*Advances in Engineering Research, volume 209*
**International Joint Conference on Science and Engineering 2021 (IJCSE 2021)**

# Monitoring System of Self-Isolation COVID-19 Patient Based on Internet of Things

Arif Widodo [1,*] Zain Bahaul Anwar [2]

[1] *Department of Electrical Engineering, Universitas Negeri Surabaya, Surabaya, Indonesia*
[2] *Student of Electrical Engineering Department, Universitas Negeri Surabaya, Surabaya, Indonesia*
[*] *Corresponding author. Email:* arifwidodo@unesa.ac.id

**ABSTRACT**

The emergence of Coronavirus Disease 2019 (COVID-19) has made many people affected by this disease, making hospital facilities limited as a place of isolation, ranging from mild symptoms to asymptomatic ones such as Happy Hypoxia (severe pneumonia) causing oxygen saturation levels, heart rate, and temperature to increase. Abnormal changes will occur, this makes the government recommend that those with mild symptoms to self-isolate. So that this study creates a system that can monitor patients who are self-isolating so that the hospital burden does not increase. By using the ESP32 microcontroller as an IoT (Internet of Things) device, data such as SpO2 (Oxygen Saturation), heart rate, and body temperature as monitoring parameters can be sent to the ThingSpeak server. In this study, the measurement results obtained from 10 subjects were measured. The SpO2 value reading error is obtained from the lowest 0.11% and the highest 1.20%. Then the lowest reading heart rate error is 0.89% and the highest is 1.54%. Then the body temperature value obtained the lowest reading error of 0.19% and the highest 2.78%. And the interval of sending data from the smartphone to the ThingSpeak server is <20 seconds. So that this system can be used properly by patients who are self-isolating because they have a low SpO2 error, not exceeding the standard SpO2 accuracy of 4%.

***Keywords:*** *ESP32, IoT ThingSpeak, MAX30102, DS18B20, Bluetooth.*

## 1. INTRODUCTION

In 2019 a new disease that has never been identified by humans emerged, which was named Coronavirus Disease 2019 (COVID-19). For clinical symptoms of COVID-19 sufferers, most cases have difficulty breathing and X-ray results of extensive pneumonia infiltrates in both lungs [1]. With the trend of positive cases of COVID-19 in Indonesia continuing to increase over time and limited hospital facilities for isolation, patients with SWAB test results show positive results with mild or even asymptomatic symptoms such as Happy Hypoxia, it is recommended to self-isolate at home - respectively. This can reduce the burden on the hospital which is increasing every day for isolated COVID-19 patients.

With this, hospitals need to have a system that can connect with patients who are self-isolating. Several journals have conducted research to conduct remote monitoring. Several studies use ESP8266 as wireless [2,3]. Many of these studies have been developed, because the ESP8266 itself is very familiar with being used in IoT research.

In other studies obtained using Zigbee [4,5], Wi-Fi Module Roving RN-Xvee [6], BlueSMiRF module [7], research from Sakshi D. Ambadkar, Shobha S. Nikam., using nRF24L01 as a separator between sensors (RX) and the receiving part (TX) which contains the SIM900 module as an SMS sender to send information [8]. Some studies also still use a local network system to send information, such as what Sugondo Hadiyoso, Achmad Rizal, and Rita Magdalena did using a LAN network (802.11b) as a data sender from sensors to the user's laptop and obtained a range of 70 meters [9].

But in order to maximize the condition of patients who are in isolation, apart from sending them to the hospital, the family also needs to know without direct contact with the patient. By utilizing an Android smartphone, it can be used as monitoring by the family. For research that uses an Android smartphone for its interface, to make the interface using Android Studio by transmitting data from the device using the HC-06 Bluetooth module [10], then other research uses the HC-05 module with an interface on an Android smartphone [11]. Of the two delivery systems, namely paralleling between sending data to an android smartphone and to a server/cloud.

For this reason, in this research activity, we will do how to detect HeartRate, blood oxygen saturation levels (SPO2), and patient body temperature, using ESP32 as a microcontroller and Bluetooth which later on the data read will be displayed via an android smartphone which will then be forwarded to ThingSpeak server for remote monitoring.

So that later this tool can be used by families to find out the patient's condition without direct contact and can be monitored by medical personnel from the ThingSpeak server side as remote monitoring, this can reduce the burden on the hospital so that patients with mild symptoms can self-isolate and of course remain in the hospital. monitored by medical personnel.

## 2. PULSE OXIMETRY

Oxygen flows through the circulatory system in the body. The deoxygenated blood goes to the heart and then the blood is pumped to the lungs to be oxygenated. The process of oxygenation, blood passes through the alveoli of the lungs where gas exchange (diffusion) occurs. So that carbon dioxide ($CO_2$) is released so that the blood is oxygenated, after which the blood will be pumped back into the aorta.

Pulse oximeter itself is a non-invasive way of measuring the level of oxygen saturation in the body ($SpO_2$). This measurement is based on the detection of Hemoglobin and Deoxyhemoglobin dissolved in the blood, it is necessary to use two wavelengths of light in this measurement, namely 660nm and 940nm. Figure 1 provides a graph of light absorption by Hb and $HbO_2$.

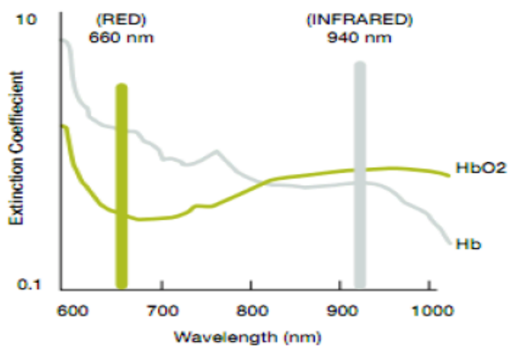


**Figure 1** Graph of light absorption by Hb and $HbO_2$ [12, p.4].

Deoxygenated hemoglobin (Hb) has a higher absorption at 660nm and oxygenated hemoglobin ($HbO_2$) has a higher absorption at 940nm. Then on the receiving side of the reading there is a photodetector which later the reading signal will be amplified by an operational amplifier (OpAmp) until it can be read, then the signal will be divided into two, namely, the AC signal component and the DC signal component, which can be seen in Figure 2. DC component the result of absorption of light from skin tissue (nails, muscles, bones), venous blood, and arterial blood is not pulsed. The AC component results in pulsating arterial blood.

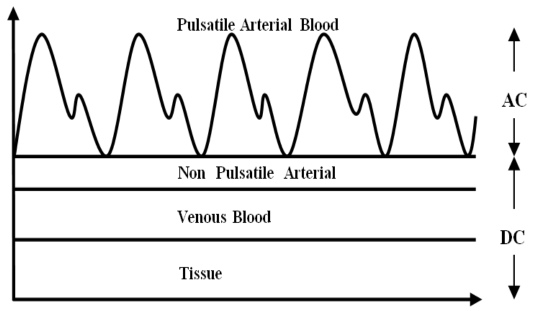


**Figure 2** Light absorption by body tissue [12, p.4].

The Pulse Oximeter analyzes the light absorption of two wavelengths of LED (Light Emitting Diode) from an additional volume of oxygenated pulsed arterial blood (AC/DC) and calculates the absorption ratio (1) [12]. Equation 2 is used to get the ratio value from the above equation, it can be calculated for the $SpO_2$[13] value.

$$R = \frac{(AC_{red}/DC_{red})}{(AC_{ir}/DC_{ir})} \quad (1)$$

$$SpO_2 = 110 - 25 \cdot R \quad (2)$$

In practice, there are two ways to place one on the finger, illustrated in Figure 3. By transmitting a beam of light, then a reflection will be obtained, where some of the light that has been absorbed will then be captured by the receiver, this method can be seen in Figure (a). Then for the second method by transmitting light on the top of the finger, this makes the light penetrate the finger and some will be absorbed by the body so that the rest of the light will be received by the receiver on the other side, this method can be seen in Figure (b).

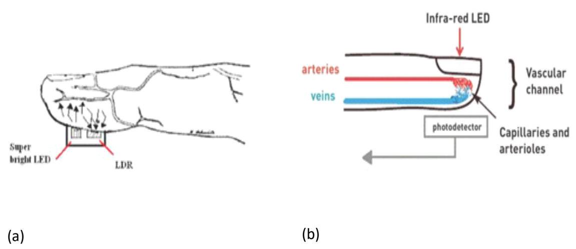


**Figure 3** Light transmit method [10, p.2].

## 3. RESEARCH METHODOLOGY

### *3.1. SpO2 Measurement*

In getting the $SpO_2$ value, using the MAX30201 module. In the circuit it is necessary to add a pull-up resistor on the $I_2C$ line of 1k5$\Omega$. Then from the programming side by using the Arduino IDE, by utilizing the existing libraries to run the module. The value of the red LED and IR obtained will be calculated using the formula applied to the programming code(3).

$$R = \frac{(\sqrt{(sum_{red})} \,/\, ave_{red})}{(\sqrt{(sum_{ir})} \,/\, ave_{ir})} \qquad (3)$$

Equation 4 is the result of the calibrated SpO2 value.

$$SpO_2 = -23.3 * (R - 0.4) + 100; \qquad (4)$$

### *3.2. HeartRate Measurement*

In practice, this heart rate measurement only requires one value from the LED. By using one of these values, we get the time value between each beat. To get the time in the program code, use the micros time function to get a better value. Then the time obtained is converted to the value of BPM(5).

$$BPM = 59/\left(\frac{time\ interval}{1000000.0}\right) \qquad (5)$$

After obtaining the BPM value with the above calculation, the value will be stored first in a variable. BPM value obtained will be summed with the values obtained further, the data as much as 20 BPM. After that, the average value of the 20 data is calculated so that a stable BPM value will be obtained.

### *3.3. Body Temperature Measurement*

To get the body temperature value, the DS18B20 sensor is used as a measuring tool. This sensor requires a pull-up resistor of 4k7$\Omega$ installed on the data and voltage lines. Then for the programming code on the Arduino IDE the data conversion resolution is set at 12bit.

By using the Zotek ZT-C4 multimeter which has a temperature measurement, calibration calculations are carried out using the help of Microsoft Excel. Utilizing existing functions, namely using add trendline in order to obtain a linear value and its equation, here are the results of the calculation:

### *3.4. Sending Data to the Server*

Data from the sensor will be sent to the Smartphone via Bluetooth, where the application that has been created has been installed. Data will be sent serially, with the arrangement of $SpO_2$, HeartRate, Temperature with a separator between values using the symbol "|" (data parsing). And the sending interval to the smartphone is 100ms.

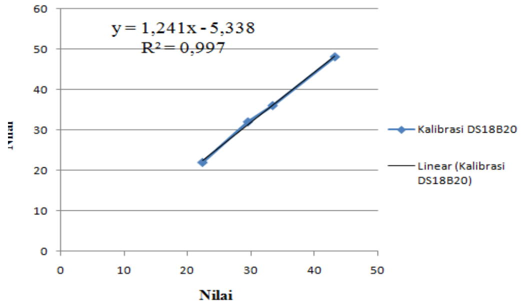


**Figure 4** DS18B20 calibration results.

To connect the Smartphone to the server, an Application Programming Interface (API) is needed to connect to the server. By copying the API that is on the ThingSpeak server section and then entering it in the compiler section of the application that is made, here is the section for sending the server.

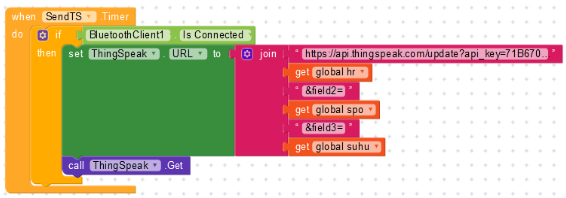


**Figure 5** Command sending to server with API.

### *3.5. Monitoring System Work*

The following is a block diagram of the monitoring system circuit made in Figure 6, the two sensors, namely the MAX30102 and DS18B20, will read the value controlled by the ESP3 microcontroller if a finger is placed on the MAX30102 sensor. After that, the data will be sent to the SmartPhone application via the Bluetooth network so that it can be displayed on the SmartPhone screen. Then the data received will be posted to the ThingSpeak cloud as a remote monitor so that it can be seen by health workers. So that the internet network on the SmartPhone will have an important role in the process of sending data to the server. In this study using a smartphone that has a 4G network.

## 4. MEASUREMENT SETUP

In the measurement, some settings will be made for measurement. In general, measurements will take 10 people with different age ranges to obtain $SpO_2$ and body temperature data.

### 4.1. Device Design

In order to make it easier to use and reduce the amount of wiring in the circuit, the circuit is made on a PCB board that has been designed in such a way.

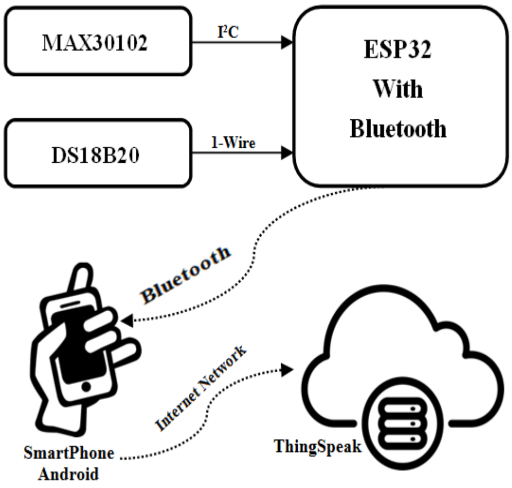


**Figure 6** System work block diagram.

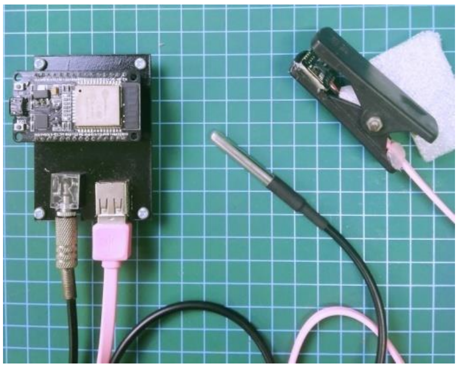

**Figure 7** Prototype design.

In the MAX30102 sensor section, the wiring uses a Micro USB cable whose ends are connected directly using only the USB type A section as the connector. For the DS18B20 sensor using a 3.5mm audio jack connector, the arrangement of the ends of the connector is voltage, data, ground.

### 4.2. SpO2 and HeartRate Measurement

Then for the method of reading the SpO2 and HeartRate parameters by placing the MAX30201 sensor and comparison tool on the subject's finger. Then the reading is done by waiting a few minutes to get a stable value. When measuring, the subject must be still, not moving a lot in order to get a good value.

### 4.3. Body Temperature Measurement

For data collection, the sensor will be placed on the body, namely in the folds of the arm, where the part is usually used to measure body temperature. So the DS18B20 sensor and multimeter temperature sensor will be put together to get readings from both.

### 4.4. Comparison of Value Obtained

To determine whether the tools made are the same as standardized measuring instruments, an error calculation is carried out with measuring instruments that are already on the market. Using Serenity Pulse Oximetry SP090 as a comparison and looking for reading errors. Serinity SP090 itself has been standardized by the Indonesian Ministry of Health with the code of KEMENKES RI AKL: 20502914772.

To provide the level of accuracy you want to get, it is necessary to standardize the accuracy of a tool. By the P2 SMTP LIPI institution, which according to ISO 80601-2-61 the recommended $SpO_2$ accuracy is 4% in the range of 70% - 100%[14]. To find the percentage error value between values created with Serenity SP090, use the relative error formula (6).

$$Error\ (\%) = \left| \frac{(\text{Value Serenity/Zotek} - \text{Value Device})}{\text{Value Serenity/Zotek}} \right| * 100\% \quad 6)$$

### 4.5. Display on SmarthPhone

In the smartphone display itself, an application is made that can receive data from devices via Bluetooth on the ESP32 MCU. When the Bluetooth smartphone and Bluetooth device are connected, the data will be displayed on the smartphone screen and will be posted directly on ThingSpeak using the smartphone internet network. In this case the application is made via the Kodular.io web, then it can be installed on an android smartphone with a file in the form of .apk.

### 4.6. ThingSpeak Server

The ThingSpeak section itself, the data received from the smartphone will be turned into a graph every time. With ThingSpeak, it is easier to read the received data, and it can also be downloaded to recap the data in CSV form. The following display has been created which contains some subject information.

## 5. RESULTS

### 5.1. Reading Result SpO2, Heart Rate and Body Temperature

The following are the measurement results obtained from 10 subjects with different ages. Figure 10 will present $SpO_2$ data from 10 measured subjects. Comparison of data was obtained between $SpO_2$ from the serenity tool and $SpO_2$ from the research tool. Then look for the relative error, where the smallest value is 0.11% and the highest is 1.20%.

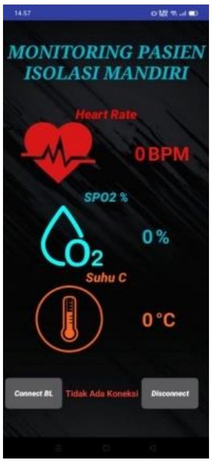


**Figure 8** Interface on smartphone.

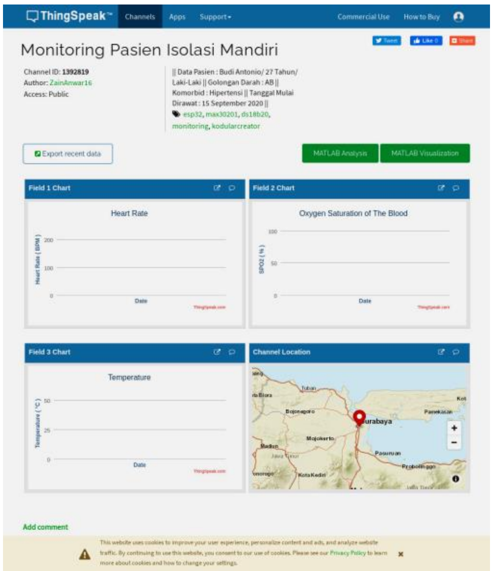


**Figure 9** ThingSpeak server section view.

Figure 11 presents the results of the heart rate readings of the 10 subjects that have been measured. Comparison of data was obtained between the heartrate from the serenity tool and the heartrate from the research tool. Then look for the relative error, where the smallest value is 0.89% and the highest is 1.54%.

Figure 12 presents the results of the body temperature readings of 10 subjects who have been measured. Comparison of data was obtained between body temperature from the Zotek ZT-C4 device and body temperature from the research instrument. Then look for the relative error, where the smallest value is 0.19% and the highest is 2.78%. The DS18B20 sensor is less sensitive because the area that should be subjected to temperature is too large, while the comparison used uses a thermocouple which has a fast data change rate.

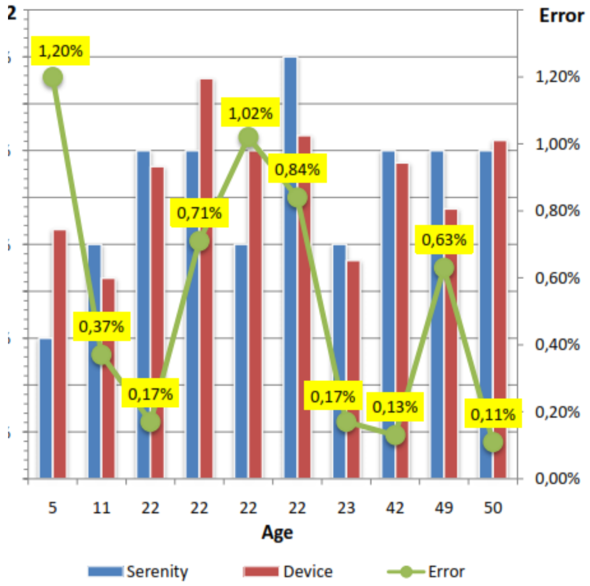


**Figure 10** $SpO_2$ measurement results.

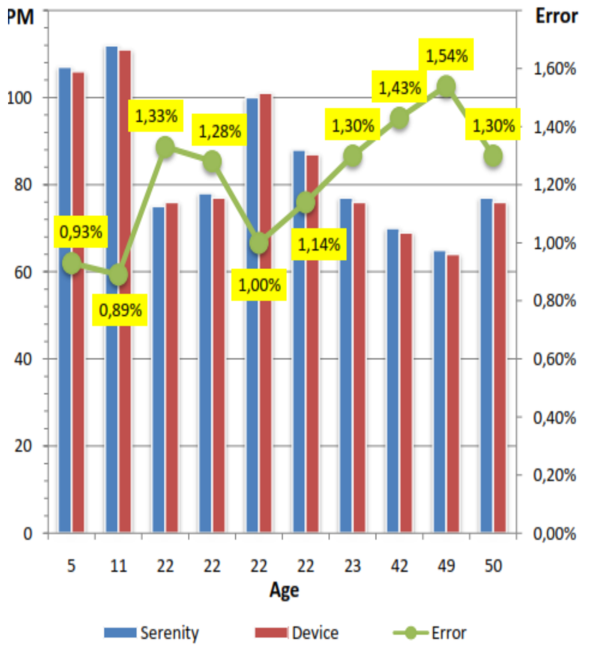


**Figure 11** Heartrate measurement results.

### *5.2. Bluetooth dan ThingSpeak Result*

To display the results of sending data to a smartphone using Bluetooth, it can be seen in Figure 13. On the smartphone data display, the data will change according to the programming code that has been created, which is every 100 milliseconds. For display in the form of images and values sent from the microcontroller.

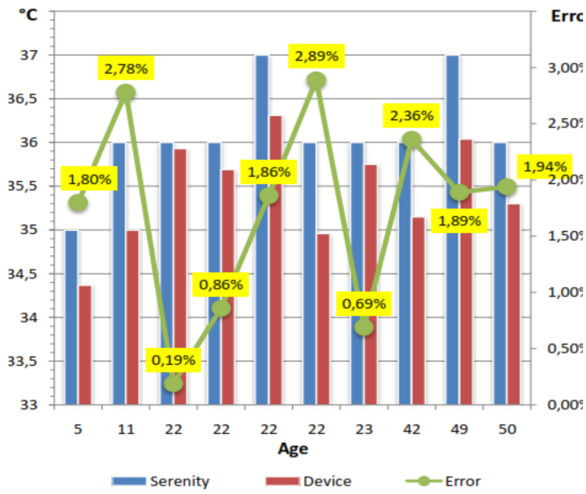


**Figure 12** Body temperature measurement results.

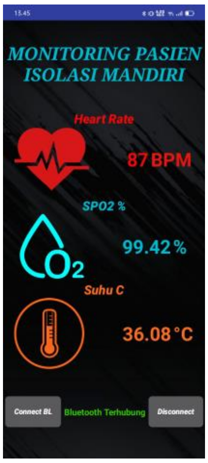


**Figure 13** Smartphone display when used.

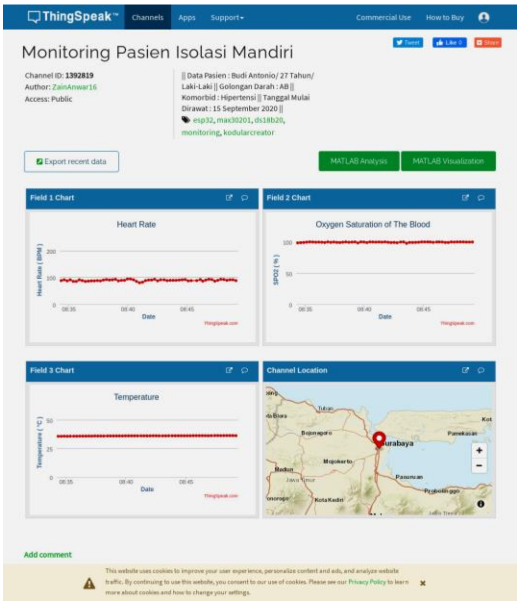


**Figure 14** Web server display.

The results of sending data from the smartphone to the web server are presented in Figure 14. Where the data will be displayed in the form of a red line graph. The map display can be changed according to the position of the patient who is self-isolating.

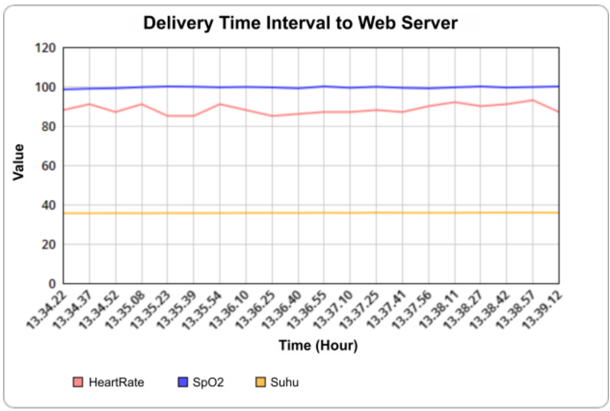


**Figure 15** Delivery interval to server.

While Figure 15 provides an overview of the time interval for sending data to the server. By downloading part of the CSV file, you get a time interval of < 20 seconds. This is influenced by the fast or slow internet network on the smartphone that is used.

## 6. CONCLUSION

From the research conducted, several results can be drawn, some conclusions can be drawn. The error results obtained for the $SpO_2$ measurement start from the lowest value of 0.11% and the highest value of 1.20%. The error result for heart rate measurement is the lowest value is 0.89% and the highest is 1.54%. The results for the error obtained by measuring body temperature are the lowest

0.19% and the highest 2.78%. The results for sending data from a smartphone to the ThingSpeak server are obtained at an interval of < 20 seconds, this depends on the internet network on the smartphone, and the data can be downloaded in CSV form. This system can be used properly by patients who are self-isolating because they have a low $SpO_2$ error, not exceeding the standard $SpO_2$ accuracy of 4.

## REFERENCES


[1] Pedoman kesiapsiagaan menghadapi infeksi novel corona virus (2019-nCoV), infeksiemerging.kemkes.go.id, 2020. [Online]. Available: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwj0776kj_jwAhW68HMBHZGTA_kQFjAKegQIFBAD&url=https%3A%2F%2Finfeksiemerging.kemkes.go.id%2Fdownload%2FDOKUMEN_RESMI_Pedoman_Kesiapsiagaan_nCoV_Indonesia_28_Jan_2020.pdf&usg=AOvVaw3mPbZzEbO3G.

[2] M. Niswar, M. Nur, and I. Mappangara, A Low Cost Wearable Medical Device for Vital Signs Monitoring in Low-Resource Settings, Int. J. Electr. Comput. Eng. IJECE, vol. 9, no. 4, p. 2321, Aug. 2019, doi: 10.11591/ijece.v9i4.pp2321-2327.

[3] S. Marathe, D. Zeeshan, T. Thomas, and S. Vidhya, A Wireless Patient Monitoring System using Integrated ECG module, Pulse Oximeter, Blood Pressure and Temperature Sensor, in 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), Vellore, India, Mar. 2019, pp. 1–4. doi: 10.1109/ViTECoN.2019.8899541.

[4] H. Cao, Mémoire présenté à l'université du québec à chicoutimi comme exigence partielle de la maîtrise en ingénierie, p. 76, 2017.

[5] Hamdala, Y. Rahayu, I. Yasri, Sistem Monitoring Photoplestysmograph Berbasis Zigbee dan Labview, Jom FTEKNIK, vol. 2, no. 2, pp. 1-10, Okt. 2015.

[6] M. Omar, M. Omar, and F. Abdallah, Merging and Designing Biomedical devices with Telecommunication, For Creating Tele Pulse Oximeter, vol. 6, no. 4, p. 6, 2015.

A. Aithal, Wireless Sensor Platform for Pulse Oximetry, p. 75, 2015.

[7] S. D. Ambadkar and S. S. Nikam, NRF Transceiver based Saline Level, Health Monitoring & Control System, vol. 07, no. 10, p. 7, 2020.

[8] S. Hadiyoso and A. Rizal, Monitoring photoplethysmograph digital dengan wireless LAN (802.11b), p. 5, 2011.

[9] R. Sameh, M. Genedy, A. Abdeldayem, and M. H. Abdel azeem, Design and Implementation of an SPO2 Based Sensor for Heart Monitoring Using an Android Application, J. Phys. Conf. Ser., vol. 1447, p. 012004, Jan. 2020, doi: 10.1088/1742-6596/1447/1/012004.

[10] M. M. Ali, S. Haxha, M. M. Alam, C. Nwibor, and M. Sakel, Design of Internet of Things (IoT) and Android Based Low Cost Health Monitoring Embedded System Wearable Sensor for Measuring SpO2, Heart Rate and Body Temperature Simultaneously, Wirel. Pers. Commun., vol. 111, no. 4, pp. 2449–2463, Apr. 2020, doi: 10.1007/s11277-019-06995-7.

[11] S. Lopez, Pulse Oximeter - Fundamentals and Design, Rev. 2, p. 39. Nov, 2012.

[12] L. Umar, I. Firmansyah, and R. N. Setiadi, Design of Pulse Oximetry Based on Photoplethysmography and Beat Rate Signal Using DS-100 Probe Sensor for SpO 2 Measurement, in 2018 3rd International Seminar on Sensors, Instrumentation, Measurement and Metrology (ISSIMM), Depok, Indonesia, Dec. 2018, pp. 25–29. doi: 10.1109/ISSIMM.2018.8727725.

[13] P2 SMTP LIPI. 2016. Jaminan Akurasi Pulse Oximiter Melalui Standar dan Pengujian [Online]. Available: http://www.smtp.lipi.go.id/berita441-Jaminan-Akurasi-Pulse-Oximiter-Melalui-Standar-dan-Pengujian.html