JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 1, Maret 2025, Pp. 774-780

ANALISA PERBANDINGAN STEMMING DOKUMEN TEKS BERBAHASA JAWA DENGAN ALGORITMA LEVENSHTEIN DISTANCE DAN JARO-WINKLER

Wachid Daga Suryono*[1)], Ema Utami[2)], Dhani Ariatmanto[3)]
1. Magister Teknik Informatika, Universitas Amikom Yogyakarta, Indonesia
2. Magister Teknik Informatika, Universitas Amikom Yogyakarta, Indonesia
3. Magister Teknik Informatika, Universitas Amikom Yogyakarta, Indonesia

Article Info
Kata Kunci: NLP, stemming, bahasa jawa, levenshtein distance, jaro-winkler

Keywords: NLP, stemming, javanese language, levenshtein distance, jaro-winkler

Article history:
Received 5 October 2024
Revised 7 November 2024
Accepted 3 December 2024
Available online 1 March 2025

DOI :
https://doi.org/10.29100/jipi.v10i1.6092

* Corresponding author.
Corresponding Author
E-mail address:
wachiddaga@students.amikom.ac.id

ABSTRAK
Bahasa Jawa merupakan salah satu bahasa yang paling banyak digunakan di Indonesia, namun penelitian terkait bahasa Jawa dalam bidang informatika masih terbilang terbatas. Penelitian ini bertujuan untuk membandingkan kinerja algoritma Levenshtein Distance dan Jaro-Winkler dalam proses stemming dokumen teks berbahasa Jawa. Stemming adalah proses penting untuk pemrosesan teks yang bertujuan untuk mengubah kata-kata menjadi bentuk dasarnya. Bahasa Jawa memiliki tantangan tersendiri karena keterbatasan sumber daya. Dalam penelitian ini, kami menggunakan dataset dokumen teks bahasa Jawa yang telah melalui tahap pre-processing sebelumnya serta kamus bahasa Jawa sebagai acuan. Kedua algoritma diterapkan untuk melakukan stemming pada dokumen teks, dan hasilnya dievaluasi berdasarkan akurasi. Hasil penelitian menunjukkan bahwa rata-rata akurasi keduanya adalah 43%. Penelitian ini memberikan kontribusi dalam pengembangan algoritma stemming bahasa Jawa dan dapat menjadi landasan untuk penelitian lebih lanjut dalam meningkatkan kinerja stemming bahasa Jawa. Selain itu, penelitian ini juga memberikan wawasan baru dalam pemrosesan teks berbahasa Jawa yang dapat bermanfaat dalam berbagai aplikasi NLP dan pengolahan bahasa alami lainnya.

ABSTRACT
Javanese is one of the most widely spoken languages in Indonesia, yet research in Javanese informatics remains limited. This study aims to compare the performance of the Levenshtein Distance and Jaro-Winkler algorithms in stemming Javanese text documents. Stemming is crucial for text processing, aiming to convert words to their base form. Javanese presents challenges due to resource scarcity. In this research, we utilized preprocessed Javanese text datasets and a Javanese language dictionary as reference. Both algorithms were applied for stemming the text documents, and the results were evaluated based on accuracy. The findings revealed an average accuracy of 43%. This suggests Jaro-Winkler's better capability in finding nearest words and providing more accurate results. The research contributes to the development of Javanese stemming algorithms and may serve as a foundation for further studies in improving Javanese stemming performance. Moreover, it offers new insights into Javanese text processing, which can be beneficial for various NLP applications and other natural language processing tasks.

I. PENDAHULUAN

Bahasa Jawa merupakan salah satu bahasa daerah di Indonesia yang memiliki jumlah penutur terbanyak, diperkirakan mencapai sekitar 84 juta jiwa [1]. Meskipun era digital telah memasuki zaman yang semakin maju, penggunaan bahasa lokal seperti Bahasa Jawa masih tetap relevan dan menjadi bagian penting dari kehidupan sehari-hari masyarakat. Bahasa Jawa, dengan basis pengguna yang luas, memiliki kekayaan budaya dan tradisi yang mendalam yang terwujud dalam penggunaannya sehari-hari[16]. Namun, keberadaannya dalam dunia teknologi terkadang terabaikan, terutama dalam konteks pemrosesan bahasa alami (Natural Language Processing/NLP)[11]. Bahasa Jawa memiliki kekayaan dan kompleksitas gramatikal yang unik[13][17][18],

menjadikannya penting untuk dilestarikan dan diteliti.

Meskipun bahasa Jawa banyak digunakan, penelitian tentang pemrosesan bahasa alaminya (NLP) masih tertinggal dibandingkan bahasa lain seperti bahasa Inggris[3][4]. Salah satu area yang masih minim penelitian adalah stemming, yaitu proses untuk menemukan bentuk dasar kata (kata dasar) dari kata-kata yang memiliki imbuhan[5]. Stemming merupakan langkah penting dalam berbagai aplikasi NLP seperti information retrieval, text mining, dan machine translation. Dengan menemukan kata dasar, aplikasi NLP dapat memahami makna kata dengan lebih baik dan meningkatkan performanya.

Algoritma Levenshtein Distance dan Jaro-Winkler merupakan dua algoritma yang umum digunakan untuk string matching[10][12] dan spell correction[2]. Algoritma ini dapat digunakan untuk menghitung jarak antara dua string dan menemukan string yang paling mirip. Penggunaan algoritma Levenshtein Distance dan Jaro-Winkler untuk stemming bahasa Jawa memiliki beberapa keuntungan:

Akurasi: Algoritma ini memiliki tingkat akurasi yang tinggi dalam menemukan kata dasar.
Efisiensi: Algoritma ini relatif efisien dan dapat diimplementasikan dengan mudah.
Fleksibilitas: Algoritma ini dapat diadaptasi untuk berbagai jenis bahasa dan dialek[6][7].

Penelitian ini bertujuan untuk mengembangkan algoritma stemming bahasa Jawa yang menggunakan algoritma Levenshtein Distance dan Jaro-Winkler. Algoritma ini diharapkan dapat meningkatkan akurasi dan efisiensi proses stemming bahasa Jawa.

Meskipun Bahasa Jawa memiliki basis pengguna yang besar, penelitian tentangnya dalam konteks NLP masih terbatas, terutama dalam bidang stemming[3]. Stemming, sebuah proses penting dalam NLP, bertujuan untuk mengubah kata-kata ke dalam bentuk dasarnya atau kata dasar, sehingga memungkinkan analisis lebih lanjut seperti klasifikasi teks, pencarian informasi, dan penggabungan dokumen[14].

Stemming Bahasa Jawa memiliki tantangan tersendiri karena perbedaan struktur dan kaidah linguistik yang khas. Dalam Bahasa Jawa, variasi kata-kata terjadi karena adanya awalan, akhiran, dan perubahan fonologis yang kompleks[14][15][19][20]. Oleh karena itu, pengembangan algoritma stemming yang efektif dan akurat menjadi krusial dalam meningkatkan kinerja sistem NLP untuk Bahasa Jawa.

Algoritma Levenshtein Distance dan Jaro-Winkler telah terbukti efektif dalam berbagai aplikasi pemrosesan string, seperti pengejaan kata yang benar (spell correction)[8][19], pencocokan kata (string matching)[19], dan identifikasi kemiripan antar string[9][20]. Penerapan algoritma ini dalam konteks stemming Bahasa Jawa diharapkan dapat memberikan kontribusi yang signifikan dalam pengembangan teknologi NLP untuk Bahasa Jawa.

Penelitian ini bertujuan untuk membandingkan kinerja algoritma stemming Levenshtein Distance dan Jaro-Winkler dalam konteks bahasa Jawa. Dengan melakukan perbandingan ini, diharapkan dapat diperoleh pemahaman yang lebih baik tentang kinerja kedua algoritma tersebut. Memang dalam penelitian sebelumnya, penggunaan algoritma Levenshtein Distance dan Jaro-winkler untuk stemming bahasa jawa belum banyak dilakukan. Penelitian terdahulu pernah dilakukan untuk stemming bahasa indonesia yang memberikan hasil yang bagus[26]. Untuk algoritma Jaro-Winkler sendiri juga pernah dilakukan penelitian untuk bahasa Indonesia yang memberikan hasil akurasi yang bagus[27].

Pada penelitian[3] penulis mengajukan algoritma untuk stemming bahasa jawa yang menggunakan kamus kata sebagai acuan untuk stemming. Pada [23] menggunakan algoritma levenshtein distance untuk Bahasa jawa dan Analisa morfologi menunjukkan hasil 63% untuk akurasinya.

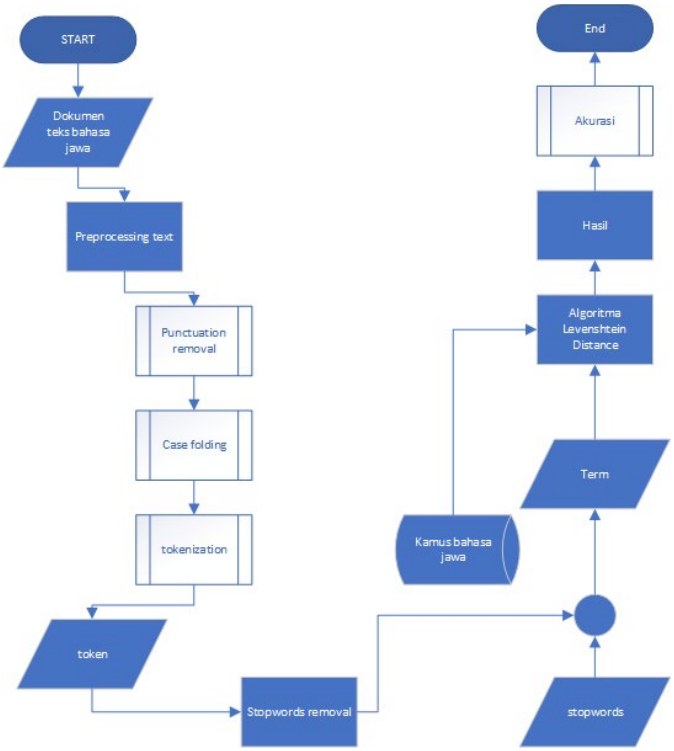
Penelitian berbeda [2] dilakukan untuk bidang spell correction dan menunjukkan hasil bahwa Levenshtein distance optimal sebagai koreksi kata dalam preprocessing. Penelitian [8] juga menyarankan bahwa algoritma jaro-winkler lebih baik akurasinya dibandingkan algoritm levenshtein distance untuk mendeteksi plagiarism dokumen.

II. METODE PENELITIAN

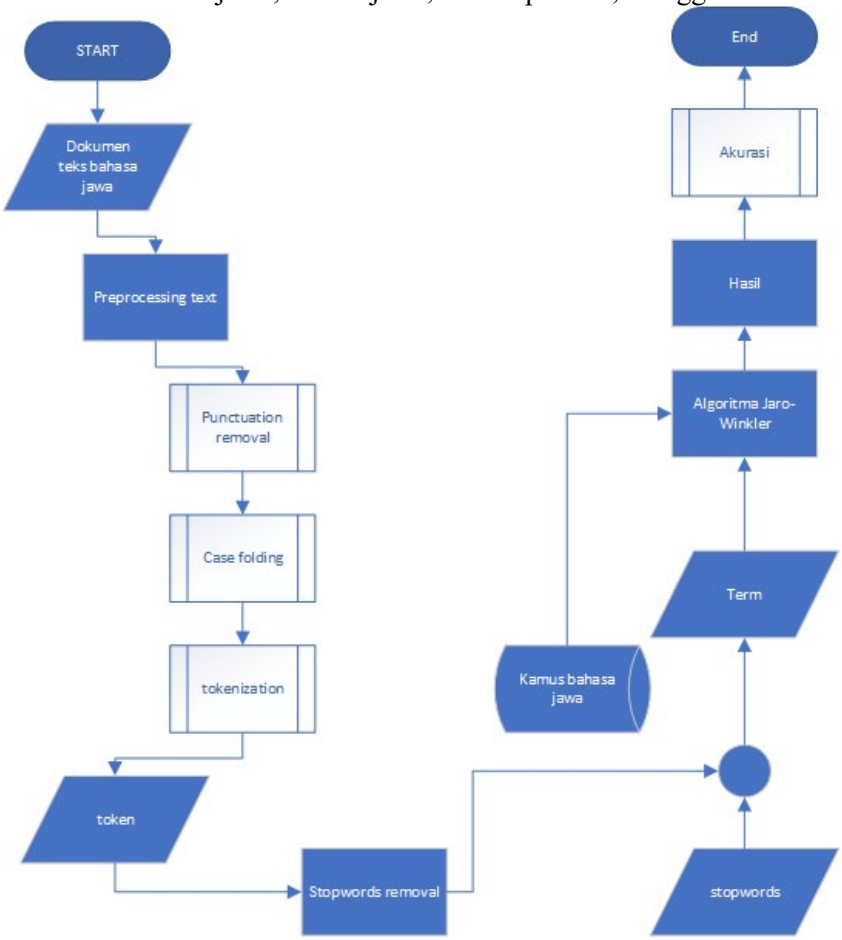
Alur proses penelitian dijelaskan pada gambar 1. Proses dimulai dari *start* sampai dengan *end* dan melalui tahapan-tahapan yang akan dijelaskan setiap prosesnya. Gambar 1 menjelaskan proses penelitian menggunakan algoritma levenshtein distance. Sedangkan pada gambar 2 adalah gambar yang menunjukkan alur penelitian dengan menggunakan algoritma jaro-winkler.

Gambar 1. Alur Penelitian Menggunakan Algoritma Levenshtein Distance

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 1, Maret 2025, Pp. 774-780



Dalam prosesnya, tidak ada perbedaan proses. Hanya perbedaan di bagian penggunaan algoritmanya. Mulai dari proses *start* sampai dengan *end* tidak ada perbedaan sama sekali kecuali di bagian penggunaan algoritmanya saja. Penggunaan data dokumen teks bahasa jawa, kamus jawa, dan stopwords, menggunakan data yang sama.



Gambar 2. Alur Penelitian Menggunakan Algoritma Jaro-Winkler

A. Pengumpulan Data

Pada pengumpulan data ini terdapat 2 data yang dikumpulkan data kamus Bahasa jawa serta data stopwords. Masing-masing dari data yang dikumpulkan memiliki sumber masing-masing.

Data Dokumen Bahasa Jawa dikumpulkan dari beberapa majalah berbahasa jawa dan juga naskah drama berjudul rajapati.

Dokumen Bahasa Jawa berasal dari majalah Panjebar Semangat, Djaka Lodhang, dan juga Jayabaya sebanyak 5 dokumen. Dari majalah Jaya Baya sebanyak 2 dokumen, dari Djaka Lodhang 2 dokumen dan juga dari majalah Panjebar Semangat sebanyak 1 dokumen. Dokumen tersebut dipilih dikarenakan sudah divalidasi oleh pihak editor. Dokumen ini dalam penelitian diberikan nama text6 sampai dengan text10.

Dokumen naskah drama Rajapati terdapat sebanyak 5 dokumen. Dokumen rajapati ini merupakan seri naskah drama karangan Ahmad Bakri. Dokumen teks ini diberikan nama untuk percobaan dengan nama text1 sampai dengan text5.

Data kamus Bahasa jawa berasal dari https://www.sastra.org/ [21] yang memiliki koleksi terkait dengan kesastraan Bahasa jawa. Kamus yang akan digunakan penulis dalam penelitian ini adalah Bausastra Jawa dari Poerwadarminta tahun 1939 yang bisa diakses di https://www.sastra.org/koleksi?cid=12&sid=41/ [21]. dari hasil crawling kamus bahasa jawa tersebut, terdapat 43155 kata Bahasa jawa.

Data stopwords berasal dari beberapa penelitian terdahulu terkait dengan stemming Bahasa jawa.

Data dokumen ini merupakan data utama yang akan diproses untuk stemming pada penelitian ini.

B. Pre-Processing Data

Pre-processing data pada penelitian ini digunakan untuk memproses terlebih dahulu data yang akan digunakan untuk penelitian. Data dari dokumen teks Bahasa jawa tersebut akan mengalami beberapa proses untuk meminimalisir fraud atau membersihkan data dari data-data yang tidak diperlukan dalam penelitian.

Proses yang pertama dalam *pre-processing data* adalah punctuation removal. Tahap ini digunakan untuk membersihkan data dokumen teks Bahasa jawa dari tanda baca yang ada pada dokumen. Beberapa dokumen memiliki tanda baca seperti '.'(titik), ','(koma), '?' (tanda tanya), ataupun tanda seru '!' yang berada di dalamnya. Tidak memungkiri juga terdapat tanda baca lain yang terdapat pada dokumen yang tidak diperlukan atau perlu dibersihkan dalam penelitian ini.

Proses berikutnya adalah case folding. Proses ini adalah mengubah semua huruf pada dokumen menjadi huruf kecil. Sebagai contoh apabila kita mempunyai kalimat : "Madhegani ing Ngamarta" maka Ketika memasuki proses ini, kalimat akan berubah menjadi "mandhegani ing ngamarta"

Memasuki proses pre-processing berikutnya adalah tokenization. Tokenization adalah proses untuk memecah kalimat yang ada di dokumen tersebut menjadi kata. Dalam penelitian ini, salah satu rule yang diberikan oleh penulis adalah, apabila ada kata dengan karakter kurang dari 3, maka dianggap tidak ada. Sehingga, apabila dalam dokumen teks Bahasa jawa tersebut terdapat kata yang mempunyai 1 atau 2 huruf, maka dalam proses preprocessing akan langsung dieliminisasi atau dihilangkan. Dari proses ini, akan didapatkan kata-kata yang sudah menjadi bentuk token untuk masuk ke dalam implementasi.

C. Implementasi

Implementasi pada tahap ini adalah keseluruhan proses dari setelah pre-processing sampai dengan perhitungan akurasi. Pada proses implementasi ini dilakukan beberapa tahapan lebih lanjut dari tahap pre-processing.

Stopwords removal adalah proses untuk menghilangkan kata-kata tidak perlu (stopwords) pada dokumen Bahasa jawa yang sudah berubah menjadi bentuk token. Stopword ini bisa berupa kata-kata seperti : 'iki', 'iku', 'ing' yang tidak mempunyai makna dalam konteks NLP.

Dari proses tersebut akan didapatkan term. Term adalah token yang sudah dilakukan pembersihan kata-kata stopwords.

Proses berikutnya adalah term akan dilakukan stemming menggunakan algoritma Levenshtein Distance maupun Jaro-Winkler yang diperbandingkan dengan kamus Bahasa jawa dari bausastra Bahasa jawa poerwodarminta.

Cara kerja untuk stemming ini adalah dengan cara membandingkan term dengan kamus Bahasa jawa.

Pada algoritma levenshtein distance, term akan masuk ke dalam proses, kemudian akan dicari kandidat yang sesuai dengan kata dasar berdasarkan score level kedekatannya. Skor kedekatakan ini dimulai dari 0 yang berarti tidak dekat dan 1 yang berarti dekat. Score kedekatan ini akan dipilih yang paling mendekati 1 sebagai kandidat kata dasar sesuai dengan kamus. Algoritma Levenshtein Distance digunakan untuk menemukan kata terdekat dalam

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 1, Maret 2025, Pp. 774-780

bahasa Jawa dengan menghitung jumlah operasi (penyisipan, penghapusan, dan penggantian karakter) yang diperlukan untuk mengubah satu kata menjadi kata lainnya. Setiap kata dalam dokumen teks dipetakan ke kamus bahasa Jawa, kemudian dihitung jaraknya terhadap setiap kata dalam kamus. Kata terstem dipilih berdasarkan jarak Levenshtein Distance terkecil, yang menunjukkan kesamaan yang paling besar dengan kata dalam dokumen teks.

Sedangkan pada algoritma jawo winkler mempunyai mekanisme yang berbeda. Algoritma Jaro-Winkler digunakan untuk menemukan kata terdekat dalam bahasa Jawa dengan menghitung tingkat kesamaan antara setiap kata dalam dokumen teks dan kata-kata dalam kamus bahasa Jawa. Langkah awalnya adalah memetakan setiap kata dalam dokumen teks ke kamus kata dalam bahasa Jawa. Kemudian, algoritma menghitung kesamaan antara kedua kata dengan mempertimbangkan jumlah karakter yang sama dan urutan karakter yang mirip. Nilai kesamaan dihitung berdasarkan beberapa komponen, seperti jumlah karakter yang cocok, jarak transposisi, dan skala penalti. Kata terstem dipilih berdasarkan nilai kesamaan tertinggi dengan kata dalam kamus, yang dianggap sebagai representasi terbaik dari kata dalam dokumen teks.

Hasil yang diperoleh adalah berupa, term asal, hasil stemming dan label untuk kategori ‘sesuai’ dan ‘tidak sesuai’. Label ‘sesuai’ akan diberikan nilai ‘1’ dan label ‘tidak sesuai’ akan diberikan nilai ‘0’.

Selanjutnya akan memasuki proses perhitungan akurasi untuk melihat seberapa akurat penggunaan algortima levenshtein distance dan algoritma jaro-winkler untuk stemming dokumen teks Bahasa jawa.

D. Penghitungan akurasi

Penghitungan akurasi adalah tahap untuk menghitung seberapa akurat penggunaan algoritma levenshtein distance dan juga algoritma jaro-winkler. Evaluasi algoritma stemming dilakukan untuk mengetahui tingkat keakuratan hasil stemming dan waktu untuk melakukan stemming[3][22-23]. Dalam hal ini, tingkat keakuratan ditentukan berdasarkan nilai presisi. Presisi adalah rasio jumlah record akurat yang didapatkan dengan jumlah total record baik akurat maupun tidak akurat yang didapatkan[24]. Dalam hal ini record adalah stem. Sebuah kata hasil stemming dianggap akurat jika masukan dan keluaran sesuai dengan penelitian yang dilakukan. Contoh perhitungan, jika dokumen yang distemming terdiri dari 200 kata, setelah proses stemming kata yang dianggap akurat sebanyak 50 kata. Maka presisinya adalah 50 : 200 = 0.25 (25%). Untuk lebih jelasnya rumus perhitungan presisi dapat dilihat pada Persamaan 1[3][22-24]. Dalam konteks penelitian ini, jumlah stem yang sesuai direpresentasikan dengan ‘correct_labels’ dan jumlah total record direpresentasikan dengan ‘total_labels’.

$$Accuraccy = \frac{correct\_labels}{total\_labels} * 100\% \quad (1)$$

dari persamaan tersebut, akan didapatkan akurasi dari penelitian ini.

III. HASIL DAN PEMBAHASAN

Hasil dari eksperimen ini terdapat pada tabel 1 dan tabel 2 yang memperlihatkan hasil akurasi dari setiap dokumen yang telah dilakukan proses stemming. Pada tabel 1 hasil stemming terdapat dokumen bertuliskan text1_LD yang berarti adalah dokumen teks nomor 1 yang dilakukan stemming menggunakan algoritma

TABEL I.
HASIL STEMMING MENGGUNAKAN LEVENSHTEIN DISTANCE

File	Total_Labels	Correct_Labels	Accuracy_Percentage
text1_LD.csv	838	387	46%
text10_LD.csv	18364	7659	42%
text2_LD.csv	1007	419	42%
text3_LD.csv	972	412	42%
text4_LD.csv	1029	442	43%
text5_LD.csv	1060	444	42%
text6_LD.csv	16213	6988	43%
text7_LD.csv	16128	6986	43%
text8_LD.csv	15843	6766	43%
text9_LD.csv	20478	9248	45%

levenshtein distance. Tabel 2 adalah tabel hasil stemming menggunakan algoritma Jaro-Winkler.

Dari tabel 1. hasil penelitian, dapat dilihat bahwa akurasi dari penggunaan algoritma Levenshtein Distance,

memiliki rentang antara 42% hingga 46%. Meskipun demikian, terdapat variasi kecil dalam akurasi antara berbagai dokumen, di mana beberapa dokumen memiliki akurasi yang sedikit lebih tinggi atau lebih rendah daripada yang lain. Sehingga rerata akurasi dari semua dokumen baik yang menggunakan algoritma Levenshtein Distance adalah sebesar 43%. Hasil tersebut tentu jauh lebih rendah dibandingkan penelitian sebelumnya yang menggunakan algoritma levenshtein distance dan Analisa morfologi. Begitu juga pula penggunaan algoritma levenshtein distance untuk stemming memiliki akurasi yang lebih rendah dibandingkan dengan penggunaan algoritma levenshtein distance untuk spell correction maupun deteksi plagiarism.

TABEL II.
HASIL STEMMING MENGGUNAKAN ALGORITMA JARO-WINKLER

File	Total_Labels	Correct_Labels	Accuracy_Percentage
text1_JW.csv	838	387	46%
text10_JW.csv	18364	7659	42%
text2_JW.csv	1007	419	42%
text3_JW.csv	972	412	42%
text4_JW.csv	1029	442	43%
text5_JW.csv	1060	444	42%
text6_JW.csv	16213	6988	43%
text7_JW.csv	16128	6986	43%
text8_JW.csv	15843	6766	43%
text9_JW.csv	20478	9248	45%

Dari tabel 2. hasil penelitian, dapat dilihat bahwa akurasi dari penggunaan algoritma Jaro-Winkler, memiliki rentang antara 42% hingga 46%. Meskipun demikian, terdapat variasi kecil dalam akurasi antara berbagai dokumen, di mana beberapa dokumen memiliki akurasi yang sedikit lebih tinggi atau lebih rendah daripada yang lain. Sehingga rerata akurasi dari semua dokumen baik yang menggunakan algoritma Jaro-Winkler adalah sebesar 43%.

Kolom *correct_labels* pada tabel 1 dan tabel 2 adalah kolom jumlah kata yang sesuai dengan hasil stemming sedangkan *total_labels* pada tabel 1 dan tabel 2 adalah jumlah semua term pada dokumen tersebut.

Dalam penelitian ini juga terdapat beberapa kata yang tidak terstem dengan benar meskipun seharusnya terstem. Hal ini menunjukkan bahwa kedua algoritma, Levenshtein Distance (LD) memiliki keterbatasan dalam menangani variasi morfologis dan struktur bahasa Jawa yang kompleks. Sebagai contoh : term ‘pawakane’ yang kemudian dilakukan proses stemming menggunakan algoritma LD maupun JW memberikan hasil yang paling dekat dengan ‘pawakan’. Tentu secara hasil stemming menunjukkan bahwa hasilnya benar, akan tetapi dari model memberikan keterangan ‘tidak sesuai’. Begitu juga ketika ada term ‘banget’ yang kemudian mendapatkan hasil stemming ‘bangêt’ yang seharusnya benar akan tetapi model juga memberikan hasil ‘tidak sesuai’.

Algoritma jaro-winkler sendiri menunjukkan kinerja akurasi 43% untuk stemming Bahasa jawa. Akurasi tersebut tentu tidak sebaik Ketika algoritma jaro-winkler digunakan sebagai algoritma untuk deteksi plagiarism ataupun spell correction.

Pembahasan ini menunjukkan bahwa baik algoritma Jaro-Winkler maupun Levenshtein Distance dapat digunakan untuk stemming dokumen teks berbahasa Jawa dengan tingkat akurasi yang serupa, namun jika dibandingkan dengan penelitian terdahulu menggunakan metode yang lain[23], tentu akurasi ini belum sepenuhnya lebih baik. Selain itu, peningkatan akurasi mungkin dapat dicapai dengan melakukan penyesuaian parameter atau dengan mengkombinasikan kedua algoritma untuk meningkatkan kinerja.

IV. KESIMPULAN DAN SARAN

Penelitian ini menunjukkan penggunaan algoritma Levenshtein Distance dan Jaro winkler untuk string matching ataupun spell correction memang baik, akan tetapi tidak terlalu baik ketika diterapkan untuk proses stemming. Berdasarkan penelitian ini, saran untuk penelitian ke depannya, bisa melakukan penelitian lanjutan atau kombinasi dengan algoritma yang lain untuk bisa mendapatkan hasil yang lebih baik. Berdasarkan hasil penelitian yang diberikan, dapat disimpulkan bahwa kedua algoritma, Jaro-Winkler dan Levenshtein Distance, memiliki tingkat akurasi yang serupa dalam melakukan stemming pada dokumen teks berbahasa Jawa. Meskipun terdapat variasi kecil dalam akurasi antara berbagai dokumen, secara keseluruhan tidak ada algoritma yang secara signifikan lebih unggul daripada yang lain. Dari penelitian sebelumnya, penggunaan levenshtein distance dengan metode Analisa morfologi memiliki akurasi yang lebih baik dibandingkan penelitian ini. Namun untuk algoritma Jaro-Winkler tidak ada penelitian terdahulu sebagai pembanding dengan penelitian ini. Sehingga penggunaan algoritma Jaro-Winkler untuk stemming dokumen teks Bahasa jawa memerlukan penelitian lanjutan agar memperoleh akurasi

lebih baik lagi. Meskipun demikian, peningkatan akurasi masih merupakan area yang dapat dieksplorasi lebih lanjut. Penyesuaian parameter, penggunaan kombinasi algoritma, atau pengembangan metode baru dapat membantu meningkatkan kinerja stemming untuk bahasa Jawa. Secara keseluruhan, penelitian ini memberikan kontribusi penting dalam pemahaman tentang kinerja algoritma stemming untuk bahasa Jawa dan memberikan landasan untuk penelitian lebih lanjut dalam pengembangan algoritma yang lebih efektif dan efisien dalam pemrosesan teks berbahasa Jawa.

DAFTAR PUSTAKA

[1] Badan Pengembangan dan Pembinaan Bahasa, Kementerian Pendidikan dan Kebudayaan Republik Indonesia. (2017). Kamus Besar Bahasa Indonesia (KBBI) Online. Diperoleh dari https://kbbi.kemdikbud.go.id
[2] Adnan Nur, M. (2021). Perbandingan Levenshtein Distance Dan Jaro-Winkler Distance Untuk Koreksi Kata Dalam Preprocessing Analisis Sentimen Pengguna Twitter. Jurnal Fokus Elektroda : Energi Listrik, Telekomunikasi, Komputer, Elektronika Dan Kendali), 6(2), 88–93. https://doi.org/10.33772/JFE.V6I2.17751
[3] Agusta, L., & Harjoko, A. (2009). Algoritma stemming untuk dokumen teks Bahasa Jawa. http://etd.repository.ugm.ac.id/penelitian/detail/41269
[4] Amin, F., & Purwaningtyas. (2016). Stemmer Bahasa Jawa Ngoko dengan Metode Affix Removal Stemmers(Rule Based Approach). Jurnal Teknologi Informasi DINAMIK, 21, 16–24. https://www.unisbank.ac.id/ojs/index.php/fti1/article/view/6076/1943
[5] Asian, J., Williams, H. E., Tahaghoghi, S. M. M., Nazief, B., & Adriani, M. (2005). Stemming Indonesian : A Confix-Stripping Approach. Conferences in Research and Practice in Information Technology Series, 38, 307–314. https://doi.org/10.1145/1316457.1316459
[6] Bacchin, M., Ferro, N., & Melucci, M. (2005). A probabilistic model for stemmer generation. Information Processing and Management, 41(1), 121–137. https://doi.org/10.1016/j.ipm.2004.04.006
[7] Cahyani, D. E., Utami, L. M. T., & Setiadi, H. (2019). Clustering of Javanese News in Krama Alus Level with Javanese Stemming. ICOIACT, 462–467.
[8] Damerau Levenshtein dan Jaro-Winkler, K. (2020). Kombinasi Damerau Levenshtein dan Jaro-Winkler Distance Untuk Koreksi Kata Bahasa Inggris. Jurnal Teknik Informatika Dan Sistem Informasi, 6(2), 2443–2229. https://doi.org/10.28932/JUTISI.V6I2.2493
[9] Indriyono, B. V. (2020). Kombinasi Damerau Levenshtein dan Jaro-Winkler Distance Untuk Koreksi Kata Bahasa Inggris. Jurnal Teknik Informatika Dan Sistem Informasi, 6(2). https://doi.org/10.28932/jutisi.v6i2.2493
[10] Julian Tannga, M., & Rahman, S. (2017). ANALISIS PERBANDINGAN ALGORITMA LEVENSHTEIN DISTANCE DAN JARO WINKLER UNTUK APLIKASI DETEKSI PLAGIARISME DOKUMEN TEKS. JTRISTE, 4(1), 44–54.
[11] Kartika, H. C., & Suharso, W. (2013). PENERAPAN TEKNIK STEMMING PADA BAHASA JAWA NGOKO BERBASIS ALGORITMA PORTER.
[12] Kastowo, D., Saputra, A., Suryono, W. D., & Setyowati, E. (2022). Comparative analysis of the Nazief Adriani and Levenshtein Distance algorithms to measure the level of similarity of Javanese news using Rabin Krap. JNANALOKA, 3(1), 1–10. https://doi.org/10.36802/JNANALOKA.2022.V3-NO1-1-10
[13] (Bentuk dan Struktur Bahasa Jawa). https://staffnew.uny.ac.id/upload/132006198/penelitian/Morfologi%20Bahasa%20Jawa.pdf
[14] Nq, M. A., Manik, L. P., & Widiyatmoko, D. (2020). Stemming Javanese: Another Adaptation of the Nazief-Adriani Algorithm. 2020 3rd International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2020, 627–631. https://doi.org/10.1109/ISRITI51436.2020.9315420
[15] Sugiarto, Diyasa, I. G. S. M., & Diana, I. N. (2020). Levenshtein distance algorithm analysis on enrollment and disposition of letters application. Proceeding - 6th Information Technology International Seminar, ITIS 2020, 198–202. https://doi.org/10.1109/ITIS50118.2020.9321030
[16] Sumarlam. (2004). Aspektualitas bahasa jawa : kajian morfologi dan sintaksis. Surakarta Pustaka Cakra.
[17] Uhlenbeck, E. M. (1949). The structure of the Javanese morpheme. Lingua, 2(C), 239–270. https://doi.org/10.1016/0024-3841(49)90027-3
[18] Uhlenbeck, E. M. (1982). Kajian Morfologi Bahasa Jawa (1982) (4th ed., Vol. 4). Pusat Pembinaan dan Pengembangan Bahasa Departemen Pendidikan dan Kebudayaan. Aspektualitas bahasa Jawa: kajian morfologi dan sintaksis
[19] Winkler, W. E. (1990). String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage Cleaning and Analyzing Sets of Files View project. https://www.researchgate.net/publication/243772975
[20] Yulianingsih, Y. (2017). Implementasi Algoritma Jaro-Winkler dan Levenstein Distance dalam Pencarian Data pada Database. STRING (Satuan Tulisan Riset Dan Inovasi Teknologi), 2(1), 18–27. https://doi.org/10.30998/STRING.V2I1.1720
[21]]"Leksikon." Accessed: Mar. 19, 2024. [Online]. Available: https://www.sastra.org/leksikon
[22] M. Fauziyah, "STEMMING BAHASA JAWA MENGGUNAKAN ALGORITMA LEVENSHTEIN DAN ANALISA MORFOLOGI," Malang, 2019. Accessed: Apr. 07, 2023. [Online]. Available: http://etheses.uin-malang.ac.id/16387/1/12650132.pdf
[23] A. P. Wibawa and M. N. Hakim, "STEMMING BAHASA JAWA MENGGUNAKAN DAMERAU LEVENSHTEIN DISTANCE (DLD)," JURNAL TEKNIK INFORMATIKA, vol. 14, no. 1, pp. 22–27, Sep. 2021, doi: 10.15408/jti.v14i1.15010.
[24] Jizba, R. (2000). Measuring search effectiveness. Creighton University Health Sciences Library and …. http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle: Measuring+Search+Effectiveness#2
[25] Navarro, G. (2001). A guided tour to approximate string matching. ACM Computing Surveys (CSUR), 33(1), 31-88.
[26] Putra, Rahardyan. (2018). Optimalisasi Stemming Kata Berimbuhan Tidak Baku Pada Bahasa Indonesia Dengan Levenshtein Distance. Jurnal Informatika: Jurnal Pengembangan IT. 3. 200-205. 10.30591/jpit.v3i2.877.
[27] M. Qulub et al., "Stemming Kata Berimbuhan Tidak Baku Bahasa Indonesia Menggunakan Algoritma Jaro-Winkler Distance," Creative Information Technology Journal, vol. 5, no. 4, pp. 254–263, Mar. 2020, doi: 10.24076/CITEC.2018V5I4.218.