

Penerapan Metode Jaro Winkler Untuk Mencari Kemiripan Kata Pada Aplikasi English Pronunciation Test

Arthur Glenn Tambahani¹, Ardi Sanjaya², Danang Wahyu Widodo³

^{1,2,3}Teknik Informatika, Fakultas Teknik, Universitas Nusantara PGRI Kediri

E-mail: ¹arthur.tambahani@gmail.com, ²dersky@gmail.com, ³danayudo@yahoo.com

Abstrak – Bahasa Inggris merupakan bahasa internasional yang hampir digunakan di seluruh dunia. Di Indonesia sendiri, Bahasa Inggris merupakan mata pelajaran pokok yang sudah diperkenalkan dari tingkat pendidikan TK hingga perguruan tinggi. Dalam pembelajaran Bahasa Inggris banyak sekali unsur- unsur yang bisa diajarkan, antara lain vocabulary (kosa kata), pronunciation (pengucapan) dan grammar (tata bahasa). Permasalahan yang sering terjadi adalah banyak siswa yang belum bisa mengucapkan kata dalam Bahasa Inggris dengan benar. Untuk meningkatkan skill siswa dalam belajar pronunciation (pengucapan) maka diperlukan suatu media pembelajaran interaktif untuk melatih skill pronunciation (pengucapan) yang telah dipelajari siswa di sekolah. Metode yang digunakan untuk membangun media pembelajaran tersebut adalah Jaro Winkler. Didalam aplikasi terdapat 3 buah level, antara lain : mudah, sedang, dan sulit. Dalam setiap level terdapat beberapa vocabulary (kosa kata) yang sudah disesuaikan dengan tingkatan level. Kemudian siswa akan mengucapkan vocabulary tersebut dengan memanfaatkan Google Speech API untuk mendeteksi suara. Algoritma Jaro Winkler bekerja dengan cara membandingkan antara hasil deteksi suara dan data acuan. Kemudian dicari kemiripannya untuk memunculkan skor perbandingannya. Dengan adanya aplikasi pembelajaran pronunciation (pengucapan) ini dapat membantu para siswa lebih mahir dalam pronunciation (pengucapan) serta mempermudah proses pembelajaran.

Kata Kunci — Bahasa Inggris, Jaro Winkler, Pronunciation, Speech to Text

1. PENDAHULUAN

Teknologi di era saat ini memiliki manfaat sebagai sumber informasi yang dapat menunjang berbagai macam aktivitas. Dengan adanya teknologi, seseorang dapat menemukan berbagai informasi yang dibutuhkan hanya dengan hitungan menit tanpa harus kesusahan mencari informasi di media cetak maupun di media lainnya. Perkembangan teknologi yang pesat hingga saat ini membuat pengaruh yang sangat besar pada berbagai bidang, salah satunya adalah bidang pendidikan. Dalam dunia pendidikan, teknologi memiliki dampak terbesar pada pembelajaran ketika diintegrasikan ke dalam kurikulum untuk mencapai tujuan yang jelas [1].

Kemajuan pengembangan teknologi di bidang Pendidikan merupakan suatu solusi bagi para tenaga pengajar dan peserta didik. Dengan adanya teknologi kinerja tenaga pengajar justru semakin dimudahkan dan membuat penyampaian materi ke peserta didik menjadi lebih mudah diterima. Begitu pula untuk peserta didik di era sekarang teknologi bukan merupakan hal yang asing. Justru dengan adanya teknologi peserta didik menjadi lebih tertarik dan mudah memahami menangkap materi yang disampaikan.

Bahasa Inggris merupakan bahasa internasional yang hampir digunakan di seluruh dunia. Lebih dari 2,4 milyar orang diseluruh dunia menggunakan

Bahasa Inggris sebagai bahasa utama mereka. Di Indonesia sendiri Bahasa Inggris sudah menjadi mata pelajaran utama dalam dunia Pendidikan yang dimulai dari sejak TK hingga SMA bahkan sampai perguruan tinggi. Namun pada kenyataannya masih banyak diluar sana peserta didik yang kesulitan membaca maupun mengucapkan kata – kata dalam Bahasa Inggris dengan benar.

Dalam penelitian yang dilakukan oleh EF, indeks kemampuan berbahasa Inggris (English Proficiency Index) EPI tahun 2019 masyarakat Indonesia masih menduduki peringkat 61 di dunia [2]. Sedangkan untuk tahun 2020 justru terjadi penurunan yang cukup signifikan yang membuat EPI tahun ini membuat Indonesia menduduki peringkat 74 di dunia [3]. Dapat disimpulkan bahwa dari tahun ke tahun terjadi penurunan kemampuan berbahasa inggris.

Berawal dari masalah tersebut, peneliti mencoba untuk mengembangkan sistem yang berguna untuk mendukung kemampuan berbahasa inggris khususnya pada bidang pendidikan SD. Banyak sekali referensi dengan kata kunci *Jaro Winkler* telah dibuat untuk memudahkan pengembangan sistem ini.

Pada tahun 2018, penelitian yang dilakukan oleh Youllia Indrawaty, Andriana Zulkarnain, Reza Rianto dengan judul “Pengembangan Pembelajaran Pengenalan Kata Dalam Bahasa Indonesia

Menggunakan Multimedia Interaktif dan Speech Recognition”[4]. Dalam penelitian ini memanfaatkan *speech recognition* yang menggabungkan antara dua algoritma untuk mencari hasilnya, yaitu algoritma *Jaro Winkler Distance* dan algoritma *levenshtein distance*. Aplikasi ini menghasilkan tampilan yang menarik sehingga cocok digunakan untuk anak PAUD.

Kemudian penelitian lain pada tahun 2017 yang dilakukan oleh Panji Novantara, Opin Pasruli dengan judul “Implementasi Algoritma Jaro-Winkler Distance Untuk Sistem Pendeteksi Plagiarisme Pada Dokumen Skripsi”[5]. Pada penelitian ini menggunakan metode *Jaro Winkler*. Metode *Jaro Winkler* merupakan sebuah algoritma untuk mencari kemiripan kata. Implementasi algoritma *Jaro Winkler Distance* pada penelitian ini adalah untuk pendeteksi plagiarisme pada dokumen skripsi dengan cara membandingkan antara dokumen asli dengan dokumen uji untuk mencari tingkat kemiripan (*similarity*) dari dokumen yang diujikan.

2. METODE PENELITIAN

Pada bagian ini akan menjelaskan tentang metode yang akan digunakan dalam penelitian ini. Metode yang digunakan adalah metode *waterfall*. Metode *waterfall* sendiri adalah metode yang menggunakan pendekatan sistematis danurut dimulai dari level kebutuhan sistem lalu menuju ke tahapan analisis, desain, *coding*, *testing/verification* dan *maintenance* [6].

2.1 Teknologi Pemrosesan Bahasa

Pada prinsipnya bahasa alami adalah suatu bentuk representasi dari suatu pesan yang ingin dikomunikasikan antara manusia [7]. Bentuk utama representasinya adalah berupa suara atau ucapan.

Menurut Pramono (dalam penelitian Nurpadila Muslimin, 2017) bahasa dapat dibedakan menjadi dua, yaitu bahasa alami dan bahasa buatan. Bahasa alami adalah bahasa yang biasa digunakan untuk berkomunikasi antar manusia, misalnya bahasa Indonesia, Inggris, Jepang dan sebagainya. Bahasa buatan adalah bahasa yang dibuat secara khusus untuk memenuhi kebutuhan tertentu, misalnya bahasa pemrograman komputer. [8]

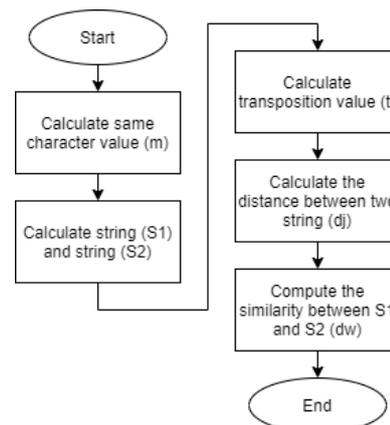
2.2 Speech Recognition

Speech recognition merupakan sebuah pengembangan sistem yang memungkinkan untuk sebuah komputer dapat menerima suatu inputan berupa kata – kata lalu dicocokkan dengan pola tertentu.

Menurut Jurafsky, D., & Martin, J. H (dalam penelitian Ayu Dhiya Mufidah, 2017) *Speech Recognition* adalah proses otomatis mengenali kata - kata yang diucapkan kemudian mengubahnya ke dalam bentuk teks. Dengan kata lain *Speech Recognition* merupakan suatu proses yang memungkinkan untuk mengenali apa yang diucapkan oleh seseorang lalu mengubahnya ke bentuk tulisan. Input data yang diterima berupa suara atau ucapan manusia, kemudian sistem akan mengidentifikasi kata atau kalimat yang diucapkan dan menghasilkan keluaran atau *output* berupa teks yang sesuai dengan apa yang diucapkan. [9]

2.3 Algoritma Jaro Winkler Distance

Jaro Winkler distance merupakan varian dari sebuah metrik *Jaro Distance* yang biasanya digunakan dalam mencari sebuah keterkaitan dokumen atau duplikasi dengan cara diukur kesamaan antara kedua stringnya. *Jaro Winkler* memiliki kemampuan terbaik dalam mengenali *string* pendek. Jika diperoleh skor 0 menandakan tidak ada kesamaan dengan data pembanding, dan jika skor 1 berarti sama persis.



Gambar 1. Algoritma *Jaro Winkler*

Dalam algoritma *Jaro Winkler* terdapat ke bagian yang menjadi dasar, yaitu :

1. Menghitung Panjang *string*.
2. Menentukan jumlah karakter yang sama di dalam dua *string*.
3. Menentukan jumlah transposisi.

Pada algoritma *Jaro Winkler* digunakan rumus untuk menghitung jarak yang disimbolkan dengan (d_j) antara dua string S_1 dan S_2 [10]. adalah

$$d_j = \frac{1}{3} * \left(\frac{m}{S_1} + \frac{m}{S_2} + \frac{m-t}{m} \right) \quad (1)$$

Dimana:

m = jumlah karakter yang sama persisi

S_1 = panjang string 1

$$S_2 = \text{panjang string 2} \quad (5)$$

$$t = \text{jumlah transposisi (karakter yang sama pada string)}$$

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

$$\left(\frac{\max(|S_1|, |S_2|)}{2} \right) - 1 \quad (2)$$

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.

Algoritma *Jaro Winkler distance* menggunakan skala awalan atau *prefix scale (p)* menurut Winkler, nilai standar untuk konstanta ini adalah $p = 0.1$. Dan *prefix length (l)* yang menyatakan panjang karakter yang sama dari string dan dibandingkan sampai ditemukan ketidaksamaan, maka perhitungan *Jaro Winkler Distance (d_w)* adalah :

$$d_w = d_j + (l \times p(1 - d_j)) \quad (3)$$

Dimana :

d_j = *Jaro distance* untuk string S1 dan S2

l = Panjang *prefix* (panjang karakter yang sama sebelum ditemukan ketidaksamaan nilai maksimum 4 karakter).

p = konstanta *scaling factor*. Nilai standar untuk konstanta ini menurut Winkler adalah 0.1

Berikut ini merupakan contoh simulasi perhitungan dari algoritma *Jaro Winkler Distance*. Jika string S1 MARTHA dan string S2 MARHTA maka :

$$m = 6$$

$$S1 = 6$$

$$S2 = 6$$

Karakter yang bertukar 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 S1 dan string S2 dapat diketahui nilai $l = 3$, dengan nilai konstan $p = 0.1$. Maka nilai *Jaro Winkler Distance* adalah :

$$d_w = 0.944 + (3 \times 0.1(1 - 0.944)) = 0.961$$

Berikut ini adalah contoh *Jaro Winkler* jika tidak ditemukan karakter yang sama tapi bertukar urutannya. Jika string S1 BEAUTY dan string S2 BIUTY maka :

$$m = 4$$

$$S1 = 6$$

$$S2 = 5$$

Pada contoh diatas tidak ditemukan karakter sama yang bertukar 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 S1 dan string S2 dapat diketahui nilai $l = 1$, dengan nilai konstan $p = 0.1$. Maka nilai *Jaro Winkler Distance* adalah :

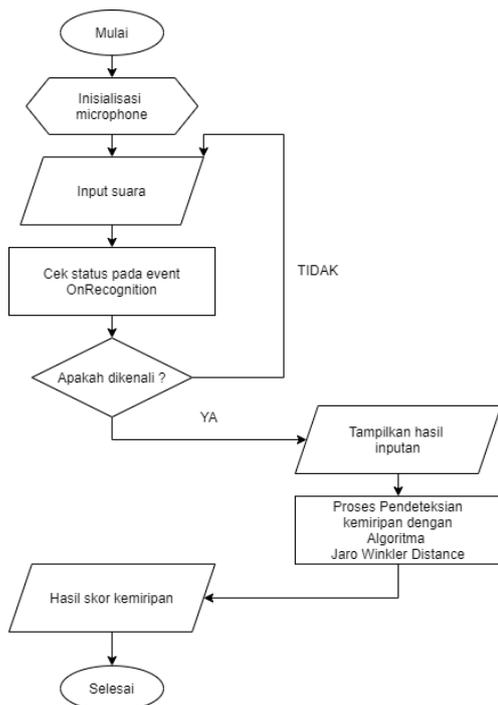
$$d_w = 0.822 + (1 \times 0.1(1 - 0.822)) = 0.840 \quad (7)$$

2.4 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*.

2.4.1 Flowchart

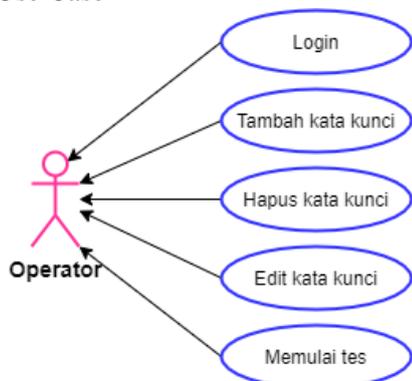
Untuk menggambarkan langkah-langkah yang dilalui dalam proses mencari kemiripan antara kata kunci dengan inputan suara digunakan *flowchart* atau diagram alir. Untuk *flowchart* sistemnya dapat dilihat pada Gambar 2.



Gambar 2. Flowchart sistem

Proses diawali dengan keadaan inisialisasi microphone, kemudian pengguna mengucapkan kata – kata sesuai teks. Kemudian input berupa suara tersebut akan di proses pada even *OnRecognition*. Jika suara yang diproses tadi dikenali maka akan menampilkan *output* hasil suara berupa text. Jika suara tidak dikenali maka, akan Kembali ke proses input suara. Pada tahap selanjutnya akan dihitung menggunakan algoritma *Jaro Winkler* untuk mendeteksi kemiripan antara suara inputan dengan kata kunci. Jika proses perhitungan tersebut sudah selesai maka akan memunculkan sebuah *output* berupa skor kemiripan antara kata kunci dengan inputan suara.

2.4.2 Use Case



Gambar 3. Use Case

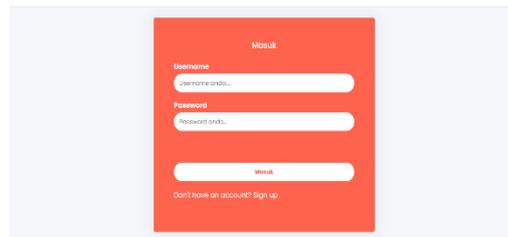
Dalam use case diatas dapat diketahui bahwa operator memiliki kendali sistem seutuhnya. Mulai dari *login*, tambah kata kunci, hapus kata

kunci, edit kata kunci, dan memulai tes English Pronunciation.

3. HASIL DAN PEMBAHASAN

3.1 Halaman *Login*

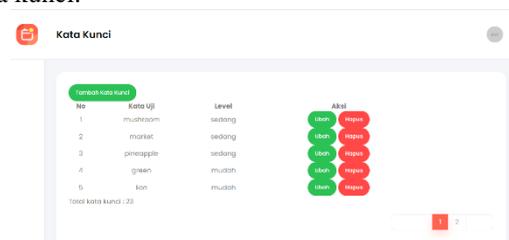
Pada tampilan login ini digunakan oleh operator atau admin untuk masuk ke dalam aplikasi dengan tujuan untuk mengolah data kata kunci dengan memasukkan *username* dan *password* terlebih dahulu. Berikut tampilan dari halaman *login*.



Gambar 4. Halaman login

3.2 Dashboard Admin

Halaman dashboard ini adalah halaman pertama setelah admin atau operator berhasil *login*. Dalam halaman dashboard ini terdapat tabel dari keseluruhan kata kunci.



Gambar 5. Halaman dashboard admin

3.3 Halaman Tambah Kata Kunci

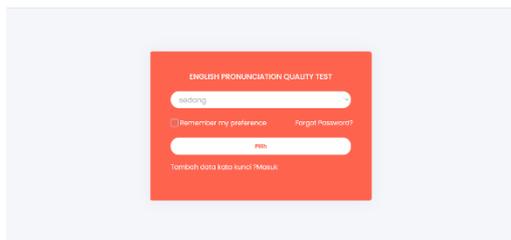
Pada halaman ini digunakan oleh operator atau admin untuk menambah data kata kunci yang berisi kata kunci dan level.



Gambar 6. Halaman tambah kata kunci

3.4 Halaman Level

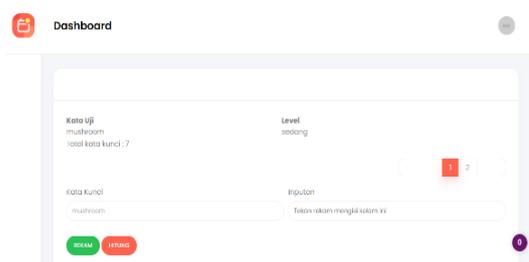
Pada halaman ini operator memilih level yang sesuai yang ingin diujikan. Terdapat tiga pilihan level, yaitu mudah, sedang, sulit.



Gambar 7. Halaman level

3.5 Halaman tes

Halaman utama adalah halaman yang ditampilkan ketika sudah memilih level. Pada halaman ini terdapat kata kunci yang harus disebutkan dan *form input* untuk memasukkan ucapan.



Gambar 8. Halaman tes

3.6 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 data kata kunci

No	Kata Kunci	Level
1	Green	Mudah
2	Lion	Mudah
3	Table	Mudah
4	Apple	Mudah
5	Banana	Mudah
6	Car	Mudah
7	Tiger	Mudah
8	Broom	Sedang
9	Market	Sedang
10	Seaweed	Sedang
11	Flour	Sedang
12	Money	Sedang
13	Island	Sedang
14	Bicycle	Sedang
15	Espresso	Sulit
16	Penguin	Sulit
17	Squirrel	Sulit
18	Weird	Sulit
19	Pharaoh	Sulit
20	Asterisk	Sulit
21	Temperature	Sulit
22	Dictionary	Sulit

3.7 Uji Coba Hasil Kemiripan Level Mudah

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

Tabel 2. Uji Coba Level Mudah

Kata Kunci	Level	Deteksi Inputan Suara Uji Coba 1	Hasil Jaro Winkler Uji Coba 1	Deteksi Inputan Suara Uji Coba 2	Hasil Jaro Winkler Uji Coba 2
Green	Mudah	Green	100 %	Green	100 %
Lion	Mudah	Lion	100 %	Leon	85 %
Table	Mudah	Table	100 %	Table	100 %
Apple	Mudah	Apple	100 %	Apple	100 %
Banana	Mudah	Banana	100 %	Banana	100 %
Car	Mudah	Car	100 %	Car	100 %
Tiger	Mudah	Teager	72,75 %	Tiger	100 %

3.8 Uji Coba Hasil Kemiripan Level Sedang

Tabel 3. Uji Coba Level Sedang

Kata Kunci	Level	Deteksi Inputan Suara Uji Coba 1	Hasil Jaro Winkler Uji Coba 1	Deteksi Inputan Suara Uji Coba 2	Hasil Jaro Winkler Uji Coba 2
Broom	Sedang	Broom	100 %	Broom	100 %
Market	Sedang	Market	100 %	Market	100 %
Seaweed	Sedang	She With	52.11 %	See With	64.57 %
Flour	Sedang	Flower	87.56 %	Floor	87.75 %
Money	Sedang	Money	100 %	Money	100 %
Island	Sedang	Islam	89.33 %	Eastland	65.28 %

Bicycl e	Sedan g	Bicycl le	100 %	Bicycl e	100 %
-------------	------------	--------------	-------	-------------	-------

3.9 Uji Coba Hasil Kemiripan Level Sulit

Tabel 4. Uji Coba Level Sulit

Kata Kunci	Level	Deteksi Inputan Suara Uji Coba 1	Hasil Jaro Winkler Uji Coba 1	Deteksi Inputan Suara Uji Coba 2	Hasil Jaro Winkler Uji Coba 2
Espresso	Sulit	Espresso	100 %	Espresso	100 %
Penguin	Sulit	Penguin	100 %	Penguin	100 %
Squirrel	Sulit	Karela	45.83%	Squirrel	100 %
Weird	Sulit	Reality	56.19%	Wait	67%
Pharaoh	Sulit	Parl	75.36%	Parl	75.36%
Asterisk	Sulit	Asterisk	100 %	Asterisk	100 %
Temperature	Sulit	Temperature	96.36%	Temperature	100 %
Dictionary	Sulit	Dictionary	100 %	The Ordinary	61.83%

4. SIMPULAN

Berdasarkan dari hasil analisa, perancangan, hingga tahap pembuatan aplikasi dan pengujian aplikasi. Maka diperoleh kesimpulan bahwa dalam mencari kemiripan kata dengan menggunakan metode *Jaro Winkler Distance* dapat digunakan dalam mencari kemiripan kata dalam bahasa Inggris dengan baik. Sehingga aplikasi English Pronunciation Test dapat memudahkan para pengajar dalam melatih pengucapan (*pronunciation*) dari peserta didiknya. Dalam mengenali kemiripan antara kata kunci dengan inputan suara metode *Jaro Winkler* ini memperoleh skor terbaiknya dengan nilai 100% dan nilai terendahnya 45.83%.

5. SARAN

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

- 1) Kedepannya diharapkan aplikasi ini dapat dibuat kedalam versi mobile sehingga memudahkan dalam melakukan tes *pronunciation* kapan saja.
- 2) Melakukan pengoptimalan pada *Speech Recognition* dalam mendeteksi suatu ucapan.
- 3) Menambahkan algoritma lain untuk membandingkan keakuratan skor dari algoritma

Jaro Winkler Distance dengan metode lainnya dalam mendeteksi kemiripan kata

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] EF Education First. (2019). EF EPI Indeks Kecakapan Bahasa Inggris EF. Hal. 24.
- [3] EF Education First. (2020). EF EPI Indeks Kecakapan Bahasa Inggris EF. Hal. 4.
- [4] Nurhasanah, Y. I., & Zulkarnain, A. (2018). Pengembangan Pembelajaran Pengenalan Kata Dalam Bahasa Indonesia Menggunakan Multimedia Interaktif dan Speech Recognition. <http://eprints.itenas.ac.id/207/>
- [5] Novantara, P. (2017). Implementasi Algoritma Jaro-Winkler Distance Untuk Sistem Pendeteksi Plagiarisme Pada Dokumen Skripsi. *Buffer Informatika*, 3(2). <http://journal.uniku.ac.id/index.php/buffer/article/view/960>
- [6] Mufidah, D.A. 2017. Speech Recognition Dengan Query Barbahasa Indonesia Untuk Pengiriman. http://digilib.unhas.ac.id/uploaded_files/temporary/DigitalCollection/NWUxZDgwNTY3YWJINjY2MjFiZDQwMDZlZTNhZTYxMGFmZjA1ZTYlOQ==.pdf
- [7] Susanto, R., & Andriana, A. D. (2016). Perbandingan Model Waterfall dan Prototyping Untuk Pengembangan Sistem Informasi. *Majalah Ilmiah UNIKOM*. <https://repository.unikom.ac.id/30459/>
- [8] Wangsanegara, N. K., & Subaeki, B. (2015). Implementasi Natural Language Processing Dalam Pengukuran Ketepatan Ejaan Yang Disempurnakan (EYD) Pada Abstrak Skripsi Menggunakan Algoritma Fuzzy Logic. *Jurnal Teknik Informatika*, 8(2). <http://journal.uinjkt.ac.id/index.php/ti/article/view/3185/2474>
- [9] Muslimin, N. 2017. Sistem Pembaca Pesan Text To Speech Berbasis Android Menggunakan Framework Google API. http://digilib.unhas.ac.id/uploaded_files/temporary/DigitalCollection/ZGM1ZGNkM2UxNTk0NTljMDZlMDhhMzJiMTcxNTBjMjZkMjNlZjQyZg==.pdf
- [10] Sanjaya, A. 2020. Optimasi Pencarian Data Menggunakan Text Filtering dan Algoritma Jaro Winkler, *Jurnal Nero Vol 5 No 1*, Fakultas Teknik Univ. Trunojoyo