the sensor data.

**Keywords:** wireless sensor networks, internet of things, heartbeat sensor

## 1. Introduction

Internet of Things (IoT) technology plays a role in Industry 4.0 that allows electronic devices, in this case, sensors, to send their data through communication networks [1]. The IoT technology can be implemented in electronic devices, such as a telemetry system, to send medical data from patients to the nearest health facility. This approach can support the development of telemedicine applications and e-healthcare systems. Bakolo explains that telemedicine system applications and e-healthcare technology implementation increased during the COVID-19 pandemic [2]. Singh et al. conclude that IoT technology implementation could help health service management, especially in the hospital, and help to quickly identify infected patients [3]. The implementation of IoT technology in the medical field allows the patients to be monitored through internet networks, where the patients' location should not be near the medical facility. Wu et al. implemented the wireless body area networks in healthcare applications, resulting in the IoT technology that being implemented very well in medical applications [4]. However, Wu et al. only used Bluetooth communication as a medium to send their sensor data. Another researcher, Guillen et al., explains that some IoT protocol models could be used in medical applications, mainly to monitor human health, like heartbeat [5]. According to the research done by Rohman and Salahuddin, Wi-Fi communication can be used as a medium to send sensor data from the sensor node to the coordinator node [6]. Apart from the communication system, a visualization system is also needed to visualize the received sensor data result by a coordinator node [7]. Hidayaturrohman et al. also utilized a web server as a coordinator node to visualize the sensor data, so that the sensor data could be presented well [8]. Based on those several references, in this *paper*, an IoT system was designed for medical application, especially to monitor human heartbeat through Wi-Fi networks and a simple website as an interface. The microcontroller used is ESP-32 [9].

## 2. Literature Review

### 2.1. Related papers

Bakolo explains that telemedicine and e-healthcare had increased during the COVID-19 pandemic [2], and Singh et al. conclude that the IoT technology implementation could help the health service management [3]. IoT technology in the medical field allows patients to be monitored over an internet network, so that the patients do not need to visit the health facility. Wu et al. implemented wireless body area networks (WBAN) in healthcare applications, resulting in IoT technology being implemented well in medical applications [4]. However, the study of Wu et al. just only used Bluetooth (BLE) communication system as a medical sensor data medium.

Furthermore, Guillen et al. explained that some IoT protocol models could be used in medical application needs, especially to monitor human health conditions, such as heartbeat [5]. Refer to the study by Rohman and Salahuddin, where they used Wi-Fi as a communication medium to send sensor data from the node sensor to the coordinator node [6]. Aside from the communication system, a visualization system is also required to visualize received sensor data results. Hidayaturrohman et al. implemented a web server as a coordinator node in their study to store the sensor data and visualize it so that the sensor data can be represented well [8].

### 2.2. Wireless Sensor Networks

The sensor is an electronic device that responds to physical changes, such as light, heat, sound, and others, and produces an electronic signal based on the physical changes. For several years, the sensor system has already been used in many industries, such as the military, manufacturing, and medical applications. Sensor networks are sensor nodes connected to the coordinator node, where sensor data is sent through a particular communication medium with determined protocol communication. Generally, the sensor node is in the form of an embedded system, allowing the sensor to read environmental and physical changes and send those data to the coordinator node, which is commonly a server computer or minicomputer like Raspberry Pi [9]. Another type of sensor network, wireless sensor networks or WSNs, is a set of a sensor nodes connected wirelessly to the coordinator node through wireless medium and protocol communication. The commonly used protocol in wireless sensor networks applications is Wi-Fi or 802.11 a/g protocol. Wireless sensor networks allow the sensor node to connect over a long distance to the coordinator node; in some cases, it could be up to 1 kilometer if the position between the sensor node and the coordinator node is point-to-point [9].

## 3. Research Method

### 3.1. Block Diagram of The System

In this study, the proposed system involves several electronic devices such as an ESP-32 microcontroller, a heartbeat sensor, a computer enabled as a server, a router, and another computer or smartphone as a client. Those electronic devices are presented in a block diagram of the system presented in Figure 1.

In this paper, the tree topology is used as a network topology in the IoT system, and all of them are connected wirelessly through wireless local area network (WLAN). The topology consists of sensor node of ESP-32 and a heartbeat sensor connected to the ESP-32. The pulse sensor is used in this study [10]. The coordinator node is a computer that has already installed with the webserver to receive (POST) the sensor data and save the data to the database, network devices such as a router to manage the network traffic, and a client node or end user that aims to access the web interface of the system. In the sensor node, the sensor will read the human heartbeat data change in analog value, and further the value is converted to digital bit value by using the Arduino command. After that, the digital bit is converted into BPM data. All the processes have been proceeded in ESP-32. After obtaining the BPM value, the ESP-32 sends the data sensor to the hosting server as a coordinator node through Wi-Fi that connected locally with the ESP-32. The sensor data is received (GET) by the web server on the coordinator node side and then saved to the database (POST). After the data is inputted into the database, the computer uses a PHP script to visualize the data in the database by using a web interface. The end user is used as a third party to view the visualization of the interface.

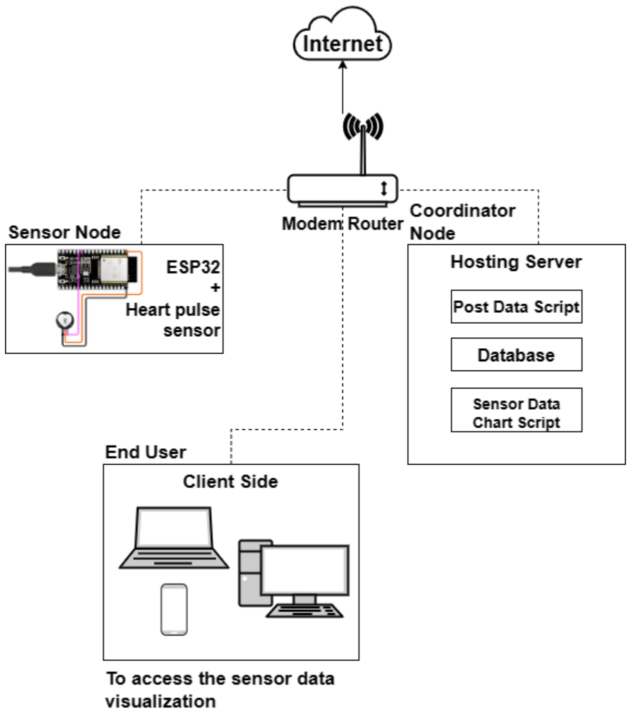


Figure 1. Block diagram of the system

According to the block diagram, the system uses two algorithms. The first algorithm is to send the data from the sensor node to the coordinator node. Further, the coordinator node receives sensor data and stores it in the database to be visualized in the next step. Figure 2 shows the flowchart of the system.

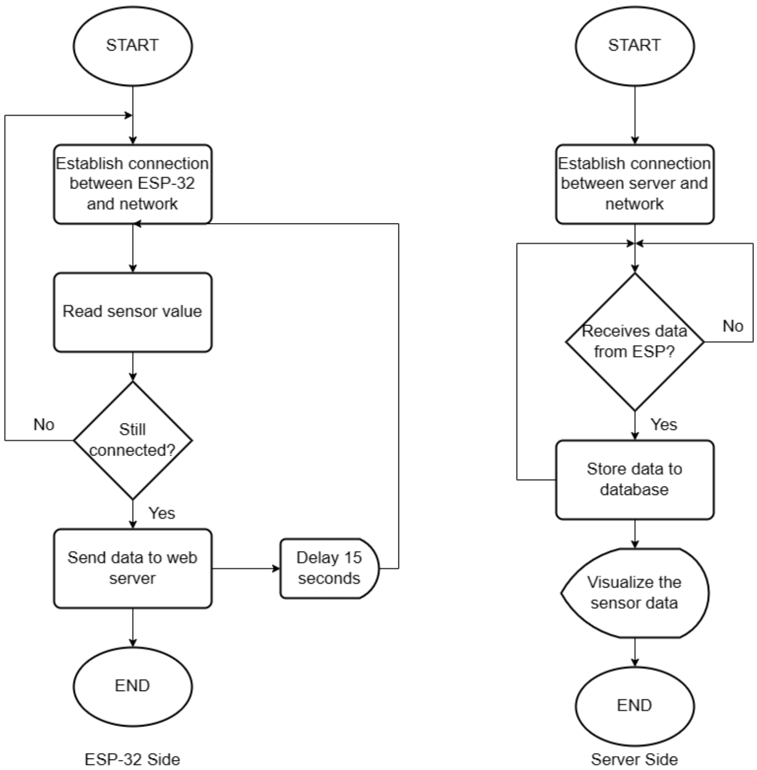


Figure 2. Flowchart of the system

### 3.2. System Algorithms

In the sensor node, the sensor reads changes in blood flow that occur in the human finger [11]. The blood flow change will be converted into an electric signal unit (in current or voltage). The electric signal will be sent to the ESP-32 microcontroller to be converted into digital bit data. After that, the ESP-32 will convert the digital bit data into a BPM unit. In the actual process, the BPM data needs about 10 seconds to be stable.

Furthermore, the BPM data will be sent to the coordinator node to be stored in the database. This applied method is the GET POST method on HTTP protocol [12]. Table 1 presents the algorithm of the node sensor side:

Table 1. First algorithm of the system

| Algorithm 1: Sending sensor data to the web server | |
|---|---|
| 1. | **while** Sensor read the heartbeat condition **do** |
| 2. |     **print** the sensor data to the serial monitor |
| 3. |     **if** ESP-32 connected to the hotspot **then** |
| 4. |         **send** the sensor data to web server ip address over postdata PHP script |
| 5. |     **else** |
| 6. |         perform reconnect to the hotspot |
| 7. |     **end if** |
| 8. |     **wait** 15 seconds |
| 9. | **end while** |

The coordinator node receives data from the sensor node by using the Post Data script. The script receives the sensor data and then stores it in the database by using the POST method. The database stores the sensor data further can be gotten by the PHP script to be visualized. This algorithm is presented in Table 2.

Table 2. Second algorithm of the system

| Algorithm 2: Web server receive the sensor data | |
|---|---|
| 1. | **while** web server is connected to the hotspot **then** |
| 2. |     **if** the web server receives the data from ESP-32 over postdata PHP script **then** |
| 3. |         **save** the sensor data to the database |
| 4. |     **else** |
| 5. |         **no data** will be saved to the database |
| 6. |     **end if** |
| 7. |     **visualize** the sensor data to the website interface over PHP script |
| 8. | **end while** |

## 4. Result

### 4.1. Hardware Testing

Hardware design involves ESP-32 as a microcontroller that already supported by a Wi-Fi communication module and pulse sensor to read the human heartbeat. In this testing step, the pulse sensor is connected to the ESP-32 over the jumper cable and gets power from Laptop. Figure 3 is the part of the hardware used.

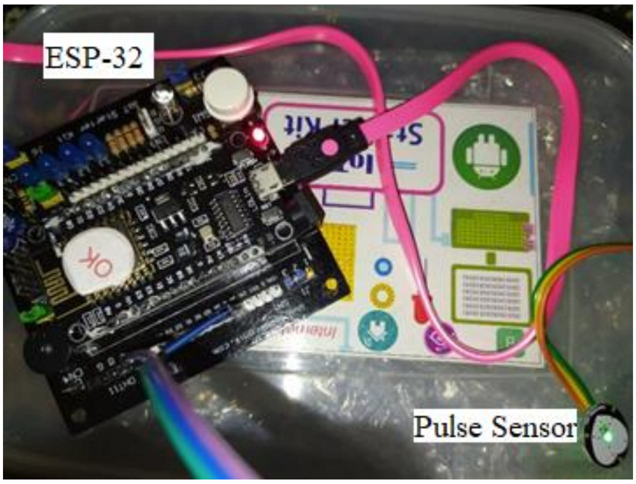

Figure 3. Hardware of the system

In order to make sure that the sensor works well, firstly, the sensor was tested by placing the connected pulse sensor to the thumb. As depicted in figure 4, a pulse sensor is placed on the thumb to read the blood flow that occurs in the human's thumb, then the sensor can generate the heartbeat value based on the sensor reading. The pulse sensor has a working principle of detecting the signal changes that resulting from blood flowing continuously on the human finger by using an infrared source. The signal changes produce an output of analog signal that converted into digital bit by ESP-32 [13].

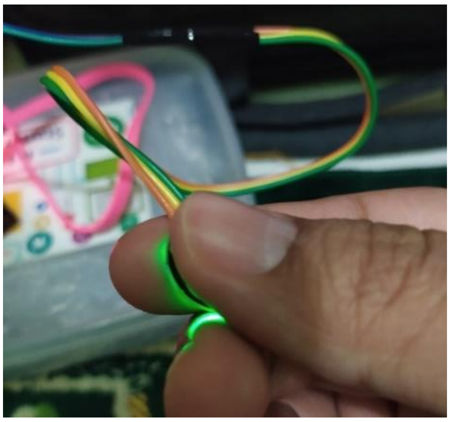
Figure 4. Sensor testing

In order to read the sensor value, a Serial Plotter provided by Arduino IDE was used, as shown in Figure 5. As a result, it shows that the sensor can read the heartbeat.

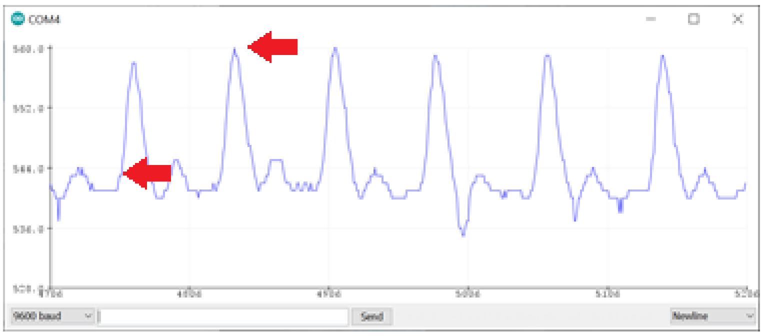

Figure 5. Serial plotter on tested pulse sensor

Figure 5 shows that there is a threshold value of 550. The threshold value is defined before the sensor testing, then plot the sensor value to see whether the threshold value is suitable and acceptable to read the pulse. To get the suitable threshold value, the author ran a default Arduino code for the pulse sensor provided by Arduino IDE to see the graph, as depicted in Figure 5. The suitable threshold value is shown from the feedback value of the low point (red arrow) that shows the heartbeat reading by the pulse sensor on that point. Based on that testing, the average value is 550, and it is used as a threshold value.

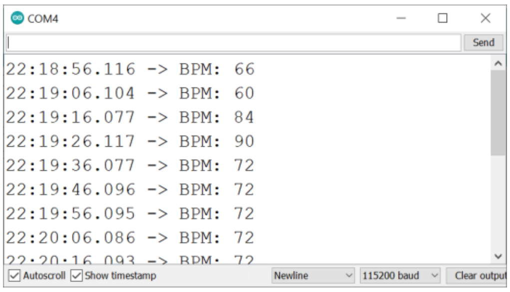

Figure 6. Serial monitor values of tested pulse sensor

After obtaining the threshold value, the sensor was tested by reading the heartbeat in the BPM unit. In this program, 550 was used as threshold value, and the sensor can read the heartbeat changes. In this testing, the pulse sensor reads every 10 seconds, and the data was calculated to obtain the BPM value. This testing involves a Serial Monitor to observe the result from the sensor reading.

Figure 6 shows in the first stage, the sensor needs at least 60 seconds to read the heartbeat steadily until it shows 72 BPM as a stable sensor reading. While testing the sensor, the 72 BPM value was matched to the manual calculation for one minute.

## 4.2. Sensor Values Comparison

In order to determine the sensor's accuracy, the heartbeat value from the sensor was compared with the heartbeat value from the smartwatch. A smartwatch from Xiaomi Mi Band 5 was used, which includes a heartbeat sensor. The test was conducted ten times with the time of each test is not sequential (pause on testing is about 2-5 minutes). Only the fixed value of heartbeat from the smartwatch (in this study is used 72 BPM) was taken, so the accuracy and error value can be obtained well. Table 2 presents the result of the sensor value comparison.

Table 3. Heartbeat values comparison between applied sensor and smartwatch

| Testing | Sensor Value | Smartwatch Value | Accuracy % | Error % |
|---|---|---|---|---|
| 1 | 72 | 72 | 100 | 0.00 |
| 2 | 72 | 72 | 100 | 0.00 |
| 3 | 72 | 72 | 100 | 0.00 |
| 4 | 71 | 72 | 98.61 | 1.39 |
| 5 | 70 | 72 | 97.22 | 2.78 |
| 6 | 70 | 72 | 97.22 | 2.78 |
| 7 | 71 | 72 | 98.61 | 1.39 |
| 8 | 70 | 72 | 97.22 | 2.78 |
| 9 | 72 | 72 | 100 | 0.00 |
| 10 | 72 | 72 | 100 | 0.00 |
| | | Average Value | 98.89 | 1.11 |

### 4.3. Simple Visualization Result

After the sensor was successfully tested, it means that it can send its data to the coordinator node. The web service package (XAMPP web service) has already been installed in the coordinator node, so that the coordinator node can receive the sensor data by using the POST and GET methods on the web service. HTML and CSS are used to make a simple visualization of the sensor data. Also, the database has already been included in the XAMPP web service application by using MariaDB. Figure 7 shows that the sensor data has already been received by the web server and stored in the database.

After the sensor data was received and stored in the database, the sensor data was visualized by using HTML and CSS script to show the received sensor data by the coordinator node. A simple line plot visualization was implemented to show the sensor data changes at a specific time. Figure 8 is the result of a simple visualization of the sensor data.

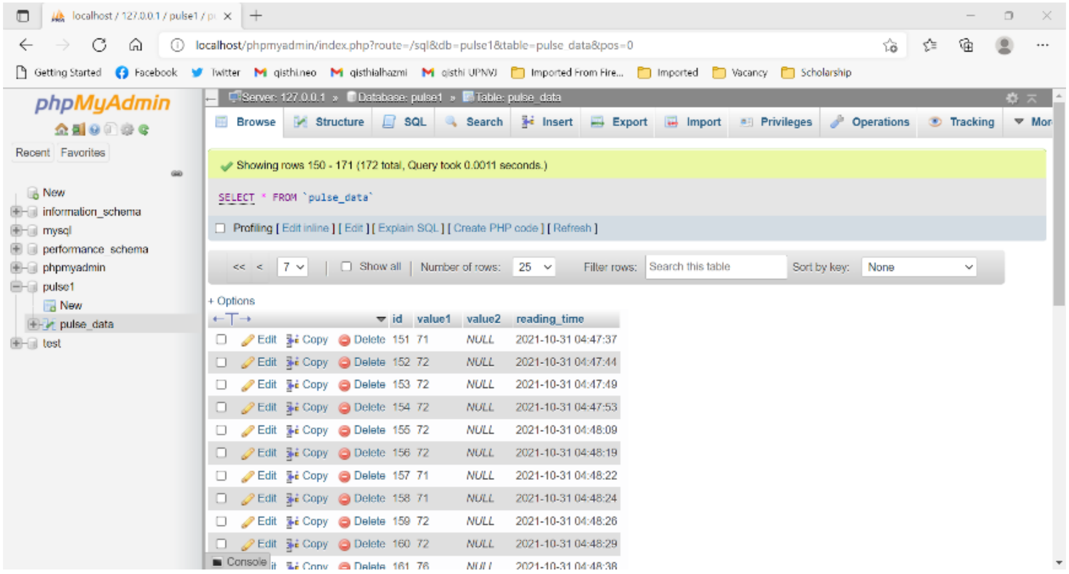


Figure 7. Database stored the sensor data

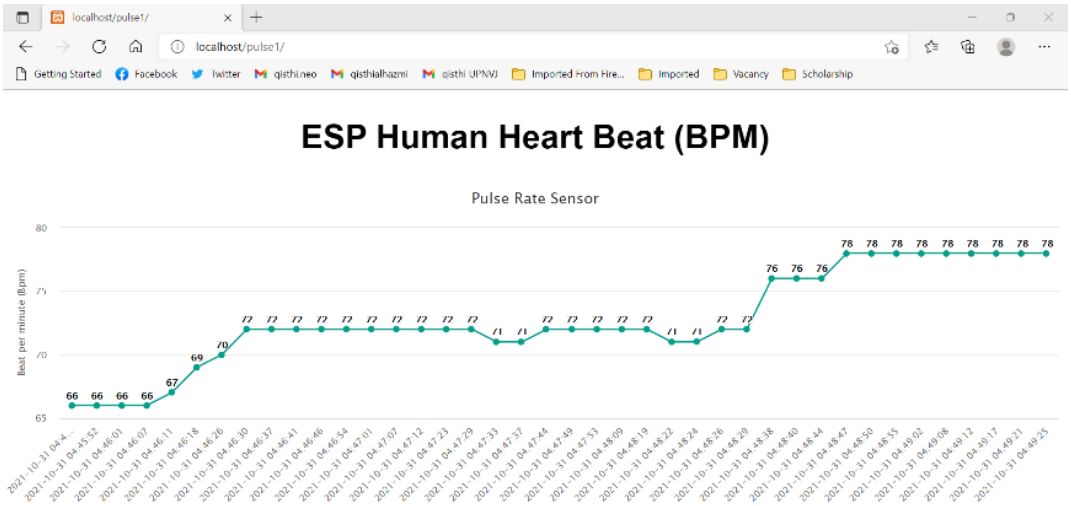


Figure 8. Simple visualization result

## 5. Conclusion

The ESP-32 microcontroller could be used as a sensor node in wireless sensor network applications, especially in medical applications. The pulse sensor used in this study allows the reading of human pulse or heartbeat with 1.11 persen error percentage and 98.89 persen accuracy compared to the heartbeat values in the smartwatch. The ESP-32 microcontroller could be used to send sensor data over wireless local area networks to the coordinator node. Sensor data that received by the coordinator node could be stored and visualized through the web interface using the XAMPP web service. It can be concluded that the XMPP web server could be used to store and visualize the sensor data through a website interface on a wireless sensor networks application.

## 6. Acknowledgment

The authors would like to thank Faculty of Engineering Universitas Pembangunan Nasional Veteran Jakarta that facilitated author to conduct this study under Riset Dosen Pemula (RDP) research scheme from Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) UPN Veteran Jakarta.

## References

[1] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0," *IEEE industrial electronics magazine*, vol. 11 no. 1, pp. 17-27, 2017.

[2] A. J. Bokolo, "Application of telemedicine and eHealth technology for clinical services in response to COVID-19 pandemic," *Health and Technology*, vol. 11 no. 2, pp. 359-366, 2021.

[3] R. P. Singh, M. Javaid, A. Haleem, and R. Suman, "Internet of things (IoT) applications to fight against Covid-19 pandemic," *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*, vol. 14 no. 4, pp. 521-524, 2020.

[4] T. Wu, F. Wu, J. M. Redoute, and M. R. Yuce, "An autonomous wireless body area network implementation towards IoT connected healthcare applications," *IEEE access*, vol. 5, pp. 11413-11422, 2017.

[5] E. Guillén, J. Sánchez, and L. R. López, "IoT protocol model on healthcare monitoring," *VII Latin American Congress on Biomedical Engineering CLAIB 2016*, Bucaramanga, Santander, Colombia, October 26th-28th, 2016 (pp. 193-196), Springer, Singapore, 2017.

[6] Q. A. H. H. Rohman and N. S. Salahuddin, "Rancang bangun prototipe mobil penjelajah dengan kendali jarak jauh melalui jaringan Wi-Fi berbasis antarmuka web," Teknika, vol. 7 no. 1, pp. 1-7, 2018.

[7] Q. Zhang, "Web-based medical data visualization and information sharing towards application in distributed diagnosis," *Informatics in Medicine Unlocked*, vol. 14, pp. 69-81, 2019.

[8] Q. A. Hidayaturrohman, P. Yuliantoro, and R. Rizal, "Visualisasi data sensor pada sistem pemantauan suhu dan kelembaban ruangan berbasis antarmuka web," *Bina Teknika*, vol. 16 no. 2, pp. 73-77, 2021.

[9] L. Shkurti, X. Bajrami, E. Canhasi, B. Limani, S. Krrabaj, and A. Hulaj, "Development of ambient environmental monitoring system through wireless sensor network (WSN)

using NodeMCU and "WSN monitoring"," *6th Mediterranean Conference on Embedded Computing (MECO)*, pp. 1-5, IEEE, 2017.

[10]M. Islam and A. Rahaman, "Development of smart healthcare monitoring system in IoT environment," *SN computer science*, vol. 1 no. 3, pp. 1-11, 2020.

[11]H. Kemis, N. Bruce, W. Ping, T. Antonio, L. B. Gook, and H. J. Lee, "Healthcare monitoring application in ubiquitous sensor network: Design and implementation based on pulse sensor with Arduino," *6th International Conference on New Trends in Information Science, Service Science and Data Mining (ISSDM2012)*, pp. 34-38, IEEE, 2012.

[12]D. Glaroudis, A. Iossifides, and P. Chatzimisios, "Survey, comparison and research challenges of IoT application protocols for smart farming," *Computer Networks*, vol. 168, pp. 107037, 2020.

[13]A. S. Khalaf, "Remote heart rate monitor system using NodeMCU microcontroller and easy pulse sensor v1. 1," *IOP Conference Series: Materials Science and Engineering*, vol. 518, no. 5, pp. 052016, IOP Publishing, 2019.