Speech Application Programming Interface for English Pronunciation Learning

Achmad Teguh Wibowo[1], Moch Yasin[1], Mohammad Khusnu Milad[1], Muhammad Andik Izzuddin[1] and Faris Mushlihul Amin[1]
[1]*Departement of Information System, UIN Sunan Ampel, Jl. A Yani 117, Surabaya, Indonesia*

Keywords: Learning, Teaching, Media, Speech Application Programming Interface, Human-Computer Interaction.

Abstract: Teaching and learning activity is a compulsory activity that should be done by all citizens in Indonesia for at least nine years. This activity involves a role of a teacher in teaching students. A learning material, such as English language, should be practiced by students for better understanding, whereas the quality of English teachers in Indonesia varies between people. This tends to make students have differing levels of skill mastery in this language learning depending on the teacher, especially in pronunciation. Current development of technologies allows people to be more interactive. One of these technologies is the *Speech Application Programming Interface*. This technology allows people to *talk* to the computer. This research proposes a new learning method that involves a computer system that displays a set of attractive images and has the ability to listen to student's voice. Voice input is converted into text and will be matched to the data using an if-then method. If the text is in the data, the student's answer is correct. Thus, human-computer interaction can improve the learning process because students become happier and have more standardized pronunciation skill results.

1 INTRODUCTION

In the learning and teaching activities at a school, interactions take place between a teacher and students or between a student and another when there is a group learning activity. In these interactions, a learning process occurs. Learning is generally defined as a process that brings together cognitive, emotional, and environmental influences and experiences to obtain, improve, or make changes, knowledge, skills, values, and worldviews (Illeris, 2000).

A teaching and learning process is a process of action that is done intentionally, which then leads to a change, the situation of which is different from the actions caused by others (Suryabrata, 2001).From the definition, not all changes refer to teaching and learning. Some changes contain a conscious effort to achieve certain goals. A teaching and learning process will be more effective if it is not only taught, but also practiced directly.

Current technological developments have been able to facilitate teaching and learning activities to become more interactive. One of such developments is the Speech Application Programming Interface (SAPI) technology. SAPI is the result of research conducted by Microsoft Corporation in the form of technology and standards for speech ingestion that can be processed by a computer (speech-to-text) (Syarif et al., 2011).

This research's product is a computer-/application-based learning medium. This application will try to translate a user's speech into text, which will be matched with the existing database. If the text matches the data that appears, the user will get an additional point. If it does not match, it will switch to the next question without giving the user additional points.

The use of the SAPI technology for the teaching and learning of English pronunciation in the early childhood education can be an interesting solution. Because by utilizing computers a learning process can be more interactive and less boring, the principle of Human-Computer Interaction (HCI) is to make the system engage in dialogues with the user in as friendly and efficient a manner as possible.

Wibowo, A., Yasin, M., Milad, M., Izzuddin, M. and Amin, F.
Speech Application Programming Interface for English Pronunciation Learning.
DOI: 10.5220/0008903800002481
In *Proceedings of the Built Environment, Science and Technology International Conference (BEST ICON 2018)*, pages 321-325
ISBN: 978-989-758-414-5

2 BASIC THEORY

2.1 Natural Language

Natural language is an interactive communicative mechanism. The computer cannot understand the instructions that humans use in everyday language. Natural language can understand written input and speech input (Dix, 2009). However, there are still deficiencies in terms of ambiguity in syntactic and semantic aspects. The oral dialogue system consists of several components needed so that the system can function successfully, including speech recognition and text-to-speech systems (Mc Tear, 2004).

2.2 Speech Recognition

Each speech recognition system essentially runs a process for recognizing human speech and turning it into something that is understood by computers (Amundsen, 1996). Research into effective speech recognition algorithms and processing models has lasted almost since computers were created. There are 4 key operations of listening and understanding human speech.

1. *Word separation* is the process of creating *discreet* from human speech. Each section can be a large or small phrase as a single syllable or a word part.
2. *Vocabulary* is a list of sound items that can be identified by speech recognition engines.
3. *Word matching* is a speech recognition system method that is used to search for voice parts in the vocabulary system.
4. *Speaker dependence* is the extent to which speech recognition machines depend on vocal tones and individual speech patterns.

The last element of speech recognition is grammar rules. Grammar rules are used by speech recognition software to analyze human speech input in the process of trying to understand what someone is saying. There are many grammar rules, each of which consists of a set of pronunciation rules. One of them is the context-free grammar (CFG). The main elements of CFG are the following:

1. *Word*, a list of valid cases to say;
2. *Rules*, greeting structures for words used; and
3. *List*, word list for use in rules.

2.3 *Speech Application Programming Interface (SAPI)*

Two basic types of SAPI machines are text-to-speech and speech recognition. The TTS system synthesizes text strings and files into speech audios using synthetic sounds. On the other hand, speech recognition converts human-spoken audios into readable text strings. The application can control text-to-speech by using ISpVoice's Component Object Model (COM). After the application is created, call ISpVoice: Speak to generate speech output from some text data.

An application has two choices of speech recognition engine types (ISpRecognizer). A shared recognition that might be shared with other speech recognition applications is recommended for most speech recognition applications. To create an ISpRecoContext for a shared ISpRecognizer (shared), the application only needs to make a CoCreateInstance COM call to the CLSID_SpSharedRecoContext component. For large server applications that will run alone on a system, where performance is the key, the InProc speech recognition engine is more appropriate. In order to create ISpRecoContext for ISpRecognizer InProc, firstly, the application must call CoCreateInstance on the CLSID_SpInprocRecoInstance component to create its own ISpRecognizer InProc. Next, the application can call ISpRecognizer: CreateRecoContext to get an ISpRecoContext.

Finally, a speech recognition application must create, load, and activate ISpRecoGrammar, which basically shows the type of utterance to be recognized, namely dictation or command and control. First, this application creates ISpRecoGrammar using ISpRecoContext: CreateGrammar. Then, it loads the appropriate grammar application either by calling ISpRecoGrammar: loadDictation for dictation or one of the methods ISpRecoGrammar: LoadCmdxxx for commands and controls. Finally, this grammar rule is activated so that speech recognition data starts the application for commands and controls (Syarif et al., 2011).

3 RESEARCH CONCEPT

3.1 Thinking Framework

The framework for thinking in this research is shown in Figure 1.

3.2 Flow Chart and Problem Solution

Flow charts and problem solutions from the *Speech Application Programming Interface* for *English Pronunciation Learning* appear in Figure 2 starting

from the user giving the voice captured by the microphone. The data captured by the microphone is processed by the SAPI which changes speech to text. The resulting text will be captured into the input form that will be matched into the database. If the text matches the database, the points will increase and the next question will be shown. However, if it does not match, the user is given time to say the word again within a specified time period (±15 seconds). If the data input until the time is over is still wrong, the user should proceed to the next question without getting additional points.

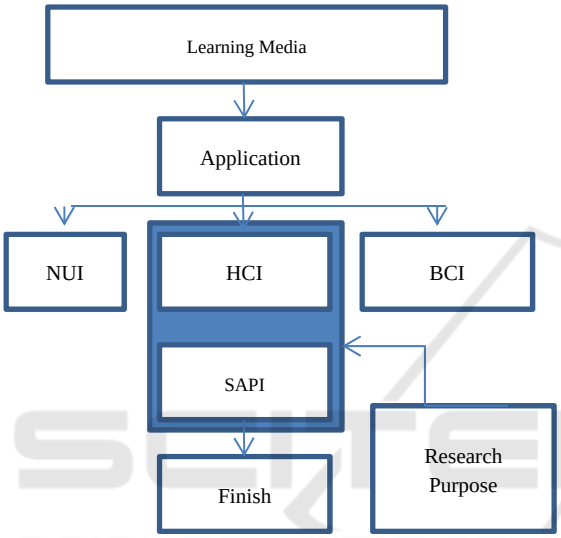

Figure 1: Framework for Thinking.

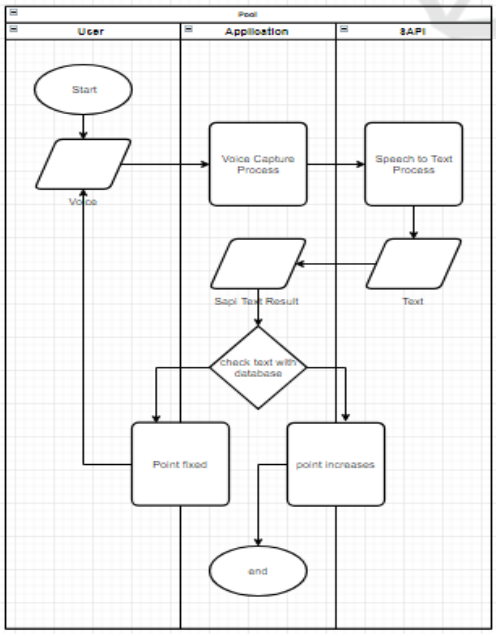

Figure 2: Flowchart and Problem Solution.

4 RESULTS

4.1 View of Home

Figure 3 is the *view of home* of the application. In the application view, there is a navigation that shows the total points that are used for displaying the points obtained by the user in using this application. There is also a folder that is used to navigate the questions in the application. There are *other games* to choose and a *timer* that functions as a *counter next* to the problem if the user does not answer correctly.

Figure 3: *View of Home.*

4.2 Wrong Answer

Figure 4 is the view of wrong answer of the application. In the application display, it can be seen that the map number 1 is red, which indicates that the answer is wrong. Figure 4 can be seen below.

Figure 4: *View* of Wrong answer.

4.3 Right Answer

Figure 5 is the view of correct answer of the application. In the application, it can be seen that maps number 1, 2, and 3 are blue, which indicates that the *user's* answers are correct. Correct answers will affect the total points. In this view, the user gets total points of 60, which come from 3 correct answers.

Each answer is worth 20 points. Figure 5 can be seen below.

Figure 5: *View of Right answer*.

4.4 Finish View of the Application

Figure 6 is the *finish view* of the application. In the *application view*, the total points that were successfully obtained by the *user* are displayed, and in the map, it is shown that there are 3 correct answers, with the maps turning blue, and 2 incorrect answers, with the maps turning red. In addition, there is a message of total wages earned. Figure 6 can be seen as follow.

Figure 6: *Finish View of the* Application.

4.5 System Testing

Application testing was conducted using computer devices in the form of laptops and all-in-one desktop computers with the following specifications: For laptop we used Lenovo IdeaPad Z480 laptops, hereinafter referred to as Laptop (specification as seen on table 1) and Lenovo IdeaCentre desktop PCs, hereinafter referred to as Desktop, with the the specification can be seen on table 2/.

Table 1: Laptops specification.

Processor	Intel Core i7
GPU	Nvidia GeForce Cuda
Internal	RAM 8 Gb
Wi-fi	Ready
Bluetooth	Ready
USB	USB 2 and USB 3
Connectivity	HDMI, camera, microphone, Ethernet and USB
Storage	HDD 500 GB
OS	Windows 7

Table 2: Desktop computers specification.

Processor	Intel Core i7
GPU	Nvidia GForce Cuda
Internal	RAM 4 Gb
Wi-fi	Ready
Bluetoth	Ready
USB	USB 2 and USB 3
Connectivity	HDMI, camera, microphone, Ethernet and USB
Storage	HDD 250 GB
OS	Windows 10

The application testing result can be seen on table 3.

Table 3: Results of application testing.

Features	Laptop	Desktop
Display the start page	✓	✓
Change points if the answer is correct	✓	✓
The application keeps countering if the answer is wrong	✓	✓
Change the map to blue if the answer is correct	✓	✓
Change the map to red if the answer is wrong	✓	✓
Display the total points if all questions are done	✓	✓
Use a man’s voice	✓	✓
Use a woman's voice	✓	✓

5 CONCLUSIONS

From the test results, it can be concluded that the Speech Application Programming Interface for English pronunciation learning has been completed. It is suggested that future research should add

facilities to make more applications such as vehicles, animals, etc. and that this application should be developed into a mobile-based application. The SAPI technology can be used not only to learn English but also to learn other languages such as Javanese, Chinese, Arabic, etc.

REFERENCES

Amundsen, Michael C. MAPI, (1996) . *SAPI and TAPI Developer's Guide,* Indianapolis, Sams Publishing

Dix, Alan, Janet Finlay, Gregory D. Abowd, and Russle Beale. (2009). *Human Computer Interaction,* Third Edition, New Delhi, Pearson Education

Hadi, Rahadian. (2005). *Panduan Bagi Pemrograman Microsoft Windows Common Control,* Jakarta, Alex Media Komputindo

Hendrayudi. (2009). VB 2008 *untuk Berbagai Keperluan Programming,* Jakarta, Alex Media Komputindo

McTear, Michael F. (2004). *Spoken Dialogue Technology Toward the Conversational User Interface,* London, Springer

Syarif A, Daryanto T, Arifin MZ. (2011). *Aplikasi Speech Application Programming Interface (SAPI) 5.1 Sebagai Perintah Untuk Pengoperasian Aplikasi Berbasis Windows,* Yogyakarta, SNATI 2011