e-ISSN: 2549-7952
p-ISSN: 2580-3336

Implementasi Algoritma Jaro Winkler Distance Untuk Pendeteksi Kesamaan Kata Dalam Pengembangan Aplikasi English Conversation

Eureka Jeremy Aritomatika[1], Ardi Sanjaya[2], Danang Wahyu Widodo[3]
Teknik Informatika, Fakultas Teknik, Universitas Nusantara PGRI Kediri
E-mail: [1]*eurekajeremy@gmail.com* , [2]*dersky@gmail.com*, [3]*danayudo@yahoo.com*

Abstrak *– Pendidikan Bahasa Inggris, dalam hal percakapan, memang bukanlah hal yang mudah bagi para siswa Sekolah Dasar. Meskipun merupakan tingkat dasar, namun terkadang mereka merasa kesulitan pada saat melafalkan kata dalam bahasa inggris. Maka di butuhkan media pendidikan yang menjadi alat bantu mereka dalam berlatih percakapan dalam Bahasa Inggris. Masalah ini dapat diatasi dengan cara mengembangkan sebuah aplikasi sistem belajar interaktif dalam bentuk website menjadi jembatan bagi siswa Sekolah Dasar dalam belajar dan berlatih pelafalan percakapan Bahasa Inggris. Dengan sistem ini, siswa tidak hanya mengenal kosa kata dalam Bahasa Inggris. Namun juga bagaimana melafalkan secara tepat dan mengetahui nilai kemiripan kata yang dilafalkan. Sistem ini dikembangkan dengan bahasa pemrograman php, javascript, ajax dengan memanfaatkan Google Speech API serta menggunakan database MySQL. Pemilihan Algoritma yang digunakan dalam pengembangan media pembelajaran tersebut adalah Jaro Winkler. Berdasarkan kecepatan pemrosesan dan keefektifan dalam membandingkan string pendek sehingga sangat cocok digunakan. Fungsi algoritma Jaro Winkler Distance disini adalah membandingkan antara hasil dari deteksi ucapan suara dan kata kunci, Dimana hal ini berguna untuk mencari kemiripan untuk mengetahui nilai perbandingannya. Dalam aplikasi ini memiliki bebrapa kode soal yang di dalamnya terdapat kata kunci yang harus di ucapkan oleh siswa agar dapat mengetahui jawaban dengan memanfaatkan Google Speech API . Hasil pengujian aplikasi English Conversation dengan Algoritma Jaro Winkler Distance memperoleh hasil nilai perbandingan kemiripan kata dengan hasil nilai maksimal sebesar 100% serta nilai minimal sebesar 68.1%.*

Kata Kunci *— Speech to Text, Conversation, Jaro Winkler, Pembelajaran, Bahasa Inggris*

1. PENDAHULUAN

Pendidikan tak ubahnya membantu siswa dalam rangkaian proses pembelajarannya, salah satunya tidak menghabiskan waktu untuk melacak semua informasi tentang pendidikan. Saat ini, teknologi memberikan sejumlah alat untuk membantu pendidik dalam kemudahan pengelolaan informasi sehingga mereka dapat berkonsentrasi pada peningkatan prestasi siswa. Teknologi akan memiliki dampak terbesar pada pembelajar ketika diintegrasikan ke dalam kurikulum untuk mencapai tujuan yang jelas, tujuan pendidikan yang terukur [1]. Maka jelas terlihat bahwa teknologi merupakan alat dan sumber daya yang harus menjadi bagian integral dari proses belajar mengajar jika mereka memiliki dampak pada prestasi siswa.

Bukan hal yang istimewa lagi bahwa sekarang tidak sedikit pula beredar aplikasi edukasi berbentuk permainan yang menjadi sarana pembelajaran bagi siswa. Banyak pengembang beramai-ramai menciptakan media pembelajaran berbentuk permainan edukasi.

Pelajaran Bahasa Inggris memang bukan pelajaran wajib di Sekolah Dasar. Namun, hampir seluruh Sekolah Dasar memberikan mata pelajaran tersebut. Hal ini dikarenakan Bahasa Inggris dianggap penting sebagai bahasa pendamping Bahasa Indonesia. Dalam prakteknya, pembelajaran Bahasa Inggris memang tidak begitu menemui banyak kendala. Satu kendala bagi siswa Sekolah Dasar adalah melakukan percakapan menggunakan Bahasa Inggris.

Berangkat dari maraknya beragam permainan edukasi berbasis perangkat lunak, peneliti mencoba untuk mengembangkan sistem dalam pentuk aplikasi. Banyak sekali alat bantu yang disebut dengan *speech recognition* telah diciptakan untuk memudahkan pembuatan sistem ini.

Metode yang digunakan dalam yang di gunakan pengembangan media pembelajaran ini adalah *Jaro Winkler*. Dimana algoritma *Jaro Winkler* merupakan algoritma yang tepat dalam mengukur kesamaan dari dua string atau dokumen

Penelitian terdahulu tentang analisis perbandingan algoritma *Levenshtein distance* dan *Jaro Winkler* untuk aplikasi deteksi plagiarisme dokumen teks [2], dimana Perbandingan kinerja algoritma *Jaro Winkler Distance* dan algoritma

e-ISSN: 2549-7952
p-ISSN: 2580-3336

Levensthein Distance diukur dengan menggunakan 2 data uji yaitu data uji untuk mengukur nilai *similarity* dan data uji untuk mengukur waktu proses. Selanjutnya data uji tersebut diujikan menggunakan aplikasi deteksi plagiarisme untuk mengetahui berapa nilai *similarity* dan waktu proses yang dihasilkan dari pengujian tersebut. Selanjutnya dari hasil pengujian tersebut dianalisis perbandingannya, dan analisis perbandingan yang didapat dari pengujian nilai *similarity*, yaitu algoritma *Jaro Winkler* memiliki nilai *similarity* yang lebih akurat yaitu sebesar 80.92% dibanding dengan algoritma *Levensthein Distance* yaitu hanya sebesar 49.43%. Sedangkan untuk pengujian waktu proses, algoritma *Jaro Winkler* memiliki rata-rata waktu proses yang lebih cepat yaitu 0.054 detik, dibandingkan algoritma *Levensthein Distance* yaitu 0.138 detik.

Selanjutnya, yaitu penelitian tentang implementasi algoritma *Jaro Winkler Distance* untuk sistem pendeteksi plagiarisme pada dokumen skripsi [3], dapat disimpulkan bahwa, hasil dari Sistem pendeteksi plagiarisme dengan menggunakan algoritma *Jaro Winkler Distance* dapat digunakan untuk mendeteksi plagiarisme dokumen skripsi dengan cara melakukan perbandingan antara dokumen asli dan dokumen uji yang diinputkan untuk mengetahui tingkat kemiripan (*similarity*) dari dokumen skripsi yang diuji.

Kemudian, penelitian tentang perbaikan kata pada sistem chatbot dengan metode *Jaro Winkler* [4], penelitian ini menghasilkan. Perbaikan kata dengan menggunakan metode Jaro-Winkler dapat bekerja dengan hasil yang cukup baik dengan memperoleh rata-rata nilai kemiripan kata sebesar 95,21% dan tingkat keakuratan dalam pemberian saran kata sebesar 76%. Penambahan fitur perbaikan kata pada sistem *chatbot* juga telah dibuktikan dapat meningkatkan nilai akurasi sistem *chatbot* dalam memberikan respons dengan akurasi 96% dibandingkan sistem *chatbot* tanpa menggunakan fitur perbaikan kata dengan akurasi 36%. Nilai peningkatan akurasi sistem *chatbot* dengan penambahan fitur perbaikan kata mencapai 60%.

2. METODE PENELITIAN

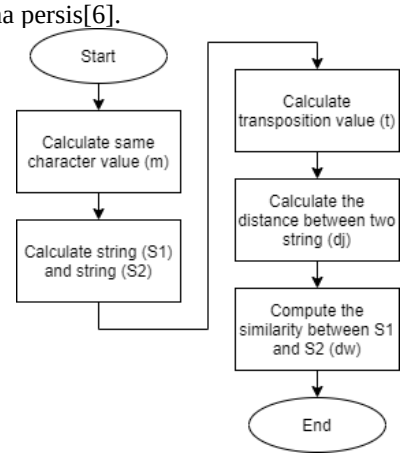
Proses pengembangan sistem yang digunakan adalah metode *Waterfall*. Metode *Waterfall* adalah salah satu jenis model pengembangan aplikasi dan termasuk ke dalam *classic life cycle* (siklus hidup klasik), yang mana menekankan pada fase yang berurutan dan sistematis. Untuk model pengembangannya, dapat dianalogikan seperti air terjun, dimana setiap tahap dikerjakan secara berurutan mulai dari atas hingga ke bawah.

2.1 Speech Recognition

Sistem Pengenalan Ucapan (*Speech Recognition Sistem*) adalah sistem yang berfungsi untuk mengubah bahasa lisan menjadi bahasa tulisan. Masukan sistem adalah ucapan manusia, selanjutnya sistem akan mengidentifikasikan kata atau kalimat yang diucapkan dan menghasilkan teks yang sesuai dengan apa yang diucapkan. Sinyal ucapan pertama kali akan dilewatkan pada bagian penganalisis ucapan untuk mendapatkan besaran-besaran atau ciri-ciri yang mudah diolah pada tahap berikutnya[5].

2.2 Jaro Winkler

Jaro-Winkler merupakan varian dari metrik Jaro Distance biasanya digunakan pada bidang keterkaitan rekaman (duplikat) dirancang dan paling sesuai untuk string pendek. Pada JaroWinkler untuk dua string semakin tinggi jarak, semakin mirip data yang diperoleh dengan skor 0 sama dengan tidak ada persamaan dan 1 sama persis[6].



Gambar 1 *Algoritma Jaro Winkler*

Algoritma *Jaro Winkler distance* memiliki kompleksitas waktu *quadratic runtime complexity* yang sangat efektif pada string pendek dan dapat bekerja lebih cepat dari algoritma *edit distance* [6]. Dasar dari algoritma ini memiliki tiga bagian:

a. Menghitung panjang string.
b. Menemukan jumlah karakter yang sama di dalam dua string.
c. Menemukan jumlah transposisi.

Pada algoritma Jaro-Winkler digunakan rumus untuk menghitung jarak (dj) antara dua

e-ISSN: 2549-7952
p-ISSN: 2580-3336

string yaitu s1 dan s2 dapat dilihat p pada persamaan (1) :

$$d_j = \frac{1}{3} \times \left(\frac{m}{s1} + \frac{m}{s2} + \frac{m-t}{s}\right) \quad (1)$$

Dimana:
m = jumlah karakter yang sama persisi
S_1 = panjang string 1
S_2 = panjang string 2
t = jumlah transposisi (karakter yang sama pada string)

Jarak teoritis dua buah karakter yang disamakan dapat dibenarkan jika tidak melebihi:

$$\left(\frac{max(|S1|,|S2|)}{2}\right) - 1 \quad (2)$$

Tetapi, jika mengacu pada nilai yang dihasilkan dari algoritma ini, maka nilai jarak maksimalnya adalah 1 yang menandakan adanya kesamaan string yang dibandingkan mencapai seratus persen atau sama persis.

Dimana m adalah jumlah karakter yang sama persis, |s1| adalah panjang string 1, |s2| adalah panjang String 2, T adalah jumlah transposisi, dan dj adalah Nilai jarak antara dua buah string yang dibandingkan [6].

Jaro-Winkler *distance* menggunakan *prefix scale (p)* yang memberikan tingkat penilaian yang lebih, dan *prefix length (l)* yang menyatakan panjang awalan yaitu panjang karakter yang sama dari *string* yang dibandingkan sampai ditemukannya ketidaksamaan. Bila *string* s1 dan s2 yang diperbandingkan, maka Jaro-Winkler *distance*-nya (dw) seperti pada persamaan.

$$dw = d_j + \left(lp\left(1 - d_j\right)\right) \quad (3)$$

Dimana dw adalah nilai Jaro-Winkler *Distance*, dj adalah Jaro *distance* untuk strings s1 dan s2, l adalah panjang prefiks umum di awal *string* nilai maksimalnya 4 karakter (panjang karakter yg sama sebelum ditemukan ketidaksamaan max 4), dan p adalah konstanta *scaling factor*.

Berikut ini merupakan contoh simulasi perhitungan dari algoritma *Jaro Winkler Distance*. Jika string S_1 MARTHA dan string S_2 MARHTA maka :
m = 6
S1 = 6
S2 = 6
Karakter yang tertukar hanyalah T dan H. Maka t = 1. Maka nilai Jaro Distancennya adalah sebagai berikut :

$$d_j = \frac{1}{3} * \left(\frac{6}{6} + \frac{6}{6} + \frac{6-1}{6}\right) = 0.944 \quad (4)$$

Lalu jika diperhatikan pada susunan string S_1 dan string S_2 dapat diketahui nilai l = 3, dengan nilai konstan p = 0.1. Maka nilai *Jaro Winkler Distance* adalah :

$$d_w = 0.944 + \left(3\ x\ 0.1(1 - 0.944\ )\right) = 0.961 \quad (5)$$

Berikut ini adalah contoh *Jaro Winkler* jika tidak ditemukan karakter yang sama tapi tertukar urutannya. Jika string S_1 BEAUTY dan string S_2 BIUTY maka :
m = 4
S1 = 6
S2 = 5
Pada contoh diatas tidak ditemukan karakter sama yang tertukar maka t = 0, Karakter seperti B,U,T,Y dianggap dalam urutan yang sama. Maka nilai *Jaro Winkler Distance* adalah

$$d_j = \frac{1}{3} * \left(\frac{4}{6} + \frac{6}{5} + \frac{4-1}{4}\right) = 0.822 \quad (6)$$

Kemudian bila diperhatikan susunan string S_1 dan string S_2 dapat diketahui nilai l = 1, dengan nilai konstan p = 0.1. Maka nilai *Jaro Winkler Distance* adalah :

$$d_w = 0.822 + \left(1\ x\ 0.1(1 - 0.822\ )\right) = 0.840 \quad (7)$$

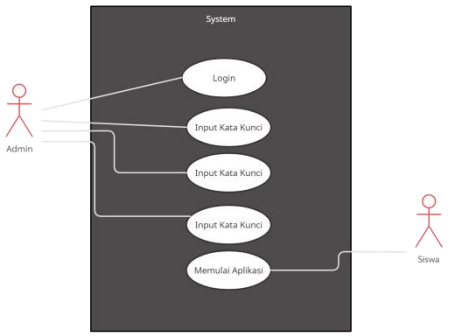
2.3 Perancangan Sistem

Perancangan sistem digunakan sebagai langkah untuk mengidentifikasi kebutuhan apa saja yang dibutuhkan untuk membangun suatu aplikasi. Pada hasil yang bersumber dari studi kepustakaan dan hasil analisa, maka pada perancangan sistem ini dibuatlah alur program dan desain *interface*.

2.4 Use Case

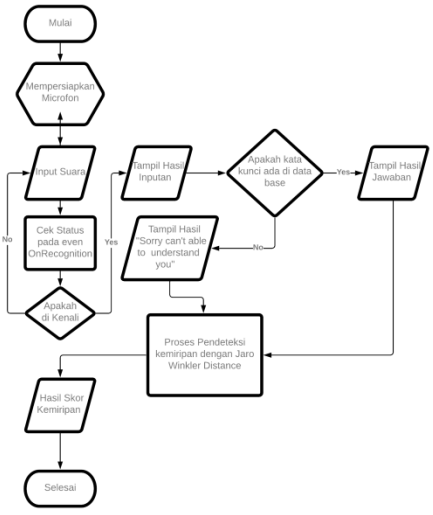
e-ISSN: 2549-7952
p-ISSN: 2580-3336

Dalam *use case* diatas dapat diketahui bahwa admin memiliki kendali sistem seutuhnya. Mulai dari *login*, tambah kata kunci, hapus kata kunci, edit kata kunci, sedangkan Siswa hanya dapat memulai aplikasi.



2.5 *Flowchart*

Untuk menggambarkan langkah-langkah yang dilalui dalam proses mencari kemiripan antara kata kunci dengan kata yang dilafalkan suara digunakan *flowchart* atau diagram alir. Untuk *flowchart* sistemnya dapat dilihat pada Gambar 2.

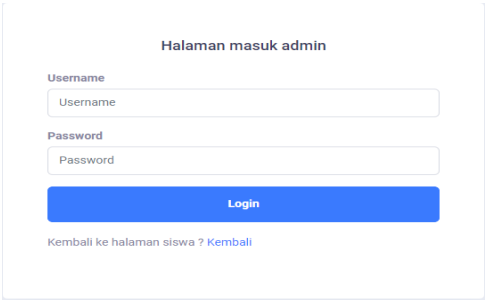


Gambar 2 Flowchart Sistem

3. HASIL DAN PEMBAHASAN

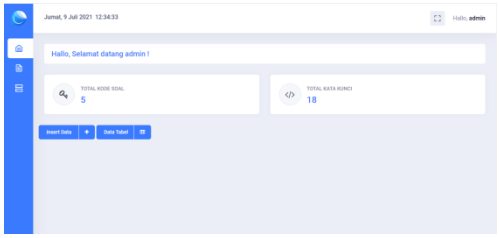
3.1 Halaman *Login*

ada tampilan login ini digunakan oleh operator atau admin untuk masuk ke dalam aplikasi dengan tujuan untuk mengolah data kata kunci dengan memasukkan *username* dan *password* terlebih dahulu. Berikut tampilan dari halaman *login*.



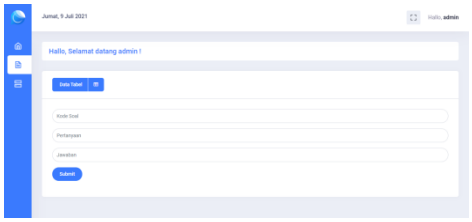
3.2 Halaman Dashboard Admin

Halaman dashboard ini adalah halaman pertama setelah admin atau operator berhasil *login*. Dalam halaman dashboard ini terdapat sebuah tampilan jumlah dari kata kunci dan jumlah kode soal.



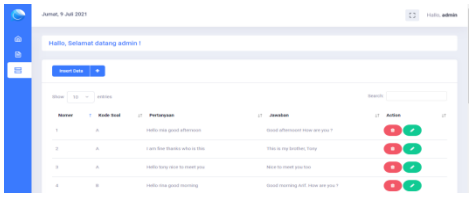
3.3 Halaman Masukan Kata Kunci

Pada halaman ini digunakan oleh admin untuk menambah data kata kunci serta jawaban yang berisi kata kunci dan kode soal serta jawaban yang berguna untuk memulai awal sebuah sistem.



3.4 Halaman Data Tabel

Dalam halaman data tabel ini terdapat tabel dari keseluruhan kata kunci, kode soal, dan jawaban.



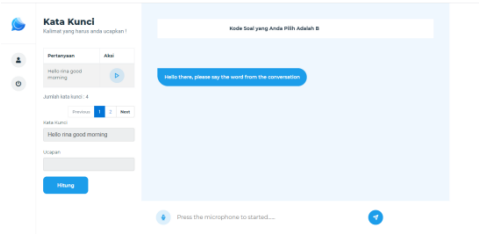
3.5 Halaman Pilih Kode Soal

Pada halaman ini pengguna memilih kode soal yang ingin diujikan. Terdapat lima pilihan kode soale, yaitu A, B, C, D, E.

e-ISSN: 2549-7952
p-ISSN: 2580-3336

3.6 Halaman Pengujian

Halaman Pengujian adalah halaman yang ditampilkan ketika sudah memilih kode soal. Pada halaman ini terdapat kata kunci yang harus disebutkan dan *form input* untuk memasukkan ucapan yang nantinya jika kata yang di ucapkan benar akan mncul jawaban dari kata kunci tersebut.



3.7 Data Kata Kunci

Dalam aplikasi ini terdapat data kata kunci yang dibagi menjadi tiga pilihan level, yakni mudah, sedang, dan sulit dan semua kata kunci tersebut digunakan sebagai acuan untuk melakukan proses mencari kemiripan menggunakan algoritma *Jaro Winkler Distance* yang ditunjukkan pada tabel berikut.

Tabel 1 Tabel kata kunci

No	Kode Soal	Pertanyaan	Jawaban
1	A	Hello mia good afternoon	Good afternoon! How are you ?
2	A	I am fine thanks who is this	This is my brother, Tony
3	A	Hello tony nice to meet you	Nice to meet you too
4	B	Hello rina good morning	Good morning Arif. How are you ?
5	B	I am fine thanks	Where are you going ?
6	B	I am going to school	Oh see you Arif
7	B	See you	.....
8	C	Hi how are you	I am fine. Thank you. You ?
9	C	I am very well thank you	What are you doing ?
10	C	Iam making an appointment	Oh. Well, have a nice day!
11	D	Halo Sammy	Hi lusi
12	D	Can you come to my house today	Sure
13	D	Okay see you	See you
14	E	Good morning i am candra	Good morning, I am Maya.
15	E	How do you do	How do you do ?
16	E	Nice to meet you	Nice to meet you too
17	E	Where are you from	I am from Lampung. And you ?
18	E	I am from Bekasi	.....

3.8 Uji Hasil Kemiripan Kode Soal A

Dalam mencari hasil kemiripan maka dibutuhkanlah sebuah kata kunci yang sebelumnya terdapat pada tabel 1. Untuk mencari hasil kemiripan digunakanlah algortima *Jaro Winkler Distance* untuk membandingkan antara kata kunci dengan *kata yang dilafalkan* yang memanfaatkan *Speech Recognition*. Dalam uji coba kali ini dilakukan sebanyak dua kali agar mendapat hasil yang lebih akurat dan diperolehlah hasil yang dapat dilihat pada tabel berikut ini.

Tabel 2 Uji Coba Kode Soal A

Kode Soal	Kata Kunci	Uji Coba 1	Hasil Uji Coba 1	Uji Coba 2	Hasil Uji Coba 2
A	Hello mia good afternoon	Hello mia good afternoon	100 %	Hello araya good afternoon	88.98 %
A	I am fine thanks who is this	Iam fine thank who is sit	81.79 %	Iam fine thanks who is this	83 %
A	Hello tony nice to meet you	Halo rony nice to meet you	79.91 %	Hello tony nice to meet you	100 %

3.9 Uji Hasil Kemiripan Kode Soal B

Tabel 3 Uji Coba kode Soal B

Kode Soal	Kata Kunci	Uji Coba 1	Hasil Uji Coba 1	Uji Coba 2	Hasil Uji Coba 2
B	Hello rina good morning	Hello dina good morning	91.44 %	Hello rina good morning	100 %
B	I am fine thanks	I am fine thanks	100 %	Ayam fine thanks	78.13 %
B	I am going to school	Ayam going to school	80 %	I am going to school	100 %
B	See you	Seyu	68.1%	See you	100%

e-ISSN: 2549-7952
p-ISSN: 2580-3336

3.10 Uji Hasil Kemiripan Kode Soal C

Tabel 4 Uji Coba Kode Soal C

Kode Soal	Kata Kunci	Uji Coba 1	Hasil Uji Coba 1	Uji Coba 2	Hasil Uji Coba 2
C	Hi how are you	Hi how are you	100 %	Hi where are you	86.25 %
C	I am very well thank you	I am very well thanks	94.76 %	I am very well	91.25 %
C	Hello tony nice to meet you	Halo rony nice to meet you	79.91 %	Hello tony nice to meet you	100 %

3.11 Uji Hasil Kemiripan Kode Soal D

Tabel 5 Uji Coba Kode Soal D

Kode Soal	Kata Kunci	Uji Coba 1	Hasil Uji Coba 1	Uji Coba 2	Hasil Uji Coba 2
D	Halo sammy	Halo sami	91.56 %	Halo semi	87.06 %
D	Can you come to my house today	Can you come too my house today	100 %	Can you come to my home today	96.3 %
D	Okay See You	Oke See you	96.57 %	Okay See You	100 %

3.12 Uji Hasil Kemiripan Kode Soal E

Tabel 6 Uji Coba Kode Soal E

Kode Soal	Kata Kunci	Uji Coba 1	Hasil Uji Coba 1	Uji Coba 2	Hasil Uji Coba 2
E	Good morning i am candra	Good morning i am sandra	98.33 %	Good morning im mandra	94.42 %
E	How do you do	How do you do	100 %	How are you	77.76 %
E	Nice to meet you	Nice to meet you	100 %	Nais to meet you	79.5 %
E	Where are you from	Where you from	87.32 %	Where you come from	90.9 %
E	I am from Bekasi	I am from Bekasi	100 %	Ayam from Bekasi	94.72 %

4. SIMPULAN

Berdasarkan dari hasil analisa, perancangan, hingga tahap pembuatan aplikasi dan pengujian aplikasi. Maka diperoleh kesimpulan dalam mencari kemiripan kata dengan metode *Jaro Winkler Distance* dapat digunakan dalam mencari kemiripan kata dalam bahasa Inggris dengan baik serta dapat di gunakan dalam pencarian data dengan baik. Sehingga aplikasi pembelajaran *English Conversation* dapat memudahkan para pengajar dalam melatih pengucapan dari para siswa dalam kehidupan sehari – hari. Kemiripan antara kata kunci dengan kata yang dilafalkan menggunakan metode *Jaro Winkler Distance* ini memperoleh nilai maksimal dengan nilai 100 % dan nilai minimal 68.1 %.

5. SARAN

Berdasarkan dari hasil analisa hingga tahap pengujian aplikasi ini, terdapat beberapa saran yang perlu dilakukan agar aplikasi ini dapat berjalan lebih baik kedepannya, antara lain :

1) Diharapkan aplikasi ini dapat dikembangkan menggunakan versi mobile sehingga memudahkan dalam pembelajaran *English Conversation* kapan saja.
2) Menambahkan algoritma lain untuk membandingkan keakuratan skor dari algoritma *Jaro Winkler Distance* dengan metode lainnya dalam mendeteksi kemiripan kata.
3) Menambahkan perbaikan kata pada sistem tersebut agar mendapatkan hasil yang lebih maksimal

DAFTAR PUSTAKA

[1] Hawkins, J., Panush, E. and Spielvogel, R. (1996). Education Development Center, Center for Children and Technology. National study tour of district technology integration (summary report) CCT Reports Issue No. 14 December 1996. http://cct.edc.org/publications/national-study-tour-district-technology-integration-summary-report.

[2] Tannga, M. J., & Rahman, S. (2017). Analisis Perbandingan Algoritma Levenshtein Distance dan Jaro Winkler Untuk Aplikasi Deteksi Plagiarisme Dokumen Teks. *JTRISTE*, *4*(2), 44-54.

[3] Novantara, P. (2017). Implementasi Algoritma Jaro-Winkler Distance Untuk Sistem Pendeteksi Plagiarisme Pada Dokumen Skripsi. *Buffer Informatika*, *3*(2).

[4] Pinajeng, P. T. K. I., Sukarsa, M. I., dan Putra, S. M. I. (2020) Perbaikan Kata Pada Sistem Chatbot dengan Metode Jaro Winkler. *JITTER*, 4(2).

e-ISSN: 2549-7952
p-ISSN: 2580-3336

[5] Setiawan, A. B., & Rabi, A. IMPLEMENTASISISTEM SPEECH RECOGNITION MENGGUNAKAN WEB SERVER SECARA REAL TIME BERBASIS ANDROID.

[6] Sanjaya, A. (2020). OPTIMASI PENCARIAN DATA MENGGUNAKAN TEXT FILTERING DAN ALGORITMA JARO WINKLER. *Network Engineering Research Operation*, *5*(1), 24-29.