

Penerapan Metode *Jaro-Winkler* Distance Untuk Mendeteksi Kesalahan Penulisan Pada Makalah Bahasa Indonesia Berbasis *Mobile*

Muhammad Rafly Al Ajid, Rudi Heriansyah, Dewi Sartika

Prodi S1 Teknik Informatika, Fakultas Ilmu Komputer dan Sains

Universitas Indo Global Mandiri Palembang

Penulis Korespondensi: raflygantajid12@gmail.com³

rudi@uigm.ac.id¹, dewi.sartika@uigm.ac.id²,

Informasi Artikel:

Received: 15 Maret 2024, Accepted: 22 April 2024, Published: 31 Juli 2024

Abstract

A research paper is one of the types of scientific works created by academics to demonstrate their level of expertise in a field. Since a research paper is a scientific work, its writing must be accurate in terms of knowledge and language. Errors in writing scientific works, especially research papers, often occur, and sometimes due to the large number of words in the paper, authors often have difficulty finding writing errors. Based on this problem, an application is needed to find writing errors in a research paper. In searching for incorrect words, a word matching process will be performed between the paper and a dataset, where the dataset will be in JSON format to facilitate the search process. After matching the words from the document and the dataset, words that are different from those in the dataset will be obtained, and then a matching process between the incorrect words and the words in the dataset will be conducted using the Jaro-Winkler Distance method to match the words and evaluate their similarity. The words in the dataset will be derived from the KBBI (Great Dictionary of the Indonesian Language) as a reference for Indonesian words. The application to be developed will use the Android Studio software, which is useful for creating mobile-based applications. The Jaro-Winkler Distance method has a high level of accuracy and precision, around 80.6%. This level of accuracy and precision indicates that the Jaro-Winkler Distance method is quite suitable for use in providing suggestions for incorrectly spelled words or typographical errors.

Keywords: Android Studio, Jaro-Winkler Distance, Mobile, Word Matching

1. Pendahuluan

Makalah adalah karya tulis ilmiah yang menyajikan suatu masalah yang pembahasannya berdasarkan data di lapangan yang bersifat empirisobjektif. Umumnya, makalah ditampilkan dalam sebuah seminar atau dipresentasikan di depan kelas sebagai tugas perkuliahan (A. Pratomo, 2018). Makalah sama seperti karya ilmiah lainnya yang bersifat ilmiah, objektif, netral, sistematis, logis, dan berisi fakta. Sering terjadi kesalahan dalam pembuatan makalah,

Published by:

salah satunya adalah *typographical error* atau kesalahan penulisan. *Typographical error* merupakan masalah dalam pembuatan dokumen dan tidak terkecuali dalam pembuatan makalah yang bersifat ilmiah.

Typographical error adalah kesalahan pengejaan yang terjadi saat melakukan typing (pengetikan). Masalah *typographical error* ini dapat diatasi dengan menggunakan teknik *spelling correction*. *Spelling correction* atau dalam bahasa Indonesia yang berarti Koreksi Ejaan merupakan sebuah teknik yang digunakan untuk mencari kata-kata yang salah berdasarkan dataset kata yang dimiliki oleh aplikasi, dan saran kata diberikan dengan melalui perhitungan menggunakan algoritma-algoritma atau metode yang digunakan pada aplikasi. Metode yang akan digunakan pada aplikasi ini adalah *Jaro-Winkler Distance*, yang merupakan salah satu varian dari *string metri*.

Metode *Jaro-Winkler Distance* merupakan pengembangan dari metode *Jaro Distance Metric* yang berfungsi untuk menghitung tingkat kemiripan dua kata. Metode ini biasa dipakai untuk menemukan duplikasi kata. Nilai *Jaro-winkler* yang semakin besar menunjukkan bahwa dua kata semakin cocok (Friendly, 2017). *Jaro-Winkler Distance* dapat menghitung seberapa mirip dua string yang biasa dipakai untuk mendeteksi duplikasi dokumen. Nilai *Jaro-Winkler Distance* yang semakin besar menunjukkan bahwa dua string semakin sama. Nilai standarnya adalah 0 yang berarti tidak ada kemiripan dan 1 yang berarti ada kemiripan (Tinaliah & Triana, 2018). Kata dicari dengan mencocokkan satu persatu, jika tidak ditemukan maka kata tersebut terindikasi sebagai *Typographical Error* dan akan dilakukan proses pencarian kata yang disarankan oleh program dengan menghitung bobot kemiripannya menggunakan *Jaro-Winkler Distance*.

Batasan masalah yang ditentukan untuk penelitian ini adalah :

1. Bahasa yang digunakan pada *dataset* adalah bahasa Indonesia.
2. Teks yang dilakukan pengecekan tidak berupa simbol, dan angka.
3. Aplikasi dijalankan pada *smartphone Android*.
4. File yang digunakan bertipe *PDF, doc, dan docx*.
5. Pengujian dilakukan pada setiap 1 halaman dokumen berekstensi PDF, DOC, dan DOCX
6. Kata yang diuji merupakan kata baku bahasa Indonesia, dan tidak termasuk nama orang, nama organisasi, dan kata lainnya

2. Tinjauan Literatur

2.1 *Typographical Error*

Typographical error atau dalam bahasa Indonesia yang berarti kesalahan ejaan merupakan sebuah istilah untuk kesalahan dalam pengetikan. *Typographical error* biasa terjadi saat proses mengetik teks yang dapat mengakibatkan perubahan arti dari sebuah kalimat. *Typographical error* terjadi karena kesalahan mekanis atau slip tangan atau jari, dan juga terjadi karena ketidaktahuan penulis terhadap ejaan kata baku yang dimaksud (Arina I. F. et al, 2018). *Typographical error* memiliki beberapa variasi, seperti kesalahan ketik biasa hingga kesalahan tatanan bahasa yang digunakan atau bahkan arti dari kata tersebut. Kesalahan-kesalahan tersebut dapat dikategorikan ke dalam 2 (dua) jenis yaitu *non-word error* dan *real-word error*. Kesalahan yang tidak memiliki makna didalamnya disebut *non-word error*, sedangkan

Published by:

kesalahan *real-word error* terjadi saat kata yang tertulis bernilai benar atau bisa disebut memiliki arti dalam kamus tetapi tidak dimaksudkan pada kalimat tersebut maupun memiliki arti yang berbeda (Arina I. F. et al, 2018).

2.2 Jaro-Winkler Distance

Jaro-Winkler Distance merupakan sebuah algoritma yang digunakan untuk mengukur kemiripan antara 2 (dua) kata atau string (Dewi & Qoiriah, 2021). Algoritma *JaroWinkler Distance* memiliki kompleksitas waktu yang sangat efektif pada *string* pendek dan dapat bekerja lebih cepat daripada algoritma edit distance yang lainnya. Untuk menghitung jarak kedekatan 2 *string* digunakan rumus berikut ini:

$$d_j = \begin{cases} 0 & , m = 0 \\ \frac{1}{3} \times \left(\frac{m}{s_1} + \frac{m}{s_2} + \frac{m-t}{m} \right) & , m > 0 \end{cases} \quad (2.1)$$

Keterangan:

d_j = Jaro Distance dari S1 dan S2.

m = Jumlah karakter yang sama

$|S1|$ = Panjang *string*/kata pertama

$|S2|$ = Panjang *string*/kata kedua

t = Jumlah transposisi (perpindahan huruf yang tidak sesuai dengan tempatnya tetapi tidak mengubah jumlah huruf dari *string*)

Jika jarak kedekatan dari rumus diatas tidak melebihi maksimal, maka dapat diperbolehkan. Maksimal nilai dari jarak kedekatan *string* d_j dihitung dengan menggunakan rumus berikut:

$$maxd = \left(\frac{\max(|S1|, |S2|)}{2} \right) - 1 \quad (2.2)$$

Keterangan:

$maxd$ = jarak maksimum yang diperbolehkan untuk mencocokkan karakter dari dua *string*

$\max(|S1|, |S2|)$ = Panjang maksimal dari kedua *string*

2.3 Confusion Matrix

Confusion Matrix adalah metode untuk mengukur kinerja dalam masalah klasifikasi pembelajaran mesin di mana hasilnya bisa berupa 2 atau lebih kelas. Ini adalah tabel yang terdiri dari 4 kombinasi yang berbeda dari nilai prediksi dan nilai aktual. *Confusion Matrix* juga sering disebut sebagai matriks kesalahan. Pada dasarnya, *Confusion Matrix* memberikan data perbandingan antara hasil klasifikasi yang dicoba oleh sistem dengan hasil klasifikasi yang sebenarnya (Hozairi et al., 2021).

Published by:

3. Metode Penelitian

3.1 Tahap Penelitian

3.1.1 Analisis Masalah

Terdapat banyak cara untuk menyelesaikan masalah *typographical error* salah satunya adalah dengan menggunakan aplikasi *spelling correction*. Aplikasi ini bekerja dengan menerapkan beberapa metode dan algoritma untuk mendeteksi kesalahan-kesalahan *typographical error* seperti kekurangan huruf, kekurangan huruf, dan huruf yang ada pada kata tidak sesuai dengan kata yang seharusnya. Berbagai metode dan algoritma yang bisa digunakan pada aplikasi *spelling correction*, metode *Jaro-Winkler Distance* merupakan salah satu metode yang populer digunakan pada pengembangan aplikasi ini.

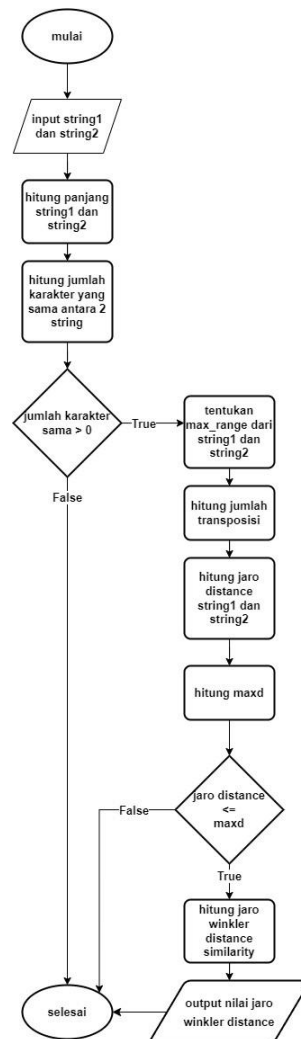
Penelitian ini bertujuan untuk menyelesaikan masalah *typographical error* dengan cara melakukan pendeteksian setiap kata pada dokumen makalah terhadap korpus. Hasil kata-kata yang tidak sesuai dengan korpus nantinya akan dilakukan pendeteksian kata yang benar dengan cara menghitung *Jaro-Winkler Distance* antara kata yang salah dan kata yang ada pada korpus. Kata pada korpus dengan nilai yang tinggi akan dianggap sebagai kata yang paling sesuai untuk disarankan

3.1.2 Analisis Perangkat Lunak

Pada pengembangan perangkat lunak diperlukan analisis untuk mengetahui kebutuhankebutuhan yang harus tercapai atau terpenuhi setelah pengembangan dilakukan. Kebutuhan-kebutuhan ini menjadi acuan pada pengembangan perangkat lunak, berikut ini adalah beberapa kebutuhan-kebutuhan pada penelitian ini.

3.2 Desain Perangkat Lunak

Dengan menggunakan rumus *Jaro-Winkler Distance* 2.1, maka *flowchart* atau alur metode adalah seperti di Gambar 1.



Gambar 1 Flowchart Jaro-Winkler Distance

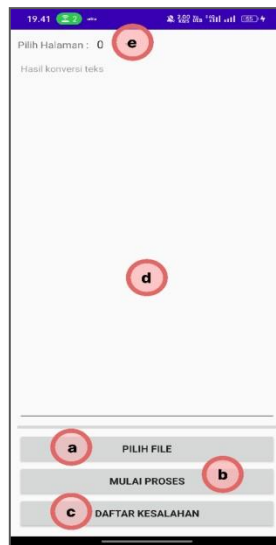
4. Hasil dan Pembahasan

Metode pengembangan perangkat lunak yang digunakan pada penelitian ini adalah SDLC (*Software Development Life Cycle*). Pada metode pengembangan perangkat lunak ini dilakukan beberapa tahap, yaitu perancangan, desain, implementasi / pengkodean, dan pengujian. Tahap perancangan dan desain telah dijelaskan pada bab sebelumnya, sehingga pada bab ini akan dijelaskan tahap implementasi dan pengujian. pengujian akan dilakukan 2 (dua) jenis, yaitu pengujian perangkat lunak dengan menggunakan *blackbox testing* dan pengujian sistem dengan menggunakan *confusion matrix* untuk mendapatkan tingkat akurasi serta presisi dari sistem yang telah dibangun. Berikut ini adalah tahap implementasi perangkat lunak. 4.1 Implementasi Tampilan Antarmuka

Tampilan Antarmuka Halaman Utama

Saat aplikasi dijalankan maka akan langsung menampilkan halaman utama. Pada antarmuka halaman utama terdapat beberapa elemen, seperti edittext, button, dan spinner. Tampilan antarmuka halaman utama aplikasi dapat dilihat pada Gambar 2 di bawah ini

Published by:



Gambar 2. Tampilan Antarmuka Halaman Utama

- Tombol Pilih File berfungsi untuk membuka dialog pemilihan file dokumen makalah yang akan diproses.
- Tombol Mulai Proses berfungsi untuk menjalankan proses textpreprocessing serta perhitungan Jaro-Winkler Distance untuk mencari kata yang disarankan berdasarkan nilai Jaro-Winkler Distance.
- Tombol Daftar Kesalahan berfungsi untuk menampilkan bottomsheets yang berisi daftar kata yang dianggap salah oleh sistem beserta kata yang disarankan benar dan tingkat kemiripannya.
- Bagian edittext ini berguna untuk menampilkan isi dari halaman dokumen Makalah yang dipilih.
- Dropdown atau spinner pilih halaman berfungsi untuk memilih halaman dokumen makalah yang ingin diproses.

Tampilan Antarmuka setelah Pilih *File*

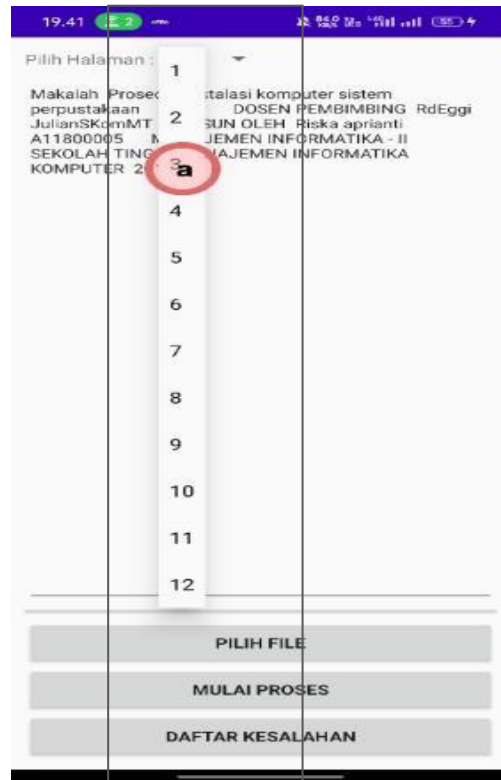
Setelah *file* dokumen makalah dipilih, maka isi dari halaman pertama makalah tersebut akan ditampilkan pada *edittext*.



0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99

Tampilan Antarmuka Pilih Halaman

Pengguna dapat memilih halaman dokumen Makalah yang diinginkan untuk dilakukan proses pendeteksian kesalahan. Setelah halaman dipilih, isi dari halaman tersebut akan ditampilkan pada bagian *edittext*.

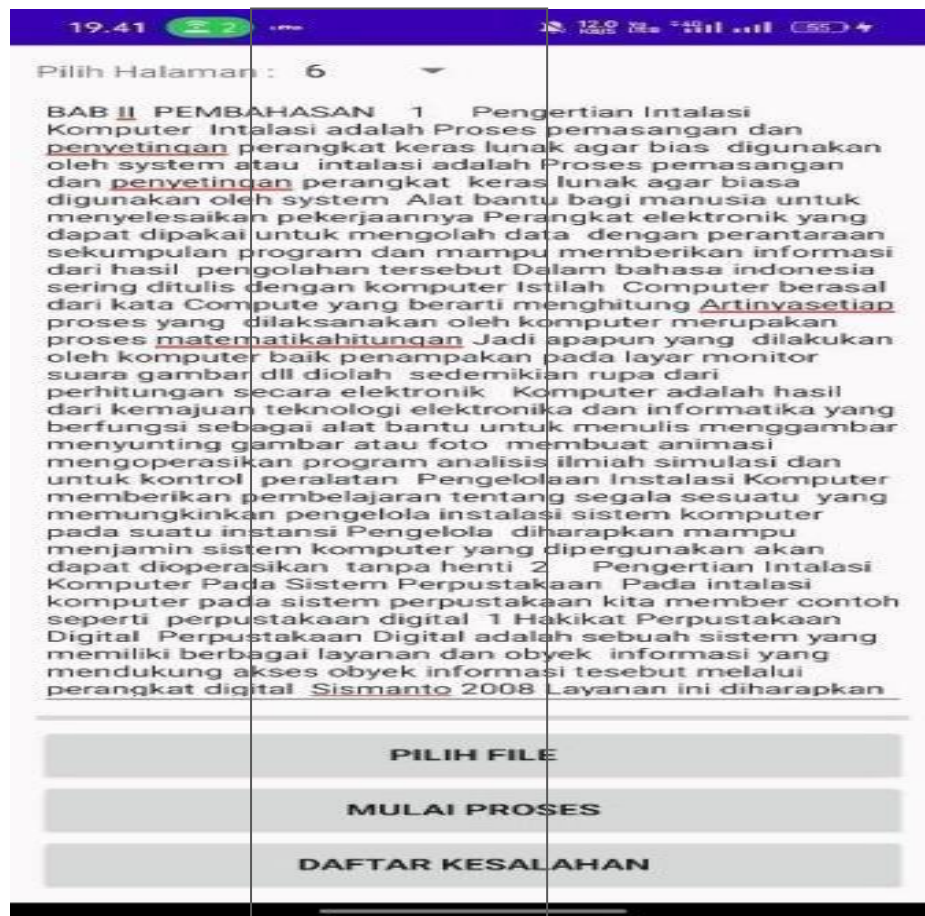


Gambar 4. Tampilan Antarmuka Pilih Halaman

- a. Dropdown atau Spinner berfungsi untuk memilih halaman dokumen makalah yang diinginkan, seperti halaman 2, halaman 3, dan lainnya

Tampilan Antarmuka setelah Pilih Halaman

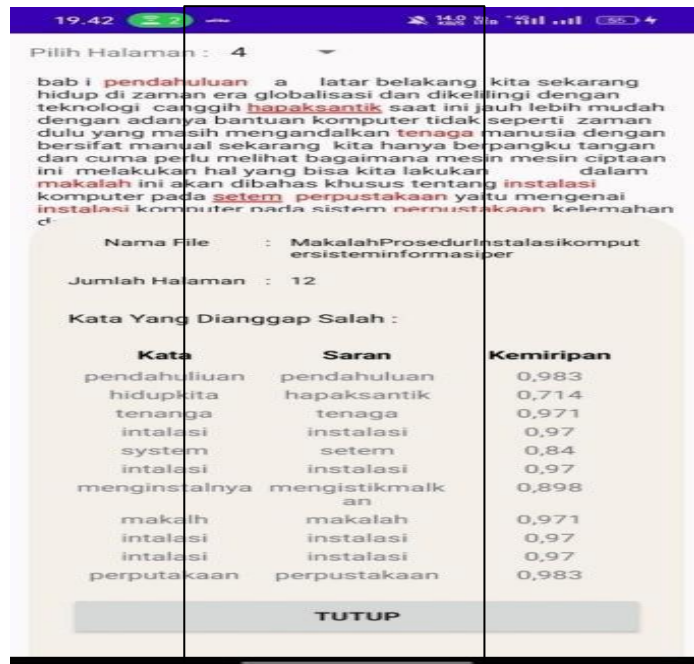
Setelah pengguna memilih halaman dokumen makalah yang diinginkan, maka isi dokumen makalah tersebut akan ditampilkan pada *edittext* dan selanjutnya siap untuk dilakukan proses pendeteksian. Tampilan antarmuka setelah pilih halaman dapat dilihat pada Gambar 5.



Gambar 5. Tampilan Antarmuka setelah Pilih Halaman

Tampilan Antarmuka Daftar Kesalahan

Pengguna dapat melihat daftar kata yang dianggap salah oleh sistem beserta kata yang disarankan oleh sistem berdasarkan tingkat kemiripannya. Setelah pengguna menekan tombol daftar kesalahan maka nantinya akan muncul bottomsheets yang berisikan informasi-informasi tersebut. Pada tampilan antarmuka ini akan ditunjukkan kata-kata yang salah dan telah diberikan saran kata dengan ditandai berwarna merah.



Gambar 6. Tampilan Antarmuka Daftar Kesalahan

4.2 Pengujian

Terdapat 10 makalah yang akan digunakan pada proses pengujian, dan hanya 2 makalah yang akan dibuatkan rincian hasil proses pengujian.

Makalah	Halaman pengujian																						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Makalah 1	0	7	0	0	3	0	3	0	0														
Makalah 2	0	0	0	3	0	1	0	4	1	2	11	11	1	1	1	0	0						
Makalah 3	0	2	0	2	3	0	1	0	0	2	0	2	3	1									
Makalah 4	0	0	0	1	1	0	1	1	0	1	1	3	0										
Makalah 5	0	0	2	10	2	1	4	6	1	0	1	2	0	0	0								
Makalah 6	0	1	0	1	0	0	0	0	0	1	1	0	3	0	1	1	1	0	0	0	0		
Makalah 7	0	1	0	0	0	2	0	2	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Makalah 8	0	0	0	1	0	3	0	0	5	7	1	0	4	1	5	0							
Makalah 9	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	0						
Makalah 10	0	0	0	6	3	6	2	1	3	1	2	0											

Table 2. Jumlah Kata Saran Benar dsn Salah

No.	Makalah	Kata Saran Benar	Kata Saran Salah
1	Makalah 1	10	3
2	Makalah 2	36	1
3	Makalah 3	9	7
4	Makalah 4	9	0
5	Makalah 5	24	5

Published by:

6	Makalah 6	8	2
7	Makalah 7	5	3
8	Makalah 8	19	8
9	Makalah 9	2	1
10	Makalah 10	20	4
Total		142	34
Total kata		176	

Table 3. Confusion Matrix Pemberian Kata Saran

	Data hasil prediksi		
		True	False
	Data sebenarnya		
	True	142	0
	False	34	0

4.2.1 Perhitungan Akurasi, Presisi, dan Recall

Akurasi mengukur sejauh mana sistem klasifikasi benar dalam mengklasifikasi data.

$$\text{Akurasi} = \frac{(142 + 0)}{(142 + 0 + 34 + 0)} = 0.806 \times 100 = 80.6\%$$

Presisi mengukur sejauh mana hasil positif yang diklasifikasikan oleh sistem klasifikasi adalah benar.

$$\text{Presisi} = \frac{(142)}{(142 + 34)} = 0.806 \times 100 = 80.6\%$$

Recall mengukur sejauh mana sistem klasifikasi mampu mengidentifikasi secara benar data yang sebenarnya positif.

$$\text{Recall} = \frac{(142)}{(142 + 0)} = 1 \times 100 = 100\%$$

Berdasarkan pengujian yang telah dilakukan dapat dilihat bahwa tingkat akurasi dan presisi dari sistem aplikasi yang telah dibuat menunjukkan sekitar 80.7% (tujuh puluh lima persen) untuk akurasi serta presisi. Hal ini membuktikan bahwa tingkat ketepatan kata yang direkomendasikan oleh sistem melebihi 50% (lima puluh persen) atau lebih dari setengah. Tetapi masih ada banyak sekali kata yang tidak tepat disarankan oleh sistem aplikasi ini yang disebabkan oleh beberapa faktor, berikut adalah beberapa faktor yang menyebabkan kurangnya tingkat akurasi dan presisi yang dihasilkan oleh sistem: 1. Kata yang salah atau kata yang mengalami typographical error ini memiliki terlalu banyak kata yang kurang ataupun berlebih, sehingga menyebabkan tingkat kemiripan Jaro-Wankler Distance yang telah dihitung oleh sistem menjadi rendah, dan membuat sistem memberikan kata saran yang salah. Contoh, kata yang salah adalah “aym” sedangkan kata yang sebenarnya adalah “bayam”, sistem akan memberikan kata saran seperti “ayam”, atau “ayo” karena memiliki nilai Jaro-Winkler Distance yang lebih tinggi dibandingkan kata yang sebenarnya yaitu “ayam”.

2. Masih kurangnya kosa kata pada dataset korpus yang menyebabkan terdapat beberapa kata yang tidak dapat diberikan saran dengan benar.

Dari faktor-faktor diatas dapat disimpulkan bahwa alasan utama kurangnya tingkat akurasi dan presisi dari sistem aplikasi yang telah dibuat dikarenakan kurangnya dataset korpus kata, dan selain itu juga penulisan kata yang terlalu tidak lengkap dapat mengakibatkan kesalahan dalam pemberian kata yang direkomendasikan oleh sistem aplikasi.

5. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, maka dapat ditarik kesimpulan sebagai berikut:

1. Aplikasi perangkat lunak yang dibangun berhasil mendeteksi *Typographical Error* pada makalah dengan menggunakan metode *Jaro-Winkler Distance*, serta mendapatkan hasil akurasi dan presisi sekitar 80.6% (delapan puluh koma enam persen).
2. Hasil yang didapatkan dari aplikasi perangkat lunak yang dibangun berupa daftar kata yang terdeteksi sebagai kesalahan beserta kata yang disarankan untuk memperbaikinya dan persentase kemiripannya.
3. Aplikasi perangkat lunak berhasil dibangun dengan menerapkan teknik spelling correction dan menggunakan metode *Jaro Winkler Distance*.

Referensi [1] Adhe, D., Rachman, C., Goejantoro, R., Deny, F., & Amijaya, T. (2020). Implementasi Text Mining Pengelompokkan Dokumen Skripsi Menggunakan Metode K-Means Clustering Implementation Of Text Mining For Grouping Thesis Documents Using K-Means Clustering. *Jurnal EKSPONENSIAL*, 11(2), 167–174.

[2] Ahamed, I., Jahan, M., Tasnim, Z., Karim, T., Reza, S. M. S., & Hossain, D. A. (2021). Spell corrector for Bangla language using Norvig 's algorithm and Jaro-Winkler distance. 10(4), 1997–2005. <https://doi.org/10.11591/eei.v10i4.2410>

[3] Alita, D., & Isnain, A. R. (2020). Pendeteksian Sarkasme pada Proses Analisis Sentimen Menggunakan Random Forest Classifier. *Jurnal Komputasi*, 8(2), 50–58. <https://doi.org/10.23960/komputasi.v8i2.2615>

[4] Anoesyirwan, A., Madiistriyatno, H., & Mutmainnah, S. (2020). Improving the Quality of Scientific Publication Management Using Agile Methods. *ADI Bisnis Digital Interdisiplin Jurnal*, 1(2 Desember), 31–39.

[5] Augusfian, F., Mawardi, V. C., Hendryli, J., & Naga, D. S. (2019). Sistem Pengoreksian Ejaan Teks Bahasa Indonesia Dengan Damerau Levenshtein Distance Dan Recurrent Neural Network. *Computatio : Journal of Computer Science and Information Systems*, 3(2), 144.

[6] <https://doi.org/10.24912/computatio.v3i2.6038>

[7] Dewi, N. C., & Qoiriah, A. (2021). Implementasi Algoritma Jaro-Winkler Distance dan N-Gram untuk Deteksi dan Prediksi Perbaikan Kesalahan Penulisan Kata Bahasa Indonesia pada Karya Tulis Ilmiah Mahasiswa. *Journal of Informatics and Computer Science (JINACS)*, 2(03), 169–177. <https://doi.org/10.26740/jinacs.v2n03.p169-177>

Published by:

- [8] Endang, R., Jafar, S., & Dony, O. (2017). Pembelajaran Pemrograman Berorientasi Objek (Object Oriented Programming) Berbasis Project Based Learning. *Informatics for Educators and Professionals*, 2(1), 95 – 104. <http://ejournal-binainsani.ac.id/index.php/ITBI/article/view/668>
- [9] Erdani, B., Aditia, F. D., Rodiah, S., Ciptyasih, & Santi, I. H. (2019). Application System Dictionary of PHP Programming Language Terms Using Brute Force Algorithm. *Multimedia & Artificial Intelligence*, 3(1), 1–8.
https://jurnal.itats.ac.id/wp-content/uploads/2015/10/6.-Anita-T.-Kurniawati-Afrilyan-Ruli-Dwi-Rama_EDITED.pdf
- [10] Fahrudin, T. M., Sa'diyah, I., Latipah, L., Atha Illah, I. Z., Bey Lirna, C. C., & Acarya, B. S. (2021). KEBI 1.0: Indonesian Spelling Error Detection System for Scientific Papers using Dictionary Lookup and Peter Norvig Spelling Corrector. *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, 12(2), 78.
<https://doi.org/10.24843/lkjiti.2021.v12.i02.p02>
- [11] Frando, J., Ruslianto, I., & Rahmi, H. (2019). Penerapan Jaro Winkler Distance Dalam Aplikasi Pengoreksi Kesalahan Penulisan Bahasa Indonesia Berbasis Web. 07(03), 44–53.
- [12] Hozairi, Anwari, & Alim, S. (2021). Implementasi Orange Data Mining Untuk Klasifikasi Kelulusan Mahasiswa Dengan Model K-Nearest Neighbor, Decision Tree Serta Naive Bayes. *Network Engineering Research Operation*, 6(2), 133.
<https://doi.org/10.21107/nero.v6i2.237>
- [13] Jtik, J., Teknologi, J., Ridwan, M., & Fitri, I. (2021). Rancang Bangun Marketplace Berbasis Website menggunakan Metodologi Systems Development Life Cycle (SDLC) dengan Model Waterfall. 5(2).
- [14] Kurniawati, A., Puspitodjati, S., & Rahman, S. (2010). Implementasi Algoritma Jaro-Winkler Distance untuk Membandingkan Kesamaan Dokumen
- [15] Berbahasa Indonesia. *Proceeding, Seminar Ilmiah Nasional Komputer Dan Sistem Intelijen KOMMIT 2008*, Depok, Indonesia.
- [16] Martin, R., Naga, D. S., & Mawardi, V. C. (2021). Penggunaan Spelling Correction Dengan Metode Peter Norvig Dan N-Gram. *Jurnal Ilmu Komputer Dan Sistem Informasi*, 9(1), 175. <https://doi.org/10.24912/jiksi.v9i1.11591>
- [17] Misbullah, A., Mutiawani, V., & Mulyani, C. S. (2022). Deteksi Kata Tak Baku Dan Kesalahan Penulisan Kata Pada Tugas Akhir Mahasiswa Menggunakan Metode Dictionary Lookup. *Cyberspace: Jurnal Pendidikan Teknologi Informasi*, 6(2), 130. <https://doi.org/10.22373/cj.v6i2.15423>
- [18] Muhamad, S., & Wahyu, N. (2020). Pemodelan Diagram Uml Sistem Pembayaran Tunai Pada Transaksi E-Commerce. 4(1).
- [19] Muhammad, S., & Azrai, S. (2018). Pengembangan Sistem Informasi Administrasi Akademik Dengan Rancangan Modul Program Menggunakan. 2(1), 51–57.

- [20] Prasetyo, A., Baihaqi, W. M., & Had, I. S. (2018). Algoritma Jaro-Winkler Distance: Fitur Autocorrect dan Spelling Suggestion pada Penulisan Naskah Bahasa Indonesia di BMS TV. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 5(4), 435. <https://doi.org/10.25126/jtiik.201854780>
- [21] Ratnasari, C. I., Kusumadewi, S., & Rosita, L. (2017). A Non-Word Error Spell Checker for Patient Complaints in Bahasa Indonesia. *International Journal of Information Technology, Computer Science and Open Source*, 1(1), 18–21.
- [22] Sari, N., Gustriasyah, R. and Mair, Z.R., 2023. Analisis User Experience pada Aplikasi Prodeskel Menggunakan Metode System Usability Scale (SUS), User Experience Questionnaire (UEQ), dan Heart Metrics (Doctoral dissertation, Universitas Indo Global Mandiri)
- [23] S., R. A., & Shalahuddin, M. (2018). Rekayasa Perangkat Lunak : Terstruktur dan Berorientasi Objek Edisi Revisi. *Informatika*.
- [24] Silitonga, P. D. ., & Purba, D. E. R. (2021). Implementasi System Development Life Cycle Pada Rancang Bangun Sistem. *Jurnal Sistem Informasi Kaputama (JSIK)*, 5(2), 196–203.
- [25] Suendri. (2018). Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem Informasi Remunerasi Dosen Dengan Database Oracle (Studi Kasus: UIN Sumatera Utara Medan). 6341(November), 1–9.
- [26] Sujaini, H., Muhardi, H., & Jefri Hasiholan, S. (2022). Aplikasi Pengoreksi Ejaan (Spelling Correction) pada Naskah Jurnal Bidang Informatika dengan N- Gram dan Jaro-Winkler Distance. 8(2), 235–244.
- [27] Syarif, M. (2022). Waterfall Sebagai Model Pengembangan Sistem. 6(1), 44–52.
- [28] Yunahar, H. (2018). Perancangan Sistem Informasi Rental Mobil Berbasis Web Pada PT.APM Rent Car. 2(2), 64–77.
- [29] Z. R. Mair, W. Cholil, E. Yulianti, D. Marcelina, Theresiawati and I. N. Isnainiyah, "Convolutional Neural Network Analysis on Handwriting Patterns and Its Relationship to Personality: A Systematical Review," 2023 International Conference on Informatics, Multimedia, Cyber and Informations System (ICIMCIS), Jakarta Selatan, Indonesia, 2023, pp. 308-312, doi: 10.1109/ICIMCIS60089.2023.10348999.

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>)



Published by:

