

Classification of Types A and B Volcanic Earthquakes using Neural Network for Monitoring Mount Merapi Activities

Dwi Lastomo¹, Wisnu Waskito Aji², Lalu Muhammad Jaelani³, Agus Budi Santoso⁴, Arif Musthofa⁵,

¹*Electrical Engineering, Universitas Kebangsaan Republik Indonesia, Indonesia*

^{2,5}*Electrical Automation Engineering, Institut Teknologi Sepuluh Nopember, Indonesia*

³*Department of Geomatics Engineering, Institut Teknologi Sepuluh Nopember, Indonesia*

⁴*BPPTKG, Ministry of Energy & Minerals Resources, Indonesia*

Abstract—Mount Merapi is one of the most active volcanoes in the world. Seismic activity at Mount Merapi was divided into tectonic, volcanic A and B, avalanches, and multiphase. At the volcanology and mitigation station in the area of Mount Merapi, the vibrations received by the seismometer installed on the mountain were then transmitted to the station to be interpreted in the form of a seismograph. Determination of the type of earthquake at the Mount Merapi station was done manually by analyzing the shape of the wave formed. Utilizing the earthquake waveform, it could be used to determine the type of earthquake using the Nntool toolbox in Matlab. So that the determination of the type of earthquake no longer needed to be done manually. The classification process began with training the system to understand earthquake-type classes. After the system understood the earthquake data belonging to each category, the classification process could be carried out. The types of earthquakes analyzed in this research were volcanic earthquakes of types A and B. The results of these classifications were used as a determinant of the type of earthquake that occurred. The results were obtained in the form of an introduction to the types of volcanic earthquakes type A and type B which were carried out automatically with a total accuracy of 91.83%. Type A earthquake recognition accuracy was 100%, type B earthquake recognition accuracy was 84.21%, the earthquake recognition accuracy of non-A & B type was 94.44%.

Keywords—Artificial Intelligence, Classification, Mount Merapi, Neural Network, and Volcanic Earthquake

I. INTRODUCTION

Mount Merapi is administratively located between the Province of the Special Region of Yogyakarta (DIY) and the Province of Central Java. To be precise, it limits four regencies, namely Sleman Regency and Magelang Regency, Boyolali Regency, and Klaten Regency. With the intensity of eruptions every 2-7 years, Mount Merapi is one of the most active volcanoes in the world and increases the risk of disaster in the surrounding area. The National Disaster Management Agency (BNPB) recorded that the victims of the 2010 eruption reached 347 people. The eruption of Mount Merapi also

resulted in material losses in various sectors such as settlements, infrastructure, telecommunications, electricity, and clean water [1].

Therefore, the mitigation system becomes a great concern to governments, non-governmental organizations, and people in general. The mitigation system according to Law Number 24 of 2007 is a series of efforts to reduce disaster risk, both through physical development as well as awareness and capacity building to face disaster threats [2].

The upstream mitigation system is volcanic activity monitoring (monitoring) because all mitigation actions are carried out based on the state of volcanic activity and the predicted scenario. Prediction of volcanic activity is usually based on two aspects, namely historical data from volcanic activity and current activity data. There, the important role of the volcano monitoring system is needed. Any volcanic activity monitoring system aims to identify the current level of volcanic activity (normal, standby, cautious, alert), identify the current process and predict the next process, predicting potential eruptions including time and magnitude, and understand volcanic activity/eruptive characteristics and cycles.

Until now, monitoring seismic activity (seismic) on Mount Merapi is the most powerful method to determine the level of activity of Mount Merapi. Seismic monitoring is carried out with seismometers installed at 4 stations around Merapi, namely Pusunglondon, Klatakan, Plawangan, and Deles. The signal recorded by the seismometer is sent to the receiving station at the Jogjakarta BPPTKG Office using VHF radio telemetry. At the receiving station, this signal is then recorded on seismogram paper, and also stored in digital data for analysis [3]. One method for understanding and predicting volcanic activity is earthquake classification and counting the number of each type of earthquake. In the case of the eruption of Merapi, the increasing number of Volcano-tectonic (VT) earthquakes is a reference for increased volcanic activity.

Ratdompurbo (1995) classified volcanic earthquakes based on shape and frequency into 7 types. The types of earthquakes are volcanic type-A, volcanic type B, multiphase,

Corresponding Author:

Dwi Lastomo
Electrical Engineering, Universitas Kebangsaan Republik Indonesia

dtomo23@gmail.com

low-frequency event (LF), low high-frequency event (LHF), and avalanches [4]. Currently, the classification of earthquakes carried out by the Yogyakarta Geological Disaster Technology Research and Development Center (BPPTKG) is still manual, namely by direct observation on a seismometer. To facilitate the process, a system is needed to perform earthquake classification automatically.

Matlab is software that is suitable for use as a computational tool involving matrices and columns. The function of Artificial Neural Networks (ANN/JST) in Matlab can be used for information processing with characteristics similar to biological neural networks. Because of the ability that is close to human biological neurons, artificial neural networks can be applied to perform classification. This classification capability can be used to identify vibration data/earthquake signals at Mount Merapi so that the Volcano-tectonic (VT) earthquake classification results are obtained automatically.

II. FUNDAMENTAL THEORY

A. Monitoring the Seismicity of Mount Merapi

Seismicity that occurs due to the activity of a volcano is one of the most important monitoring parameters. Monitoring of the seismicity of Mount Merapi is carried out by BPPTKG through seismic data obtained in the field [4].



Figure 1. Monitoring the seismicity of Mount Merapi

Figure 1 is a seismic monitoring system at Mount Merapi. Seismic monitoring with analog seismic telemetry consists of two parts, namely the field system and the receiving system. The main components of the field system are a seismometer (4) and a VCO-amplifier (5) which function to amplify the signal and convert the voltage into a frequency that will be superimposed on a radio carrier wave (6) with a yagi antenna (7). The power supply consists of a solar panel (1) and a regulator (2) to disconnect and connect the current from the battery (3) to the solar panel. At the receiving station, the signal will be received by the radio receiver (a) which is then forwarded to the discriminator (b). From here the signal can be routed directly to a recorder (seismograph) (d) or stored and displayed digitally on a PC (e) with the help of an ADC (analog to digital converter) (c).

B. Volcanic Earthquake

Volcanic earthquakes are earthquakes caused by the movement of magma out of the volcano. This magma activity causes ground vibrations to be felt only in the area around the mountain. Volcanic earthquakes can be used as an early sign of increased volcanic activity [5].

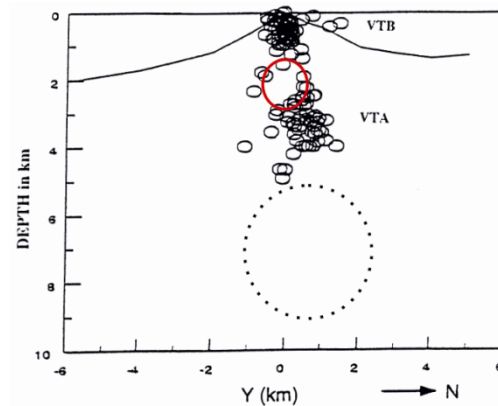


Figure 2. Volcanic earthquake mechanism

Figure 2 describes the types of volcanic earthquakes type A and B that often occur when Mount Merapi approaches eruption. The primary magma chamber of Mount Merapi is estimated to be located at a depth of more than 5 kilometers below the mountain peak and the secondary magma chamber at a depth of between 1.5 – 2.5 km. At Mount Merapi, volcanic earthquakes are divided based on earthquake signal data from a network of seismic stations. According to Ratdomopurbo & Poupinet (2000), the grouping of earthquake types in Mount Merapi consists of volcanic type-A earthquakes, volcanic types B, multiphase earthquakes, low-frequency earthquakes, LHF earthquakes (low high frequency), and tremors earthquakes.

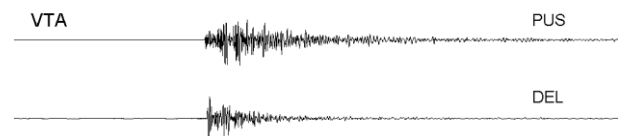


Figure 3. Type A volcanic earthquake record

Figure 3 is an example of a deep volcanic earthquake recording at the Pusunglondon (PUS) and Deles (DEL) monitoring stations. Deep volcanic earthquakes (type A) are characterized by a hypocenter (earthquake epicenter) that occurs at a depth of more than 2 km below the summit of the volcano, usually appearing on an active volcano. The cause of this earthquake is the magma that rises to the surface accompanied by fractures. The main characteristic of a type-A volcanic earthquake is that it has a very clear primary (longitudinal) and secondary (transverse) wave arrival time [5].

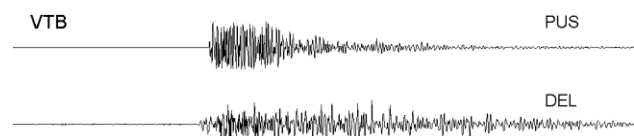


Figure 4. Type B volcanic earthquake record

Figure 4 is an example of a recording of a type B volcanic earthquake at the Pusunglondon (PUS) and Deles (DEL) monitoring stations. Shallow volcanic earthquakes (Type B) are less than 2 km deep and have more seismometer arrival times for seismic waves that appear at stations farther from the summit. In some cases, secondary waves (transverse waves) are indistinguishable from primary waves (longitudinal waves) [3].

III. SYSTEM PLANING AND DEVELOPMENT

A. Classification System Design

This section explained the description of the process flow that occurred with the proposed method to produce output. The design determination of this application described what and how the proposed method was applied in solving the problem. The stages of the earthquake classification process can be seen in Figure 5.

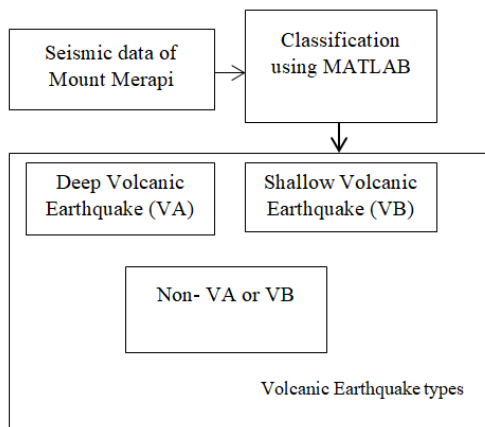


Figure 5. Earthquake type clasification process

Figure 5 describes the process of classifying volcanic earthquakes in monitoring volcanic activity. Starting from the seismic data of Mount Merapi in the form of data from the seismometer which was converted into digital data to be processed with Matlab software. Then built coding to classify the type of earthquake. In the output section, the results were in the form of a classification of volcanic earthquake types on volcanoes, namely deep volcanic earthquakes (VA), shallow volcanic earthquakes (VB), or not both types of earthquakes.

B. ANN Training Method Design

Artificial Neural Network (ANN) architecture was influenced by several factors, including the number of inputs, hidden screens, weights, and outputs. In this Final Project, supervised learning was carried out first to produce a pattern from each monitoring data with a high level of accuracy.

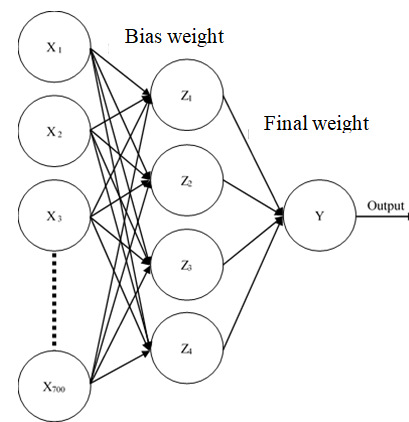


Figure 6. Backpropagation ANN architecture

Figure 6 is the backpropagation ANN architecture. The network architecture in the backpropagation ANN method for the classification of the type of Mount Merapi earthquake had 3 main parts, namely input (X), hidden layer (Z), and output (Y). Each input neuron (X) was connected to each neuron in the hidden layer (Z) with the bias weight of each connected neuron having a different value. All neurons in the hidden layer (Z) were connected to the output layer (Y) with a different final weight for each neuron. The propagation from the input to the output was called feedforward propagation if the value at Y was not appropriate then backpropagation was carried out (backpropagation) so that the appropriate weight value was obtained for the output in the form of an earthquake type.

	1	2	3	4	5	6	7	8	9
1	49x1 struct	39x1 struct	38x1 struct	21x1 struct	21x1 struct	15x1 struct	14x1 struct	12x1 struct	5x1 struct

Figure 7. Class o each type of earthquake

Figure 7 is a class for each type of earthquake. Columns 1 and 2 were non-earthquake classes of type A and B. Columns 4, 5, and 9 were earthquake classes of type A. Columns 3,6,7, and 8 were earthquake classes of type B.

Fields	scnl	Fs	start	data	units	version	misc_fields	misc_value
1	1x1 struct	50	7.3440e+05	700x1 double	'Counts'	1.2000		
2	1x1 struct	50	7.3440e+05	700x1 double	'Counts'	1.2000		
3	1x1 struct	50	7.3440e+05	700x1 double	'Counts'	1.2000		
4	1x1 struct	50	7.3440e+05	700x1 double	'Counts'	1.2000		
5	1x1 struct	50	7.3440e+05	700x1 double	'Counts'	1.2000		
6	1x1 struct	50	7.3442e+05	700x1 double	'Counts'	1.2000		
7	1x1 struct	50	7.3442e+05	700x1 double	'Counts'	1.2000		
8	1x1 struct	50	7.3442e+05	700x1 double	'Counts'	1.2000		
9	1x1 struct	50	7.3442e+05	700x1 double	'Counts'	1.2000		
10	1x1 struct	50	7.3442e+05	700x1 double	'Counts'	1.2000		
11	1x1 struct	50	7.3442e+05	700x1 double	'Counts'	1.2000		
12	1x1 struct	50	7.3442e+05	700x1 double	'Counts'	1.2000		

Figure 8. Earthquake class

Figure 8 is an example of an earthquake class. Wf {1,1} in the table title shows the row in the earthquake class (1) then the selected earthquake class column (1) In the data column contains earthquake amplitude data that had been sampled every 700 seconds. The field column shows the sequence of earthquake events. The number of rows in the earthquake class shows the number of earthquake events in each earthquake

class. In each earthquake class data was divided into training data and test data. Training data was used to train the system and test data was used to test the system. The amount of test data could be set according to the user's wishes. To obtain a more accurate system, the amount of training data was usually more than the amount of test data.

Table 1. Total training and test data

Earthquake Class	Amount of Training Data	Amount of Test Data
Wf {1,1}	40	9
Wf {1,2}	30	9
Earthquake Class	Amount of Training Data	Amount of Test Data
Wf {1,3}	30	8
Wf {1,4}	15	6
Wf {1,5}	15	6
Wf {1,6}	10	5
Wf {1,7}	10	4
Wf {1,8}	10	2
Wf {1,9}	5	0
Total	165	49

Table 1 is the amount of training data and test data for each earthquake class. The number of training data used was 165 data and 49 test data so that the total data used in this Final Project was 214 earthquake data.

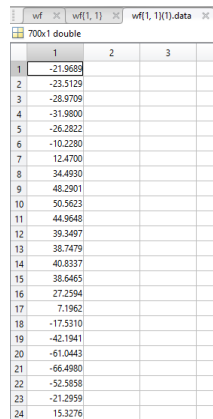


Figure 9. Preview of earthquake data sampling results

Figure 9 is a preview of the results of earthquake data sampling. Column 1 row 1 shows the magnitude of the amplitude in the first second. The number of rows indicates the amount of data to form one earthquake signal. In this final project, the number of data used for the classification of one earthquake event was 700 data. From this 700 data, it was used as an input unit in the earthquake type classification system using Matlab.

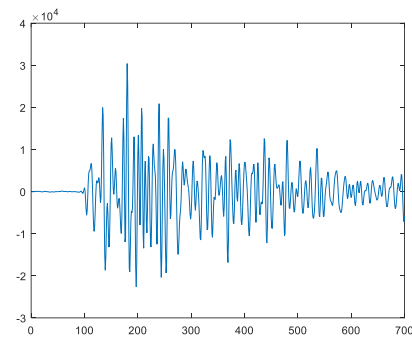


Figure 10. Earthquake recording data plot

Figure 10 is a plot of earthquake recording data. The horizontal section shows the time for an earthquake event while the vertical section shows the magnitude of the amplitude.

Hidden Layer was an independent variable so that users could change as needed. The number of hidden layers could affect the system's classification capability. The output of the ANN was the output data of the classification results which were compared with the target data.

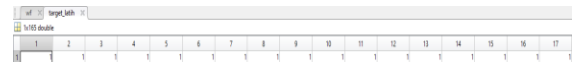


Figure 11. Training target

Figure 11 is the training target. The training target was the class that should be obtained in the output unit. A value of 1 was the initialization of the target class for an earthquake of type A, a value of 2 for the initialization of an earthquake of type B, and a value of 0 for the initialization of classes for non-A and B types.

C. Training Process

The training was the initial procedure that must be carried out on an Artificial Neural Network (ANN) to obtain the best ability from the ANN architecture. The training flow is depicted as in Figure 12.

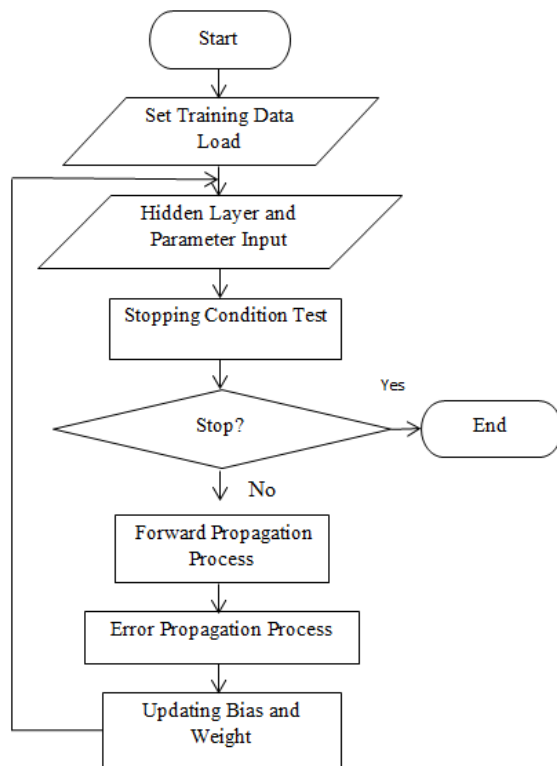


Figure 12. Flowchart of backpropagation ANN training process

Figure 12 is a flow diagram of the backpropagation ANN training process. The process began by entering monitoring data into the system (in Matlab). Then entered the number of hidden layers, the type of activation function, and parameters covering error goal, maximum epoch, learning rate, and training function. After that, the stopping condition test process was carried out, namely testing to get the best weight on each neuron so that the best network capability was obtained. The test began with forward-propagation, then the system performed an error propagation process, namely calculating the error in reading the data. If the error was still high, the system would carry out the process of updating the bias and weights to test the stopping condition again. The process of feedforward and back-propagation ANN took place repeatedly until the training objectives were achieved, namely the number of epochs (the number of repetitions of the feedforward process and back-propagation ANN) was greater than or equal to the maximum epoch, or the MSE (Mean Square Error) value was smaller or equal to the target error. In addition, in the backpropagation neural artificial network, an activation function was used which served to determine the output of a neuron or network. The training weights and network architecture that had achieved high accuracy were later be used to carry out the testing process.

3.4. Testing Process

System testing was carried out using a backpropagation artificial neural network. The weight value of the artificial neural network used was the weight value at the time of training. The system testing process aimed to test the accuracy of the training process. The final weight value in the training process was used to test the data sets that had been prepared,

which were 49 data sets for volcanic earthquakes of type A, volcanic types B, and not volcanic earthquakes of type A or B.

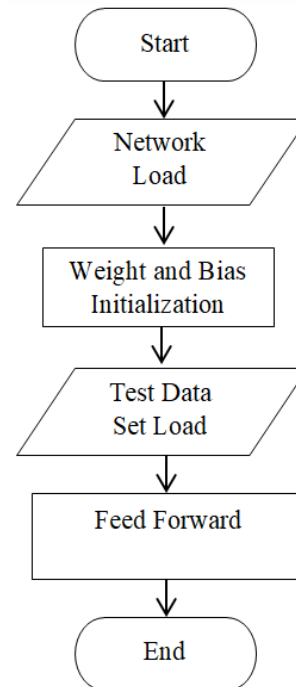


Figure 13. Flowchart of the testing process

The flow of the backpropagation neural network testing process is as depicted in Figure 3.10. The process began by loading the network architecture of the training process. Then the system initialized the weights and biases. After that entered the test data set. Followed by the feed-forward process in the testing process, namely the system would read the data with the network architecture that was formed.

In the process of testing this system, the level of accuracy was generated as a measure of the success of the system. The level of accuracy in the testing process could be calculated through the equation:

$$accuracy = \frac{total\ number\ of\ data - number\ of\ incorrect\ data}{total\ data} \times 100\% \quad (1)$$

IV. RESULTS AND DISCUSSION

A. System Display Analysis

The main menu interface displays at a glance title information about the program being created.

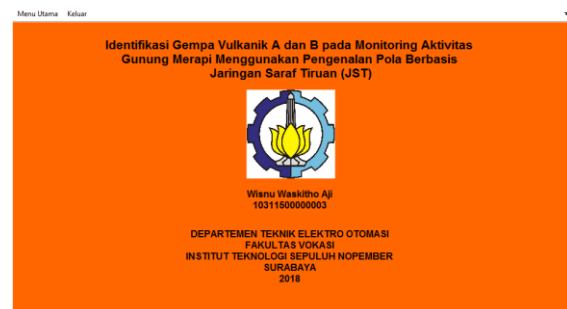


Figure 14. System main page display

Figure 14 is a display of the main system page. There are 2 sub-menus that users can choose according to their needs. In the "Main Menu" sub, when pressed it will display the "Training" and "Test" options.

The training interface is a window used to input the data needed for the training process on the system that is formed.

Figure 15. Training window display

Figure 15 is a display of the training window. In the training window, there are several sections, namely "Weights", "Load Data", "Hidden Layer", "Output Layer", "Parameters", "Data Processing", and "Training Accuracy". The following is an explanation of each part and how it operates.

Weight. This section contains 2 button options, namely "Random" to display the network weights randomly and the "Load Weight" option to call up the previously stored training weights.

Load Data. Load Data has 6 selection buttons, namely "Load Data", "Load Weights", "Training", "Training Results", "Save Network", and "Save Weights". When the button in the weight selection is random, in the load data section, the load data button option is active to call the data used for training. Meanwhile, if in the weight section, load weight is selected, the weight load button will be active which functions to call the training weights. The training button is used to start the system training process. The training results button is used to display the training results on the training accuracy panel. The save network button functions to save the network architecture in the form of parameters in the hidden layer panel, output layer, and parameter panel to be called during the testing process. The save button is used to save the weight of the training process results.

Hidden Layer. The hidden layer panel is useful for setting the parameters contained in the hidden screen. In the edit text the number of neurons, the user can enter the number of neurons that appear on the hidden screen. The more neurons that are entered, the longer the system time to carry out the training process. The activation function pop-up menu has two options, namely binary sigmoid and bipolar sigmoid.

Output Layer. Based on the architecture built in this final project as shown in Figure 3.4, the number of neurons in the output screen is only one. Therefore, in the output layer, only the type of activation function of the output layer can be changed. The activation function pop-up menu has 3 options, namely linear, binary sigmoid, and bipolar sigmoid.

Parameters. After setting the hidden screen and output screen, before the training process is carried out, it is better to

set the training parameters. The parameter section has 4 sections that can be set by the user. The first is the training function, which is a function to minimize performance. There is a choice of traingdx (gradient descent with momentum and adaptive learning rate), traingd (gradient descent), trainlm (Levenberg-Marquardt Backpropagation), trainrp (Resilient Propagation), traingda (gradient descent with adaptive learning rate). Next is setting the error goal or the least number of errors that make the system declared complete training. User-selectable error goals from 1e-6 (0.000001) to 1e-3 (0.001). The next edit text that can be adjusted is the number of steps in learning both forward (feedforward) and (backpropagation) or called epochs. The last parameter is the learning rate which can be set by the user (usually using 0.1 or 0.01).

Data Processing. The data processing section is the section that displays the training data. There is an inactive edit text in the path section to show the training data directory on the laptop. The table in the data processing section displays training data. The table column shows the data for one earthquake event and the row shows the number of earthquake events.

Training Accuracy. Training accuracy is the part that displays the training results data after the training results button on the data load is pressed. All text input is inactive because it only serves to display data. The total amount of data indicates the amount of data used to train the system. The number of correct data is the number of data that has the same output class as the target class. While the number of incorrect data is the amount of data that has a different output class from the target class. The training accuracy displays the percentage of the system's accuracy in recognizing the pattern being trained. Accuracy is obtained through Equation 2:

$$\text{accuracy} = \frac{\text{number of correct data}}{\text{number of incorrect data}} \times 100\% \quad (2)$$

The test interface is a window that is used to carry out the system testing process. The testing process is carried out after the training process is carried out.

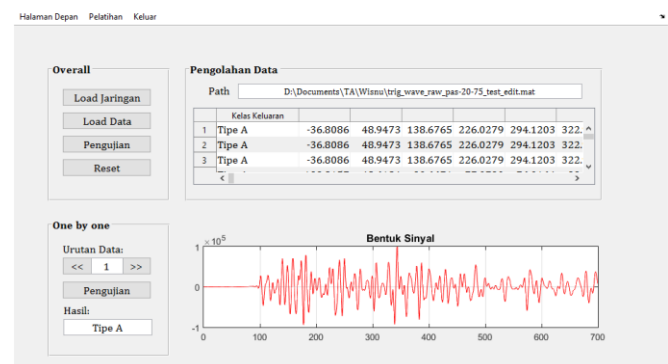


Figure 16. Test window display

Figure 16 is the display of the test window after the testing process is complete. There are 4 sections in the window, namely "Overall", "Data Processing", "One by One", and "Signal Form". The following is an explanation of each part and how it operates.

Overall. The overall section contains buttons for carrying out the testing process. There are 4 buttons namely "Load Network", "Load Data", "Test", and "Reset". Network load is a

button to call the network architecture and the resulting weights. The network architecture includes the number of neurons, activation functions, epochs, and other parameters contained in the training process. The load data button is used to call data that will be used for system testing. The test button is used to perform the system testing process. And the reset button is used to reset the testing process.

Data processing. In the data processing section, in the path writing there is edit text that has been disabled so that it only has a function to display the location of the test data taken on the laptop. The data processing section also displays a test data table where the "Output Class" column is the result of the classification of earthquake types carried out by the system. The row in the table shows the data used for testing and the column shows the input data to the system for one earthquake event.

One by One . The one-by-one section serves to display the test results one by one for earthquake events. In this section, there are left and right push buttons to shift the earthquake data displayed. Inactive edit text shows the currently displayed earthquake data. The "Testing" button has a function to perform testing on one earthquake data and the inactive edit text displays the type of earthquake for one data selected by the user.

Signal Form. The signal form section will contain a table that displays the earthquake shape plot based on the data in the data processing table for one earthquake event. The signal form will be displayed when the test button in the one-by-one section is pressed.

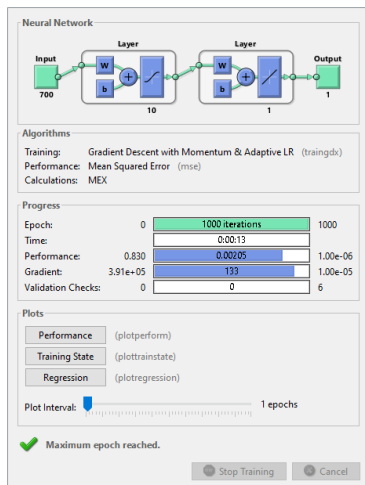


Figure 17. Display of system training process

Figure 17 is a display of the system training process. It can be seen that the system stops after reaching the 1000th epoch and has not yet reached the error goal set by the user, which is $10e-6$. In this experiment, the type of training used was Gradient Descent with Momentum and Adaptive Learning (Trainingdx). Performance, which states the method to stop the process if it was close to Real was the Mean Squared Error (MSE). Epoch, states the number of repetitions of the training. Time, states the time taken by Matlab to conduct training. Performance, states the quality of the training results where the closer to 0 the better the quality. The gradient is the slope between one iteration and the next. Validation Check is used to check the training process leads in the right direction or deviates. In the plot section, there are 3 buttons, namely

"Performance", "Training State", and "Regression". The three buttons are useful for displaying a graph of learning outcomes.

B. Analysis of ANN Training Parameters

ANN training parameter analysis was carried out to obtain the best weights by changing the parameters used in the training process or by repeating the training process. The system was set with the parameters:

Number of neurons in the hidden layer: 10

The activation function in hidden layer: sigmoid bipolar

Number of neurons in the output layer: 1

Activation function on output layer: linear

Training Function: Trainingdx

Error Goal: $1e-6$

Epochs: 1000

Learning Rate: 0.1

In the training process, the system accuracy was obtained by 100%. Then testing the system built with these parameters, the results obtained were:

Table 2. Test results

Order of Test Data	Target Class	Test Results
1 st	Type A	Type A
2 nd	Type A	Type A
3 rd	Type A	Type A
The 4 th	Type A	Type A
The 5 th	Type A	Type A
The 6 th	Type A	Type A
The 7 th	Type A	Type A
The 8 th	Type A	Type A
The 9 th	Type A	Type A
The 10 th	Type A	Type A
The 11 th	Type A	Type A
The 12 th	Type A	Type A
The 13 th	Type B	Type B
The 14 th	Type B	Type B
The 15 th	Type B	Type B
The 16 th	Type B	Not Type A&B
The 17 th	Type B	Type B
The 18 th	Type B	Type B
The 19 th	Type B	Type B
The 20 th	Type B	Type A
The 21 st	Type B	Type B
Order of the Test Data	Target Class	Test Results
The 22 nd	Type B	Type B
The 23 rd	Type B	Type B
The 24 th	Type B	Type B
The 25 th	Type B	Type B
The 26 th	Type B	Type A
The 27 th	Type B	Type B
The 28 th	Type B	Type B
The 29 th	Type B	Type B
The 30 th	Type B	Type B
The 31 st	Type B	Type B
The 32 nd	Not Type A&B	Not Type A&B
The 33 rd	Not Type A&B	Type A

The 34 th	Not Type A&B	Not Type A&B
The 35 th	Not Type A&B	Not Type A&B
The 36 th	Not Type A&B	Not Type A&B
The 37 th	Not Type A&B	Not Type A&B
The 38 th	Not Type A&B	Not Type A&B
The 39 th	Not Type A&B	Not Type A&B
The 40 th	Not Type A&B	Not Type A&B
The 41 st	Not Type A&B	Not Type A&B
The 42 nd	Not Type A&B	Not Type A&B
The 43 rd	Not Type A&B	Not Type A&B
The 44 th	Not Type A&B	Not Type A&B
The 45 th	Not Type A&B	Not Type A&B
The 46 th	Not Type A&B	Not Type A&B
The 47 th	Not Type A&B	Not Type A&B
The 48 th	Not Type A&B	Not Type A&B
The 49 th	Not Type A&B	Not Type A&B

Table 2 is a table of test results. the data order column shows the order of the test data. The target class column contains the type of earthquake that the system should classify. The test results column contains the results of the classification performed by the system. The classification was declared correct if the test results were the same as the target class. The correct amount of data was used to calculate system accuracy. From Table 4.1, namely the results of system testing obtained an accuracy of:

$$\text{accuracy} = \frac{\text{total number of data} - \text{number of incorrect data}}{\text{total data}} \times 100\% \quad (3)$$

$$\text{total accuracy} = \frac{49-4}{49} \times 100\% = 91,83\% \quad (4)$$

$$\text{VTA accuracy} = \frac{12-0}{12} \times 100\% = 100\% \quad (5)$$

$$\text{VTB accuracy} = \frac{19-3}{19} \times 100\% = 84,21\% \quad (6)$$

$$\text{non - A\&B accuracy} = \frac{18-1}{18} \times 100\% = 94,44\% \quad (7)$$

The tests carried out using training weights with training accuracy of 100%, resulted from system testing with an accuracy of 91.83%, where the system could recognize type-A earthquakes of 100%, type B earthquakes of 84.21%, and non-type earthquakes A&B of 94.44%.

V. CONCLUSION

The results of the earthquake classification carried out by the artificial neural network can be classified according to the earthquake type class with an average accuracy of 91.83%, where the accuracy of the recognition of type A earthquakes is 100%, the accuracy of type B earthquakes is 84.21%, recognition of non-A&B earthquakes by 94.44%. Parameters entered in the training process affect the ability of the system.

REFERENCES

- [1] *Gunung Merapi*, pembahasan mengenai Gunung Merapi, [http://www.vsi.esdm.go.id/index.php/gunungapi/data-](http://www.vsi.esdm.go.id/index.php/gunungapi/data-dasar-gunungapi/542-g-merapi)

- dasar-gunungapi/542-g-merapi (diakses 12 April 2018)
- [2] Ariyadi Nugoho Susilo dan Iwan Rudiarto, "Analisis Tingkat Resiko Erupsi Gunung Merapi Terhadap Permukiman di Kabupaten Klaten", *Jurnal Teknik PWK vol: III no: 34-49*, 2014.
- [3] Agus Budi Santoso, "*The Seismic Activity Associated With The Large 2010 Eruption Of Merapi Volcano, Java: Source Location, Velocity Variation, And Forecasting*", *Thesis*, Université Grenoble Alpes, Grenoble, 2006
- [4] *Informasi Merapi Pemantauan*, pembahasan tentang informasi pemantauan Merapi, <http://merapi.bgl.esdm.go.id/pub/page.php?idf=11> (accessed 12 April 2018).
- [5] Sepry Dawid, Ferdy, dan Guntur Pasau, "Penentuan Lokasi Pergerakan Magma Gunung Api Soputan Berdasarkan Studi Sebaran Hiposenter Gempa Vulkanik Periode Mei 2013-Mei 2014", *Jurnal Ilmiah Sains vol: XV no: 88-93*, 2015.