# Heart Rate and Body Temperature Tracking Application Based on Fuzzy Logic

**A N A Thohari[*1], A W Wibowo[2], A Karima[3], I Hestiningsih[4], K Santoso[5], F Abdollah[6]**

[1-4]Informatic Engineering Study Program, Politeknik Negeri Semarang

[5]Computer Engineering Technology Study Program, Politeknik Negeri Semarang

[6]Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka

E-mail: afandi@polines.ac.id[1]

**Submitted: 16 April 2023, revised: 16 June 2023, accepted: 4 August 2023**

**Abstract.** Temperature and heart rate are indicators of health. It is necessary to monitor heart rate and body temperature to prevent the spread of a virus or disease. In most cases, heart rate and body temperature are monitored independently, and the patient cannot view a record of previous examinations. This study uses fuzzy logic to develop an application for monitoring heart rate and body temperature. It can monitor heart rate and body temperature in real-time, store a history of previous examinations, and use fuzzy logic to diagnose body conditions based on heart rate and body temperature data. Based on test results, the sensor reading error rate for heart rate is minimal at 0.68 and for body temperature at 0.18. The accuracy of fuzzy diagnosis of the patient's body condition is one hundred percent. The performance indicator for the application is excellent, the completion rate is 100 percent, and the time-based efficiency is 93%. The results of the user satisfaction test indicate that most users are pleased with the application's usability. The average value for measuring user satisfaction is 80%, with the highest result of the five measurement criteria being 89.6% for the ease-of-use criterion.

**Keywords:** monitoring, heart rate, body temperature, fuzzy logic, usability.

## 1. Introduction

The human body can give notifications if there are viruses, germs, or diseases that enter. One of the parameters measured when viruses and diseases enter the body is heart rate and body temperature [1]. For example, a COVID-19 patient must regularly control his heart rate to determine the amount of oxygen flowing in the blood [2]. Viruses and infections will cause an increase in body temperature, so it is necessary to monitor body temperature frequently.

The temperature of the body will affect the heart rate. Increasing body temperature accelerates the heart rate [3]. This occurs due to a higher metabolic rate, so increasing the oxygen supply and releasing carbon dioxide is necessary. The average human heart rate ranges between 60 and 100 beats per second (BPM). At the same time, the average adult body temperature ranges between 35.8 and 37.5 degrees Celsius [4]. Given the significance of health indicators such as heart rate and body temperature, it is imperative to have an application for monitoring heart rate and body temperature.

Several researchers have developed prototypes for measuring heart rate and body temperature. Using Raspberry Pi and Xbee, [5] monitored web-based heart rate and body temperature. Some develop Arduino and Bluetooth-based monitoring of heart rate and body temperature [6][7][8]. The Internet of Things (IoT) is the most prevalent technology for monitoring heart rate and body temperature. Using NodeMCU and wifi, [9], [10] and [11] monitor heart rate and body temperature using IoT to monitor heart rate and temperature.

This study monitored heart rate and body temperature using a smartphone application for Android. The easy pulse sensor is used to measure heart rate, while the DS18260 sensor is used to measure body temperature. Data from sensor readings are stored in Firebase. In contrast to previous research, the contribution of this study is the processing of heart rate and body temperature data to determine the user's health condition. In data processing, fuzzy logic is utilized. In addition, this research aims to develop a monitoring application with an exemplary user interface and usability, making it simple for users to operate.

## 2. Method

The Android operating system runs the heart rate and body temperature monitoring application. The data originates from sensor readings transmitted over a wifi network by NodeMCU. Figure 1 depicts the flow of prototyping and monitoring applications for heart rate and body temperature.

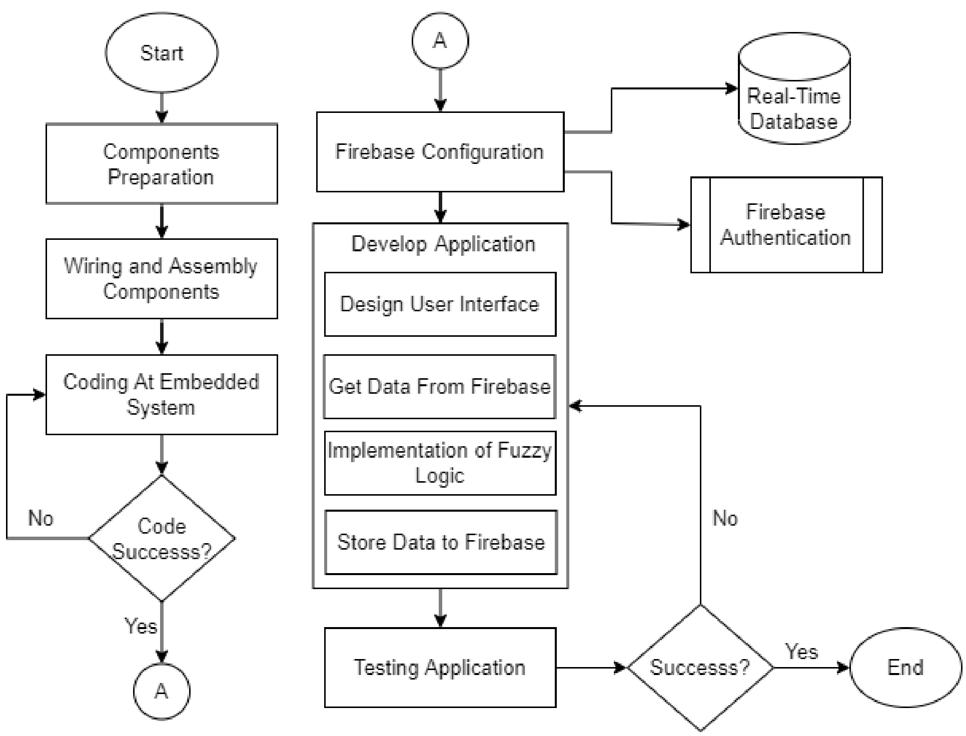


**Figure 1.** Our Proposed Method

### *2.1. Component Preparation*

Creating a prototype for monitoring heart rate and body temperature begins with preparing the necessary components. Six components must be ready for prototyping. Table 1 displays the components utilized and the rationale for their utilization.

**Table 1.** Prototyping Components

| No | Component | Justification |
|---|---|---|
| 1 | NodeMCU | Process and transmit sensor data |
| 2 | Easy Pulse Sensor | Read heart rate |
| 3 | DS18260 Sensor | Read body temperature |
| 4 | LCD 12C 16x2 | Displays the results of sensor data readings |
| 5 | Resistor 10 ohm | Adjust signal level |
| 6 | Jumper Cable | Transporting electric current from a voltage source |

NodeMCU is chosen because it is simple to configure, includes a WiFi module, and has numerous ports for connecting electronic components [12]. The heart rate detection sensor is an easy pulse sensor. The easy pulse sensor is utilized because it provides an accurate heart rate reading. The easy pulse sensor operates on the principle of photoplethysmography (PPG), a non-invasive method for measuring heart rate (cardiovascular) by detecting the volume of blood flow in the pulse close to the skin.

The easy pulse sensor is activated by placing one's finger on the sensor. Then, the finger's pulse affects the passage of light from the IR LED to the photodetector, which is transformed, filtered, and amplified by the sensor module before being processed by the microcontroller. Figure 2 depicts the straightforward pulse sensor data reading procedure.

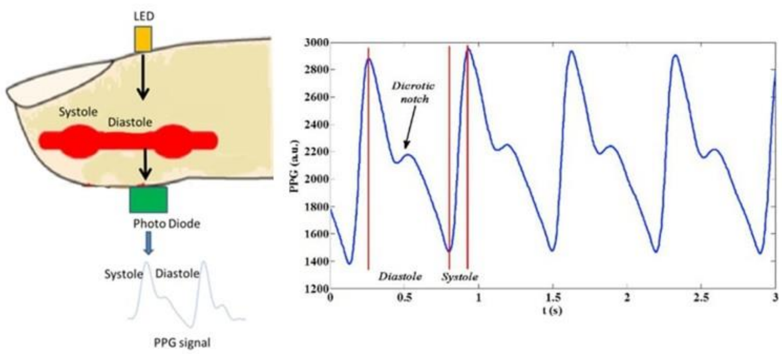


**Figure 2.** Heart Rate Sensor Data Reading [13]

### *2.2. Firebase Configuration*

Google's Firebase is a cloud computing service that provides several functions. Firebase provides automatic authentication and a real-time database as one of its offerings [14]. The constructed monitoring application uses the Firebase authentication capability to obtain the user's email address and password. Data from sensor readings will be recorded in the Firebase real-time database and displayed on the application's interface. The firebase configuration procedure is performed on the console.firebase.com website and additionally configures the Arduino IDE and Android Studio application code.

### *2.3. Wiring Components and Coding Program*

After all, the components have been assembled, connecting each component to the NodeMCU is the next step. Each component is attached to the NodeMCU's ports. LCD I2C is connected to NodeMCU pins D1 and D2, the DS18B20 sensor on the tile is connected to NodeMCU pins D4, and the simple pulse sensor is connected to pin A0. Figure 3 illustrates the component wiring method in further detail.

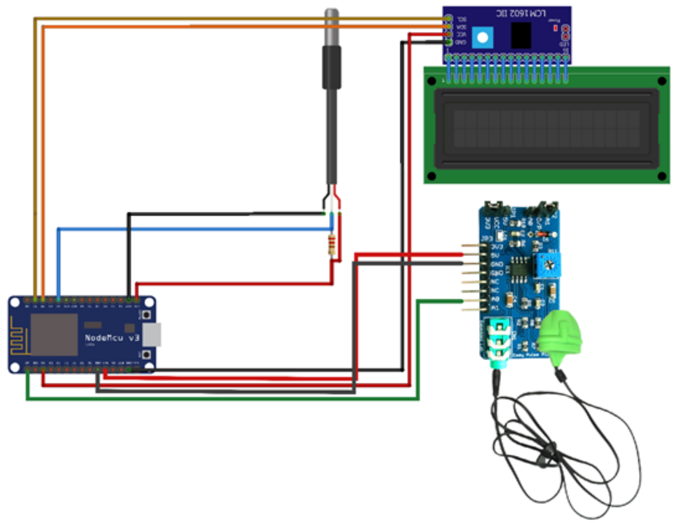

**Figure 3.** Wiring Components

After assembling all the components, the next step is developing the NodeMCU control code. Using the Arduino IDE editor, I am composing program code. The output of the written program code is uploaded to NodeMCU and then evaluated. The serial monitor displays the test results in the form of sensor readings. If the transmitted data is illegible or erroneous, it will be rectified.

*2.4. Develop Mobile Application*

Designing a use case to identify the application's operations and features is the first step in developing an application. Use case diagrams are utilized to determine what application functions exist and who has access to these functions. Figure 4 depicts the use case diagram for the heart rate and body temperature monitoring application.

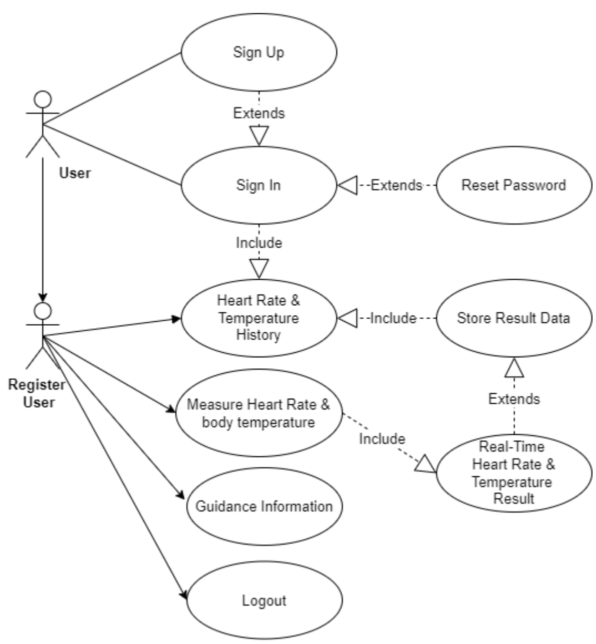


**Figure 4.** Application Use Case Diagram

The use case diagram depicts the workflow of the heart rate and body temperature monitoring application. Figure 4 According to the use case diagram, the user must sign in before entering the application dashboard. If you do not have an account, you can create one on the sign-up page. After successfully logging in, the user will be directed to the application dashboard. On the application's dashboard screen, the user can view the patient's heart rate and body temperature data from the past.

Registered users can access all of the application's capabilities, including measuring their heart rate and body temperature and browsing the user guide. The timer will appear when the measurement button is pressed. The measurement data will be displayed based on the patient's physical condition. The data displayed in the application and on Firebase are the same. After the user presses the stop button, the timer stops, and a new page featuring graphs of heart rate and body temperature reading data and the fuzzy-processed results of the diagnosis of heart problems and body temperature appears. Data collected from heart rate and body temperature measurements are kept in Firebase and displayed as history on the dashboard page.

After creating the application's features using use case diagrams. In addition, the development of a mobile application involves the following procedures:

a. Design User Interface
   Creating user interfaces that increase usability for the end user. The design is created in Figma and then implemented in Android Studio using XML code.
b. Get Data from Firebase
   NodeMCU transmits sensor-reading data to Firebase. Firebase stores sensor data in a real-time database. In addition, the monitoring application will retrieve data from the real-time database for graphical representation.
c. Implementation of Fuzzy Logic
   Fuzzy logic is fuzzy logic because this fuzzy set has imprecise boundaries and is not represented in terms of true or false logic but rather in terms of degrees of membership [15]. Figure 5 is a graph illustrating the membership function of heart rate. Included in the collection of heart rate data are heart rate values for adults older than 15 years. As indicated in Table 2, there are three conditions, namely slow, normal, and fast BPM levels.

**Table 2.** Heart Rate Input

| Condition | Heart Rate (BPM) |
|---|---|
| Slow | 0 - 60 |
| Normal | 50 - 110 |
| Fast | 100 - 200 |

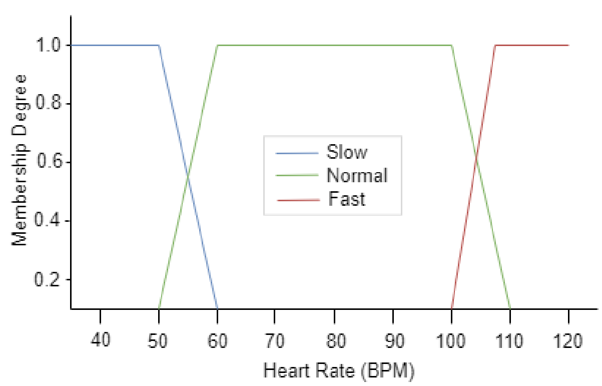


**Figure 5.** Heartbeat Membership Function

While the membership function of body temperature is depicted in Figure 6's graph, Cold, normal, and high body temperature measurements are utilized. Table 3 displays the values of every data criterion. Based on the sensor readings, the resulting temperature data may vary.

**Table 3.** Body Temperature Condition

| Condition | Temperature (C) |
|---|---|
| Cold | 20 – 28 |
| Normal | 25 – 37,5 |
| Hot | 38 - 40 |

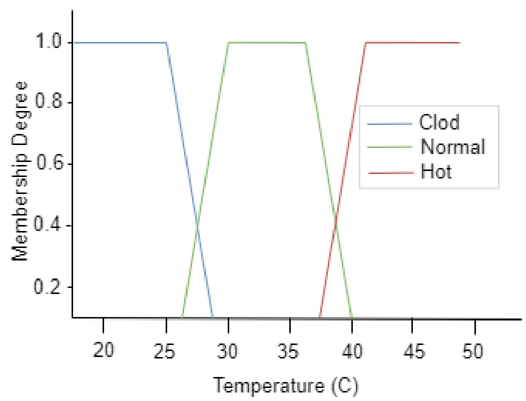


**Figure 6.** Body Temperature Membership Function

d. Store Data to Firebase
   Based on sensor data that has been analyzed by fuzzy logic, decisions regarding the patient's status are made. After the user enters the patient's name, gender, and age, decisions regarding the patient's health state will display. The diagnostic data for the application is then sent back to Firebase.

*2.5. Testing Application*

Several tests are undertaken to determine the application's effectiveness and efficiency. Among the application tests performed on the monitoring application are the following:

1. Test of Measurement Accuracy
   This test evaluates the precision of sensor readings. To determine the accuracy of sensor values, a comparison is made using commercially available heart rate measuring instruments, such as an oximeter. The standard error is used to calculate sensor data readings' error rate [16]. Before calculating the value of the standard error, the standard deviation is computed.

$$Sd = \sqrt{\frac{\sum(y - y')^2}{n - 1}} \quad (1)$$

$$Se = \frac{Sd}{\sqrt{n}} \quad (2)$$

   Standard Deviation is computed by subtracting the sensor readings from the oximeter readings, as shown in Equation (1). The Standard Deviation value is recalculated to derive the Standard Error formula displayed in equation (2).

2. Evaluation of Performance
   This measurement seeks to determine the application's level of efficiency. Measuring the extent to which a work has been completed enables the calculation of efficiency (task success). The equation can be used to express effectiveness as a percentage (3) [17].

$$E = \frac{\sum_{j=1}^{R}\sum_{i=1}^{N} n_{ij}}{RN} \; x\ 100\% \tag{3}$$

   $E$ is the task completion rate, while $N$ is the number of tasks. The $R$ symbol represents the total number of participants. Lastly, $n_{ij}$ represents the outcome of the participant j's performance on task $i$ $n_{ij}$ will be valued as 1 or 0. In addition to measuring the application's efficacy, the application's efficiency is also measured. Task duration serves as a metric for measuring productivity. Task time is the amount of time (in seconds or minutes) it takes the user to successfully perform a task. Equation demonstrates calculating the application efficiency level (4) [18].

$$P = \frac{\sum_{j=1}^{R} \cdot \sum_{i=1}^{N} n_{ij} t_{ij}}{\sum_{j=1}^{R}\sum_{i=1}^{N} t_{ij}} \; x\ 100\% \tag{4}$$

   $P$ represents the final measurement result or efficiency, depending on time. $R$ represents the number of individuals who utilized the application. $n_{ij}$ is the outcome of task $i$ by participant $j$; $n_{ij}$ will be worth 1 if the task is completed successfully or 0 if the task is not completed. $t_{ij}$ is the time necessary for participant $j$ to complete task scenario $i$.

3. Test for User Satisfaction
   A structured questionnaire displaying a satisfaction scale is used to determine the level of satisfaction of application users. Each questionnaire comprises several statements that show the subjectivity of the user's perspective regarding interactions with the program. A psychometric scale known as the Likert scale is utilized while constructing a questionnaire and analyzing its results [19]. In equations (5) and (6), the calculation of the survey results using the Likert scale is depicted (6).

$$Total\ Score = \sum_{i=1}^{x} w_i \,.\, n_i \tag{5}$$

$$Index\ Percentage = \frac{Total\ Score}{Maximal\ Score} \; x\ 100\% \tag{6}$$

   $w_i$ is the weight of each scale, $n_i$ is the number of respondents for each scale, and $x$ is the number of scales. The final formula to determine the percentage of user satisfaction is (6).

**3. Result and Discussion**

The NodeMCU port is connected to all components, such as sensors and an LCD. Soldering on the PCB is the method for connecting components. All components are assembled in a 3D-printed box, making the product neater and easier to use. Figure 7 depicts the prototype display of the monitoring device for heart rate and body temperature.

**Figure 7.** Prototype of Heart Rate and Body Temperature Monitoring Device

The user measures his heart rate by placing his finger on the simple pulse sensor's clamp. In the meantime, the user clamps the DS18260 sensor in his armpit to detect body temperature. The 16x2 LCDs are the results of measuring heart rate and body temperature. When the user enables the monitoring function, the application will display measurement data. Figure 8 depicts the monitoring application's graphical user interface.

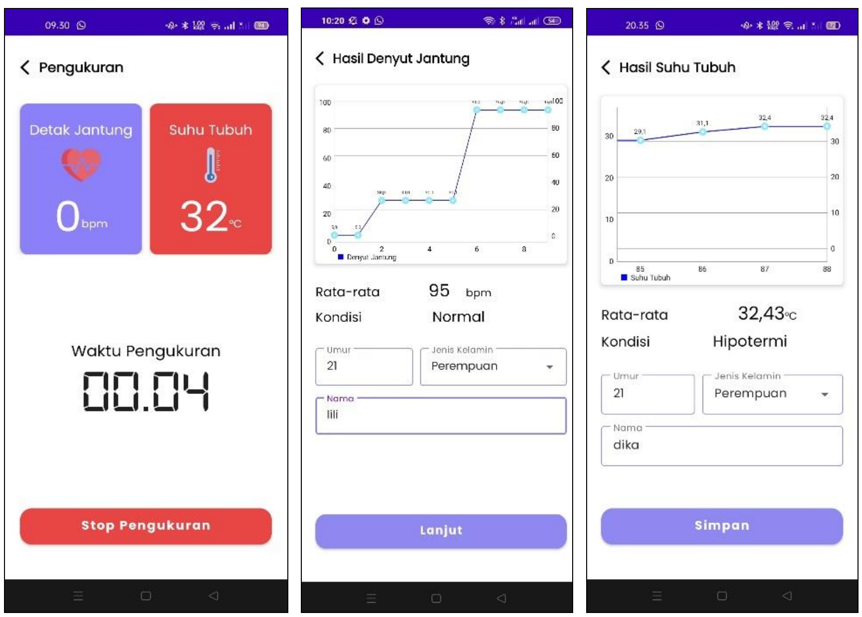


**Figure 8.** Monitoring of Heart Rate and Body Temperature is Displayed via the Application

When the user takes measurements, the timer automatically begins to run and calculate the duration. When the timer runs, the heart rate and body temperature numbers displayed on the application's screen will vary based on the user's physical state. To halt the measurement, press the stop button. When the user presses the stop button, they are automatically redirected to a new page containing a graph of their heart rate measurement history. Users can enter their name, age, and gender to determine the status of their heart.

When the subsequent button is pressed, the body temperature monitoring results will be displayed. The temperature monitoring findings are identical to that of heart rate monitoring in that a graph of the measurement history and the average value of the measurement are displayed. In addition to entering their body temperature, users must also provide their name, age, and gender.

### *3.1. Test of System Calibration*

The prototype measured heart rate and body temperature accurately. In addition, it successfully transmitted data to Firebase for display on the smartphone screen. In addition, the system calibration test is conducted by comparing the outcomes of data readings with commercially available measurement devices. During system calibration testing, the prototype's standard error will be determined.

Using an equation to calculate the standard error of heart rate and body temperature (2). The oximeter will be used to compare the heart rate sensor values. While the results from the temperature sensor will be compared to those from a thermometer, Figures 9 and 10 depict a line graph displaying the results of a comparison between heart rate and body temperature values.

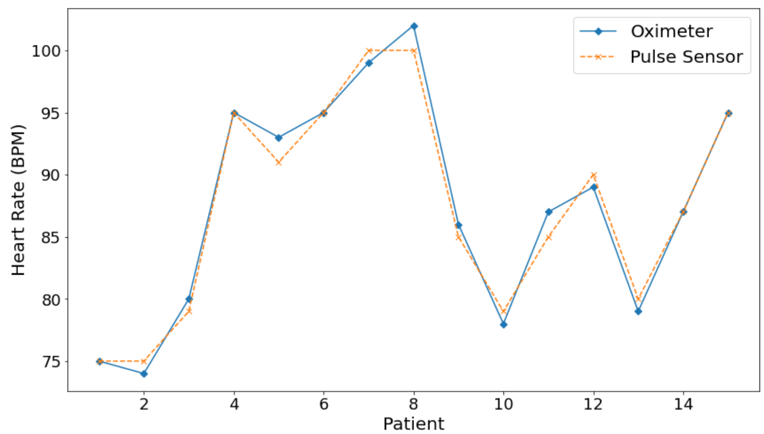


**Figure 9.** The Results of Comparing the Oximeter Data with the Easy Pulse Sensor Data

The sensor data reading test was performed fifteen times by a different user. A wide range of ages and genders is represented among people who use measurement equipment. The line chart depicted in Figure 7 demonstrates that the data provided by the pulse sensor have a value that is very close to the same as the data obtained by the oximeter. There is also a BPM value that is the same as the BPM value that the oximeter generates for you.

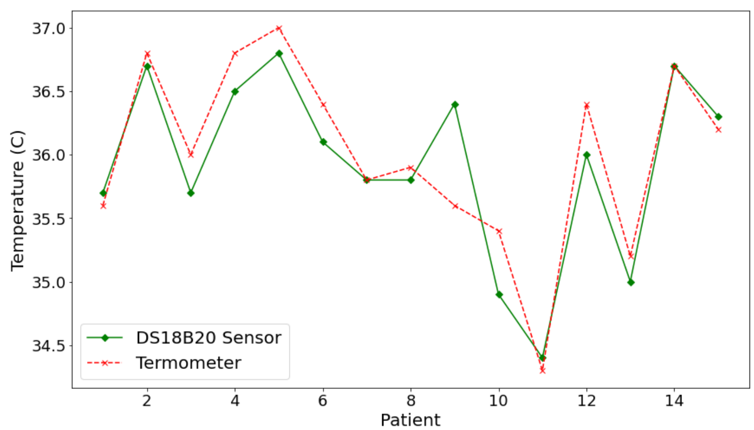


**Figure 10.** The Results of Comparing the Thermometer Data with the DS18260 Sensor Data

The same scenario applies to the results of the body temperature sensor test and the pulse sensor test. Fifteen individuals tested the prototype [20]. As a result, the temperature values produced by the two instruments are nearly identical. Table 4 displays the error rate or standard error values of the pulse sensor and DS18260 sensor.

**Table 4.** Outcomes of Standard Measurement Error

| Measurement | Heart Rate | Body Temperature |
|---|---|---|
| Standard Deviation | 2,65 | 0,77 |
| Standard Error | 0,68 | 0,18 |

Table 5 displays the outcomes of data processing derived from heart rate and temperature. Based on Table 5's experiments, the prototype has proved 100 percent effective at identifying body problems using heart rate and temperature.

**Table 5.** Prototype Test Data with Information from Medical Officers

| Patient | Temperature (◦C) | Heart Rate (BPM) | Testing | | Information |
|---|---|---|---|---|---|
| | | | Prototype | Medical Officer | |
| Patient 1 | 35.7 | 75 | Normal | Normal | Success |
| Patient 2 | 36.7 | 75 | Normal | Normal | Success |
| Patient 3 | 35.7 | 79 | Normal | Normal | Success |
| Patient 4 | 36.5 | 95 | Normal | Normal | Success |
| Patient 5 | 36.8 | 91 | Normal | Normal | Success |
| Patient 6 | 36.1 | 95 | Normal | Normal | Success |
| Patient 7 | 38.5 | 100 | Hipertermia | Hipertermia | Success |
| Patient 8 | 35.8 | 100 | Normal | Normal | Success |
| Patient 9 | 36.4 | 85 | Normal | Normal | Success |
| Patient 10 | 34.9 | 79 | Hipotermia | Hipotermia | Success |
| Patient 11 | 34.4 | 85 | Hipotermia | Hipotermia | Success |
| Patient 12 | 36.0 | 90 | Normal | Normal | Success |
| Patient 13 | 35.0 | 80 | Hipotermia | Hipotermia | Success |
| Patient 14 | 36.7 | 87 | Normal | Normal | Success |
| Patient 15 | 36.3 | 95 | Normal | Normal | Success |

### *3.2. Test of Application Performance Metrics*

Fifteen users utilize the heart rate and body temperature tracking application. The age of the youngest user is 16. The oldest individual is 60 years old. Two tests are used to determine an application's performance: the effectiveness test and the efficiency test. Equation 3 is used to calculate the level of application effectiveness, whereas equation 4 is used to calculate the efficiency level. Table 6 displays the results of the application performance measurement calculation.

**Table 6.** Result of Application Performance Measurement

| Measurement | Result (%) |
|---|---|
| Completion Rate | 100 |
| Time-Based Efficiency | 93 |

Application performance measurement reveals a 100 percent application completion rate. These results indicate that all users have finished the given task satisfactorily. The application is extremely user-friendly for those who have never used it before. In addition, the relative efficiency of the application as a whole is 93%. This indicates that the ratio of time required to complete the task by the user is quite low. The user can comprehend the application's operation within a short period.

### *3.3. User Acceptance Testing of the Application*

Utilizing a survey, testing application user satisfaction is determined. There are 20 questions organized into five categories. Respondents are required to respond to each question. There is a 1 to 5 satisfaction scale for each question. The pleasure value is measured using a Likert scale, as indicated in Equation 5 and Equation 6. Table 7 displays the findings of a survey about user satisfaction with applications.

**Table 7.** Result of User Satisfaction Survey for Applications

| Measurement | Result (%) |
|---|---|
| Reliability | 79,8 |
| Timeliness | 81,6 |
| Information Format | 83,7 |
| Informativeness | 85,4 |
| Easy to Use | 89,6 |

According to the information in Table 7, there is a very high level of user satisfaction with the application's design and usability. The measurement's average value of 80% serves as a clue. Easy of use scored the highest on all the tests (89.6%), according to Table 7 test findings. Because of the user interface's design, which is in line with the user's usability, it can be said that the application is simple to use. Although the application's reliability has the lowest rating, at 79.8%, the applications created can operate without faults or interruptions and are quite infrequent.

## 4. Conclusion

Monitoring the heart rate and body temperature is the first step in identifying illnesses and viruses that have entered the body. Using fuzzy logic, a heart rate and body temperature monitoring application has been successfully constructed in this work. The program can accurately assess the state of a person's body (normal, hypothermic, or hyperthermic) based on test results. The error rate of the sensor readings is relatively low, with 0.68 percent for heart rate and 0.18 percent for body temperature. High performance is demonstrated by the application's 100% completion rate and 93% time-based efficiency. The user satisfaction survey results suggest that the user is pleased with the application's design and functionality. Based on a questionnaire-based user satisfaction survey, the application's dependability, timeliness, information format, informativeness, and usability scores are 79.8%, 81.6%, 83.7%, 85.4%, and 89.0%, respectively.

## References

[1] G. W. Kirschen, Daniel D.Singer MD;, H. ThodeJr, and A. J.SingerMDb, "Relationship between body temperature and heart rate in adults and children: A local and national study," *Am. J. Emerg. Med.*, vol. 38, no. 4, pp. 929–933, 2020, doi: https://doi.org/10.1016/j.ajem.2019.158355.

[2] S. Villapol, "Gastrointestinal symptoms associated with COVID-19: impact on the gut microbiome," *Transl. Res.*, vol. 226, pp. 57–69, 2020, doi: 10.1016/j.trsl.2020.08.004.

[3] D. Surangsrirat, S. Dumnin, and S. Samphanyuth, "Heart rate skin temperature and skin humidity and their relationship to accumulated fatigue," in *3rd International Conference on Bio-Engineering for Smart Technologies*, 2019, pp. 1–4, doi: 10.1109/BIOSMART.2019.8734230.

[4] A. Abu Bakar, S. S. A. Rahim, A. R. Razali, E. Noorsal, R. Radzali, and A. F. Abd Rahim, "Wearable Heart Rate and Body Temperature Monitoring Device for Healthcare," *J. Phys. Conf. Ser.*, vol. 1535, no. 1, 2020, doi: 10.1088/1742-6596/1535/1/012002.

[5] I. Laila, A. Arifin, and B. Armynah, "Internet of Medical Things (IoMT)-Based Heart Rate and Body Temperature Monitoring System," *Indones. Phys. Rev.*, vol. 5, no. 1, pp. 1–14, 2022, doi: 10.29303/ipr.v5i1.134.

[6] Y. Devis, Y. Irawan, Junadhi, F. Zoromi, Herianto, and M. R. Amartha, “Monitoring System of Heart Rate, Temperature and Infusion in Patients Based on Microcontroller (Arduino Uno),” *J. Phys. Conf. Ser.*, vol. 1845, no. 1, 2021, doi: 10.1088/1742-6596/1845/1/012069.

[7] A. N. N. Chamim, J. Rinaldi, Y. Ardiyanto, I. Iswanto, and A. Ma’Arif, “Heart Rate and Body Temperature Monitoring Based on Android Operating System,” in *Proceeding - 2020 2nd International Conference on Industrial Electrical and Electronics, ICIEE 2020*, 2020, pp. 143–148, doi: 10.1109/ICIEE49813.2020.9276750.

[8] F. M. Yassin, N. A. Sani, and S. N. Chin, “Analysis of Heart Rate and Body Temperature from the Wireless Monitoring System Using Arduino,” *J. Phys. Conf. Ser.*, vol. 1358, no. 1, 2019, doi: 10.1088/1742-6596/1358/1/012041.

[9] A. K. Dwivedi, Abhishri Dwivedi, Sonali Singh, and Abhijeet Tiwari, “Prototype Development for IOT Based Contactless Heart-rate and Body Temperature Monitoring System,” in *2021 7th International Conference on Signal Processing and Communication (ICSC)*, 2021, pp. 377–380, doi: 10.1109/ICSC53193.2021.9673311.

[10] A. Taqwa *et al.*, “Prototype Development of Heartbeat and Body Temperature Monitoring System Based on Internet of Things,” in *Proceedings of the 4th Forum in Research, Science, and Technology*, 2021, vol. 7, pp. 605–612, doi: 10.2991/ahe.k.210205.100.

[11] V. Muneeswaran, P. Nagaraj, M. P. Rajasekaran, N. S. Chaithanya, S. Babajan, and S. U. Reddy, “Heart (Pulse Rate) Monitoring using Pulse Rate Sensor, Piezo Electric Sensor and NodeMCU,” in *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, 2021, pp. 530–533, doi: 10.1109/ICCES51350.2021.9489052.

[12] R. Jayaysingh, J. David, M. Joel Morris Raaj, D. Daniel, and D. Blessytelagathoti, “IoT Based Patient Monitoring System Using NodeMCU,” in *5th International Conference on Devices, Circuits, and Systems*, 2020, pp. 240–243, doi: 10.1109/ICDCS48716.2020.243588.

[13] Anonim, “Heart Rate BPM Meter using Easy Pulse Sensor & Arduino,” *https://how2electronics.com/*, 2022. https://how2electronics.com/heart-rate-bpm-meter-using-easy-pulse-sensor-arduino/ (accessed Aug. 24, 2022).

[14] A. Kumar, *Mastering Firebase for Android Development*, First. Birmingham: Packt Publishing, 2020.

[15] I. Iancu, “Heart disease diagnosis based on mediative fuzzy logic,” *Artif. Intell. Med.*, vol. 89, pp. 51–60, 2018, doi: https://doi.org/10.1016/j.artmed.2018.05.004.

[16] A. Raghunathan, S. M. Xie, F. Yang, J. C. Duchi, and P. Liang, “Understanding and mitigating the tradeoff between robustness and accuracy,” in *37th International Conference on Machine Learning, ICML 2020*, 2020, pp. 7865–7875.

[17] Y.-J. Son and E. Y. Kim, “The effectiveness and usability of a novel mobile phone-based self-care intervention for patients with heart failure: a mixed-methods pilot study,” *Eur. J. Cardiovasc. Nurs.*, vol. zvac077, pp. 1474–5151, 2022, doi: https://doi.org/10.1093/eurjcn/zvac077.

[18] H. M. Az-Zahra, W. Parwaningsuci, and M. C. Saputra, “Usability Evaluation of User Interface in Badan Narkotika Nasional East Java Province Website,” in *3rd International Conference on Sustainable Information Engineering and Technology, SIET 2018 - Proceedings*, 2018, pp. 262–265, doi: 10.1109/SIET.2018.8693144.

[19] A. Joshi, S. Kale, S. Chandel, and D. Pal, “Likert Scale: Explored and Explained,” *Br. J. Appl. Sci. Technol.*, vol. 7, no. 4, pp. 396–403, 2015, doi: 10.9734/bjast/2015/14975.

[20] T. V Greer, N. Chuchinprakarn, and S. Seshadri, “Likelihood of Participating in Mail Survey Research: Business Respondents’ Perspectives,” *Ind. Mark. Manag.*, vol. 29, no. 2, pp. 97–109, 2000, doi: https://doi.org/10.1016/S0019-8501(98)00038-8.