CHAPTER 3
RESEARCH METHODOLOGY

This chapter describes in detail the steps taken on this project until later in the end find results that match what is done. This research stage discusses the workings of the system developed in this project. Here are some steps to take find the right and correct results.

3.1. Research Process

For the first step, it is necessary to know how deep neural network algorithms work on speech recognition implementation. So in this project, deep neural network algorithms are tasked to train the system to be able to recognize human voices and execute commands as said by humans. After that, the voice will be processed until finally the system can do what is ordered by the user.

In this project, the first process that must be done is:

1. Formulate backgrounds, objectives, scope, and problem formulation.
2. Research articles or journals related to topics conducted in this project.
3. Collecting datasets that match the project created, study the algorithms used, and find ways to be able to implement the algorithms used.
4. Analyze the problem and find a solution to the previously mentioned problem, after that build a design that is suitable for the project created.
5. Implementation and analysis of the results of the implementation and then make conclusions.

3.2. Collecting Datasets

Datasets on this project I took from Mini Speech Commands Dataset[1]. In the datasets I took it contained samples of human voice recordings which contains short (one-second or less) audio clips of commands, such as ‘up’, ‘stop’, ‘no’, ‘right’, ‘left’, ‘down’, ‘go’, ‘yes’. As outlined below:

1 http://storage.googleapis.com/download.tensorflow.org/data/mini_speech_commands.zip

1. Each WAV file contains time-series data with a set number of samples per second.
2. Each sample represents the amplitude of the audio signal at that specific time
3. In a 16-bit system, like the WAV files in the mini Speech Commands Dataset, the amplitude values range from -32, 768 to 32, 767.
4. The sample rate for this dataset is 16kHz.

3.3. Deep Neural Network and Convolutional Neural Network Algorithms

Deep Neural Network (DNN) is a type of neural network. DNN consists of several hidden units with connections between layers but no connections between units at each layer. This method has an architecture similar to the architecture on an Artificial Neural Network (ANN), with supervised training. By identifying the input and matching it with an existing pattern. The advantages of deep learning methods for speech recognition, namely better network architecture, can optimize many parameters, DNN is good enough for speech recognition, DNN is also faster in understanding many languages[1].

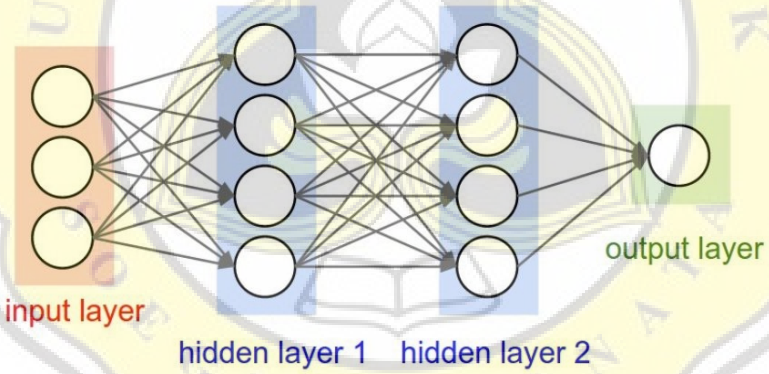

Figure 3.1: Deep Neural Network Layer

The design of the neural network in the Figure 3.1 is based on the structure of the human brain. Just like humans use their brains to identify patterns and classify different types of information. This neural network can also be trained to perform the same task based on the data used.

Individual layers on a neural network can also be analogous to filters that work to filter something from rough to soft, increasing the likelihood of detecting correctly, as well as output. The human brain works that way, every time we receive new information, the brain

will try to compare it with previously known objects. This concept is then applied by deep neural networks.

Broadly speaking architecture on Convolutional Neural Network Algorithms (CNN) with Deep Neural Networks is almost the same because both algorithms enter into one of deep learning. The difference between the two algorithms is more about how to process data or input, as explained below about Convolutional Neural Network.

Basically CNN is an excellent algorithm for processing digital image programming because CNN utilizes the convolution process by moving a certain-sized convolution kernel (filter) to an image, the computer gets new representative information from the results of multuplying that part of the image with the filter used.

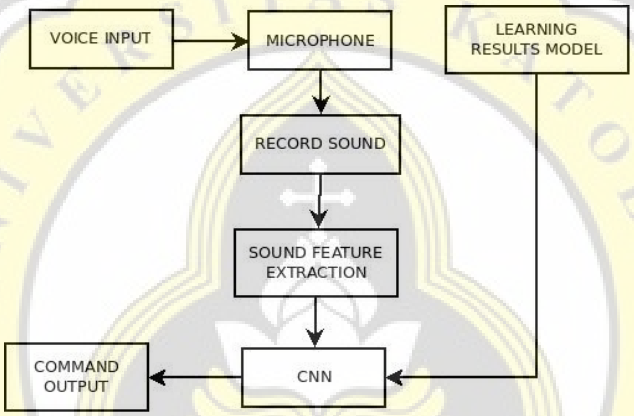

Figure 3.2: Speech Recognition on CNN

From the Figure 3.2 it can be seen that in speech recognition using Convolutional Neural Network (CNN) algorithms there is no process of classification and analysis of commands where the process is needed to build a speech recognition system. Because with the process, the computer can decipher the commands spoken by the user so that later the command can be executed correctly.

3.3.1. Connections Between Layers on Deep Neural Networks

For example, if neural network has only two layers. Where the input layer has two input neurons, then the output has three neurons as in the figure below:

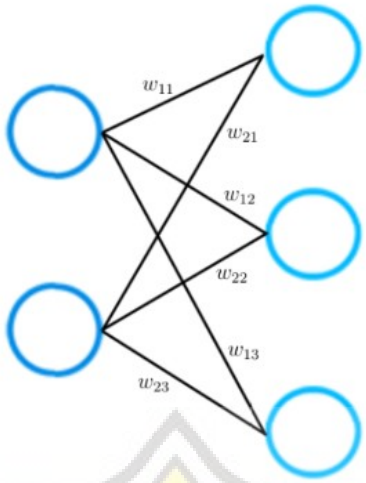

Figure 3.3: Weight Between Neurons

Each connection between neurons is presented by a numerical value that we call weight (**w**). Each of these **w** has an index. The first number in the index shows the number of neurons from the original layer, the second number is the number of neurons of the intended layer connection.

The entire weight between two neural networks can be presented in the form of a matrix below:

$$W = \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{pmatrix}$$

Figure 3.4: Weight Matrix

This weight matrix has the same number of entries as the connections between neurons. This matrix weight dimension results from the size of two layers connected based on the weight of the matrix.

The number of rows corresponds to the number of neurons from the screen, where the connection originated and the number of columns corresponds to the number of layers of the connection's destination.

In this case, the number of rows of the weight of this matrix is two (the size of the input layer) and the number of column size is three (output layer size).

3.4. Speech Recognition

Speech recognition is a technology that can convert speech in the form of sound signals into writing. In addition to recognizing speech, speech recognition can be used to give commands to a computer. For example, a user is instructed to open Google Chrome. In this project speech recognition is built with the aim of giving commands to the computer to perform as ordered by the user.

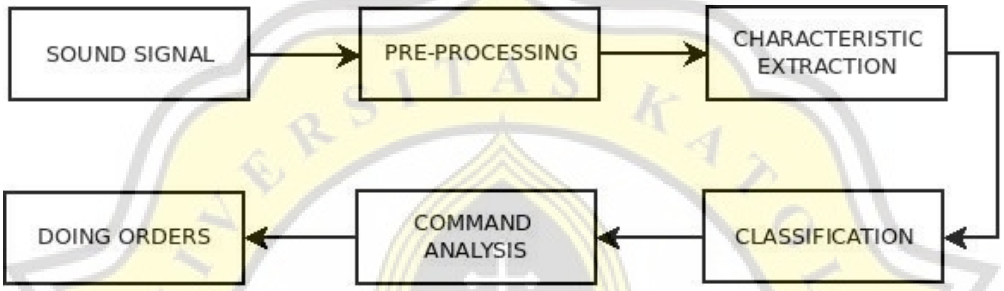

Figure 3.5: Common Process of Speech Recognition

In the Figure 3.5, voice signal input will be pre-processing first to prepare and process the initial dataset so that the dataset used is a dataset that is ready to be used and can facilitate processes in the next stage, then the dataset that has been pre-processed is extracted and separated, after that, the system will compare the extraction results that have been done with the available dataset. The parameters compared are the level of sound suppression which is then matched with the available dataset templates. Once the classification process is complete, the system will analyze what commands are coming in and what work or response the system should issue.

3.5. Success Rate of Deep Neural Network and Convolutional Neural Network

The success rate of both tested algorithms is to produce high accuracy on both algorithms. To produce high accuracy both algorithms are tested with parameters equal to the value of each parameter will be adjusted for both algorithms so as to produce high accuracy.

3.5.1. Learning Process

1. **Feed Forward:** the inputs are given into the network which pass through the hidden layer and reach the output layer to produce an output. It is used in training and also used to make predictions after the network has finished training.
2. **Cost Calculation:** the outputs produced after feedforward are compared to the desired output and we calculate how different it is from the original value. Cost shows how different the calculated outputs are from the original value. In an ideal scenario, the cost should to be 0, or very close to 0.
3. **Backpropagation:** the cost shows how much to update the weights and biases by using gradient descent. The weights and biases are updated in such a way that the outputs from the network become closer to the desire output, and the cost drops to 0 or very close to 0.
4. Repeat the steps for a fixed number of iterations called epochs. The number of epochs is decided by looking at the cost. When reach a stage where the cost is close to 0, and network is making accurate predictions, that means the network has **learned.**