MIMIC HUMAN SPEECH IN BAHASA INDONESIA USING SPEECH RECOGNITION AND SPEECH SYNTHESIS

By

Valens Prabagita Ivan Susilo

A Thesis
Submitted to the Faculty of Computing
President University
in Partial Fulfilment of the Requirements
for the Degree of Bachelor of Science
in Information Technology

Cikarang, Bekasi, Indonesia

October 2018

MIMIC HUMAN SPEECH IN BAHASA INDONESIA USING

SPEECH RECOGNITION AND SPEECH SYNTHESIS

By

Valens Prabagita Ivan Susilo

Approved:

_____	_____
Dr. Tjong Wan Sen, S.T., M.T.	Drs. Nur Hadisukmana, M.Sc.
Thesis Advisor	Program Head of Information Technology

_____

Ir. Rila Mandala, M.Eng., Ph.D.
Dean of Faculty of Computing

ABSTRACT

The Abstract …

ACKNOWLEDGMENTS

The author gratefully acknowledges the amazing supports, guidance and advices from:

1. My beloved family, Ayah, Ibu, Mas Risky and Dek Lia.
2. B35TFR13NDZ0NE, Nisa, Marsel, Bora, Fira, Rai, Pindy, Oscar, and Michel.
3. Sangar Team, Mas Andreas, Mba Diana, Nisa, Alfian, and Aci
4. DV 14, Vovo, Gojal, Ariyel, Rino, Willy, Azmi, Damara, Atisah, Risda, Sonya, Dini, and Arizha.
5. My cool thesis lecturer, Mr. Wan Sen.
6. My second family, Van Lith XXII.
7. And you, yes you.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

TABLE Page

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

1.1. Background

"Ok Google, play some music". "Siri, what should I eat for lunch?". Everyday people use their virtual assistance to boost their activities. People very like to use it because they just asked to their device and then in seconds, the wish is granted. It seems like, people are talking to the computer. The truth is, speech recognition takes big role with the help of machine learning. Google Assistance, Apple Siri, Microsoft Cortana, Amazon Alexa, and others have thousands of speech data to be analysed with the machine learning and they easily add data by collecting people speech from the assistance with permission.

If speech recognition is the process to get data by analysed speech, the opposite of speech recognition is speech synthesis, the process to produce artificial speech. Therefore, speech recognition is known as speech-to-text and speech synthesis is known as text-to-speech. "Hey Cortana, read my email" command make virtual assistance generate speech from the email text. With each technology can produce any kinds software related to speech. Combine both of can produce many more. One of the combinations is mimic human speech.

1.2. Problem Statement

This research aims to develop website application which can be used to mimic speech in Bahasa Indonesia. The application can recognize the speech and generate speech from text.

1.3. Research Objective

This research sees an opportunity to implement speech recognition and speech synthesis to create a mimic speech.

1.4. Scope and Limitation

This research focuses on developing an application which will be able to:

1. Perform speech recognition.
2. Perform speech synthesis.

The limitations of this application are as following:

1. Speech recognition data is taken from recorded speech and in human speech in Bahasa Indonesia.
2. Speech synthesis data is taken from saved speech, result from speech recognition.
3. Application is developed as website application.

1.5. Thesis Methodology

Rapid Application Development (RAD) methodology will be used in the development of this application. The RAD method, which was first developed by James Martin, is a Software Development Life Cycle method that gains its popularity in recent years due to its suitability to manage web application projects. The features of RAD were designed to overcome most of the shortcomings found in traditional waterfall model. Some of these features are: fast prototyping and capability to deal with change in requirements.

The RAD model implemented in this thesis will consists of four major phases:

1. Requirement Planning Phase

   This is the where system planning and analyses are done. System requirements are established, including the view range of the camera, the numbering schema for parking lots, and the general category of cars to be detected. Algorithms are proposed to solve the problem along with the overall outline of the program.

2. User Design Phase

   During this phase, the model of system's processes is the main focus. Models for input, output, process and user interface is built, and represented in different parts that include diagrams visualization. The system design will refer to the plan created in previous stage. The design will be continuously discussed, reviewed, and updated until the best version is found.

3. Development Phase

This phase is where the ideas and plans are executed. The application is developed according to the predefined features standard. All the components of image processing and object detection are put together into one program to perform the work from beginning to the final output. Unit testing will also be done here. It focuses on application development, including: coding, unit integration, and testing.

4. Cut Over Phase

In this final phase there will be some test to evaluate the program's ability to determine which area of parking lot is empty. There will be certain test cases of parking areas that will produce different images to be processed. An evaluation will be done to see how far the application is capable to detect the area. Bugs fixing will also be done in this step. Another part of cutover phase is to create installation and operating manuals to allow people operate the program in their environment.

1.6. Thesis Outline

The thesis consists of seven chapters, which are as follow:

1. Chapter I: Introduction

Introduction consists of Thesis Background, Problem Statement, Research Objective, Scope and Limitation, Methodology, and Thesis Outline.

2. Chapter II: Literature Study

Literature Study describes about Machine Learning, Phoneme, and Speech Recognition and also Speech Synthesis in general. It consists of Machine Learning, Neural Network, Phoneme, Speech Recognition, Pre-processing Speech, Speech Synthesis, and Related Work.

3. Chapter III: System Analysis

System Analysis describes the analysis of the behaviour and function of the mimic system. It consists of System Overview, Machine Learning, Functional Analysis, Software and System Requirements, and Software Architecture.

4. Chapter IV: System Design

System Design describes the definition of the program's architecture, components of the mimic system, and modules available in the parking finder application. It consists User Interface Design, and Class Diagram of the program.

5. Chapter V: System Implementation

System Implementation describes how the application for mimic speech is implemented. It consists of User Interface Development and Application Details.

6. Chapter VI: System Testing

System Testing contains the testing documentation of the application's ability to recognize and generate speech to mimic speech. It consists here are Testing Environment and Testing Scenarios, along with the results.

7. Chapter VII: Conclusion and Future Work

Conclusion describes the conclusion of the research on mimic speech. Future Work describes possible improvements for the speech recognition and synthesis and also more possible application in future.

CHAPTER II

LITERATURE STUDY

2.1. Machine Learning

Machine learning is a form of AI that enables a system to learn from data rather than through explicit programming [1]. Others state that machine learning usually refers to the changes in systems that perform tasks associated with artificial intelligence (AI) [2].

Basically, machine learning consists of 2 words, Machine and Learning. As a noun, there are many definitions related to machine [3] and one of the definitions, machine is a computer. Meanwhile, learning can be interpreted as the activity of obtaining knowledge or knowledge obtained by study [4]. Combine both words, machine learning can be interpreted as a computer that does an activity of obtaining knowledge.

In general, there are 2 types of machine learning Supervised Learning and Unsupervised Learning.

2.1.1 Supervised Learning

In supervised machine learning, a system is trained with data that has been labelled. The labels categorise each data point into one or more groups, such as ‘apples’ or ‘oranges’. The system learns how this data – known as training data – is structured, and uses this to predict the categories of new – or ‘test’ – data [5]. It can be considered

the learning is guided by a teacher. One has a dataset which acts as a teacher and its role is to train the model or the machine. Once the model gets trained it can start making prediction or decision when new data is given to it [6].

It can be concluded that supervised machine learning is the learning way where the machine is guided by labelled data and it needs to learn how to classify the model data to fits the labelled data.

2.1.2 Unsupervised Learning

Unsupervised learning is learning without labels. It aims to detect the characteristics that make data points more or less similar to each other, for example by creating clusters and assigning data to these clusters [5]. Suppose images of apples, bananas and mangoes are presented to the model, so what it does, based on some patterns and relationships it creates clusters and divides the dataset into those clusters. Now if a new data is fed to the model, it adds it to one of the created clusters [6].

It can be concluded that unlike supervised learning, unsupervised learning is the learning way where the machine is the figure out data by cluster the data.

2.2. Neural Network

A neural network is an approach to machine learning in which small computational units are connected in a way that is inspired by connections in the brain [5].

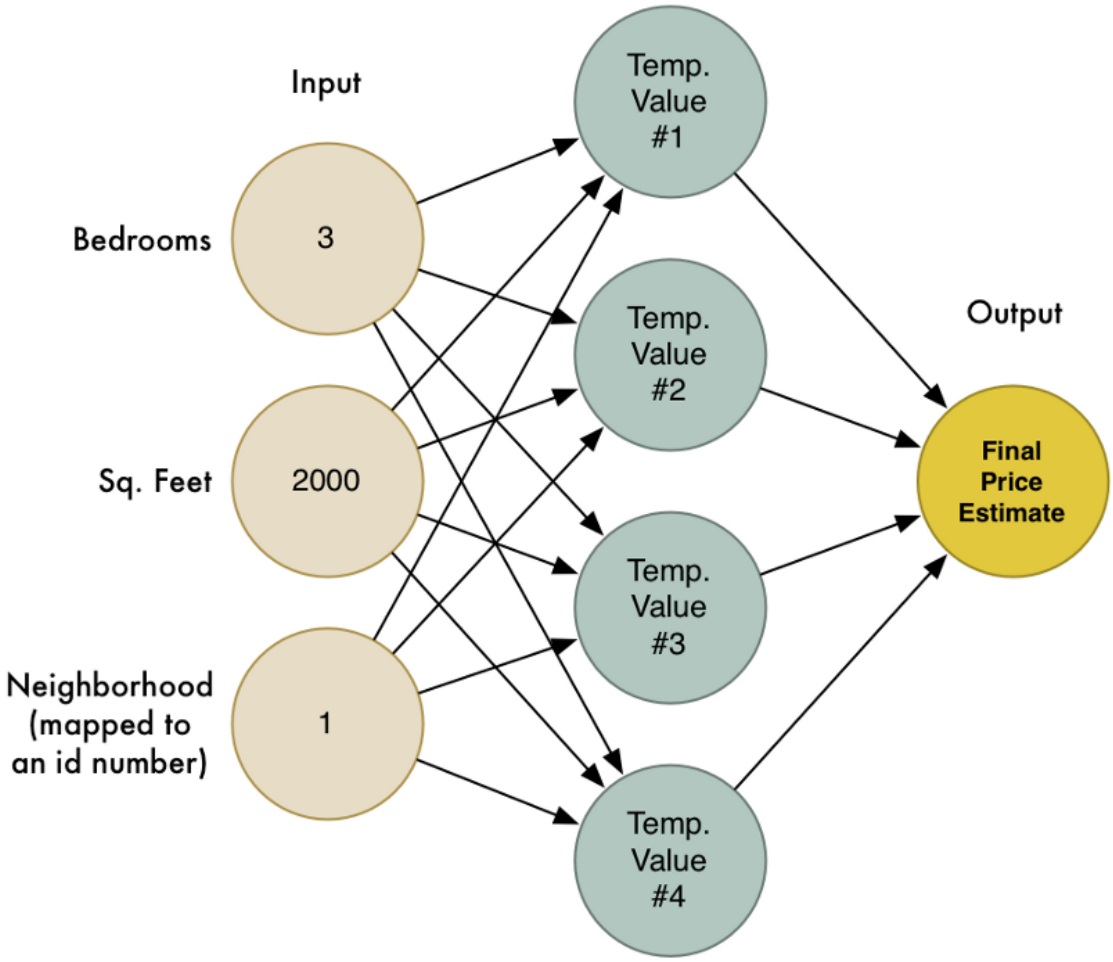

Figure 2.1. Neural network to find estimated price from specific input [7].

Every neural network model is basically a three-layered system, which are Input layer, Hidden Layer and Output Layer [8]. Input layer, where the inputs of the problem are received, hidden layers, where the relationship between the inputs & outputs are determined & represented by synaptic weights, & an output layer which emits the outputs of the problem [9].

A neural network models that has no memory with the same inputs will have the same output, which said the model is a stateless algorithm.

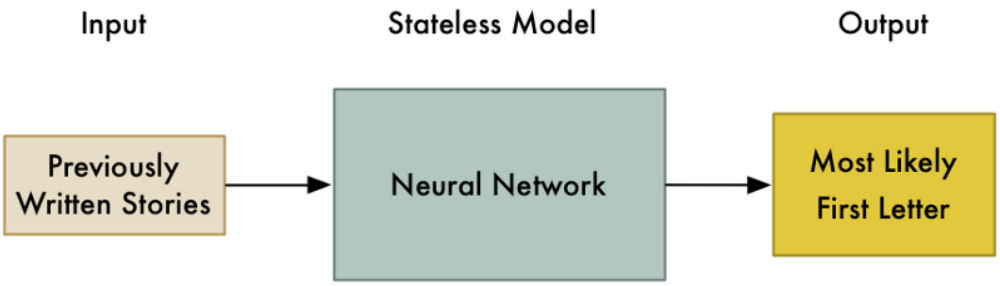

Figure 2.2. Stateless neural network model [7].

Updating the network each time when use it, basically is the idea of recurrent neural network. This allows it to update its predictions based on what it saw most recently. It can even model patterns over time as long as we give it enough of a memory [7]. So, giving the neural network memory would update the hidden layers and make the output update more accurate over the time.

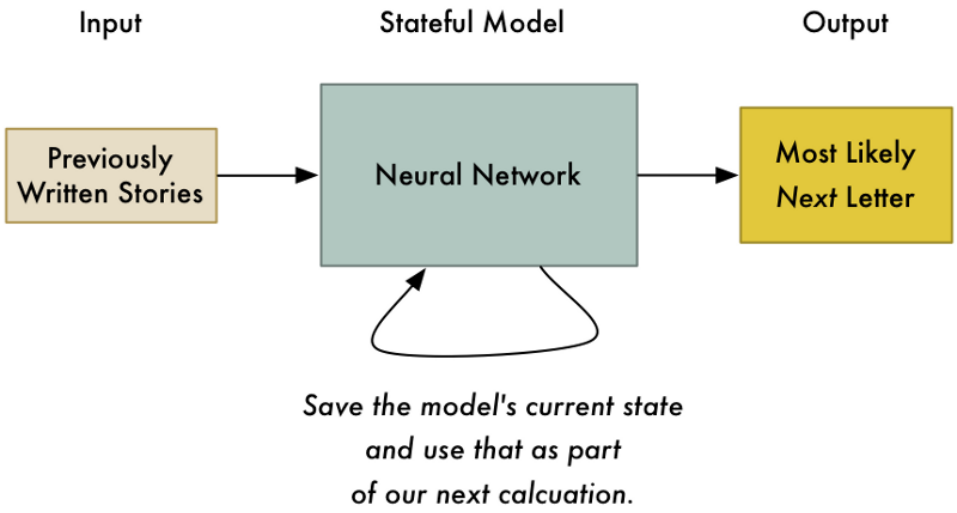

Figure 2.3. Stateful neural network model [7].

2.3. Phoneme

A phoneme is the basic unit of phonology. It is the smallest unit of sound that may cause a change of meaning within a language, but that doesn't have meaning by itself. For example, in the words' "bake" and "brake," only one phoneme has been altered, but a change in meaning has been triggered. The phoneme /r/ has no meaning on its own, but by appearing in the word it has completely changed the word's meaning [10].

In the other hand, grapheme is individual letters and groups of letters that represent single phonemes, like the "s" and the "oo" in "spoon". Understanding how letters are used to encode speech sounds in written language is crucial in learning to decode unfamiliar words [11].

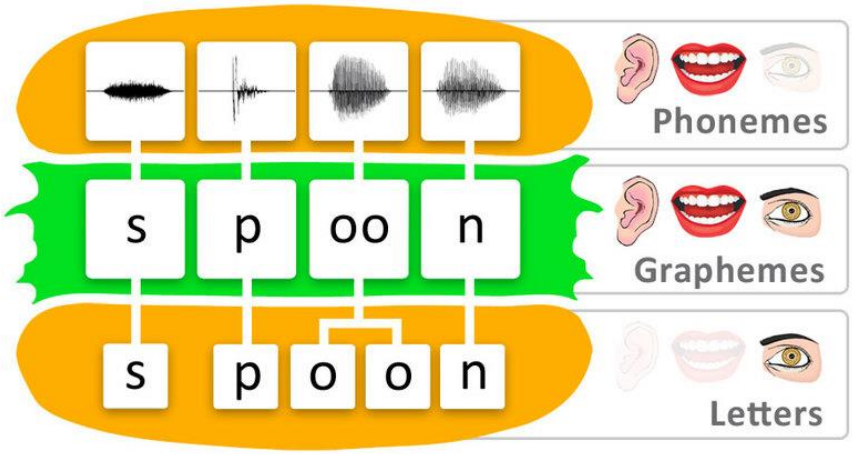

Figure 2.4. Phonemes, graphemes, and letters in the word "spoon" [11].

In Bahasa Indonesia phoneme is distributed into *vokal*, *diftong*, *konsonan*, *gugus konsonan* [12].

2.3.1 Vokal

Vokal or vowel, in English, is a speech sound produced by humans when the breath flows out through the mouth without being blocked by the teeth, tongue, or lips [13]. In Bahasa Indonesia there are 5 vowel graphemes, *a*, *i*, *u*, *e*, and *o* [14].

Table 2.1. Bahasa Indonesia vowel graphemes [14].

Huruf Vokal	**Posisi Awal**	**Posisi Tengah**	**Posisi Akhir**
a	*a*pi	p*a*di	lus*a*
e /e/	*e*nak	p*e*tak	sor*e*
e /Ɛ/	*e*mber	p*e*ndek	-
e /ə/	*e*mas	k*e*na	tip*e*
i	*i*tu	s*i*mpan	murn*i*
o	*o*leh	k*o*ta	radi*o*
u	*u*lang	b*u*mi	ib*u*

2.3.2 Diftong

Diftong or diphthong, in English, is a vowel sound in which the tongue changes position to produce the sound of two vowels [15]. In Bahasa Indonesia there are 4 diphthong graphemes, *ai*, *au*, *ei*, and *oi* [16].

Table 2.2. Bahasa Indonesia diphthong graphemes [16].

Huruf Diftong	**Posisi Awal**	**Posisi Tengah**	**Posisi Akhir**
ai	-	bal*ai*rung	pand*ai*
au	*au*todidak	t*au*fik	harim*au*
ei	*ei*gendom	g*ei*ser	surv*ei*
oi	-	b*oi*kot	amb*oi*

2.3.3 Konsonan

Konsonan or consonant, in English, is one of the speech sounds or letters of the alphabet that is not a vowel. Consonants are pronounced by stopping the air from flowing easily through the mouth, especially by closing the lips or touching the teeth with the tongue [17]. In Bahasa Indonesia there are 21 consonant graphemes, *b*, *c*, *d*, *f*, *g*, *h*, *j*, *k*, *l*, *m*, *n*, *p*, *q*, *r*, *s*, *t*, *v*, *w*, *x*, *y*, and *z* [18].

Table 2.3. Bahasa Indonesia consonant graphemes [18].

Huruf Konsonan	**Posisi Awal**	**Posisi Tengah**	**Posisi Akhir**
b	*b*ahasa	se*b*ut	ada*b*
c	*c*akap	ka*c*a	-
d	*d*ua	a*d*a	aba*d*
f	*f*akir	ka*f*an	maa*f*
g	*g*una	ti*g*a	gude*g*
h	*h*ari	sa*h*am	tua*h*
j	*j*alan	man*j*a	mikra*j*
k	*k*ami	pa*k*sa	politi*k*
l	*l*ekas	a*l*as	aka*l*
m	*m*aka	ka*m*i	dia*m*
n	*n*ama	ta*n*ah	dau*n*
p	*p*asang	a*p*a	sia*p*
q	*q*ariah	i*q*ra	-
r	*r*aih	ba*r*a	puta*r*
s	*s*ampai	asli	tangka*s*
t	*t*ali	ma*t*a	rapa*t*
v	*v*ariasi	la*v*a	moloto*v*
w	*w*anita	ha*w*a	takra*w*
x	*x*enon /s/	-	-
y	*y*akin	pa*y*ung	-
z	*z*eni	la*z*im	ju*z*

2.3.4 Gugus Konsonan

Gugus Konsonan or cluster, in English, is a group of two or more consonant sounds that are together and have no vowel sound between them [19]. In Bahasa Indonesia there are 4 cluster graphemes, *kh*, *ng*, *ny*, and *sy* [20].

Table 2.4. Bahasa Indonesia cluster graphemes [20].

Gabungan Huruf Konsonan	**Posisi Awal**	**Posisi Tengah**	**Posisi Akhir**
kh	*kh*usus	a*kh*ir	tari*kh*
ng	*ng*arai	ba*ng*un	sena*ng*
ny	*ny*ata	ba*ny*ak	-
sy	*sy*arat	mu*sy*awarah	ara*sy*

2.4. Speech Recognition

The process of automatically recognizing spoken words of speaker based on information in speech signal is called Speech Recognition [21]. Other definition of speech recognition, also known as Automatic Speech Recognition (ASR), or computer speech recognition, is the process of converting a speech signal to a sequence of words, by means of an algorithm implemented as a computer program [22].

There are many approaches to do speech recognition, one of the approaches is neural network, although neural network has almost the lowest accuracy, 70% [21]. Figure 2.5 roughly shows speech recognition with neural network approach.

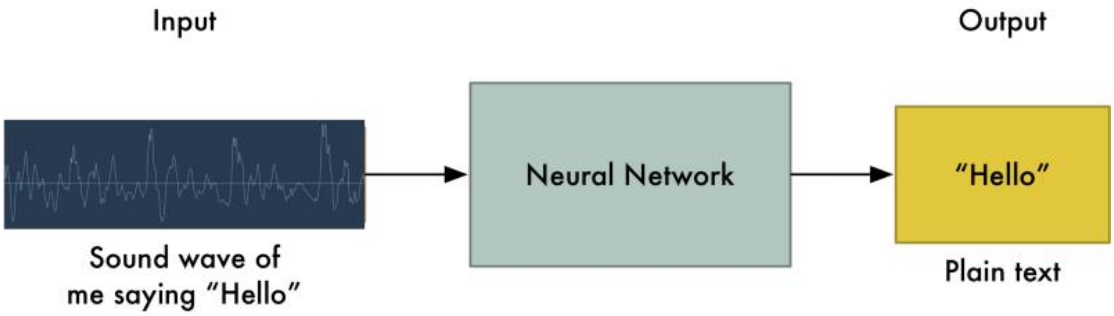

Figure 2.5. Speech recognition with neural network approach [23].

2.5. Pre-processing Speech

As sound is transmitted as waves, and computer understand numbers, it's necessary to pre-processing speech so that computer understand and can be feed as input into neural network. The pre-processing is sound sampling. Sound sampling is taking a reading thousands of times a second and recording a number representing the height of the sound wave at that point in time. Basically, all an uncompressed .wav audio file [23].

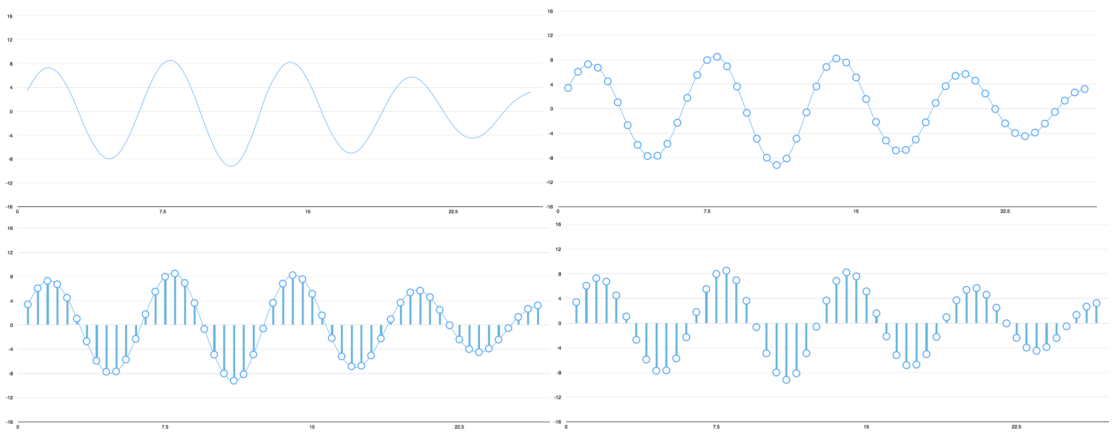

Figure 2.6. Sound sampling process [23].

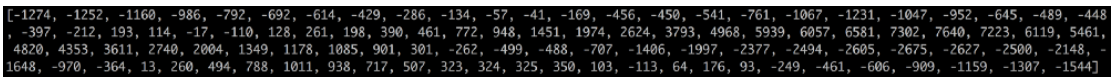

Figure 2.7. 100 samples in numbers from “Hello” sound with sample rate of 16kHz (16 samples per seconds) [23].

Nyquist sampling theorem [24] provides a prescription for the nominal sampling interval required to avoid aliasing. The sampling frequency should be at least twice the highest frequency contained in the signal. In the case, where one has f_c = 3 Hz, and so the Nyquist theorem tells that the sampling frequency, f_s, must be at least 6 Hz [25].

2.6. Speech Synthesis

Speech synthesis is the artificial production of human speech [26]. The Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database [27]. From phoneme, text can be analysed by its phonemes and then with the phonemes concatenating speech can be done.

2.7. Related Work

The following are most related work to the research. Have relation to mimic speech, both speech recognition and speech synthesis.

2.7.1 Lyrebird

Lyrebird is website application, *https://lyrebird.ai*, that has 3 products: Custom Voice, Vocal Avatar, and Vocal Avatar API [28]. Custom voice is a product to create speech based on real people's speech, it can control the intonation, expression, and the emotion of the speech. Vocal avatar is a product to create own digital speech by read some English sentences, and then generate any sentences with own digital speech. Vocal avatar API is a product to provide API to use user's own vocal avatar.

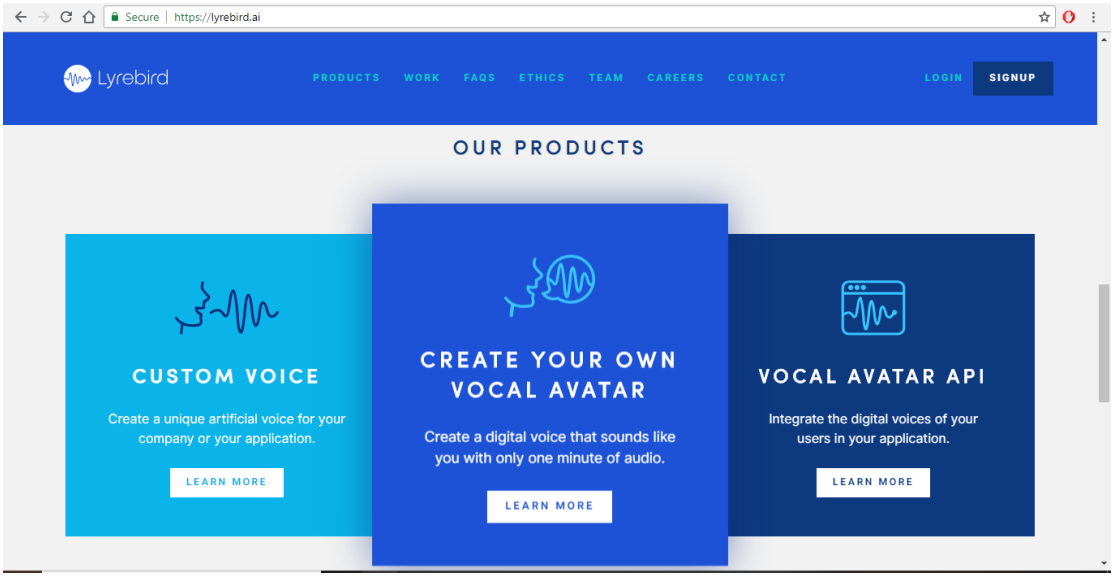

Figure 2.8. Screenshot of Lyrebird in the website [28].

2.7.2 Google Translate

Google Translate is one of Google products that is an application to translate languages. Google Translate can be access freely through *https://translate.google.com/*.

It has many features [29] and one of them is Talk feature. Talk has function to input text from speech and generate speech from text in any languages.

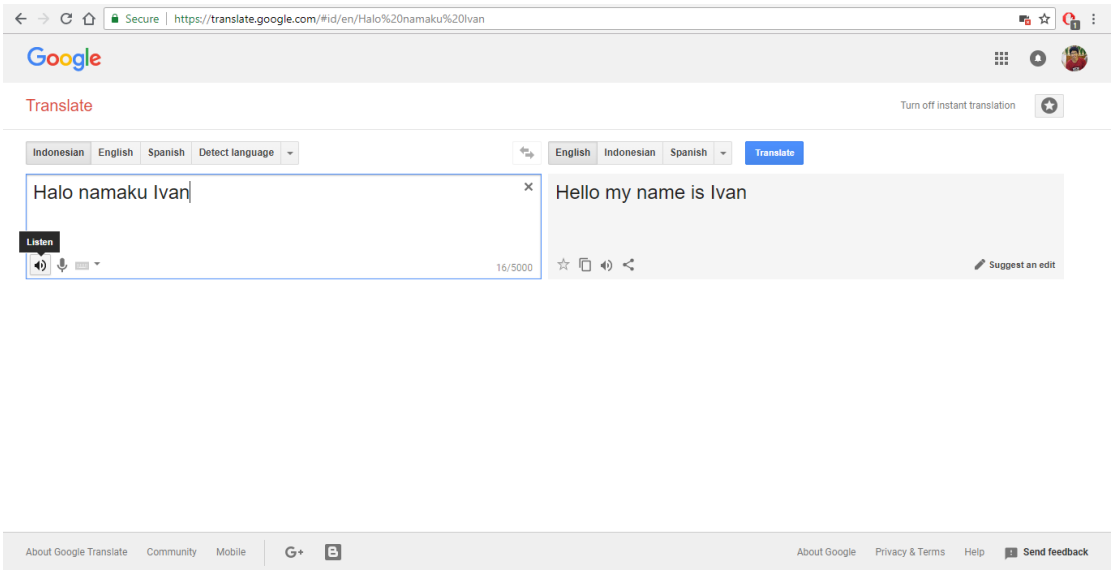

Figure 2.9. Screenshot of Google Translate in website [29].

From related work, it can be concluded that Lyrebird can mimic speech with its vocal avatar, but the speech is in English. In the other hand, Google Translate could mimic speech into any languages, but the speech vocal is from the Google Translate itself.

CHAPTER III

SYSTEM ANALYSIS

This chapter explains the analysis of the application – both in its function and behaviour, in order to fulfil the prescribed requirements.

3.1. System Overview

This research is intended to implement speech recognition and speech synthesis into this research. This application will be trained to recognize the speech before used by user. After enough training, this application will identify speech from the user based on sentences that will be displayed. Then, with speech synthesis user can generate speech from identified speech that will become mimic speech. The objective of this research is to create a web-based application for mimic speech by identify user speech and then generated them.

3.2. Machine Learning

Speech recognition in this application will implements a supervised learning, labelled sampled data will be feed on neural network. The speech data output will be saved in database. Meanwhile, speech synthesis in this application will implement by populating Bahasa Indonesia word to the database classified by the phonemes. Identified speech is concatenated based the database on corresponding with the word.

3.3. Functional Analysis

There are several functions from this application listed in the Table 3.1.

Table 3.1. Functionality Table.

No	Function Description
1	Allow user to identify user's speech.
2	Allow user to select which speech data that will be used.
3	Allow user to generated speech.

3.4. Software and System Requirements

This research and application development should be supported by the following list requirement in order to write the research, build and run the application well.

3.4.1 Laptop / Personal Computer

Laptop or Personal Computer is used as the tool where operating system is run. In this research, ASUS A455LN is used with Windows 10 as the OS.

3.4.2 Microsoft Office Word

Microsoft Office Word is used to write the research documentation. In this research, Microsoft Office Word 2016 is used.

3.4.3 Node.js, JavaScript Run-Time Environment.

Node.js is an open source server environment – Node.js is free – Node.js runs on various platform (Windows, Linux, Unix, Mac OS X, etc) – Node.js uses JavaScript on the server [30]. In this research, Node.js v8.11.4 is used.

3.4.4 NoSQL document-oriented database

Document databases pair each key with a complex data structure known as a document. Documents can contain many different key-value pairs, or key-array pairs, or even nested documents [31]. In this research, MongoDb Community Server 4.0.3 as database server and MongoDb Compass 1.15.4 as MongoDb UI.

3.4.5 Integrated Development Environment (IDE)

IDE is used as application development environment. In this research, Visual Studio Code is used.

3.4.6 Git

Git is used as version control. The repository is placed on local and cloud as preventive action. In this research, GitLab is used.

3.5. System Architecture

This sub-chapter discusses about the use-case diagram and narrative for this application in both point of view, the actors and the system.

3.5.1 Use-Case Diagram

Use-Case Diagram defines the functionality of a system and explains it in user point of view. The actors in this research is the application user. The diagram will be shown in Figure 3.1.

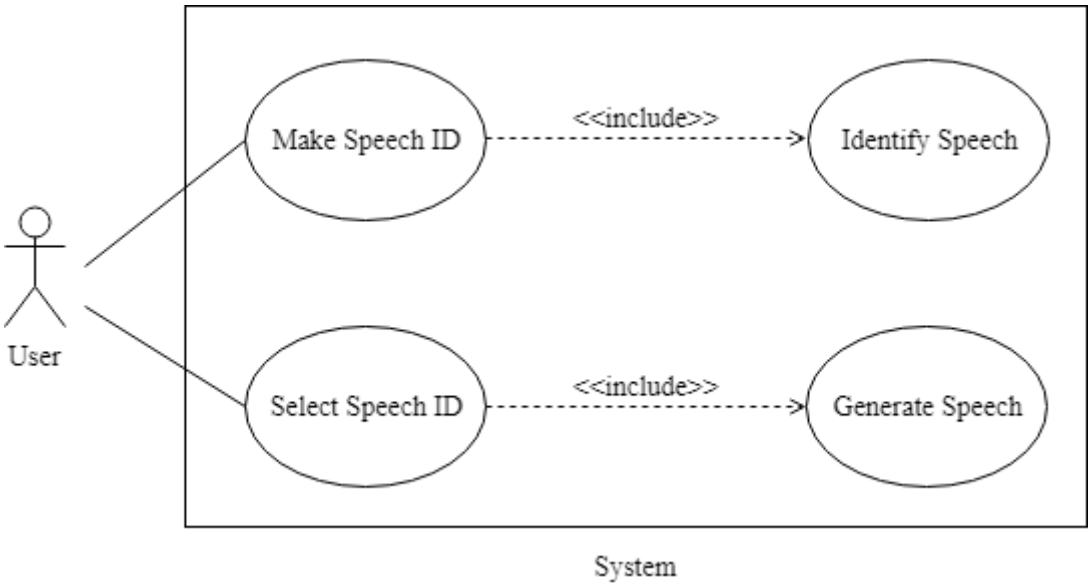

Figure 3.1. Use-Case Diagram.

3.5.2 Use-Case Narrative

Use-Case Narrative explains the interaction between the actors and the system. It describes the detail of use-cases such as name, description, pre-condition, post-condition, business rules, and the course of events that happened in the system. The Use-Case Narrative is shown in Table 3.2 and Table 3.5.

Table 3.2. Use-Case Narrative – Make Speech ID.

User Case Name	**Make Speech ID**
Use Case ID	UC01
Priority	High
Primary Business Actor	User
Primary System Actor	System
Another Participating Actor	None
Description	This use-case describes the event when user opens this application or in the home screen.

Precondition	None	
Trigger	User opens this application or user in the home screen.	
Typical Course of Event	**Actor Action**	**System Response**
	Choose Make Speech ID.	Start Make Speech ID activity.
Alternate Course	None	
Post Condition	Identify Speech screen is shown.	
Business Rule	None	
Implementation Constraint and Specifications	None	

Table 3.3. Use-Case Narrative – Identify Speech.

User Case Name	**Make Speech ID**	
Use Case ID	UC02	
Priority	High	
Primary Business Actor	User	
Primary System Actor	System	
Another Participating Actor	None	
Description	This use-case describes the event when Make Speech ID activity start.	
Precondition	User is from home screen.	
Trigger	User click Make Speech ID button in the home screen.	
Typical Course of Event	**Actor Action**	**System Response**
	Do Identify Speech.	Process Speech to Speech Data.
Alternate Course	**Actor Action**	**System Response**
	Finish Identify Speech.	Back to home screen.
Post Condition	User do identify speech again or Home screen is shown.	
Business Rule	None	
Implementation Constraint and Specifications	One speech ID for 1 speech data.	

Table 3.4. Use-Case Narrative – Select Speech ID.

User Case Name	**Select Speech ID**	
Use Case ID	UC03	
Priority	High	
Primary Business Actor	User	
Primary System Actor	System	
Another Participating Actor	None	
Description	This use-case describes the event when user opens this application or in the home screen.	
Precondition	None	
Trigger	User opens this application or user in the home screen.	
Typical Course of Event	**Actor Action**	**System Response**
	Select Speech ID.	Provide Speech ID.
Alternate Course	None	
Post Condition	Generate Speech screen is shown.	
Business Rule	None	
Implementation Constraint and Specifications	None	

Table 3.5. Use-Case Narrative – Generate Speech.

User Case Name	**Select Speech ID**	
Use Case ID	UC04	
Priority	High	
Primary Business Actor	User	
Primary System Actor	System	
Another Participating Actor	None	
Description	This use-case describes the event when Select Speech ID activity start.	
Precondition	User is from home screen.	
Trigger	User click Select Speech ID button in the home screen.	
Typical Course of Event	**Actor Action**	**System Response**
	Selected Speech ID.	Start Generate Speech Activity.
	Generate Speech.	Speech generated based on speech ID data.

Alternate Course	**Actor Action**	**System Response**
	Finish Generate Speech.	Back to home screen.
Post Condition	User do generate speech again or Home screen is shown.	
Business Rule	None	
Implementation Constraint and Specifications	None	

3.5.3 Activity Diagram

Activity Diagram presents a flowchart to represent the flow from one activity to another activity. As user open the application the home screen is shown. User could decide by choose to Make Speech ID or Select Speech ID. As long as user don't decide it stays in the home screen. If Make Speech ID is selected, Make Speech ID activity is started which is Identify Speech. Finish the activity will bring user back to home screen. If Select Speech ID is selected, Select Speech ID activity is started which is Generated Speech. Finish the activity will bring user back to home screen. The figure of the activity diagram can be seen in Figure 3.2.

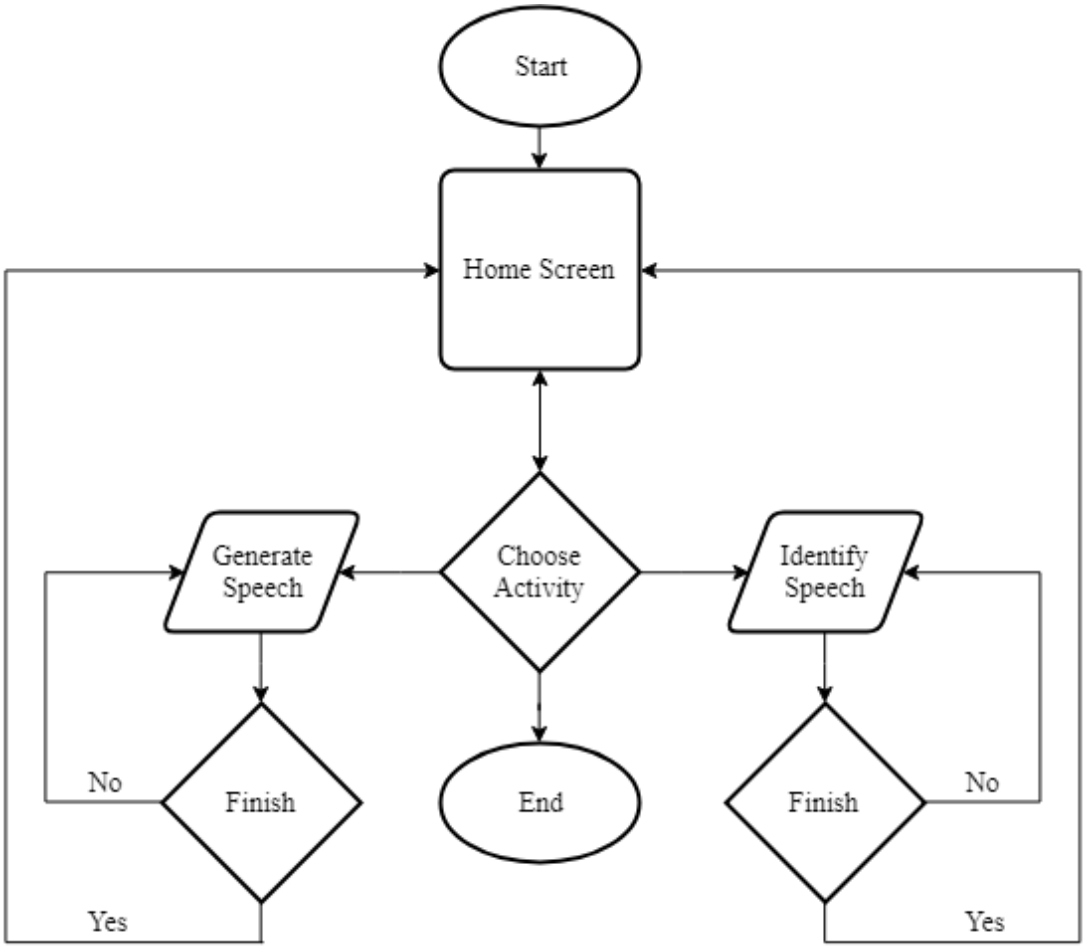

Figure 3.2. Activity Diagram.

CHAPTER IV

SYSTEM DESIGN

This chapter explains the process of system development of interfaces and class diagram based on the previous chapter.

4.1. User Interface Design

The User Interface (UI) design of this mobile application is divided into several features which are home screen, identify speech screen, and generate speech screen. The detail of every feature will be explained further below.

4.1.1 Home Screen

Figure 4.1 shows the design layout for home screen of the application. When user opens the application or finish Make Speech ID or Select Speech ID, it will show the home screen that consist of 2 buttons such as Make Speech ID and Select Speech ID. The description of the design layout is shown in Table 4.1.

Figure 4.1. Home Screen.

Table 4.1. Home Screen Description.

No	Description
1	Make Speech ID
2	Select Speech ID

4.1.2 Identify Speech Screen

Figure 4.2 shows the design layout for identify speech screen of the application. When user choose or click Make Speech ID, it will show the identify speech screen that consist of 4 element such as Random Sentences, Record Button, Identify Button, and Finish Button. The description of the design layout is shown in Table 4.2.

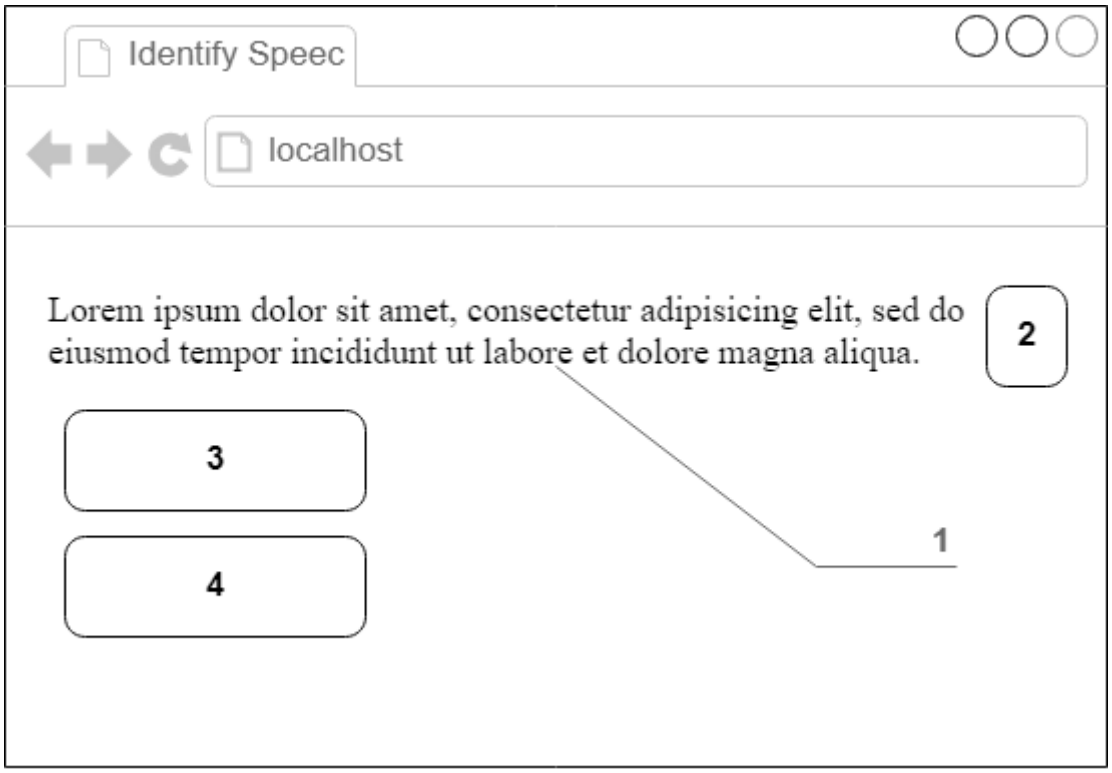

Figure 4.2. Identify Speech Screen.

Table 4.2. Identify Speech Screen Description.

No	Description
1	Random Sentences
2	Record Button
3	Identify Button
4	Finish Button

4.1.3 Generate Speech Screen

Figure 4.3 shows the design layout for generate speech screen of the application. When user choose or click Select Speech ID, it will show the generate speech screen

that consist of 4 element such as ID Selector, Textbox Form, Generate and Play Button, and Finish Button. The description of the design layout is shown in Table 4.3.

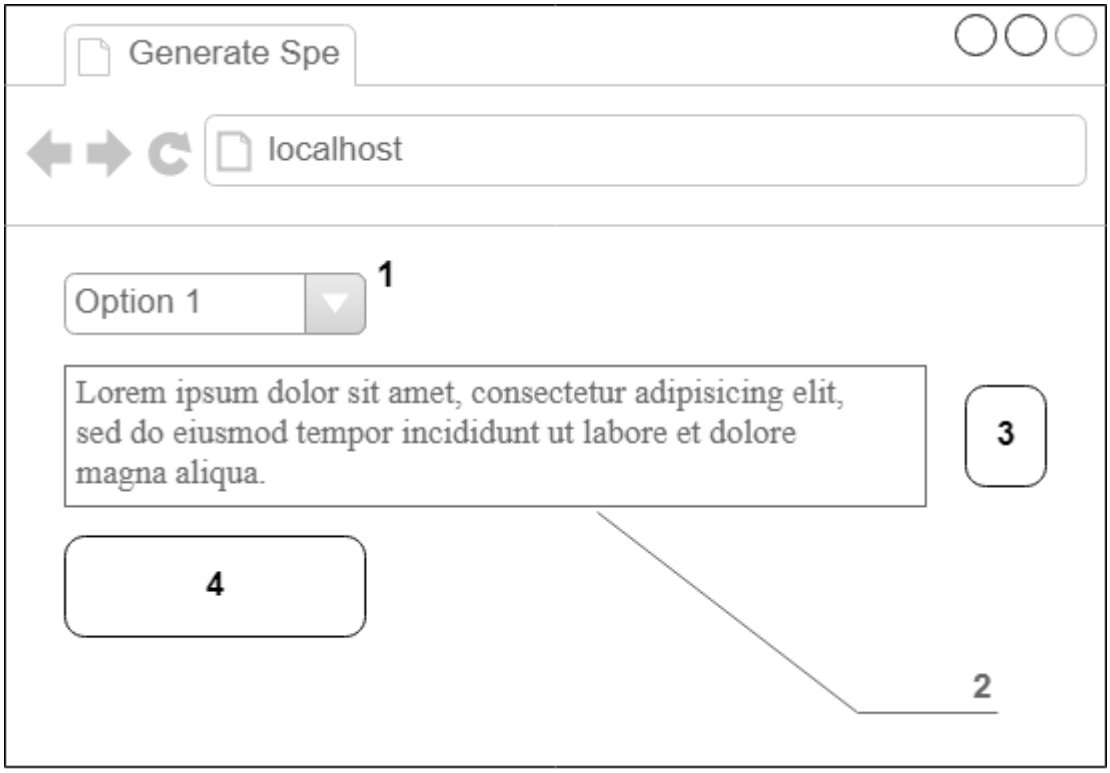

Figure 4.3. Generate Speech Screen.

Table 4.3. Generate Speech Screen Description.

No	Description
1	ID Selector
2	Textbox Form
3	Generate and Play Button
4	Finish Button

4.2. Class Diagram

The class diagram is the structure of the system used toward this research. In this application there are 2 main category libraries, front-end and back-end. The detail of every libraries will be explained further below.

4.2.1 Front-end Library

Front-end library is used as the application UI code. The front-end class diagram details are shown in Figure 4.4. Approximately there are 3 front-end classes as listed below.

1. Home

   `Home` class is class used to render Home screen. It consisting of 2 main methods, `identifyButton` and `generateButton`.

   The `identifyButton` is method to start Make Speech ID activity which open Identify Speech screen. The `generateButton` is method to start Select Speech ID activity which open Generate Speech screen.

2. IdentifySpeech

   `IdentifySpeech` class is Make Speech ID activity. It is class used to render Identify Speech screen. It consisting of 4 main methods, `randomSentences`, `recordButton`, `identifyButton`, and `finishButton`.

The `randomSentences` is method to randomize sentence that used to sentence user need to be said. The `recordButton` is method to input user recorded speech. The `identifyButton` is method to start identify user recorded speech and start `randomSentences` method again. The `finishButton` is method to finish Make Speech ID activity which open Home screen.

3. GenerateSpeech

`GenerateSpeech` class is Select Speech ID activity. It is class used to render Generate Speech screen. It consisting of 3 main methods, `selectID`, `generateButton`, and `finishButton`.

The `selectID` is method to select and define the speech ID from listed speech ID. The `generateButton` is method to generate and play speech based on inputted speech ID and text. The `finishButton` is method to finish Select Speech ID activity which open Home screen.

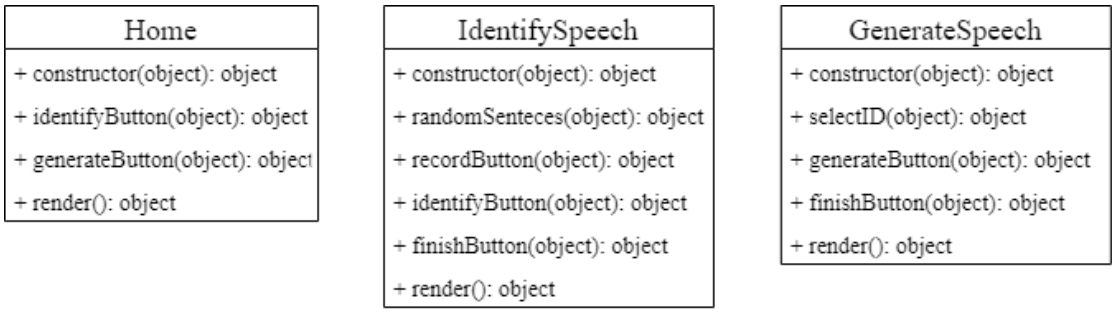

Figure 4.4. Front-end Class Diagram.

4.2.2 Back-end Library

Back-end library is used as speech recognition process, speech synthesis process, create-read-update-delete (CRUD) database process and database collections. The back-end class diagram details are shown in Figure 4.5. Approximately there are 4 back-end classes as listed below.

1. Speech Recognition

   `SpeechRecognition` class is class contains speech recognition algorithm. It containing `recognize` method to recognize speech from user and train speech.

2. Speech Synthesis

   `SpeechSynthesis` class is class contains speech synthesis algorithm. It containing `generate` method to classify inputted text then generate the speech from classified text.

3. MongoDb

   `MongoDb` class is class contains MongoDb process to established connection and data processing. It containing CRUD methods which is create, read, update, and delete data to MongoDb database.

4. MongoDb Collections

   MongoDb collections is class collection that is used as data type. It containing 4 main collections, `SpeechCollection`, `TextCollection`, `SpeechDataCollection`, and `SpeechTrainedCollection`.

The `SpeechCollection` is collection used to store all speech data classify by grapheme and gender, and store speech file location and trained speech file location. It consisting 4 main variables, `grapheme`, `gender`, `speechFile`, and `trainedFile`. The `TextCollection` is collection used to store text data classify by word and grapheme. It consisting 2 main variables, `word` and `grapheme`.

The `SpeechDataCollection` is collection used to store speech data classify by speechID and gender, and store identified speech. It consisting 3 main variables, `speechID`, `gender`, and `speech`. The `SpeechTrainedCollection` is collection used to store trained speech data classify by grapheme and gender, and store identified speech. It consisting 3 main variable, `grapheme`, `gender`, and `data`.

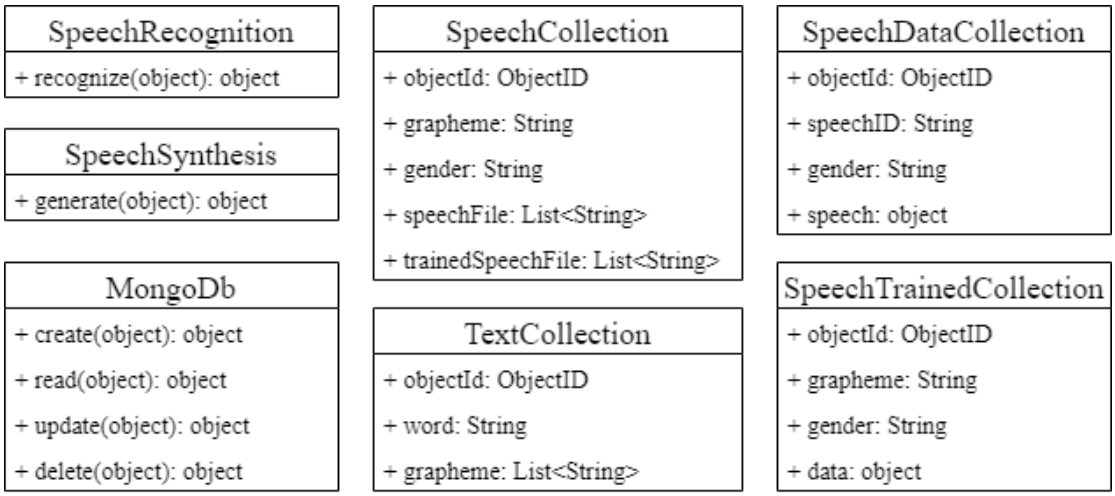

Figure 4.5. Bank-end Class Diagram.

REFERENCES

[1] Hurwitz, J., & Kirsch, D. (2018). *Machine Learning For Dummies®, IBM Limited Edition.* New Jersey: John Wiley & Sons, Inc.

[2] Nilsson, N. J., & Laboratory, R. (2005). *INTRODUCTION TO MACHINE LEARNING.* California: Nils J. Nilsson.

[3] dictionary.cambridge.org. (2018). *MACHINE*. Retrieved from Cambridge English Dictionary: https://dictionary.cambridge.org/dictionary/english/machine

[4] dictionary.cambridge.org. (2018). *LEARNING*. Retrieved from Cambridge English Dictionary: https://dictionary.cambridge.org/dictionary/english/learning

[5] Society, T. R. (2017). *Machine learning: the power and promise.* London: The Royal Society.

[6] edureka.com. (2018, October 18). *What is Machine Learning?* Retrieved from edureka: https://www.edureka.co/blog/what-is-machine-learning/

[7] Geitgey, A. (2016, January 3). *Machine Learning is Fun! Part 2*. Retrieved from Medium: https://medium.com/@ageitgey/machine-learning-is-fun-part-2-a26a10b68df3

[8] Magdi Zakaria, M. A.-S. (2014). Artificial Neural Network : A Brief Overview. *Int. Journal of Engineering Research and Applications*, 6.

[9] A.D.Dongare, R. A. (2012). Introduction to Artificial Neural Network. *International Journal of Engineering and Innovative Technology*, 6.

[10] courses.lumenlearning.com. (n.d.). *Introduction to Language*. Retrieved from lumen: https://courses.lumenlearning.com/boundless-psychology/chapter/introduction-to-language/

[11] readingdoctor.com.au. (2016). *Phonemes, Graphemes and Letters: The Word Burger*. Retrieved from Reading Doctor: http://www.readingdoctor.com.au/phonemes-graphemes-letters-word-burger/

[12] Yanti, N. T. (n.d.). FONEM BAHASA INDONESIA. *Academia.edu*, 10.

[13] dictionary.cambridge.org. (2018). *VOWEL*. Retrieved from Cambridge English Dictionary: https://dictionary.cambridge.org/dictionary/english/vowel

[14] puebi.readthedocs.io. (n.d.). *Huruf Vokal*. Retrieved from PUEBI Daring: https://puebi.readthedocs.io/en/latest/huruf/huruf-vokal/

[15] dictionary.cambridge.org. (2018). *DIPHTHONG*. Retrieved from Cambridge English Dictionary: https://dictionary.cambridge.org/dictionary/english/diphthong

[16] puebi.readthedocs.io. (n.d.). *Huruf Diftong*. Retrieved from PUEBI Daring: https://puebi.readthedocs.io/en/latest/huruf/huruf-diftong/

[17] dictionary.cambridge.org. (2018). *CONSONANT*. Retrieved from Cambridge English Dictionary: https://dictionary.cambridge.org/dictionary/english/consonant

[18] puebi.readthedocs.io. (n.d.). *Huruf Konsonan*. Retrieved from PUEBI Daring: https://puebi.readthedocs.io/en/latest/huruf/huruf-konsonan/

[19] dictionary.cambridge.org. (2018). *CLUSTER*. Retrieved from Cambridge English Dictionary: https://dictionary.cambridge.org/dictionary/english/cluster

[20] puebi.readthedocs.io. (n.d.). *Gabungan Huruf Konsonan*. Retrieved from PUEBI Daring: https://puebi.readthedocs.io/en/latest/huruf/gabungan-huruf-konsonan/

[21] Dave, B., & Pipalia, P. D. (2014). SPEECH RECOGNITION: A REVIEW. *International Journal of Advance Engineering and Research*, 7.

[22] M.A.Anusuya, & S.K.Katti. (2009). Speech Recognition by Machine: A Review. *International Journal of Computer Science and Information Security*, 25.

[23] Geitgey, A. (2016, December 24). *Machine Learning is Fun Part 6*. Retrieved from Medium: https://medium.com/@ageitgey/machine-learning-is-fun-part-6-how-to-do-speech-recognition-with-deep-learning-28293c162f7a

[24] en.wikipedia.org. (2018, October). *Nyquist–Shannon sampling theorem*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Nyquist%E2%80%93Shannon_sampling_theorem

[25] Olshausen, B. A. (2000). Aliasing. *Redwood Center for Theoretical Neuroscience*, 6.

[26] Rabiner, L. R., & Schafer, R. W. (1978). *Digital Processing of Speech Signals.* Prentice Hall: New Jersey.

[27] Hande, S. S. (2014). A Review on Speech Synthesis an Artificial Voice Production. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 8.

[28] lyrebird.ai. (2018). *Our Voice Products*. Retrieved from Lyrebird: https://lyrebird.ai/products

[29] translate.google.com. (2018). *Languages*. Retrieved from Google Translate: https://translate.google.com/intl/en/about/languages/

[30] w3school.com. (2018). *Node.js Introduction*. Retrieved from W3School: https://www.w3schools.com/nodejs/nodejs_intro.asp

[31] mongodb.com. (2018). *NoSQL Databases Explained*. Retrieved from MongoDb: https://www.mongodb.com/nosql-explained