



Aplikasi Pembelajaran Hiragana Bahasa Jepang Berbasis Android Menggunakan Speech Recognition

Ryan Anjasmara¹, Indah Lestari² dan Meilany Dewi³

¹Politeknik Caltex Riau, email: ryan15ti@mahasiswa.pcr.ac.id

²Politeknik Caltex Riau, email: indah@pcr.ac.id

³Politeknik Caltex Riau, email: meilany@pcr.ac.id

Abstrak

Alternatif mempelajari bahasa Jepang adalah dengan menggunakan mobile learning. Berdasarkan perbandingan yang dilakukan terhadap 7 aplikasi pembelajaran bahasa Jepang yang populer di playstore, belum terdapat aplikasi pembelajaran bahasa Jepang yang memiliki fitur speech recognition untuk pengucapan kalimat Jepang. Pada penelitian ini dirancang aplikasi pembelajaran hiragana Jepang untuk kalimat menggunakan speech recognition. Aplikasi ini menggunakan google cloud speech-to-text untuk fitur speech recognition. Algoritma yang diterapkan pada aplikasi ini adalah levenshtein distance untuk membandingkan inputan suara pengguna dengan teks pada database. Aplikasi berbasis android untuk pengguna dan web untuk admin. Berdasarkan hasil pengujian akurasi dan keberhasilan deteksi, jarak ideal aplikasi menerima inputan suara pengguna adalah 50 cm pada lingkungan tanpa noise dengan keberhasilan deteksi sebesar 100%. Pada pengujian sebelum dan sesudah menggunakan aplikasi terjadi peningkatan pengetahuan kosa kata sebesar 44.48% setelah pengguna menggunakan aplikasi. Berdasarkan pengujian usability menggunakan SUS diperoleh hasil 73,01 yang menunjukkan bahwa aplikasi termasuk kedalam kategori acceptable.

Kata kunci: Bahasa Jepang, Hiragana, Levenshtein Distance, Speech Recognition

Abstract

The alternative to learning Japanese is to use mobile learning. Based on comparisons made on the 7 popular Japanese language learning applications in the PlayStore, there is no Japanese language learning application that has a speech recognition feature for Japanese sentence pronunciation. In this study Japanese hiragana learning application was designed for sentences using speech recognition. This application uses google cloud speech-to-text for speech recognition features. The algorithm that is applied in this application is the levenshtein distance to compare the user's voice input with the text in the database. Android based application for users and the web for admin. Based on the results of testing the accuracy and success of detection, the ideal distance of the application receiving the user's voice input is 50 cm in a noise-free environment with a successful detection of 100%. In testing before and after using the application there was an increase in vocabulary knowledge of 44.48% after users used the application. Based on usability testing using SUS, the result is 73.01 which indicates that the application falls into the acceptable category.

Keywords: Japanese, Hiragana, Levenshtein Distance, Speech Recognition

1. Pendahuluan

Bahasa adalah hal yang diperlukan untuk berkomunikasi antar manusia, salah satunya adalah Bahasa Jepang. Bahasa Jepang dibagi kedalam 3 sistem penulisan yaitu *hiragana*, *katakana*, dan *kanji*. Ketiga sistem penulisan tersebut umumnya digunakan oleh penduduk Jepang dengan penggunaan *hiragana* yang lebih dominan.

Bahasa serta budaya jepang sudah menjadi hal yang tidak asing lagi di Indonesia. Bahasa Jepang bahkan menjadi salah satu pilihan mata pelajaran bahasa asing dari banyak SMA dan SMK di Indonesia. Menurut hasil survey oleh The Japan Foundation yang dilaksanakan pada tahun 2015, Indonesia menduduki peringkat kedua dalam jumlah orang yang mempelajari bahasa Jepang terbanyak di dunia, dengan jumlah 745,125 pelajar, dan merupakan 20.4% dari jumlah total pelajar bahasa Jepang didunia [1]. Hal ini menunjukkan tingginya minat orang Indonesia untuk mempelajari bahasa Jepang. Ada banyak cara untuk mempelajari bahasa Jepang salah satunya melalui kursus namun belajar melalui kursus memiliki kekurangan yaitu adanya batasan pertemuan, pertemuan yang dilakukan harus melalui tatap muka, serta pelajar harus menempuh jarak untuk dapat pergi ke kursus. Alternatif untuk mempelajari bahasa Jepang adalah dengan menggunakan teknologi pembelajaran.

Teknologi pembelajaran memiliki banyak model diantaranya terdapat model web, *desktop* dan *mobile*. Ketiga model tersebut memiliki karakteristiknya masing-masing. Karakteristik *mobile* yaitu memiliki ukuran yang lebih kecil jika dibandingkan dengan perangkat *desktop* sehingga perangkat *mobile* lebih mudah dibawa. Perangkat *mobile* memiliki banyak jenis sistem operasi salah satunya adalah android. Android merupakan sistem operasi *open source* yang memiliki performa cepat dan responsif. Android menggunakan *playstore* sebagai toko online untuk mendownload aplikasi. Berdasarkan perbandingan terhadap 7 aplikasi pembelajaran bahasa Jepang yang populer, belum terdapat aplikasi pembelajaran bahasa jepang yang memiliki fitur *speech recognition* untuk pengucapan kalimat Jepang.

Berdasarkan paparan di atas, dibangunlah sebuah sistem dengan judul “Aplikasi Pembelajaran Hiragana Bahasa Jepang Berbasis Android Menggunakan *Speech Recognition*”. Metode pembelajaran yang diterapkan aplikasi ini adalah melalui latihan pengucapan. *Speech Recognition* pada aplikasi ini digunakan untuk membantu pengguna dalam belajar pengucapan bahasa jepang kapan dan dimana saja tanpa perlu bertatap muka dengan pengajar. Aplikasi ini juga memiliki fitur kamus beserta arti dan cara pengucapan untuk membantu pengguna dalam menambah perbendaharaan kosa kata. Aplikasi ini juga dilengkapi dengan website admin sehingga admin dapat mengelola latihan yang diberikan kepada pengguna secara berkala.

2. Metode Penelitian

2.1 Metodologi Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah :

- 1) Studi Literatur
Mencari referensi teori yang saling berkaitan dengan masalah yang ditemukan. Pengumpulan materi dilakukan seperti buku-buku dan jurnal mengenai *katakana*, *hiragana*, *kanji*, *database*, *mobile learning*, android dan *speech recognition*.
- 2) Observasi

Melakukan pengamatan terhadap aplikasi sejenis yang sudah pernah ada di dunia dan membandingkannya dengan aplikasi yang akan dibuat.

- 3) Perancangan Sistem
Setelah melakukan studi literatur dan observasi, maka akan dilakukan perancangan-perancangan secara rinci mengenai bagaimana sistem atau aplikasi berjalan.
- 4) Implementasi Sistem
Melakukan pembuatan sistem dengan menerapkan perancangan yang telah dibuat.
- 5) Pengujian
Setelah membangun pembelajaran *hiragana* jepang berbasis android ini, maka dilakukanlah tahapan pengujian dimana akan dilakukan kuisioner pra proyek akhir untuk mengetahui hasil pembelajaran dari aplikasi.
- 6) Analisa dan Evaluasi
Setelah melakukan implementasi dan pengujian, dilakukan beberapa analisa dan evaluasi dari hasil yang telah didapat dan akan dilakukan pembuatan laporannya.

2.2 Review Penelitian Terdahulu

Penelitian mengenai pengenalan suara untuk bahasa jepang telah dilakukan oleh Prastiti dan Samopa (2015). Prastiti dan Samopa melakukan penelitian untuk menganalisis pelafalan bahasa Jepang untuk identifikasi penulisan aksara Jepang berdasarkan aksent. Metode yang digunakan adalah mendeteksi aksent menggunakan pola frekuensi yang kemudian dilakukan ekstraksi ciri menggunakan algoritma *Fast Fourier Transform* (FTT) dan *MelFrequency Cepstral Coefficient* (MFCC) untuk kata yang memiliki suku kata yang sama namun memiliki aksent yang berbeda, sehingga akan dihasilkan *kanji* dan arti yang berbeda dengan melakukan analisis pengucapan berdasarkan aksent [2].

Penelitian lainnya adalah dari Abdullah dan Erliana (2017) yang melakukan penelitian untuk pengenalan ucapan huruf Jepang menggunakan *speech recognition*. Pada perancangannya, aplikasi ini dibangun berbasis desktop. Penelitian ini dilakukan dengan 10 orang user yang mengucapkan huruf konsonan vokal Jepang *hiragana* yaitu “Ka, Ki, Ku, Ke, Ko, Sa, Shi, Su, Se, So”. Hasil dari pengenalan ucapan menunjukkan keakuratan aplikasi dengan nilai kedekatan rata-rata adalah 0,83234179 atau 83% [3].

Penelitian selanjutnya dilakukan oleh Prasetyo, Candra dan Hendra (2018) yang melakukan penelitian untuk membangun aplikasi pembelajaran bahasa jepang dengan nama *hanasu*. Aplikasi yang akan dibuat berbasis android dengan memanfaatkan *Google Speech Recognition Library*. Berdasarkan perbandingan pengujian performa antara materi yang didapat dari buku dengan materi aplikasi *hanasu*, pada pengujian pertama (menghafal nama-nama hewan dalam bahasa Jepang) didapatkan peningkatan pemahaman sebanyak 2% lebih tinggi pada android dibandingkan dengan menggunakan buku, sedangkan pada pengujian kedua (kosa kata yang sering diucapkan) didapatkan peningkatan sebanyak 8% [4].

2.3 Landasan Teori

2.3.1 Hiragana

Hiragana adalah alfabet fonetis dasar di Bahasa Jepang. Seluruh huruf *hiragana* melambangkan semua kata yang muncul di Bahasa Jepang [5]. *Hiragana* memiliki total jumlah huruf sebanyak 46 buah. *Hiragana* merupakan sebuah penyederhanaan dari huruf *kanji* yang digunakan oleh Cina di abad ke-5. Pada awal perkembangannya, *hiragana* tidak langsung

diterima oleh penduduk Jepang. Cendekiawan dan kaum *elite* di Jepang saat itu, menolak untuk menggunakan huruf *hiragana* dan hanya menggunakan huruf *kanji* saja [6].

2.3.2 Levenshtein Distance

Algoritma *Levenshtein Distance* digunakan secara luas dalam berbagai bidang, seperti halnya mesin pencari (*search engine*), pengecek ejaan (*spell checking*), pengenalan ucapan (*speech recognition*), pengucapan dialek, analisis DNA, pendeteksi pemalsuan, dan lain-lain. Algoritma ini menghitung jumlah operasi *string* paling sedikit yang diperlukan untuk mentransformasikan suatu *string* menjadi *string* lain yang meliputi penghapusan, penyisipan dan penukaran [7].

2.3.3 Mobile Learning

O'Malley telah mendefinisikan bahwa *mobile learning* merupakan suatu metode pembelajaran dimana pelajar tidak harus berada di lokasi yang telah ditentukan, pelajar juga dapat memanfaatkan semua fitur yang disediakan oleh teknologi *mobile* [8].

2.3.4 Android

Android sendiri merupakan sistem operasi *open source* yang memiliki performa cepat dan responsif, android juga dapat diaplikasikan di banyak peralatan elektronik. Namun android juga memiliki kekurangan yaitu memiliki proses kerja sistem yang cukup berat dikarenakan penggunaan RAM dalam pengoperasiannya, android juga membutuhkan spesifikasi hardware yang baik agar dapat berjalan stabil [9].

2.3.5 MySQL

MySQL adalah sebuah sistem manajemen database relasi (*relational database management system*) yang bersifat *open source*. MySQL merupakan buah pikiran dari Michael "Monty" Widenius, David Axmark dan Allan Larson yang di mulai tahun 1995. Mereka bertiga kemudian mendirikan perusahaan bernama MySQL AB di Swedia [10].

2.3.6 JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman Javascript, Standar ECMA-262 Edisi ke-3 – Desember 1999. JSON terdiri dari 2 struktur yaitu kumpulan pasangan nama/nilai (*object, record, struct, dictionary, hash table, keyed list*) dan daftar nilai terurutkan (*array, list, vector, sequence*) [11].

2.3.7 Pengujian Fungsionalitas

Pengujian fungsionalitas merupakan bagian dari pengujian *black box*. *Black box* adalah metode pengujian perangkat lunak di mana struktur internal / desain / implementasi item yang diuji tidak diketahui oleh penguji. Tes-tes ini dapat berupa pengujian fungsionalitas maupun bukan pengujian fungsionalitas, meskipun biasanya adalah pengujian fungsionalitas [12].

2.3.8 Pengujian Usabilitas

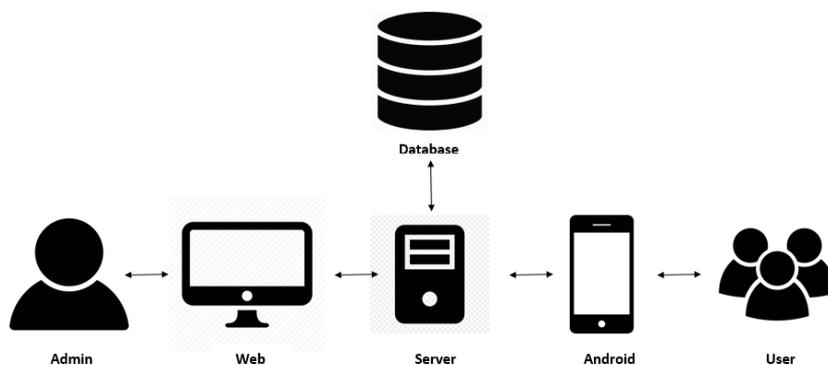
Menurut Jakob Nielsen, *Usability* meliputi: *Learnability* (fungsi-fungsi dasar produk mudah dioperasikan sejak awal), *Efficiency* (fungsi-fungsi produk dapat dengan cepat dipergunakan), *Memorability* (setelah pengguna meninggalkan sistem dalam waktu lama, lalu bisa dengan mudah mengerti lagi cara penggunaannya), *Errors* (seberapa banyak dan serius

kesalahan penggunaan produk, dan bagaimana pengguna dapat memperbaiki kesalahan), *Satisfaction* (seberapa puaskah pengguna) [13].

3. Perancangan

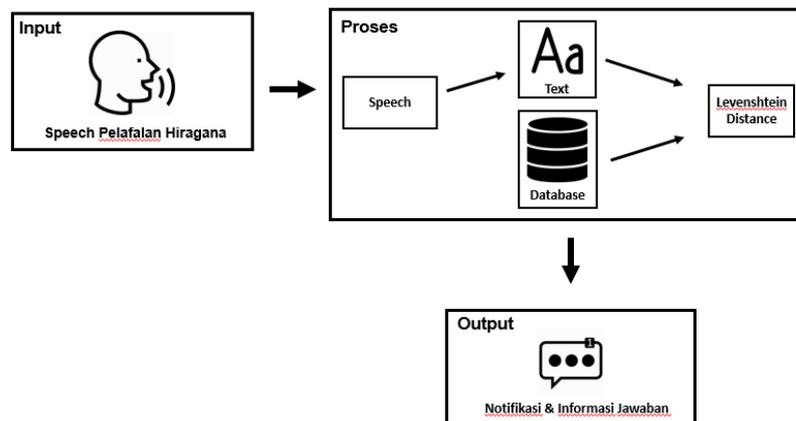
3.1 Arsitektur Aplikasi

Gambar 1 merupakan arsitektur sistem aplikasi pembelajaran *hiragana*. Pada gambar dapat dilihat bahwa terdapat 2 aktor yang bisa mengakses aplikasi. Admin menggunakan aplikasi berbasis Website sementara untuk user menggunakan aplikasi berbasis Android. Saat melakukan *request* baik menggunakan Android ataupun Website, *request* diolah oleh server dan hasil olahan akan dikembalikan berupa *response*. Selanjutnya jika diperlukan data dari *database*, maka server akan mengambil data dari *database* dan dikembalikan kembali dalam bentuk *response*.



Gambar 1. Arsitektur Sistem

3.2 Blok Diagram Android

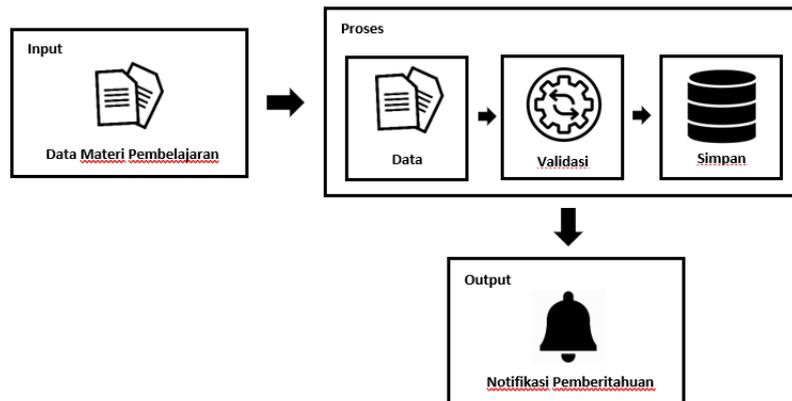


Gambar 2. Blok Diagram Android

Pada *block input*, aplikasi menangkap *speech* pelafalan *hiragana* yang diberikan oleh *user* yang berupa suara melalui perangkat *microphone*. Selanjutnya pada *block proses*, suara tersebut akan dikonversikan menjadi *text*. *Text* tersebut kemudian akan diolah bersamaan dengan *text* yang ada pada *database* menggunakan algoritma *levenshtein distance*. Berikutnya dilanjutkan pada *block output*, jika pelafalan yang berikan *user* sesuai dengan *database* maka aplikasi akan menampilkan notifikasi dan informasi jawaban yang berupa terjemahan latin dan terjemahan bahasa Indonesia.

3.3 Blok Diagram Web

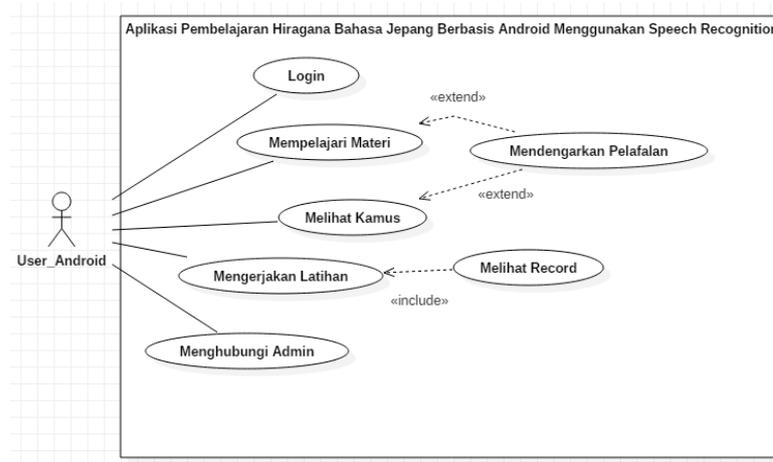
Pada *block input*, sistem menerima data materi yang diinputkan oleh admin. Selanjutnya pada *block proses*, data tersebut akan diperiksa oleh sistem apakah telah sesuai dengan format yang ditentukan. Jika telah sesuai data tersebut kemudian di simpan ke dalam *database*. Selanjutnya pada *block output*, akan dimunculkan notifikasi pemberitahuan apakah data tersebut telah berhasil di simpan atau tidak berhasil di simpan.



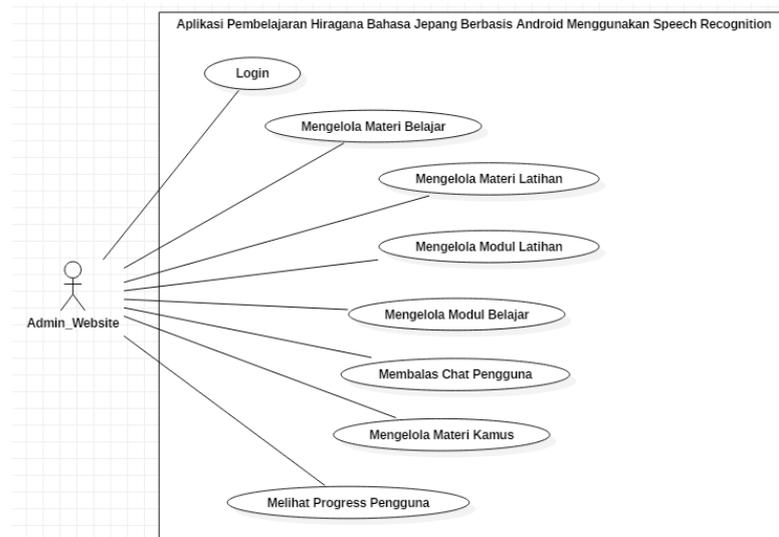
Gambar 3. Blok Diagram Web

3.4 Use Case Diagram

Use case diagram digunakan untuk mendeskripsikan kegunaan aplikasi dari sisi aktor untuk tujuan yang spesifik. Terdapat dua *use case diagram* pada aplikasi pembelajaran ini, yaitu *use case* untuk aplikasi berbasis Android dengan aktor yaitu pelajar (*user_android*) dan *use case* untuk Website dengan aktor yaitu admin.



Gambar 4. Use Case Aplikasi Android



Gambar 5. Use Case Web

3.5 Algoritma Levenshtein Distance

Pada perancangan aplikasi ini, digunakan algoritma *levenshtein distance* pada proses *speech recognition* untuk menentukan apakah pelafalan yang diucapkan oleh *user* sudah benar atau tidak. Pada algoritma *levenshtein distance* terdapat 3 operasi, yaitu *insertion*, *deletion* dan *substitution*.

Sebagai contoh, soal menampilkan kalimat *hiragana* “わたしはあに” yang dibaca “*watashi wa ani*”. Lalu user melafalkan “*watakushi wa benin*”, maka dengan menggunakan *library google*, pelafalan yang dilakukan oleh *user* akan dikonversi menjadi teks *hiragana* “*わたくしはべにん*”. Selanjutnya kalimat “*わたしはあに*” dan “*わたくしはべにん*” akan diolah menggunakan algoritma *levenshtein distance*. Contoh perhitungan *distance* akan dijabarkan pada tabel berikut ini.

Tabel 1. Contoh Perhitungan Jarak / Distance

		わ	た	し	は	あ	に
	0	1	2	3	4	5	6
わ	1	0	1	2	3	4	5
た	2	1	0	1	2	3	4
く	3	2	1	1	2	3	4
し	4	3	2	1	2	3	4
は	5	4	3	2	1	2	3
べ	6	5	4	3	2	2	3
に	7	6	5	4	3	3	2
ん	8	7	6	5	4	4	3

Kalimat yang dibandingkan adalah “*わたしはあに*” dengan “*わたくしはべにん*”. Berdasarkan hasil perhitungan diatas didapatkan nilai *distance* sebanyak 3. Dimana dibutuhkan 1 operasi *insertion* (ditandai dengan warna kuning), 1 operasi *deletion* (ditandai dengan warna

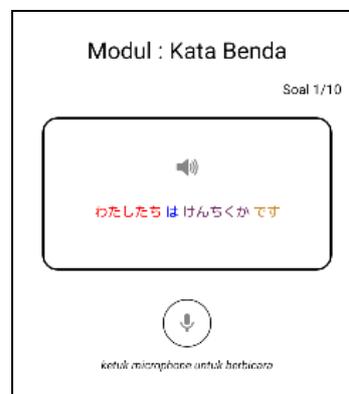
biru) dan 1 operasi *substitution* (ditandai dengan warna hitam). Selanjutnya untuk menguji keabsahannya, digunakan persentase kemiripan yang dihitung dengan rumus berikut ini. $P = 1 - D = 1 - 3 = 0,625 = 62.5\%$ Tidak Benar. Berdasarkan hasil perhitungan persentase kemiripan, didapatkan nilai sebanyak 62.5%. Jawaban dianggap salah karena persentase minimal untuk jawaban benar adalah diatas 80%. Ketika jawaban salah, maka *output* yang akan dikeluarkan oleh aplikasi adalah sebuah notifikasi bahwa jawaban salah, selain itu aplikasi akan menampilkan informasi jawaban seperti terjemahan latin dan terjemahan bahasa Indonesia.

3.6 Implementasi Hasil

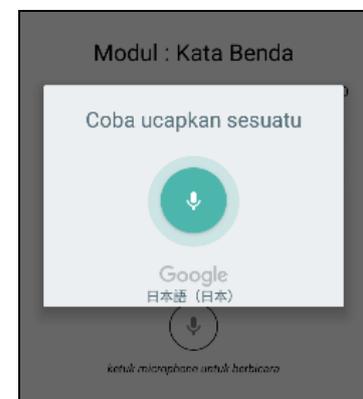
Gambar 6 adalah halaman dari materi belajar yang terdiri dari informasi pola kalimat, penjelasan kalimat, contoh kalimat beserta terjemahan latin dan terjemahan Indonesia. Pengguna juga dapat mendengarkan pengucapan kalimat dengan mengetuk *icon speaker*. Gambar 7 adalah halaman latihan. halaman ini terdiri dari hiragana soal, jumlah soal beserta nama modul yang saat ini sedang dikerjakan. Pengguna juga dapat mengetuk *icon speaker* untuk mendengarkan pengucapan yang benar. Gambar 8 merupakan halaman ketika pengguna akan melakukan proses *speech recognition*. Terdapat animasi *microphone* yang menjadi *indicator* tingkat suara pengguna.



Gambar 6. Halaman Belajar



Gambar 7. Halaman Latihan



Gambar 8. *Speech Recognition*

4. Pengujian dan Analisis

4.1 Akurasi dan Keberhasilan Deteksi

Pengujian akurasi dan keberhasilan deteksi dilakukan untuk mengukur akurasi dan keberhasilan deteksi dari perangkat android terhadap suara user. Hasil dari pengujian akurasi dan keberhasilan deteksi suara yang telah dilakukan menunjukkan bahwa aplikasi android yang telah dibangun dapat menerima inputan suara *user* laki-laki maupun suara *user* perempuan. Pengujian ini dilakukan terhadap 2 kondisi lingkungan yang berbeda yaitu pada lingkungan tanpa *noise* dan pada lingkungan yang memiliki *noise*. Berdasarkan hasil pengujian pada Tabel 2, Tabel 3, Tabel 4 dan Tabel 5 dapat dianalisis bahwa jarak sumber suara *user* terhadap perangkat android dan keadaan lingkungan mempengaruhi keberhasilan deteksi perangkat

android terhadap suara user. Selain itu suara inputan dari user laki-laki dan suara inputan dari user perempuan tidak mempengaruhi keberhasilan deteksi.

Tabel 2. Suara Laki-laki Pada Lingkungan Tanpa Noise

<i>User</i>	Jarak				
	50CM	1M	2M	3M	4M
<i>User 1</i>	100%	100%	100%	100%	100%
<i>User 2</i>	100%	100%	100%	100%	100%
<i>User 3</i>	100%	100%	100%	80%	100%
<i>User 4</i>	100%	100%	100%	100%	60%
<i>User 5</i>	100%	100%	100%	80%	80%
Rata-rata Keberhasilan	100%	100%	100%	92%	88%

Tabel 3. Suara Laki-laki Pada Lingkungan Dengan Noise

<i>User</i>	Jarak				
	50CM	1M	2M	3M	4M
<i>User 1</i>	100%	100%	100%	80%	60%
<i>User 2</i>	100%	40%	60%	80%	80%
<i>User 3</i>	100%	80%	60%	60%	80%
<i>User 4</i>	100%	100%	80%	80%	60%
<i>User 5</i>	100%	100%	100%	80%	60%
Rata-rata Keberhasilan	100%	84%	80%	76%	68%

Tabel 4. Suara Perempuan Pada Lingkungan Tanpa Noise

<i>User</i>	Jarak				
	50CM	1M	2M	3M	4M
<i>User 1</i>	100%	40%	80%	60%	100%
<i>User 2</i>	100%	100%	80%	80%	40%
<i>User 3</i>	100%	80%	80%	80%	60%
<i>User 4</i>	100%	100%	60%	60%	60%
<i>User 5</i>	100%	100%	100%	80%	60%
Rata-rata Keberhasilan	100%	84%	80%	72%	64%

Tabel 5. Suara User Perempuan Pada Lingkungan Dengan Noise

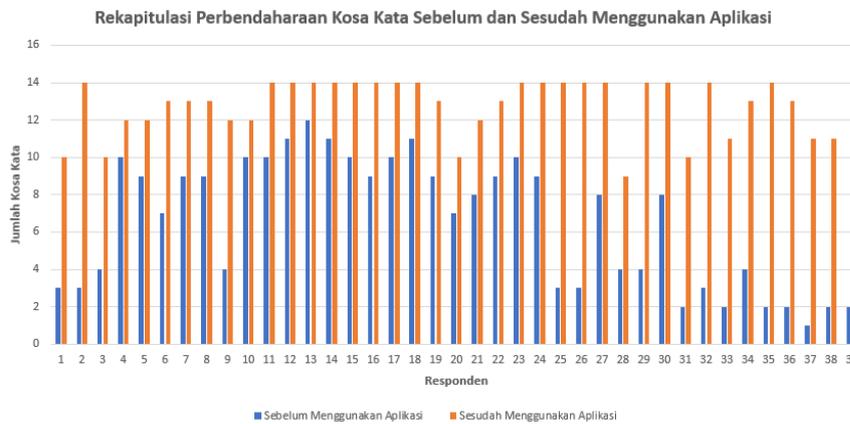
<i>User</i>	Jarak				
	50CM	1M	2M	3M	4M
<i>User 1</i>	80%	100%	60%	20%	80%
<i>User 2</i>	80%	20%	20%	40%	0%
<i>User 3</i>	100%	40%	40%	20%	0%
<i>User 4</i>	60%	60%	40%	40%	20%
<i>User 5</i>	80%	40%	40%	40%	20%
Rata-rata Keberhasilan Tiap Satuan Jarak	80%	52%	40%	32%	24%

Dari pengujian akurasi dan keberhasilan deteksi suara ini, didapatkan bahwa tidak ada perbedaan jarak dan keberhasilan deteksi untuk *user* laki-laki dan perempuan dan jarak paling ideal aplikasi dalam menerima inputan suara *user* laki-laki dan suara user perempuan adalah

sebesar 50 cm pada lingkungan tanpa *noise* dengan rata-rata keberhasilan deteksi jarak uji tersebut adalah sebesar 100%.

4.2 Sebelum dan Sesudah Menggunakan Aplikasi

Pada pengujian ini, 39 user akan diberikan 14 kosa kata Bahasa Jepang yang berhubungan dengan modul 1 pada aplikasi. *User* diminta untuk memberi tanda pada kosa kata yang telah diketahui oleh *user* sebelum menggunakan aplikasi. Selanjutnya user akan mencoba untuk belajar dan mengerjakan latihan modul 1 yang ada pada aplikasi. Setelah belajar dan mengerjakan latihan menggunakan aplikasi, *user* akan diminta untuk mengisi kembali 14 kosa kata yang sebelumnya tidak diketahui *user*. Tujuannya untuk melihat apakah aplikasi pembelajaran yang dibuat dapat membantu *user* dalam menambah kosa kata. Untuk mempermudah penganalisaan data yang diperoleh, berikut grafik yang menggambarkan hasil rekapitulasi sebelum dan sesudah user menggunakan aplikasi.



Gambar 9. Grafik Rekapitulasi Perbendaharaan Kosa Kata

Gambar 10 menunjukkan persentase hasil dari 39 *user* terhadap 14 kosa kata Bahasa Jepang yang berhubungan dengan aplikasi. Sebelum *user* menggunakan aplikasi, didapatkan rata-rata persentase kosa kata yang diketahui sebesar 46,52%, dan pada saat *user* sudah menggunakan aplikasi, rata-rata persentase kosa kata yang diketahui oleh *user* meningkat menjadi 91%. Berdasarkan hasil pengujian sebelum dan sesudah menggunakan aplikasi yang dapat dilihat pada Gambar 7 didapatkan hasil perhitungan bahwa sebelum menggunakan aplikasi, rata-rata persentase kosa kata yang diketahui user sebesar 46,52% dan rata-rata persentase setelah *user* menggunakan aplikasi meningkat menjadi 91%. Berdasarkan hasil tersebut dapat dianalisis bahwa terjadi peningkatan sebesar 44,48% setelah *user* menggunakan aplikasi dan dapat disimpulkan bahwa aplikasi pembelajaran yang telah dibuat dapat membantu *user* dalam menambah perbendaharaan kosa kata.

5. Kesimpulan dan Saran

5.1 Kesimpulan

1. Jarak paling ideal aplikasi *android* dalam menerima inputan suara *user* laki-laki dan suara *user* perempuan adalah pada lingkungan tanpa *noise*, yaitu dengan jarak sejauh 50 cm dan 1 meter dengan rata-rata keberhasilan deteksi sebesar 100%.
2. Berdasarkan pengujian sebelum dan sesudah menggunakan aplikasi yang dilakukan terhadap 39 *user* pada 14 kosa kata Bahasa Jepang, persentase kosa kata yang

diketahui oleh *user* sebelum menggunakan aplikasi adalah sebesar 46,52%. Sedangkan setelah *user* menggunakan aplikasi, persentase kota kata yang diketahui oleh *user* meningkat menjadi 91%, dimana telah terjadi peningkatan sebesar 44.48% setelah *user* menggunakan aplikasi. Hal ini menunjukkan bahwa aplikasi pembelajaran yang dibuat dapat membantu *user* untuk menambah perbendaharaan kosa kata.

5.2 Saran

1. Pada pengembangan selanjutnya diharapkan aplikasi tidak hanya sebagai media pembelajaran untuk pengucapan, tapi mampu untuk penulisan dan pendengaran.
2. Mengambil studi kasus yang berbeda untuk sistem penulisan jepang lainnya seperti *katakana* atau *kanji*.

Daftar Pustaka

- [1] Japan Foundation, "Survey report on Japanese language education abroad 2015," no. October 1972, 2017.
- [2] N. Prastiti *et al.*, "Analisis Pelafalan Bahasa Jepang Sebagai," vol. 1, no. 3, pp. 173–181, 2015.
- [3] D. Abdullah and C. I. Erliana, "Aplikasi Pengenalan Ucapan Huruf Jepang Menggunakan Hidden Markov Model(HMM)," p. 12, 2017.
- [4] E. Prasetyo, L. Nur, K. C. Brata, and A. H. Brata, "Pembangunan Sistem Aplikasi Hanasu : Pembelajaran Bahasa Jepang Android Mobile Memanfaatkan Google Speech Recognition Library," vol. 2, no. 11, pp. 4671–4678, 2018.
- [5] guidetojapanese.org, "Hiragana - Tutorial Bahasa Jepang Tae Kim." [Online]. Available: <http://www.guidetojapanese.org/indonesian/hiragana.html>. [Accessed: 11-Sep-2018].
- [6] tensai-indonesia.com, "Sejarah Huruf Hiragana," 2014. [Online]. Available: <https://tensai-indonesia.com/sejarah-huruf-hiragana/>. [Accessed: 11-Sep-2018].
- [7] Farhan, "Penerapan Algoritma Distance Untuk Fitur Autocomplete Pada Aplikasi Katalog Perpustakaan Di Universitas Almuslim," *J. TIKA*, vol. 1, no. 3, pp. 1–6, 2016.
- [8] C. O'Malley *et al.*, "Guidelines for learning / teaching / tutoring in a mobile environment To cite this version :," *Public Deliv. from MOBILEarn Proj.*, 2005.
- [9] ubaya.ac.id, "Android : Sistem Operasi Pada Smartphone," 2010. [Online]. Available: http://www.ubaya.ac.id/2014/content/articles_detail/7/Android--Sistem-Operasi-pada-Smartphone.html. [Accessed: 11-Sep-2018].
- [10] S. Arif, "Rancang bangun website program studi DIV Komputer," 2012.
- [11] binus.ac.id, "Penggunaan Format Pertukaran Data Javascript Object Notation (JSON)." [Online]. Available: <http://socs.binus.ac.id/2017/09/27/json/>. [Accessed: 11-Sep-2018].
- [12] softwaretestingfundamentals.com, "Black Box Testing - Software Testing Fundamentals." [Online]. Available: <http://softwaretestingfundamentals.com/black-box-testing/>. [Accessed: 11-Sep-2018].

- [13] S. Shahab, "Usability Requirements for GIS Application Comparative Study of Google Maps on PC and Smartphone," *Technology*, no. January, 2011.