Online-Journals.org

iJOE International Journal of Online and Biomedical Engineering

iJOE | eISSN: 2626-8493 | Vol. 20 No. 8 (2024) | OPEN ACCESS

https://doi.org/10.3991/ijoe.v20i08.48071

PAPER

Prototype Realtime Detection of Abnormal Heart Beat Using Multiple Back Propagation Neural Network (BPNN)

Suryani(✉), Faizal

Department of Informatics Engineering, Dipa Makassar University, Makassar, Indonesia

suryani187@undipa.ac.id/

ABSTRACT

Real-time heart rate monitoring and early detection of heart abnormalities are vital to determine heart health before it worsens. To achieve this goal, this project uses the backpropagation neural network (BPNN) method including its capability to classify heartbeats into normal or abnormal by inputting heartbeat values in BPM units derived from prototypes utilizing sensors like Sensor Easy Pulse and NodeMCU, along with considerations of age and sports activity. All data from sensors will be stored in Firebase. Then Firebase will connect to Android, and the normal and abnormal heart classification results will be displayed on the Android system. Simulation results successfully examined 40 people as a sample and provided information from real-time heart rate monitoring, age, and sports activity as input. This research seeks to contribute to improving health services at various public health service centers and independently in detecting heart health early.

KEYWORDS

classification, back propagation neural network, heart health, prototype, sensor easy pulse, NodeMCU

1 INTRODUCTION

Heart disease or cardiovascular disease is the number one killer disease in the world, especially among adults and the elderly. In 1990, there were 14.4 million deaths due to heart attacks. This figure increased to 17.5 million in 2005; in 2030, it is projected to rise once more, reaching 23.6 million individuals (American Heart Association, 2014).

Heart disease occurs if there is damage to the main blood vessels that supply blood to the heart. Heart disease is caused by cholesterol and inflammation of the blood vessels. Coronary heart disease is one of the highest causes of death in the world. In Indonesia, 45% of all deaths are caused by heart disease. The majority of

Suryani, Faizal. (2024). Prototype Realtime Detection of Abnormal Heart Beat Using Multiple Back Propagation Neural Network (BPNN). *International Journal of Online and Biomedical Engineering (iJOE)*, 20(8), pp. 83–99. https://doi.org/10.3991/ijoe.v20i08.48071

Article submitted 2024-01-19. Revision uploaded 2024-03-07. Final acceptance 2024-03-07.

heart disease sufferers are aged 55–64 years. Most people ignore mild symptoms (chest pain, arms that spread to the neck, jaw, shoulders, and back, weakness, cold sweat, dizziness, nausea, and vomiting), which can be caused by stress, unhealthy living such as eating unhealthy food, smoking, lack of exercise or sports activities, frequently drinking alcoholic beverages, and lack of rest, which can cause the heart to become abnormal.

The importance of knowing about heart health from an early age cannot be underestimated because this can potentially reduce the death rate caused by heart disease. To meet this challenge, we conducted research to classify heart health by examining various factors that influence heart health. We utilize input variables that impact heart health to generate a classification distinguishing between normal and abnormal hearts. The input variables are heart rate, which can be known in real-time through the prototype built, age, and sports activity.

Monitoring tools to calculate heart rate are available, both conventional and non-conventional. However, these tool are limited to checking heart rate; the challenge in monitoring heart rate lies in providing real-time heart classification capabilities as normal or abnormal by adding input variables. To overcome this challenge, we apply BPNN as a classification method to classify real-time heart rate, age, and sports activity data.

This research seeks to contribute to improving health services at various public health service centers and independently detecting heart health early.

2 LITERATURE REVIEW

2.1 Background

The main circuit of the heart rate monitoring device consists of 1 sensor, namely, the Easy Pulse Plugin sensor for monitoring heart rate to the MCU node and several other components that support a series of tools such as resistors, LEDs, buzzers, jumper cables, and push buttons.

Easy Pulse Plugin is a heart rate sensor that works using the principle of photoplethysmography (PPG), a non-invasive method for measuring heart rate (cardiovascular) by detecting the volume of blood flow in the pulse very close to the skin. This sensor uses an IR LED and photodetector, where the pulse on your finger will affect the flow of light from the IR LED to the photodetector; these changes are then converted, filtered, and amplified by the sensor module then processed by Arduino or other minsys [1].

NodeMCU is a microcontroller that is equipped with an ESP8266 WiFi module inside. NodeMCU can be analogous to the ESP8266 Arduino board, so this can save money; there is no need to have two devices, the Arduino board and the ESP8266 module [2].

The microcontroller will process data from the sensors; then, the data will enter the database and appear in the Android application system. When the essay pulse sensor detects a heartbeat and inputs other data, namely age and sports activity, these three data will be processed with a backpropagation neural network (BPNN).

Artificial Neural Network (ANN) is a technique in Machine Learning that imitates human nerves, which are a fundamental part of the brain [3]. ANN can be used to model complex relationships between input and output to find patterns in data [4]. Mathematically, ANN is like a graph that has nodes. 3 things underlie ANN:

the relationship pattern between each network or network architecture, the method for determining link weights or training method, and the function determining the output or activation function. The reason for using an Artificial Neural Network in this research is that ANN is a method that has been widely used for prediction or estimation. In this research, ANN is used because of its role in adopting a learning system in the human brain, which, with this method, can solve complex non-linear problems that are difficult to solve for conventional mathematics and statistical approaches. ANN is used to classify input data and process it effectively to obtain an output or efficient results.

2.2 Related work

Yuhefizar and colleagues presented research on a heart rate monitoring system with low-cost WiFi communication using the ESP8266 WiFi module utilizing OpenSID as a database. Heart rate values can be displayed on the LCD and website in real-time. Patients and doctors can view heart rate information anytime on the website. The SMS alert system will be active if the heart rate value is below 60 BPM and above 100 BPM and there is a press on the panic button. The time required to send the SMS is around 7 to 8 seconds [5].

Yosephine Niendy Alexandra and colleagues implemented an Artificial Neural Network on Electrocardiogram Signals to Detect Supraventricular Arrhythmia Heart Disease. The process of classifying Supraventricular Arrhythmia disease using an Artificial Neural Network was successfully implemented. Test results with a multilayer perceptron ANN structure produced accuracy, specificity, and sensitivity values of 83.3% [6].

Bimo Anjas Moro and his colleagues created a tool that can detect and count heartbeats, requiring a special sensor to read the pulse, a tool capable of processing data, and a Bluetooth module to transfer data. Then, to determine normal or abnormal heart conditions, we first have to know how many heartbeats can be said to be normal or abnormal so that when we know this, we can determine what the heart condition is. Then, the way to send heart examination results data via Bluetooth media is by connecting the Bluetooth device to the device and the smartphone. On the device side, you also need a little configuration on the Bluetooth module and Arduino Uno to send data via Bluetooth media. The research showed that the heart condition detection system consisting of hardware and software could function well. From tests carried out on seven samples, the error percentage was found to be 4.7%, so it can be concluded that the system is quite accurate in calculating heart rate and determining the heart condition [7].

The growing body of research in heart health prediction is evidence of growing interest in leveraging technology to identify heart health accurately. These studies underscore the importance of developing robust systems for predicting an individual's heart health, as early detection can address critical healthcare challenges. Given the abundance of previous research, this area remains an important focus for computational approaches to improve healthcare outcomes.

3 METHODOLOGY

The main objective of this paper is to create a system that integrates a prototype to detect heartbeats and implements a Back Propagation Neural Network to determine normal and abnormal heart classification via an Android smartphone. Supervised learning encompasses various techniques, and among the most promising ones is the Backpropagation algorithm [8]. BPNN is one of the ANN methods, an information processing method adapted from biological neurons [9]. In the context of addressing learning challenges in feedforward neural networks, the BP algorithm consistently demonstrates strong performance compared to other neural network algorithms. Empirical data indicates that approximately 80% to 90% of artificial neural network models in practical applications rely on BP networks. These algorithms serve as the foundation for advanced network architectures, encapsulating the core capabilities of neural networks [10]. The merit of the method lies in its ability to classify heart rate data (X3) generated by the prototype, along with additional data input via an Android smartphone, such as age (X1) and activity (X2). The data is processed using the BPNN method, which outputs normal heart (Y1) or abnormal heart (Y2). BPNN consumes less memory than other algorithms and can produce results with an acceptable error rate and relatively fast processing speed. Additionally, this method is advantageous because it can recognize incomplete or poor input patterns, addressing learning challenges within feedforward neural networks. The ANN architecture used in this research can be seen in Figure 1.

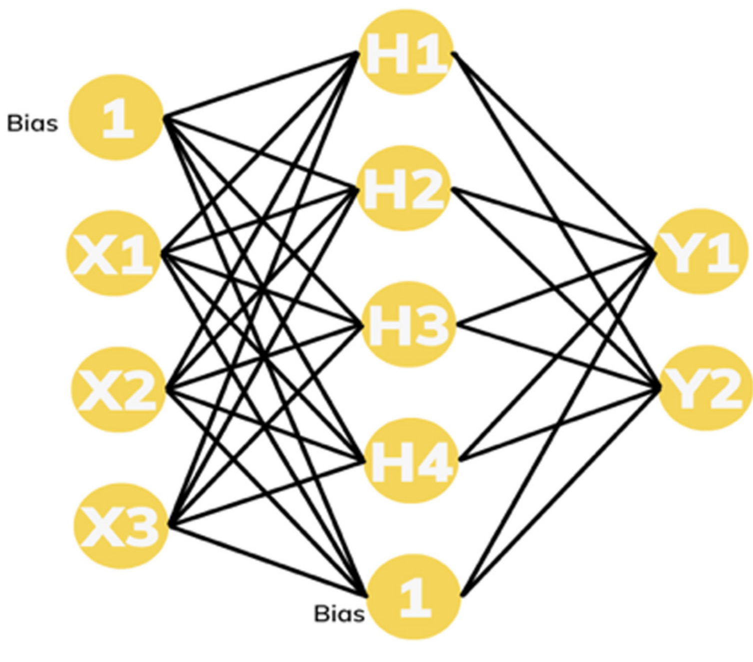

Fig. 1. ANN architecture

The ANN architecture consists of 1 input layer with three neurons (X1, X2 and X3), one hidden layer with four neurons (H1, H2, H3 and H4) using the sigmoid activation function, namely z = 1/(1 + e^(-z-in)), Apart from that, one output with two neurons (Y1 and Y2) uses the sigmoid activation function y = 1/(1 + e^(-y-in)).

The backpropagation algorithm is one of the ANN training methods that minimizes errors in the output resulting from the learning process [9].

BPNN has been widely used in many prediction fields. BPNN is a multilayer feedforward network with strong non-linear fitting capabilities and a classic network structure consisting of three layers: an input layer, one or more hidden layers, and an output layer [11]. BPNN possesses a remarkable advantage due to its robust nonlinear mapping capabilities and flexible network structure [12].

Steps on how BPNN works [13]:

1. In Forward Propagation (forward propagation), the input pattern is calculated from the input layer to the output layer for forward calculation.
2. Backpropagation is done with something directly related to the units in the output layer. Propagation is carried out because there is a difference in output and the target we want.
3. Changes in Weights and Bias are carried out on the weights so that errors or errors can be reduced or reduced.

The above phases will continue to be repeated until the conditions are met. Equality to count neurons in the hidden layer:

$$Y\_in_J = V_{0j} + \sum_{i=1}^{n} X_i V_{ij}$$

$$Y_j = f(Y\_in_j) = \frac{1}{1 + exp^{(-z\_net_j)}}$$

Y_in_j = value for calculating hidden layer, V_{0j} = bias weight between input layer and hidden layer ($V_{j0} = 0$), X_i = value of input layer i, n = number of inputs, V_{ij} = weight between input layer and hidden layer, X_i = value of input layer i, Y_j = value of hidden layer.

Equality count neurons in the output layer:

$$Z\_in_j = W_{0j} + \sum_{i=1}^{n} Y_i W_{ij}$$

$$Z_j = f(Z\_in_j)$$

Z_in_j = input value of unit j, W_{0j} = connection weight value on bias for unit Z_j(W_{0j} = 0), Y_i = activation value of unit Y_i, W_{ij} = connection weight value from Y_{ij} to unit Z_j, Z_j = jth unit in layer output, Z_in_j = input value of unit j, n = number of inputs.

Equality to count the error value:

$$\delta_j = t_j - Z_j$$

δ_j = factor setting the connection weight value in the output layer, t_j = target value and Z_j = input value unit j.

Equality for activation function:

$$f(x) = \frac{1}{1 + (e^{-x})}$$

$$f' = f(x)[1 - f(x)]$$

$f(x)$ = xth unit in the output layer, f' = input value.

The equation for changing weight:

$$\Delta W_{ij} = \alpha \delta_j Y_i$$

$$W_{jk}(new) = W_{jk}(old) + \Delta W_{ij}$$

$$\Delta V_{ij} = \alpha \delta_j X_i$$

$$V_{jk}(new) = V_{jk}(old) + \Delta V_{ij}$$

W_{jk} = connection weight value from Z_{ij} to unit Yk, ΔW_{ij} = difference between W_{jk}\{t\} and W_{jk}\{t+1\}, where j = 1, 2, …, m and k = 0, 1, …, p. V_{jk} = connection weight value from unit X_i to unit Z_i, ΔV_{ij} = difference between V_{jk}\{t\} and V_{jk}\{t+1\}, where j = 1, 2, …, p and k = 0, 1, …, n.

The system architecture built in this research can be seen in Figure 2.

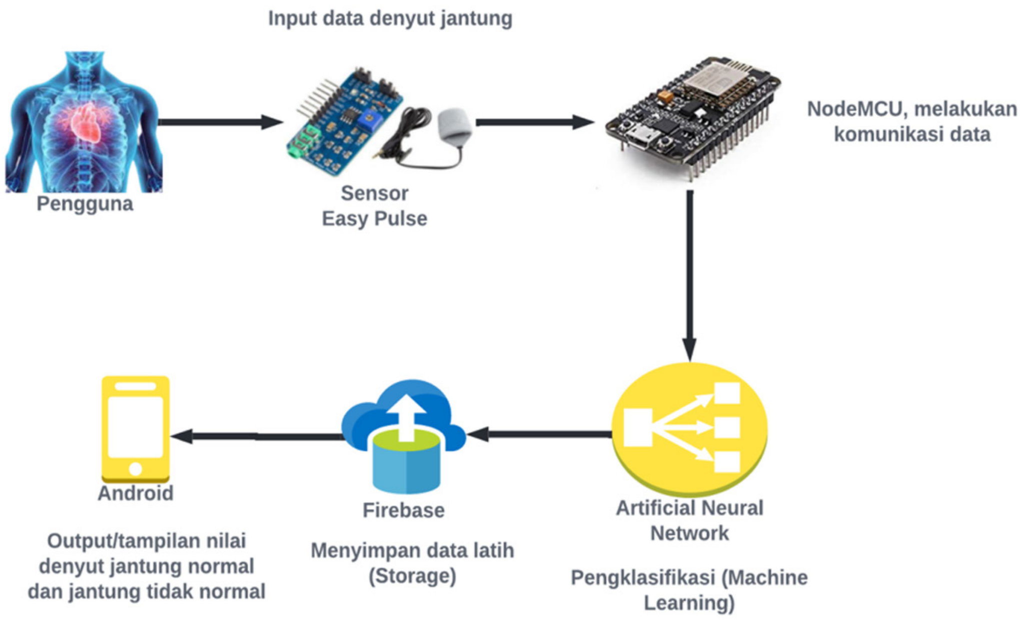

Fig. 2. System architecture

The working principle of the image above is to create a tool that can classify normal and abnormal hearts in real time. The way this tool works is:

1. The Easy Pulse Plugin sensor detects heart rate which works using the principle of photoplethysmography (PPG), which is a non-invasive method for measuring heart rate (cardiovascular) by detecting the volume of blood flow in the pulse which is very close to the skin.
2. The NodeMCU Esp8266 collects data from sensors and facilitates data communication.
3. BPNN employed to carry out a classification process that determines the normalcy of the heart.
4. All data from sensors will be stored in Firebase.
5. Then Firebase will connect to Android, and the normal and abnormal heart classification results will be displayed on the Android system.

This process is depicted in the flowchart that can be seen in Figure 3.

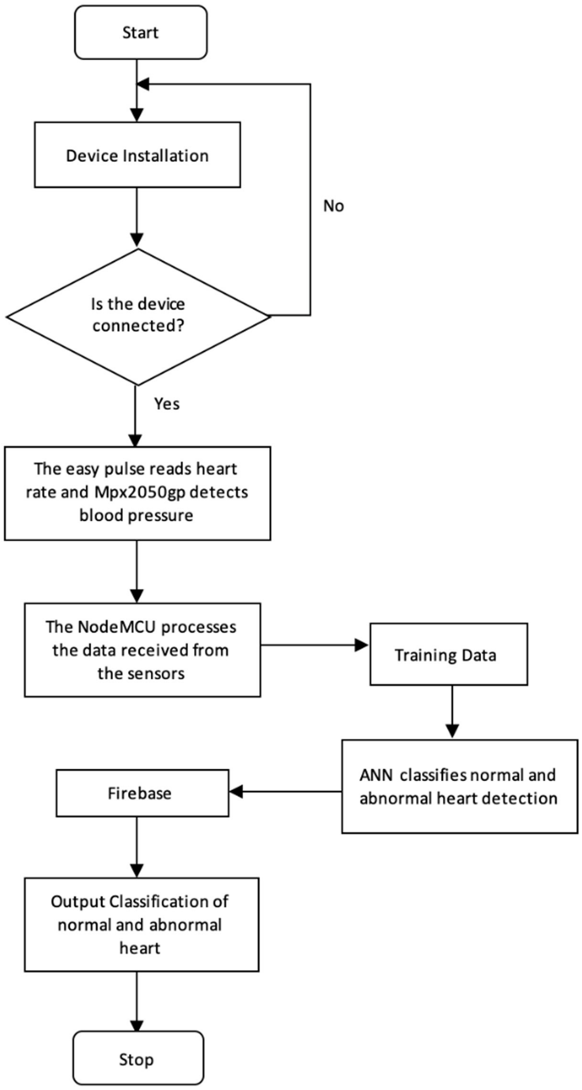

Fig. 3. Flowchart system

The process begins sensor installation when the sensor has not been installed; the process will be repeated to install the sensor; the Easy Pulse sensor detects the heart rate, and then the data is sent to the NodeMCU and will be forwarded to the ANN in the form of training data. The ANN will carry out the process of classifying normal and abnormal heart after the classification results will go to Firebase. They will be displayed on Android once the process is complete.

4 IMPLEMENTATION

4.1 Model building

We describe the steps of the proposed methodology:
Variable Initialization, Weights, and Bias.
The variables used include:

1. Input variables consist of age (X1), sports activity (X2), and bpm value (X3).
2. The output variable consists of a normal heart (Y1) and an abnormal heart (Y2).

Initial weight values for input include:

1. X1 = 0,2
2. X2 = 0,3
3. X3 = 0,5

The initial weight input to the hidden layer:

$V_{11} = 0{,}1 \quad V_{12} = 0{,}2 \quad V_{13} = 0{,}3 \quad V_{14} = 0{,}4$

$V_{21} = 0{,}1 \quad V_{22} = 0{,}2 \quad V_{23} = 0{,}3 \quad V_{24} = 0{,}4$

$V_{31} = 0{,}1 \quad V_{32} = 0{,}2 \quad V_{33} = 0{,}3 \quad V_{34} = 0{,}4$

The initial weight is biased towards the hidden layer:

$V_{01} = 0{,}1 \quad V_{02} = 0{,}2 \quad V_{03} = 0{,}3 \quad V_{04} = 0{,}4$

The initial Weight hidden layer to output layer:

$W_1 = 0{,}1 \quad W_2 = 0{,}2 \quad W_3 = 0{,}3 \quad W_4 = 0{,}4$

The initial weights are biased towards the output layer:

$$W_0 = 0{,}5$$

Network training need is carried out on:

Learning rate (α) = 1

Target error = 0,02

Network Training
Data to = 1 (X1 = 0.2, X2 = 0.3, X3 = 0.5, target T = 1)

Step 1: Forward propagation stage (forward propagation)

Operations on hidden layers:

$$\begin{aligned} Z\_in_1 &= V_{01} + V_{11} \times X_1 + V_{21} \times X_2 + V_{31} \times X_3 \\ &= 0{,}1 + 0{,}1 \times 0{,}2 + 0{,}1 \times 0{,}3 + 0{,}1 \times 0{,}5 \\ &= 0{,}2 \end{aligned}$$

$$\begin{aligned} Z\_in_2 &= V_{02} + V_{12} \times X_1 + V_{22} \times X_2 + V_{32} \times X_3 \\ &= 0{,}1 + 0{,}2 \times 0{,}2 + 0{,}2 \times 0{,}3 + 0{,}2 \times 0{,}5 \\ &= 0{,}4 \end{aligned}$$

$$\begin{aligned} Z\_in_3 &= V_{03} + V_{13} \times X_1 + V_{23} \times X_2 + V_{33} \times X_3 \\ &= 0{,}3 + 0{,}3 \times 0{,}2 + 0{,}3 \times 0{,}3 + 0{,}3 \times 0{,}5 \\ &= 0{,}6 \end{aligned}$$

$$\begin{aligned} Z\_in_4 &= V_{04} + V_{14} \times X_1 + V_{24} \times X_2 + V_{34} \times X_3 \\ &= 0{,}4 + 0{,}4 \times 0{,}2 + 0{,}4 \times 0{,}3 + 0{,}4 \times 0{,}5 \\ &= 0{,}8 \end{aligned}$$

Activation Function of the hidden layer $z_1 = \frac{1}{1+e^{-z-in}}$

$$z_1 = \frac{1}{1+e^{-0,2}} = 0,5498$$

$$z_2 = \frac{1}{1+e^{-0,4}} = 0,5987$$

$$z_3 = \frac{1}{1+e^{-0,6}} = 0,6456$$

$$z_4 = \frac{1}{1+e^{-0,8}} = 0,6899$$

Operation on the output layer:

$$\mathrm{Y\_in} = \mathrm{W}_0 + \mathrm{W}_1 \times \mathrm{Z}_1 + \mathrm{W}_2 \times \mathrm{Z}_2 + \mathrm{W}_3 \times \mathrm{Z}_3 + \mathrm{W}_4 \times \mathrm{Z}_4$$

$$= 0,5 + 0,1 \times 0,5498 + 0,2 \times 0,5987 + 0,3 \times 0,6456 + 0,4 \times 0,6899$$

$$= 1,14436$$

Function on the output layer $= \frac{1}{1+e^{-y-in}}$

$$y = \frac{1}{1+e^{-1,14436}} = 0,7584$$

Step 2: Backpropagation stage (Backpropagation)

$$\delta = (T_1 - \mathrm{y}) \times \left(\frac{1}{1+e^{-y-in}}\right) \times \left[1 - \left(\frac{1}{1+e^{-y-in}}\right)\right]$$

$$\delta = (0 - 0,7584) \times \left(\frac{1}{1+e^{-1,14436}}\right) \times \left[1 - \left(\frac{1}{1+e^{-1,14436}}\right)\right] = -0,1389$$

$$\Delta w_1 = \alpha \times \delta \times w_1$$

$$\Delta w_1 = 1 \times (-0,1389) \times 0,5498 = -0,0763$$

$$\Delta w_2 = \alpha \times \delta \times w_2$$

$$\Delta w_2 = 1 \times (-0,1389) \times 0,5987 = -0,0831$$

$$\Delta w_3 = \alpha \times \delta \times w_3$$

$$\Delta w_3 = 1 \times (-0,1389) \times 0,6456 = -0,0896$$

$$\Delta w_4 = \alpha \times \delta \times w_4$$

$$\Delta w_4 = 1 \times (-0,1389) \times 0,6899 = -0,0958$$

$$\Delta w_0 = \alpha \times \delta$$

$$\Delta w_0 = 1 \times (-0,1389) = -0,1389$$

$$\delta in_1 = \delta \times w_1 = -0{,}1389 \times 0{,}1 = -0{,}0138$$

$$\delta in_2 = \delta \times w_2 = -0{,}1389 \times 0{,}2 = -0{,}0277$$

$$\delta in_3 = \delta \times w_3 = -0{,}1389 \times 0{,}3 = -0{,}0416$$

$$\delta in_4 = \delta \times w_4 = -0{,}1389 \times 0{,}4 = -0{,}0055$$

$$\delta_1 = \delta in_1 \times \left(\frac{1}{1+e^{-z-in1}}\right) \times \left[1-\left(\frac{1}{1+e^{-y-in1}}\right)\right]$$

$$= -0{,}0138 \times \left(\frac{1}{1+e^{-0{,}2}}\right) \times \left[1-\left(\frac{1}{1+e^{-0{,}2}}\right)\right] = -0{,}0034$$

$$\delta_2 = \delta in_2 \times \left(\frac{1}{1+e^{-z-in2}}\right) \times \left[1-\left(\frac{1}{1+e^{-y-in2}}\right)\right]$$

$$= -0{,}0277 \times \left(\frac{1}{1+e^{-0{,}4}}\right) \times \left[1-\left(\frac{1}{1+e^{-0{,}4}}\right)\right] = -0{,}0066$$

$$\delta_3 = \delta in_3 \times \left(\frac{1}{1+e^{-z-in3}}\right) \times \left[1-\left(\frac{1}{1+e^{-y-in3}}\right)\right]$$

$$= -0{,}0416 \times \left(\frac{1}{1+e^{-0{,}6}}\right) \times \left[1-\left(\frac{1}{1+e^{-0{,}6}}\right)\right] = -0{,}0242$$

$$\delta_4 = \delta in_4 \times \left(\frac{1}{1+e^{-z-in4}}\right) \times \left[1-\left(\frac{1}{1+e^{-y-in4}}\right)\right]$$

$$= -0{,}0138 \times \left(\frac{1}{1+e^{-0{,}8}}\right) \times \left[1-\left(\frac{1}{1+e^{-0{,}8}}\right)\right] = -0{,}0011$$

$$\Delta v_{01} = \alpha \times \delta_1 = 1 \times (-0{,}0034) = -0{,}0034$$

$$\Delta v_{02} = \alpha \times \delta_2 = 1 \times (-0{,}0066) = -0{,}0066$$

$$\Delta v_{03} = \alpha \times \delta_3 = 1 \times (-0{,}0242) = -0{,}0242$$

$$\Delta v_{04} = \alpha \times \delta_4 = 1 \times (-0{,}0011) = -0{,}0011$$

Step 3: Weight, and Bias Change Stage

$$v_{01}(baru) = v_{01}(lama) + \Delta v_{01} = 0{,}1 - 0{,}0034 = 0{,}0966$$

$$v_{02}(baru) = v_{02}(lama) + \Delta v_{02} = 0{,}2 - 0{,}0066 = 0{,}1924$$

$$v_{03}(baru) = v_{03}(lama) + \Delta v_{03} = 0{,}3 - 0{,}0343 = 0{,}2758$$

$$v_{04}(baru) = v_{04}(lama) + \Delta v_{04} = 0{,}4 - 0{,}0011 = 0{,}3989$$

$$w_1(baru) = w_1(lama) + \Delta w_1 = 0{,}1 - 0{,}0763 = 0{,}0237$$

$$w_2(baru) = w_2(lama) + \Delta w_2 = 0{,}2 - 0{,}0831 = 0{,}1169$$

$$w_3(baru) = w_3(lama) + \Delta w_3 = 0{,}3 - 0{,}0896 = 0{,}2104$$

$$w_1(baru) = w_1(lama) + \Delta w_1 = 0{,}1 - 0{,}0763 = 0{,}0237$$

$$w_4(baru) = w_4(lama) + \Delta w_4 = 0{,}4 - 0{,}0958 = 0{,}3042$$

$$w_0(baru) = w_0(lama) + \Delta w_0 = 0{,}5 - 0{,}1389 = 0{,}3042$$

For the second data, the same operation as the first data is carried out; only the initial weight and bias values used are the weight and bias values from the calculation results of the first data, and so on until the fourth data (1 epoch), this process is conditioned to the maximum 100th epoch or squared error ≤ 0.02 (target error).

For example, after the end of literacy, the following final weight and bias values are obtained:

$$v_{11} = 5{,}8761 \quad v_{12} = 3{,}7076 \quad v_{13} = 3{,}5877 \quad v_{14} = -0{,}0804$$

$$v_{21} = -4{,}7532 \quad v_{22} = 2{,}7028 \quad v_{23} = -5{,}2943 \quad v_{24} = 0{,}7643$$

The initial weight input to the hidden layer:

$$v_{01} = 2{,}5618 \quad v_{02} = -0{,}3804 \quad v_{03} = -1{,}4358 \quad v_{04} = -0{,}6998$$

The initial weight hidden layer to output layer:

$$w_1 = -7{,}0989 \quad w_2 = 3{,}5872 \quad w_3 = 6{,}9217 \quad w_4 = 0{,}7607$$

The initial weights are biased towards the output layer:

$$w_0 = 0{,}6671$$

Now we test the network for each data.
Testing the first data (X1 = 0. 881356, X2 = 0. 830508, X3 = 0. 0, Target Y = 1)
Operation on the hidden layer =

$$Z_{in_1} = V_{01} + (V_{11} \times X_1) + (V_{21} \times X_2) + (V_{31} \times X_3)$$

$$= 2{,}5618 + (5{,}8761 \times 0{,}881356) + (-4{,}7532 \times 0{,}335714) + (3{,}7856 \times 0{,}0)$$

$$= 6{,}1450$$

$$Z_{in_2} = V_{02} + (V_{12} \times X_1) + (V_{22} \times X_2) + (V_{32} \times X_3)$$

$$= -0{,}3894 + (3{,}7076 \times 0{,}881356) + (2{,}7028 \times 0{,}335714) + (2{,}1171 \times 0{,}0)$$

$$= 3{,}7856$$

$$Z_{in_3} = V_{03} + (V_{13} \times X_1) + (V_{23} \times X_2) + (V_{33} \times X_3)$$

$$= -1{,}4358 + (3{,}5877 \times 0{,}881356) + (-5{,}2943 \times 0{,}335714) + (2{,}5788 \times 0{,}0)$$

$$= -0{,}0511$$

$$Z_{in_4} = V_{04} + (V_{14} \times X_1) + (V_{24} \times X_2) + (V_{34} \times X_3)$$

$$= -0{,}6998 + (0{,}0804 \times 0{,}881356) + (0{,}7643 \times 0{,}335714) + (0{,}5364 \times 0{,}0)$$

$$= -0{,}3725$$

Activation function on the hidden layer $Z_i = \dfrac{1}{1 + e^{-z-in}}$

$$Z_1 = \frac{1}{1 + e^{-6,1450}} = 1,0021 \qquad Z_3 = \frac{1}{1 + e^{0,0511}} = 2,0524$$

$$Z_2 = \frac{1}{1 + e^{-3,7856}} = 1,0226 \qquad Z_4 = \frac{1}{1 + e^{0,3725}} = 2,4513$$

Operations on the output layer:

$$Y_{-in} = W_0 + (Z_1 \times W_1) + (Z_2 \times W_2) + (Z_3 \times W_3) + (Z_4 \times W_4)$$

$$= 0,6671 + (1,0021 \times (-7,0989)) + (1,0226 \times 3,5872) + (2,0524 \times 6,9217) + (2,4513 \times 0,7607)$$

$$= 13,2922$$

Activation function on the output layer $Y = \dfrac{1}{1 + e^{-y-in}}$

$$Y = \frac{1}{1 + e^{-13,2922}} = 1,0000$$

Activation function $\mathrm{T} = \begin{cases} 0, jika\, Y < 0,5 \\ 1, jika\, Y \geq 0,5 \end{cases}$

Activation result T = 1 (same as target)

5 RESULT AND DISCUSSION

This study produces a prototype that monitors heart rate in bps units, which will also be an input variable in classifying the heart as normal or abnormal. The core circuit of the prototype consists of 1 sensor, namely, the Easy Pulse Plugin sensor for heart rate monitoring to the MCU node, and several other components that support the device circuit, namely resistors, LEDs, buzzers, jumper cables, and push buttons as seen in Figure 4.

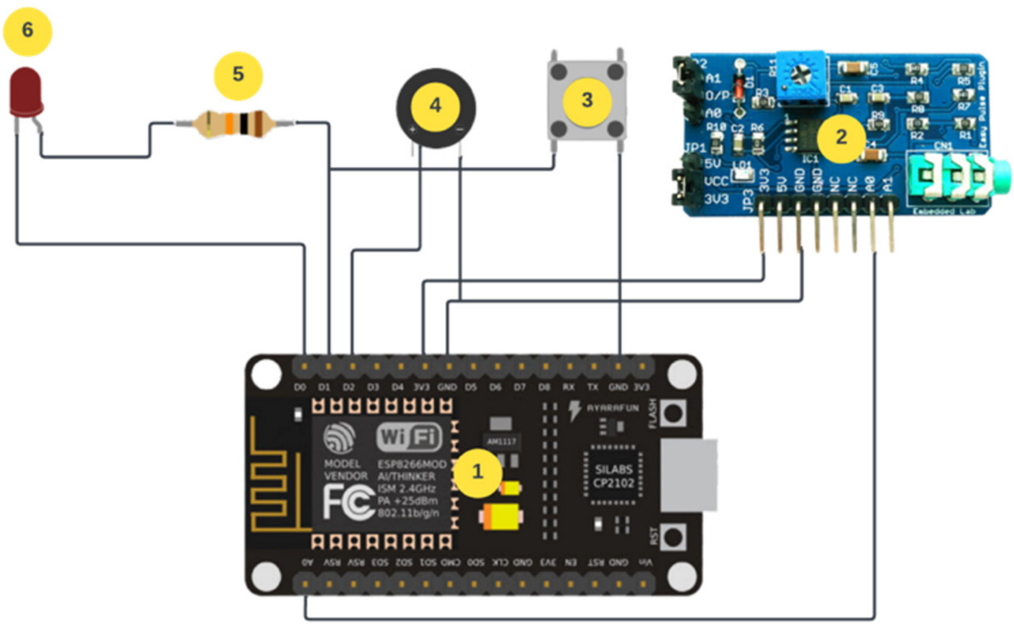

Fig. 4. Prototype tool suite

The Easy Pulse sensor is connected to the Node MCU, namely, the AO pin on the Easy Pulse sensor is connected to the AO pin on the Node MCU, and the 3V pin on the Easy Pulse sensor is connected to the VCC on the Node MCU. DO is connected to the GND pin on the LED, D1 is connected to the resistor on pin1 of the switch, pin 2 of the switch is connected to VCC on the NodeMCU, and the Buzzer pin D2 of the NodeMCU is connected to the VCC Buzzer pin.

The system built has a working principle: the heart rate value produced by the prototype and the other two inputs, namely age and sports activity, will be processed by the microcontroller. Then, the data will enter the database and appear in the Android application system. When the essay pulse sensor has detected a heartbeat and then inputs other data, namely age and sports activity, these three data will be processed by implementing BPNN to produce an output in the form of whether the heart condition is normal or abnormal, after that a notification will appear in the android application.

MCU Node Testing Results can be seen in Figure 5.

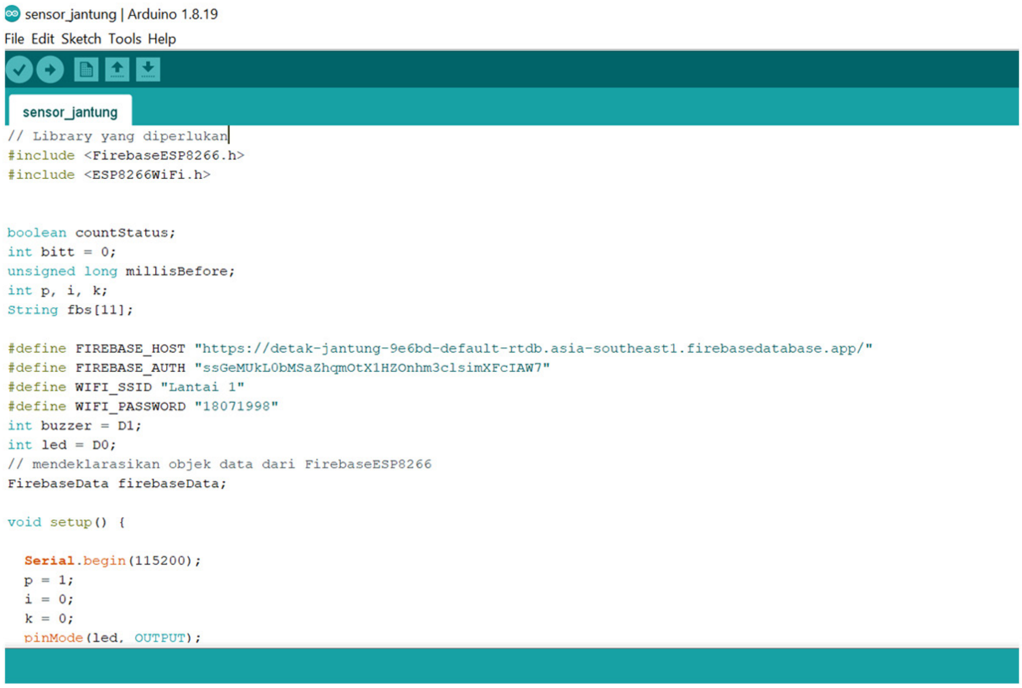

Fig. 5. MCU node compile process

In the software testing process, each goes through two processes, namely the compiler process and the upload process. No more errors exist, allowing the listing to execute as intended on the established system.

The real-time detection system, which was developed, was evaluated using 40 samples of medical record data from patients diagnosed by a heart specialist. The simulation results affirm the effectiveness of the proposed method in fulfilling the specified objectives.

Experimental verification of the theoretical results is evident in Table 1. In this table, a system test was conducted on 40 samples, revealing 2 samples with abnormal heart diagnoses, while the remaining 38 samples were diagnosed with normal hearts. These results were corroborated by the data obtained from the doctor's diagnosis of these patients.

Testing and Measurement of the Heart Detection Prototype can be seen in Table 1.

Table 1. Testing and measurement of the heart detection prototype

Respondent	Activity	Age	Tension/ Oximeter Measurement	Prototype Measurement	Classification Results
User1	No sports	36	72 bpm	74 bpm	Normal
User2	No sports	21	74 bpm	72 bpm	Normal
User3	No sports	20	74 bpm	74 bpm	Normal
User4	No sports	36	86 bpm	72 bpm	Normal
User5	No sports	36	64 bpm	62 bpm	Normal
User6	No sports	50	90 bpm	91 bpm	Normal
User7	No sports	50	91 bpm	94 bpm	Normal
User8	No sports	35	76 bpm	86 bpm	Normal
User9	After exercise	24	125 bpm	126 bpm	Normal
User10	No sports	47	90 bpm	92 bpm	Normal
User11	After exercise	20	92 bpm	94 bpm	Abnormal
User12	No sports	22	85 bpm	84 bpm	Normal
User13	No sports	22	95 bpm	96 bpm	Normal
User14	No sports	21	66 bpm	64 bpm	Normal
User15	No sports	22	84 bpm	84 bpm	Normal
User16	No sports	22	94 bpm	96 bpm	Normal
User17	After exercise	22	110 bpm	109 bpm	Normal
User18	No sports	20	80 bpm	83 bpm	Normal
User19	No sports	19	75 bpm	76 bpm	Normal
User20	After exercise	22	110 bpm	112 bpm	Normal
User21	After exercise	22	114 bpm	112 bpm	Normal
User22	After exercise	22	121 bpm	120 bpm	Normal
User23	After exercise	23	113 bpm	115 bpm	Normal
User24	After exercise	25	110 bpm	114 bpm	Normal
User25	After exercise	22	122 bpm	125 bpm	Normal
User26	After exercise	21	125 bpm	123 bpm	Normal
User27	After exercise	27	125 bpm	128 bpm	Normal
User28	After exercise	27	113 bpm	115 bpm	Normal
User29	After exercise	27	113 bpm	110 bpm	Normal
User30	After exercise	22	112 bpm	109 bpm	Normal
User31	After exercise	23	120 bpm	117 bpm	Normal
User32	After exercise	20	120 bpm	120 bpm	Normal
User33	After exercise	20	115 bpm	118 bpm	Normal
User34	After exercise	29	110 bpm	110 bpm	Normal

(Continued)

Table 1. Testing and measurement of the heart detection prototype *(Continued)*

Respondent	Activity	Age	Tension/ Oximeter Measurement	Prototype Measurement	Classification Results
User35	After exercise	27	118 bpm	115 bpm	Normal
User36	After exercise	21	120 bpm	122 bpm	Normal
User37	After exercise	24	116 bpm	119 bpm	Normal
User38	After exercise	25	89 bpm	90 bpm	Abnormal
User39	After exercise	26	122 bpm	121 bpm	Normal
User40	After exercise	27	112 bpm	110 bpm	Normal

Android Application Testing Results can be seen in Figure 6.

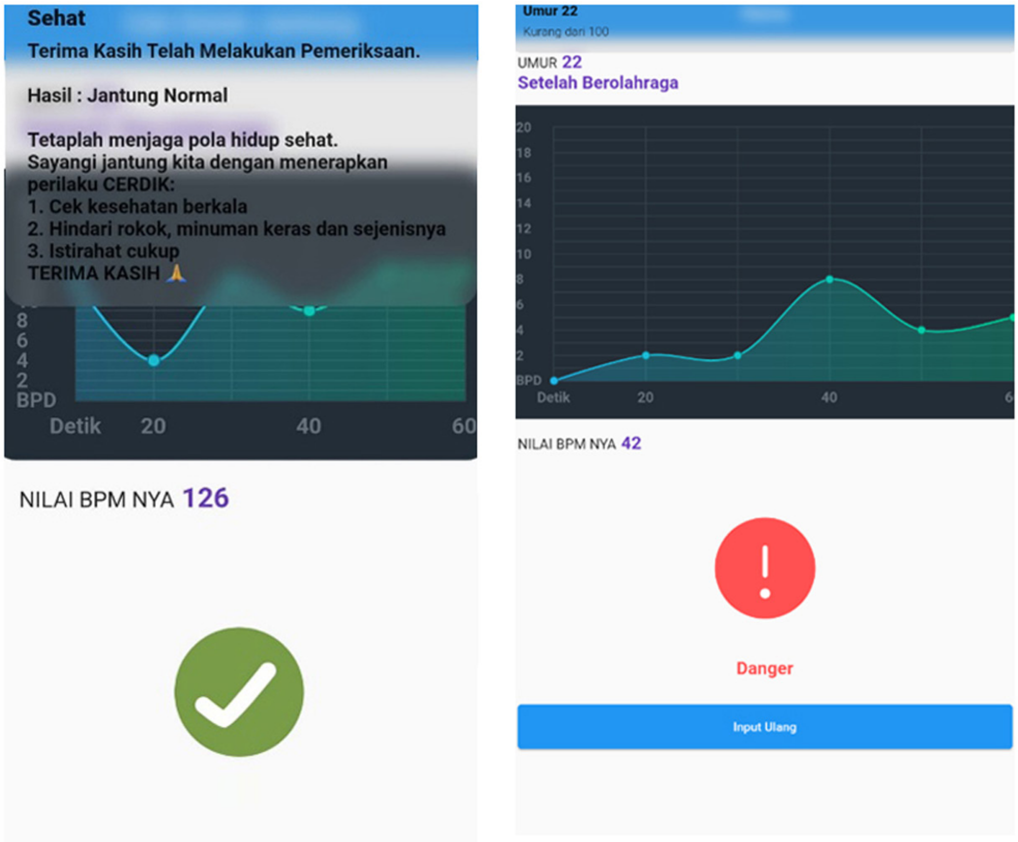

Fig. 6. Heartbeat detection results

Programs run in the Arduino IDE can connect and upload data from the tool to the Firebase database so that the Android application can be used as a monitoring application and display data from the compiler process in the Arduino IDE.

Implementing the real-time detection of Abnormal Heart Beat Using Multiple BPNN methods enables individuals to monitor their cardiac well-being autonomously. However, a notable drawback of the prototype lies in its reliance on electrical resources during operation, necessitating the availability of a power bank or a durable battery. Furthermore, the prototype's functionality heavily hinges on network connectivity, mandating a stable network environment for optimal performance.

Implementing the real-time detection of Abnormal Heart Beat Using Multiple BPNN methods enables individuals to monitor their cardiac well-being autonomously. However, a notable drawback of the prototype lies in its reliance on electrical resources during operation, necessitating the availability of a power bank or a durable battery. Furthermore, the prototype's functionality heavily hinges on network connectivity, mandating a stable network environment for optimal performance.

6 CONCLUSION AND FUTURE WORKS

The main objective of this research is to design an Android-based heart detection prototype and build a system by implementing the BPNN method to detect a person's heart health using input variables generated by the prototype and two other variables, namely age and sports activity. We have succeeded in achieving that goal. This system can improve health services in various public health service centers and independently detect heart health early.

For future research, a combination of other classification methods can be applied, and the data variables can be added to produce more accurate output. In addition, construct an IoT-centric application that integrates the prototype with a mobile application for a more flexible Detection of Abnormal Heart Beat.

7 REFERENCES

[1] R. S. Kusuma, M. Pamungkasty, F. S. Akbaruddin, and U. Fadlilah, "Prototipe alat monitoring kesehatan jantung berbasis IoT," *Emitor: Jurnal Teknik Elektro*, vol. 18, no. 2, pp. 59–63, 2018. https://doi.org/10.23917/emitor.v18i2.6353

[2] P. Prasetyawan, S. Samsugi, and R. Prabowo, "Internet of Thing Menggunakan Firebase dan Nodemcu untuk Helm Pintar," *Jurnal ELTIKOM: Jurnal Teknik Elektro, Teknologi Informasi Dan Komputer*, vol. 5, no. 1, pp. 32–39, 2021. https://doi.org/10.31961/eltikom.v5i1.239

[3] A. Ahmad, "Mengenal artificial intelligence, machine learning, neural network, dan deep learning," *J. Teknol. Indones.*, no. October, p. 3, 2017.

[4] F. Jefansa, "Penerapan Jaringan Syaraf Tiruan dalam Meramalkan Produksi Kopi Berdasarkan Provinsi," *Jurnal Infomedia: Teknik Informatika, Multimedia & Jaringan*, vol. 7, no. 1, pp. 1–7, 2022. https://doi.org/10.30811/jim.v7i1.2873

[5] Y. Yuhefizar, A. Nasution, R. Putra, E. Asri, and D. Satria, "Alat Monitoring Detak Jantung Untuk Pasien Beresiko Berbasis IoT Memanfaatkan Aplikasi OpenSID berbasis Web," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 3, no. 2, pp. 265–270, 2019. https://doi.org/10.29207/resti.v3i2.974

[6] N. A. Yosephine, "Penggunaan Artificial Neural Network pada Sinyal Elektrokardiogram untuk Mendeteksi Penyakit Jantung Aritmia Supraventrikular," *INFORMASI (Jurnal Informatika dan Sistem Informasi)*, vol. 13, no. 1, pp. 14–23, 2021. https://doi.org/10.37424/informasi.v13i1.69

[7] B. A. Moro, A. Febtriko, and D. Arisandi, "ALAT PENDETEKSI KONDISI JANTUNG MENGGUNAKAN Arduino Uno DAN Pulse Sensor," in *Prosiding CELSciTech*, 2017, vol. 2, p. tech-25–tech-31.

[8] M. A. Alsharaiah *et al.*, "Neural network prediction model to explore complex nonlinear behavior in dynamic biological network," *International Journal of Interactive Mobile Technologies*, vol. 16, no. 12, 2022. https://doi.org/10.3991/ijim.v16i12.30467

[9] J. Aminuddin and B. Bilalodin, “Development and testing of artificial neural network with backpropagation algorithm to predict the power ratio of savonius wind turbine,” in *Soedirman International Conference on Mathematics and Applied Sciences (SICOMAS 2021)*, Atlantis Press, 2022, pp. 84–90. https://doi.org/10.2991/apr.k.220503.018
[10] Z. Li, “A data classification algorithm of internet of things based on neural network,” *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 13, no. 9, pp. 28–37, 2017. https://doi.org/10.3991/ijoe.v13i09.7587
[11] B. Bai, J. Zhang, X. Wu, G. wei Zhu, and X. Li, “Reliability prediction-based improved dynamic weight particle swarm optimization and back propagation neural network in engineering systems,” *Expert Syst Appl*, vol. 177, p. 114952, 2021. https://doi.org/10.1016/j.eswa.2021.114952
[12] G. Zhou, Y. Ji, X. Chen, and F. Zhang, “Artificial neural networks and the mass appraisal of real estate,” *International Journal of Interactive Mobile Technologies*, vol. 14, no. 3, pp. 180–187, 2018. https://doi.org/10.3991/ijoe.v14i03.8420
[13] Z. Hasanati and D. Meidelfi, “Kajian Implementasi Jaringan Syaraf Tiruan Metode Backpropagation Untuk Deteksi Bau,” *Journal of Applied Computer Science and Technology*, vol. 1, no. 2, pp. 90–95, 2020. https://doi.org/10.52158/jacost.v1i2.113

8 AUTHORS

Suryani is a lecturer in Department of Informatics Engineering, Dipa Makassar University, Makassar, Indonesia (E-mail: suryani187@undipa.ac.id).

Faizal is a lecturer in Department of Informatics Engineering, Dipa Makassar University, Makassar, Indonesia (E-mail: f41241@undipa.ac.id).