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

Copyright by

Valens Prabagita Ivan Susilo

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

Ini abstrak

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, Marcel, Bora, Fira, Rai, Pindy, Oscar, Michel, and others.
3. Sangar Team, Mas Andreas, Mba Diana, Nisa, Alfian, Aci.
4. My friends and Sepulsa Family.

TABLE OF CONTENTS

Page

[ACKNOWLEDGMENTS i](#_Toc526567187)

[TABLE OF CONTENTS ii](#_Toc526567188)

CHAPTER

[I. INTRODUCTION 1](#_Toc526567189)

[1.1. Background 1](#_Toc526567190)

[1.2. Problem Statement 1](#_Toc526567191)

[1.3. Research Objective 1](#_Toc526567192)

[1.4. Scope and Limitation 1](#_Toc526567193)

[1.5. Thesis Methodology 1](#_Toc526567194)

[1.6. Thesis Outline 1](#_Toc526567195)

[II. LITERATURE STUDY 2](#_Toc526567196)

[2.1. Speech Recognition 2](#_Toc526567197)

[2.2. Speech Synthesis 2](#_Toc526567198)

[2.3. Related Work 2](#_Toc526567199)

[III. SYSTEM ANALYSIS 3](#_Toc526567200)

[3.1. System Overview 3](#_Toc526567201)

[3.2. Functional Analysis 3](#_Toc526567202)

[3.3. Development Process Analysis 3](#_Toc526567203)

[3.4. Use Case Diagram 3](#_Toc526567204)

[3.5. Use Case Narrative 3](#_Toc526567205)

[3.6. Comparison Overview with Related Work 3](#_Toc526567206)

[3.7. Hardware and Software Requirements 3](#_Toc526567207)

[IV. SYSTEM DESIGN 4](#_Toc526567208)

[4.1. User Interface Design 4](#_Toc526567209)

[4.2. Apa lagi 4](#_Toc526567210)

[REFERENCES iii](#_Toc526567211)

# INTRODUCTION

## 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.

## 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.

## Research Objective

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

## 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.

## 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.

1. 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.

1. 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.

1. 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.

## 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.

1. Chapter II: Literature Study

Literature Study describes about Linguistic Component and Speech Recognition and also Speech Synthesis in general. It consists of Linguistic Component, Speech Recognition and Speech Synthesis also Related Work.

1. Chapter III: System Analysis

System Analysis describes the analysis of the behaviour and function of the mimic system. It consists of System Overview, Functional Analysis, Development Process Analysis, Use Case Diagram, Use Case Narrative, Comparison Overview with Related Work and Hardware and Software Requirements.

1. 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, Physical Design, Database Design, and Class Diagram of the program.

1. Chapter V: System Implementation

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

1. 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.

1. 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.

# LITERATURE STUDY

## Linguistic Component

## Speech Recognition

## Speech Synthesis

## Related Work

### Lyrebird

Lyrebird is website application, https://lyrebird.ai, that has 3 products: custom voice, vocal avatar, and vocal avatar API. 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 1. Screenshot of Lyrebird in the website

### Google Translate

Google Translate is an application to translate languages. It has feature to input text from speech and generate speech from text in any languages.

Figure 2. Screenshot of Google Translate in website

# SYSTEM ANALYSIS

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

## System Overview

This thesis 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 trained enough, 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.

## 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. |

## 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.

### 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.

### Microsoft Office Word

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

### 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 (w3school.com, 2018). In this research, Node.js v8.11.4 is used.

### 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 (mongodb.com, 2018). In this research, MongoDb Community Server 4.0.3 as database server and MongoDb Compass 1.15.4 as MongoDb UI.

### Integrated Development Environment (IDE)

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

### Git

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

## 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.

### 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.

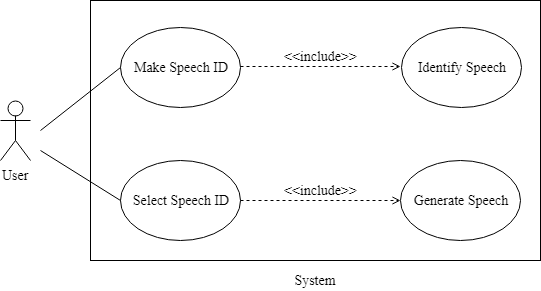


Figure 3.1. Use-Case Diagram

### 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 3.3.

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. |
| Do Identify Speech. | Process Speech to Speech Data. |
| Finish Identify Speech. | Back to home screen. |
| Alternate Course | None | |
| Post Condition | Home screen is shown. | |
| Business Rule | None | |
| Implementation Constraint and Specifications | One speech ID for 1 speech data. | |

Table 3.3. Use-Case Narrative – Select Speech ID

|  |  |  |
| --- | --- | --- |
| User Case Name | Select 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 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. |
| Selected Speech ID. | Start Generate Speech Activity. |
| Generate Speech. | Speech generated based on speech ID data. |
|  | Finish Generate Speech | Back to home screen. |
| Alternate Course | None | |
| Post Condition | Home screen is shown. | |
| Business Rule | None | |
| Implementation Constraint and Specifications | None | |

### 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.

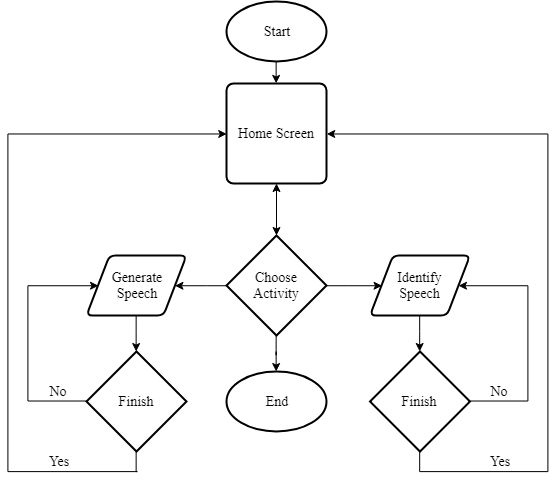


Figure 3.2. Activity Diagram

# SYSTEM DESIGN

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

## 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.

### 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 |

### 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.

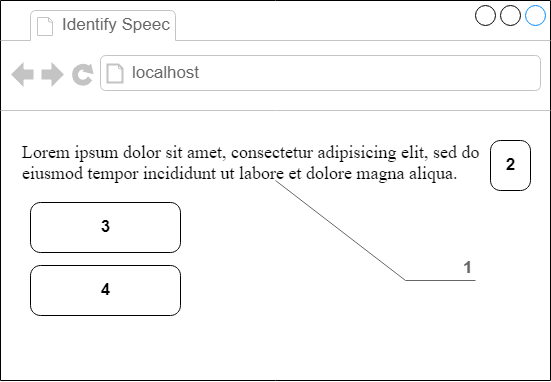


Figure 4.2. Identify Speech Screen

Table 4.2. Identify Speech Screen Description

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

### 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.

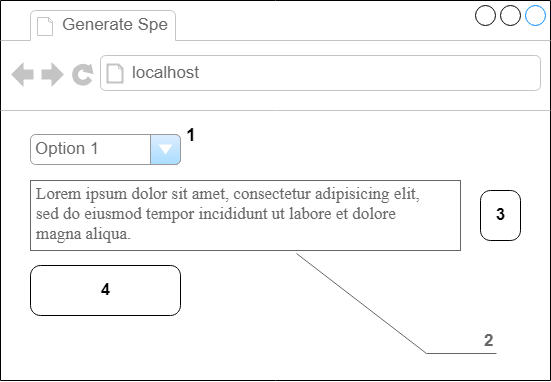


Figure 4.3. Generate Speech Screen

Table 4.3. Generate Speech Screen Description

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

## 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. In general, front-end library consists of 3 classes which are Home, IdentifySpeech, and GenerateSpeech. In general, back-end library consists of 7 classes which are SpeechRecognition, SpeechSynthesis, MongoDb, SpeechCollection, SpeechDataCollection, SpeechTrainedCollection, and TextCollection. The classes diagram details are shown in Figure 4.4 and Figure 4.5.

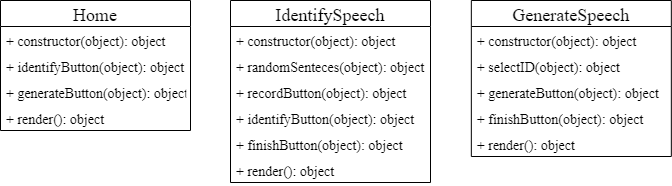


Figure 4.4. Front-end Class Diagram

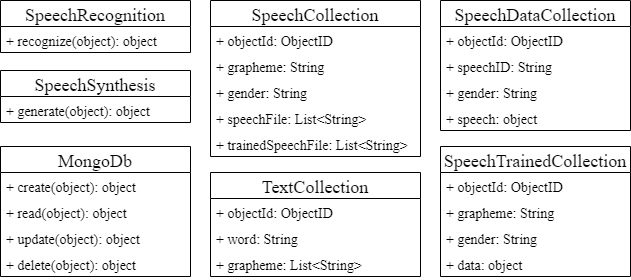


Figure 4.5. Bank-end Class Diagram

REFERENCES

w3school.com. (2018, October 18). *Node.js Introduction*. Retrieved from w3school: https://www.w3schools.com/nodejs/nodejs\_intro.asp