

CHAPTER 4

ANALYSIS AND DESIGN

4.1 Analysis

Stock investors have problems to analyze stock price movements and predict prices. There are two ways to analyze stock price movement with fundamental analysis and technical analysis. Fundamental analysis requires a lot of data that can not be accessed by the public, while technical analysis is very difficult to calculate manually. Therefore, the author will make a program that can predict the stock price for the next day. This program will be easy to use and much faster than having to calculate manually.

The program will be made using backpropagation algorithm which is a sub chapter of artificial neural network. Backpropagation algorithm can provide a prediction based on historical data. The historical data is available at finance.yahoo.com and will be taken using webscrapping.

Steps that will be done by researchers :

1. Historical data retrieval from finance.yahoo.com

Webscrapping process will be done to retrieve data from finance.yahoo.com. Stock data to be retrieved is TLKM, HMSP, BBKA, BBNI and ASII. The data will be taken from January 2016 to November 2017. Data from January to Desember 2016 will be used as training data and Januari - November 2017 will be used as data testing. The data will be stored in text file.

2. Data normalization

The data that already obtained will be normalized. Purpose of normalization is to convert all numbers into a range of 0 to 1.

formula for data normalization:

$$x' = 0.8x \frac{(x - \text{MinValue})}{\text{MaxValue} - \text{MinValue}} - 0.1$$

x : raw data.

MinValue : the lowest value of data set.

MaxValue : the highest value of data set.

3. Backpropagation process

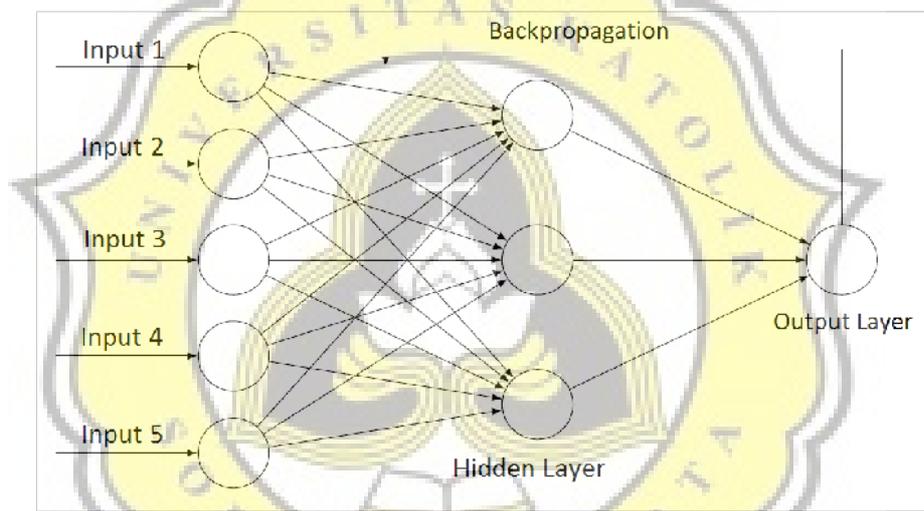


Illustration 4.1 : Diagram of Backpropagation Process

Step 0 : Initialize all weights with small random numbers.

Input bias weights (v_{0j}) = random numbers from $-\beta$ and β .

Input weight (V_{ij}) = random number of -0.5 and 0.5 .

Hidden bias weights (W_{0k}) = random numbers from -1 and 1

Hidden weights (W_{jk}) = random numbers from -1 and 1 .

Step 2 : For each training data do steps 3-8.

Step 3 : Calculate all outputs in the hidden unit ($Z_j, j = 1, \dots, p$)

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij}$$

$$z_j = f(z_in_j)$$

Step 4 : Calculate activation function on the hidden layer.

$$Z_1 = \frac{1}{1 + e^{-z_1}}$$

Step 5 : Operation in output layer.

$$y_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk}$$

$$y_k = f(y_in_k)$$

Step 6 : Calculate output layer activation function.

$$Y = \frac{1}{1 + e^{-y_in}}$$

Step 7 : Calculate δ output unit based on an error in each unit of output

$$\delta_k = (t_k - y_k) f'(y_in_k)$$

Calculate the rate of hidden weights and hidden weights with hidden learning rate α

$$\Delta w_{jk} = \alpha \delta_k z_j$$

$$\Delta w_{0k} = \alpha \delta_k$$

Step 8 : Compute δ hidden units by error in each hidden unit Z_j

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk}$$

$$\delta_j = \delta_{in_j} f'(z_{in_j})$$

Calculate the rate of change of input weight and the weight of input bias with learning rate α

$$\Delta v_{ij} = \alpha \delta_j x_i$$

$$\Delta v_{0j} = \alpha \delta_j$$

Step 9 : Calculate all weight changes

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk}$$

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij}$$

Step 10 : Test conditions of termination based on squared error.

Step 11 : Repeat step 3-10 until the squared error < 0.01

Step 12 : Calculate using another number of hidden layer cel (2,4 and 6).

4. Testing process

the results of the backpropagation will be compared with the historical data in July 2017. The accuracy of each predictions will be compared to determine which number of hidden layers cell can give the most accurate accuracy.

4.2 Desain

In this section will explain the design of the program to be created by the author. This design can provide an understanding of the program from start to finish. The program will be created using java programming language.

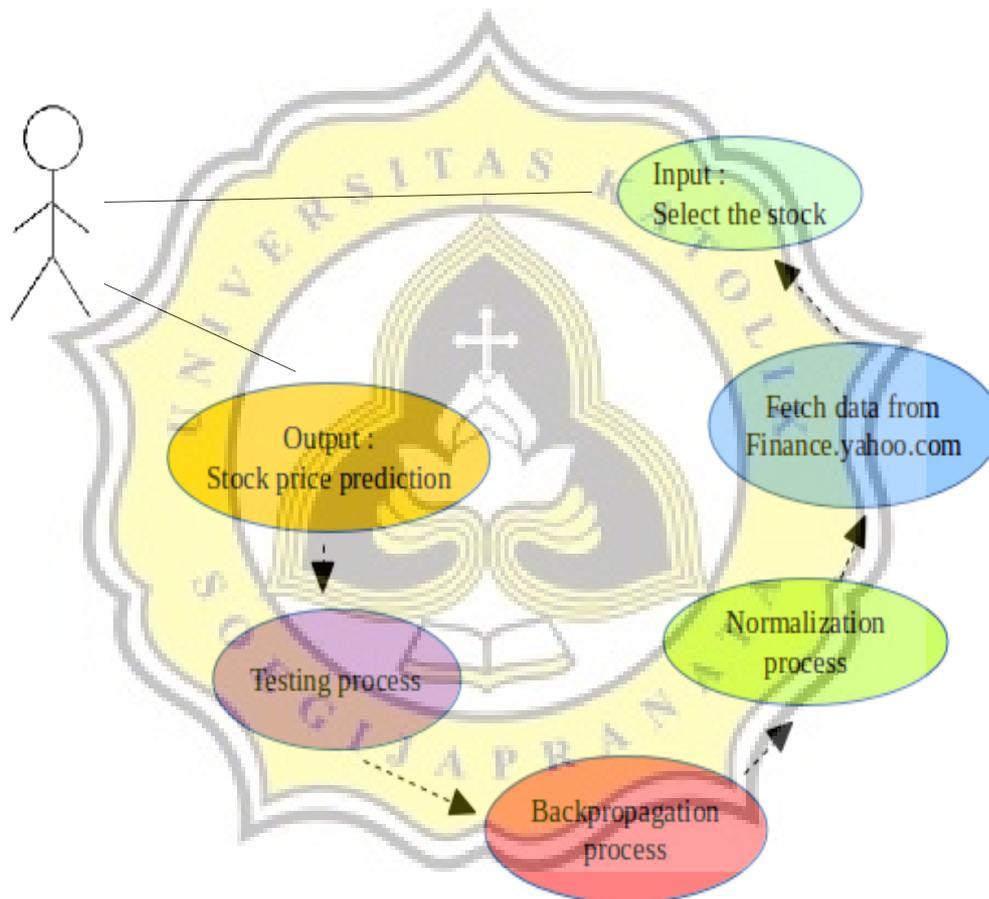


Illustration 4.2 : Use Case Diagram

Based on illustration 4.2, the program will ask the user to select the stock to be predicted. Selected stock data will be taken from finance.yahoo.com using webscrapping technique, then the fetched data will be normalized. The next

process is calculating data using backpropagation. In this process the data will be counted repeatedly. The final result is the stock price prediction.

This Page is for Class Diagram printed in A3 paper.



First, class DataSaham.java will read the list of shares owned. Then the FetchData.java class retrieves the required stock data from finance.yahoo.com. Once the data obtained is stored in the text file. From the text file the data is inserted into the linked list with the help of TextToLinkedList.java. Before entering the backpropagation formula, the linked list is normalized in the Normalisasi.java class. The data is now ready to be calculated using the backpropagation algorithm in the BackpropagationMultiLayer.java class. The result of this program is the stock price prediction of the desired date.

