

Perbaikan Kata pada Sistem Chatbot dengan Metode Jaro Winkler

I Kadek Trio Putra Pinajeng^{a1}, I Made Sukarsa^{a2}, I Made Suwija Putra^{a3}

^aProgram Studi Teknologi Informasi, Fakultas Teknik, Universitas Udayana
Bukit Jimbaran, Bali, Indonesia, telp. (0361) 701806

e-mail: ¹trioputrap@student.unud.ac.id, ²sukarsa@unud.ac.id, ³putrasuwija@unud.ac.id

Abstrak

Kesalahan pengguna saat mengetikkan kata pada pesan sangat mempengaruhi tingkat akurasi sistem chatbot dalam memberikan pesan balasan yang tepat. Masalah ini dapat diatasi dengan menambahkan fitur perbaikan kata pada sistem chatbot sehingga kesalahan penulisan kata dapat dimengerti oleh chatbot. Penelitian ini menggunakan algoritma Jaro-Winkler untuk membuat fitur perbaikan kata. Pemilihan algoritma Jaro-Winkler berdasarkan kecepatan pemrosesan dan keefektifan dalam membandingkan string pendek sehingga cocok untuk memperbaiki kata salah. Sistem chatbot dibangun pada instant messaging Telegram dengan data pengetahuan kalender Bali khususnya Dewasa Ayu dan memiliki basis percakapan Bahasa Indonesia. Pengujian algoritma Jaro-Winkler memberikan hasil nilai rata-rata similarity sebesar 95,21% dan tingkat akurasi dalam pemberian saran kata sebesar 76%. Hasil pengujian chatbot dengan perbaikan kata memperoleh hasil akurasi sebesar 96% dan sistem chatbot tanpa menggunakan fitur perbaikan kata memperoleh hasil akurasi sebesar 36%. Fitur perbaikan kata meningkatkan akurasi chatbot dalam memberikan respon sebesar 60%.

Kata kunci: Jaro-Winkler, Perbaikan Kata, Chatbot

Abstract

User errors when typing words in messages greatly affect the accuracy of the chatbot system in providing appropriate reply messages. This problem can be resolved by adding a spelling correction feature to the chatbot system so that the chatbot can understand. This research uses the Jaro-Winkler algorithm to create spelling correction feature. The selection of the Jaro-Winkler algorithm is based on processing speed and effectiveness in comparing short strings so that it is suitable for correcting wrong words. The chatbot system was built on Telegram instant messaging with Balinese calendar knowledge data, especially Dewasa Ayu and has an Indonesian conversation base. The Jaro-Winkler algorithm test gives an average similarity value of 95.21% and an accuracy rate of giving word suggestions of 76%. The results test of the chatbot with spelling correction obtained an accuracy of 96%, while the chatbot system without using the spelling correction feature obtained an accuracy of 36%. The spelling correction feature has been shown to increase the accuracy of giving chatbot responses by 60%.

Keywords : Jaro-Winkler, Spelling Correction, Chatbot

1. Introduction

Chatbot merupakan sistem perangkat lunak yang dibuat untuk berinteraksi dengan manusia melalui *chat*. Chatbot dapat membuat percakapan sederhana berdasarkan aturan pada sistem yang kompleks. Chatbot dirancang untuk mensimulasikan sebuah percakapan atau komunikasi yang interaktif kepada pengguna melalui bentuk teks, audio, maupun video. Respons yang dihasilkan merupakan hasil pemindaian kata kunci pada *input* yang dilakukan oleh pengguna dan menghasilkan respons yang dianggap paling cocok, sehingga percakapan yang terjadi seakan-akan dilakukan oleh dua pribadi manusia yang saling berkomunikasi.

Pengembangan sistem *chatbot* memiliki banyak tantangan agar *chatbot* mampu berkomunikasi secara interaktif, salah satunya adalah penanganan kesalahan pengguna dalam

mengetikkan kata pada pesan. Sistem *chatbot* sangat bergantung pada ketepatan penulisan kata yang terdapat pada pesan yang dikirimkan oleh pengguna. Kesalahan kata pada pesan akan sangat mempengaruhi hasil pencocokkan dengan data frasa pengetahuan sistem *chatbot*. Hal ini tentu juga akan mempengaruhi ketepatan sistem *chatbot* dalam memberikan respon pada pesan yang dikirimkan.

Perbaikan kata dapat dilakukan dengan menggunakan *similarity measure*. *Similarity Measure* adalah proses pengukuran kemiripan suatu objek terhadap objek acuan. Beberapa jenis *similarity measure* yang biasa digunakan antara lain *Distance-Based Similarity Measure*, *Probabilistic-Based Similarity Measure*, *Set-Based Similarity Measure*, *Feature-Based Similarity Measure* dan *Context-Based Similarity Measure* [1]. Penelitian sebelumnya menggunakan kombinasi metode Jaccard Index dan N-gram untuk mengoreksi kata berbahasa Indonesia dengan memperoleh hasil terbaik pada Bi-gram [2]. Penelitian lainnya melakukan perbandingan metode Jaro-Winkler dan Levenshtein untuk melakukan mendeteksi plagiarisme pada dokumen teks dengan memperoleh hasil Jaro Winkler lebih baik dengan nilai rata-rata keakuratan sebesar 80.92% dan waktu pemrosesan selama 0.054 detik dibandingkan Levenshtein dengan nilai-rata keakuratan sebesar 49.43% dan waktu pemrosesan selama 0.138 detik [3]. Penelitian lain yang membandingkan metode Jaro-Winkler dan Cosine Similarity untuk mendeteksi plagiarisme menyatakan metode Jaro-Winkler lebih unggul [4].

Penelitian ini berfokus pada pengukuran pengaruh perbaikan kata dengan menggunakan metode Jaro-Winkler dalam meningkatkan akurasi sistem *chatbot* ketika memproses pesan yang diterima. Algoritma Jaro-Winkler dipilih karena algoritma Jaro-Winkler dapat bekerja lebih cepat dari algoritma *edit distance* dan memiliki kompleksitas waktu *quadratic runtime complexity* yang sangat efektif pada string pendek [5]. Proses perbaikan kata pada *chatbot* dilakukan dengan membandingkan satu persatu kata pada pesan dengan kumpulan kata pada kamus kata sehingga algoritma Jaro-Winkler merupakan salah satu opsi pilihan algoritma yang tepat. Sistem *chatbot* dibangun pada *instant messaging* Telegram dengan data pengetahuan tentang kalender Bali khususnya *Dewasa Ayu* dan berbasis Bahasa Indonesia.

2. Research Method

Penelitian ini dilakukan dengan melalui serangkaian tahapan analisis permasalahan, pengumpulan data melalui studi pustaka, membuat perancangan sistem, dan uji coba rancangan sistem. Berikut merupakan rancangan sistem perbaikan kata pada sistem *chatbot* yang ditampilkan dalam diagram alur.

2.1. Data

Data yang digunakan sebagai kamus kata pada proses pencocokan kata adalah data kata dasar yang diperoleh dari pustaka Sastrawi [6]. Data yang diperoleh berjumlah 30.342 kata dasar. Data uji yang digunakan berjumlah 50 kata dan bersumber dari kamus kata yang telah dimodifikasi dengan menambahkan, mengurangi, dan mengganti beberapa karakter kata sehingga akan berbeda dengan kata aslinya. Tabel 1 menampilkan beberapa data kata dasar yang digunakan sebagai kamus kata pada sistem perbaikan kata serta data uji dari kata tersebut.

Tabel 1. Data Kamus Kata Beserta Padanan Kata Uji

Kamus Kata	Kata Uji	Modifikasi Karakter
keliling	kelilng	penghapusan
fundamental	fndamental	penghapusan
absorb	absorvb	penambahan
mikrolinguistik	mikrolinhuistil	penggantian
partisipan	partisipamna	penambahan

2.2. Diagram Alur Umum Sistem

Diagram alur umum sistem merupakan diagram yang menjelaskan bagaimana rangkaian proses sistem *chatbot* yang terdapat fitur perbaikan kata dibuat. Gambar 1 memperlihatkan bagaimana diagram alur umum sistem yang digunakan pada penelitian ini.



Gambar 1. Diagram Alur Umum Sistem

Diagram alur umum sistem terdiri dari lima proses yaitu penerimaan pesan masuk, *preprocessing* pesan, perbaikan kata, pencocokan *intent*, dan pengiriman pesan balasan. Proses penerimaan pesan merupakan proses sistem menerima pesan masuk yang dikirimkan pengguna melalui *instant messaging* (Telegram). Tahapan selanjutnya adalah *preprocessing* pesan yang merupakan tahapan awal untuk mengubah pesan menjadi data yang sesuai dengan format pemrosesan yang diperlukan pada tahapan selanjutnya. Tahapan perbaikan kata merupakan proses penggantian kata-kata pada pesan yang terdeteksi mengalami kesalahan dalam pengetikan. Proses pencocokan *intent* bertujuan untuk mencocokkan pesan dengan pola percakapan yang diketahui *chatbot* sehingga memperoleh pesan balasan yang diinginkan pengguna. Tahap terakhir adalah pengiriman pesan balasan yang diperoleh ke pengirim pesan.

2.3. Diagram Alur *Preprocessing* Pesan

Diagram alur *preprocessing* pesan merupakan diagram yang menjelaskan bagaimana rangkaian pemrosesan tahap awal sebuah pesan sehingga menjadi data yang sesuai dengan kebutuhan pada tahap pemrosesan selanjutnya. Gambar 2 memperlihatkan bagaimana diagram alur *preprocessing* pesan yang digunakan pada penelitian ini.

Diagram alur *preprocessing* pesan terdiri dari empat proses yaitu *case folding*, tokenisasi, *stop word*, dan *stemming*. Tahapan *case folding* bertujuan untuk mengubah semua huruf pada teks menjadi huruf kecil dan menghilangkan karakter yang tidak diperlukan. Tahapan tokenisasi adalah proses memecah aliran teks menjadi kata, frasa, simbol, atau elemen bermakna lainnya yang disebut token. Tahap *stop word* merupakan tahapan untuk menghilangkan kata yang sangat sering berulang pada dokumen yang tidak memiliki makna

berarti karena hanya sebagai penghubung kata-kata dalam sebuah kalimat. Tahap *stemming* adalah sebuah proses menggabungkan bentuk varian kata ke dalam representasi umum kata tersebut, yaitu suku kata.



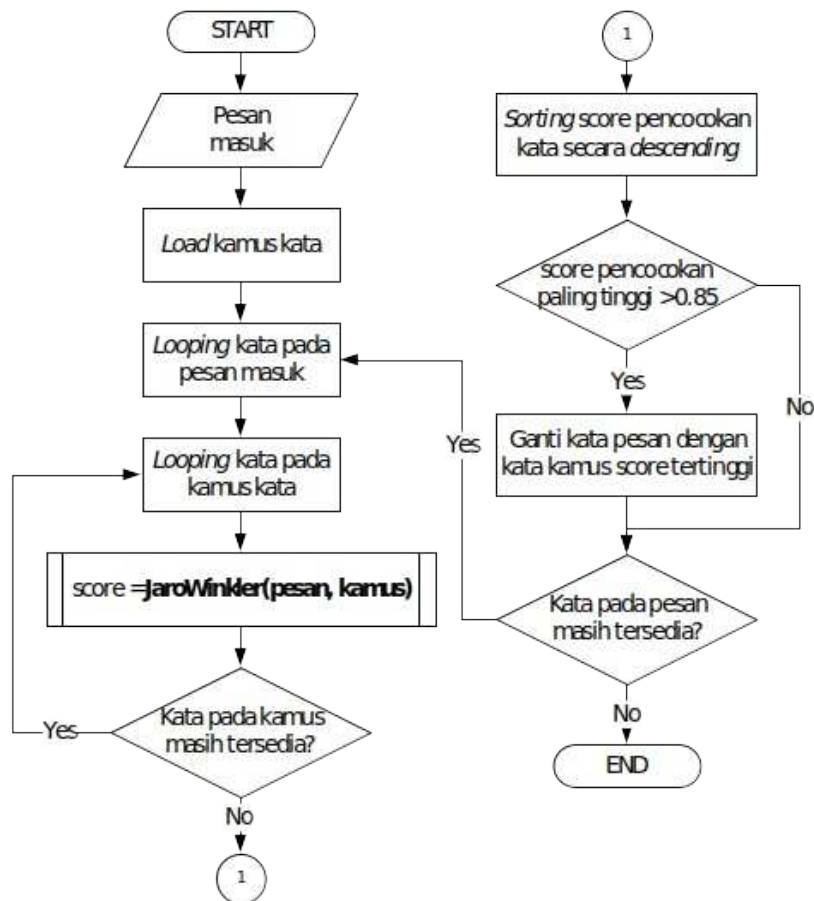
Gambar 2. Diagram Alur *Preprocessing* Pesan

2.4. Diagram Alur Proses Perbaikan Kata

Diagram alur proses perbaikan kata merupakan diagram yang menjelaskan bagaimana rangkaian pemrosesan perbaikan kata pada pesan yang terdeteksi mengalami kesalahan dalam penulisannya. Gambar 3 memperlihatkan bagaimana diagram alur perbaikan kata yang digunakan pada penelitian ini.

Diagram alur perbaikan kata salah tersebut dapat dijelaskan sebagai berikut:

- a. Masukan pada fungsi perbaikan kata salah adalah pesan masuk dari pengguna yang diterima oleh *chatbot*.
- b. Sistem kemudian akan memuat kamus kata yang benar pada dokumen KamusKata.txt.
- c. Sistem kemudian akan membandingkan kata pada pesan masuk dengan semua kata pada kamus kata benar yang telah dimuat sebelumnya. Proses perbandingan dilakukan dengan menggunakan metode JaroWinkler.
- d. Hasil perbandingan suatu kata pesan masuk dengan semua kata pada kamus kata benar kemudian diurutkan berdasarkan nilai kemiripan yang paling tinggi.
- e. Apabila kata dengan nilai kemiripan tertinggi memiliki nilai lebih dari 0.85 (*threshold*), maka kata tersebut digunakan sebagai pengganti kata yang salah pada pesan masuk.
- f. Poin (c) akan terus dilakukan oleh sistem hingga seluruh kata pada pesan masuk telah dicocokkan dengan kamus kata benar, sehingga proses perbaikan kata salah telah selesai dilakukan.



Gambar 3. Diagram Alur Perbaikan Kata

3. Literature Study

Kajian pustaka yang digunakan pada penelitian ini bersumber dari jurnal, laporan penelitian, dan situs internet yang mencakup konsep dan teori tentang metode yang terkait pada pengembangan sistem perbaikan kata pada penelitian ini.

3.1. Sastrawi

Sastrawi adalah pustaka PHP sederhana yang memungkinkan untuk mengurangi kata-kata infleksi dalam Bahasa Indonesia menjadi bentuk dasarnya (*stemming*). Algoritma yang digunakan pada pustaka ini adalah algoritma *stemming* Nazief dan Adriani [7] dan beberapa algoritma peningkatan dalam penanganan teks dan *stemming* [8], [9]. Terlepas dari kesederhanaannya, perpustakaan ini dirancang dengan kualitas tinggi dan didokumentasikan dengan baik. Sastrawi dapat digunakan untuk melakukan tahapan umum dari preprocessing teks yaitu *case folding*, *tokenisasi*, *stop word*, dan *stemming*. Pustaka sastrawi mudah diintegrasikan dengan *framework / package* lainnya dan memiliki API yang sederhana dan tentunya mudah untuk digunakan. Versi PHP yang diperlukan untuk menggunakan pustaka Sastrawi PHP adalah PHP ≥ 5.3 dan juga dapat berjalan pada HHVM. Pustaka ini juga tersedia pada bahasa pemrograman Java (Jsastrawi), C (cSastrawi), Python (PySastrawi), Go (Go-Sastrawi), dan Ruby (Sastrawi-Ruby) [6].

3.2. Jaro Index

Jaro Index merupakan metode untuk membandingkan kemiripan antara dua *string* dengan memperhitungkan penyisipan, penghapusan, dan transposisi karakter pada kedua *string* [10]. Algoritma Jaro secara umum berdasarkan pada tiga proses yaitu:

- menghitung panjang *string*,
- menemukan jumlah karakter umum dalam dua *string*, dan

- c. menemukan jumlah transposisi.

Definisi karakter umum dalam dua string adalah karakter yang sama pada kedua string dan harus berada dalam jarak setengah dari panjang *string* yang lebih pendek. Definisi transposisi adalah karakter dari satu *string* yang tidak sama dengan karakter umum yang sesuai dari *string* lainnya. Nilai kemiripan dua *string* pada metode Jaro adalah sebagai berikut.

$$Jaro(s_1, s_2) = \frac{1}{3} \cdot \left(\frac{N_C}{len_{s_1}} + \frac{N_C}{len_{s_2}} + \frac{N_C - 0.5 N_t}{N_C} \right) \quad (1)$$

Persamaan 1 merupakan perhitungan menggunakan metode Jaro Index. Penjelasan persamaan tersebut antara lain s_1 dan s_2 adalah *string* dengan panjang masing-masing len_{s_1} dan len_{s_2} , N_C adalah jumlah karakter umum antara *string* s_1 dan s_2 dengan jarak maksimal setengah dari panjang minimum string s_1 dan s_2 , dan N_t adalah jumlah transposisi.

Contoh perhitungan kemiripan dua buah *string* dengan menggunakan metode Jaro adalah sebagai berikut: $s_1 = \text{"TRACE"}$, $s_2 = \text{"CRATE"}$, $len_{s_1} = 5$, $len_{s_2} = 5$, $N_C = 3$ ('R', 'A', 'E'), dan $N_t = 0$. Jika diperlihatkan dalam persamaan 2.1 maka akan menjadi sebagai berikut.

$$Jaro(TRACE, CRATE) = \frac{1}{3} \cdot \left(\frac{3}{5} + \frac{3}{5} + \frac{3 - 0.5 \cdot 0}{3} \right) = 0.733$$

Gambar 4. Contoh Hasil Perhitungan Jaro Index

Gambar 4 merupakan hasil perhitungan metode Jaro terhadap string "TRACE" dan "CRATE". Hasil dari perhitungan rumus Jaro terhadap kedua *string* tersebut menghasilkan skor kemiripan sebesar 0,733.

3.3. Jaro-Winkler Index

Jaro-Winkler Index merupakan sebuah metode varian dari Jaro-Index. Winkler (1990a) pada penelitiannya menunjukkan bagaimana varian komparator *string* Jaro Φ secara dramatis meningkatkan hasil pencocokan dibandingkan dengan situasi ketika tanpa menggunakan pembandingan *string*. Varian metode Jaro ini menggunakan beberapa ide Pollock dan Zamora (1984) dalam sebuah studi besar untuk Layanan Abstrak Kimia. Penelitian tersebut memberikan bukti empiris yang mengkuantifikasi bagaimana probabilitas kesalahan *keypunch* meningkat ketika posisi karakter dalam *string* bergerak dari kiri ke kanan [10]. Nilai kemiripan dua *string* pada metode Jaro-Winkler adalah sebagai berikut.

$$i_w = i_j + lp(1 - i_j) \quad (2)$$

Persamaan 2 merupakan perhitungan menggunakan metode Jaro-Winkler Index. Penjelasan persamaan tersebut antara lain i_j adalah hasil kemiripan dua *string* dengan metode Jaro, l adalah panjang *prefix* karakter umum dari awal *string* dengan maksimal 4 karakter, dan p merupakan konstanta faktor penskalaan untuk seberapa banyak jumlah skor disesuaikan ke atas karena memiliki awalan yang sama. Nilai p tidak boleh melebihi 0.25 agar hasil kemiripan tidak melebihi nilai l . Standar nilai untuk konstanta ini pada metode Winkler adalah $p = 0.1$.

Contoh perhitungan kemiripan dua buah string dengan menggunakan metode Jaro-Winkler dengan kata "TRACES" dan "TRATES" adalah sebagai berikut: $i_j = 0.889$, $l = 0.1$, dan $p = 3$. Jika diperlihatkan dalam Persamaan 2 maka akan menjadi sebagai berikut.

$$i_w = 0.889 + 0.1 \cdot 3(1 - 0.889) = 0.923$$

Gambar 5. Contoh Hasil Perhitungan Jaro-Winkler Index

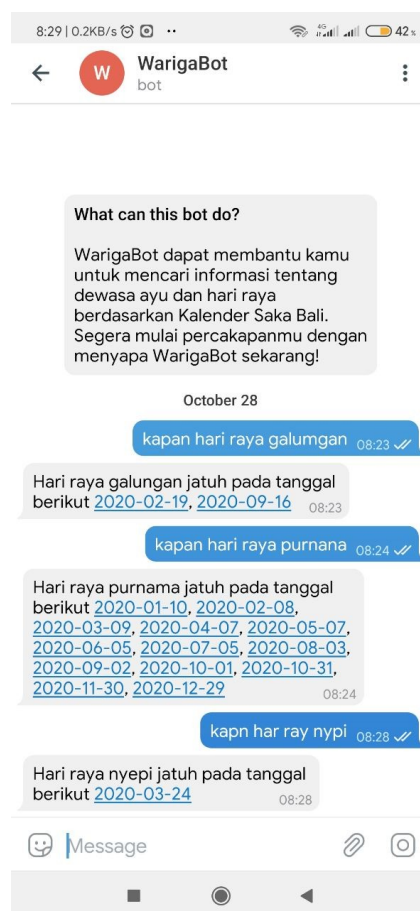
Gambar 5 merupakan hasil perhitungan metode Jaro-Winkler terhadap *string* "TRACES" dan "TRATES". Hasil dari perhitungan rumus Jaro-Winkler terhadap kedua *string* tersebut menghasilkan *index* dengan skor sebesar 0,923.

4. Result and Discussion

Bagian ini berisi tentang hasil dan pembahasan dari perancangan sistem perbaikan kata yang telah dilakukan pada penelitian ini.

4.1. Perbaikan Kata pada Chatbot

Hasil implementasi rancangan perbaikan kata dengan metode Jaro-Winkler berhasil dilakukan pada sistem *chatbot*. Implementasi perbaikan kata pada sistem *chatbot* ditampilkan pada Gambar 6 berikut.



Gambar 6. Hasil Implementasi Perbaikan Kata pada Chatbot

Perbaikan kata pada sistem *chatbot* dapat berjalan dengan baik. Gambar 6 menampilkan sistem chatbot mampu memahami kata-kata yang penulisannya salah pada pesan. Pesan pertama "kapan hari raya galumgan" terdapat satu kesalahan kata pada kata "galumgan" yang seharusnya "galungan". Pesan kedua "kapan hari raya purnana" terdapat satu kesalahan kata pada kata "purnana" yang seharusnya "purnama". Pesan ketiga "kapn har ray nypi" terdapat empat kesalahan kata yaitu pada kata "kapn" yang seharusnya "kapan", kata "har" yang seharusnya "hari", kata "ray" yang seharusnya "raya", dan kata "nypi" yang seharusnya "nyepi". Ketiga pesan yang mengandung kesalahan kata dapat diproses dengan baik oleh *chatbot* dibuktikan dengan memberikan balasan yang sesuai dengan pesan yang diberikan.

4.2. Hasil Pengujian

Pengujian pada penelitian ini dilakukan dengan dua tahap pengujian. Pengujian yang pertama yaitu pengujian algoritma Jaro-Winkler untuk perbaikan kata yang membandingkan data uji dengan kamus kata seperti yang ditampilkan pada Tabel 1. Pengujian kedua adalah pengujian untuk mengukur pengaruh perbaikan kata pada sistem *chatbot* dalam memproses pesan.

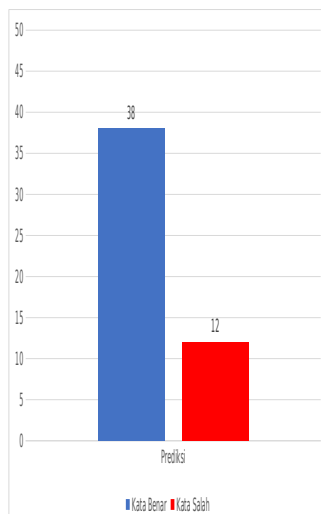
4.2.1. Pengujian Algoritma Jaro-Winkler

Pengujian algoritma Jaro-Winkler dilakukan dengan membandingkan data uji sebanyak 50 kata yang telah dimodifikasi dengan 30.342 kata dasar pada kamus kata. Pengujian dilakukan untuk mengukur tingkat kemiripan dan akurasi pencocokan kata. Beberapa hasil pengujian ditampilkan pada Tabel 2 berikut.

Tabel 2. Hasil Pengujian Tingkat Kemiripan dan Akurasi

Kata Uji	Kamus Kata	Kata Prediksi	Nilai Kemiripan(%)	Ketepatan
keliling	keliling	keliling	97,5	benar
fundamental	fundamental	fundamental	97,27	benar
absorvb	absorb	absorb	97,14	benar
mikrolinhuistil	mikrolinguistik	mikrolinguistik	94,66	benar
partisipamna	partisipasi	partisipasi	96,66	benar
karwt	karet	kartu	90,66	salah
beolang	belolang	belang	96,19	salah
kopu	kopi	kop	94,16	salah
tangi	tangis	tangsi	96,66	salah
helsentrik	heliosentrik	eksentrik	89,62	salah

Hasil pengujian dari 50 kata uji yang telah dilakukan memperoleh rata-rata nilai kemiripan kata sebesar 95,21%. Ketepatan prediksi kata yang dihasilkan berjumlah 38 kata prediksi yang benar dan 12 kata prediksi yang salah dari 50 kata uji dengan tingkat akurasi sebesar 76%.



Gambar 7. Hasil Ketepatan Perbaikan Kata

4.2.2. Pengujian Perbaikan Kata pada Chatbot

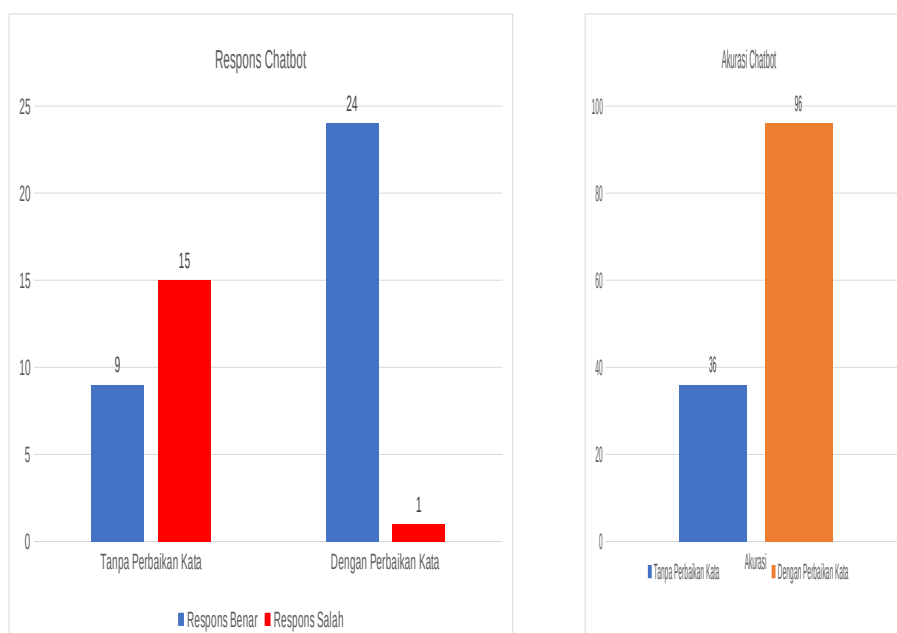
Pengujian perbaikan kata pada *chatbot* dilakukan dengan membandingkan data uji sebanyak 25 pesan yang terdapat beberapa kesalahan dalam penulisan kata. Pengujian dilakukan untuk mengukur pengaruh perbaikan kata pada sistem *chatbot* dengan membandingkan hasil akurasi pemberian respons sistem *chatbot* tanpa

perbaikan kata dengan sistem *chatbot* yang memiliki fitur perbaikan kata. Beberapa hasil pengujian ditampilkan pada Tabel 3 berikut.

Tabel 3. Hasil Pengujian Tingkat Kemiripan dan Akurasi

<i>Chatbot</i>	Jumlah Respons Benar	Jumlah Respons Salah	Akurasi
Tanpa Perbaikan Kata	9	15	36%
Dengan Perbaikan Kata	24	1	96%

Hasil pengujian dari 25 pesan yang telah dilakukan menyatakan sistem *chatbot* yang tidak menggunakan perbaikan kata memberikan 9 respons benar dan 15 respons salah. Pengujian sistem *chatbot* dengan perbaikan kata memberikan 24 respons benar dan 1 respons salah. Tingkat akurasi dari sistem *chatbot* yang tidak menggunakan perbaikan kata adalah sebesar 36% dan sistem *chatbot* yang menggunakan perbaikan kata sebesar 96%.

Gambar 8. Hasil Pemberian Respons dan Akurasi *Chatbot*

5. Conclusion

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%.

Sistem perbaikan kata ini diharapkan dapat dikembangkan dengan menggunakan dengan menggunakan metode selain *string matching* sehingga dapat diketahui kelebihan dan kekurangan dari masing-masing metode.

References

- [1] D. Nugraheny, "Metode Nilai Jarak Guna Kesamaan atau Kemiripan Ciri Suatu Citra (Kasus Deteksi Awan Cumulonimbus Menggunakan Principal Component Analysis)," *Angkasa J. Ilm. Bid. Teknol.*, vol. 7, no. 2, p. 21, Sep. 2017, doi: 10.28989/angkasa.v7i2.145.

- [2] A. Indriani, Muhammad, Suprianto, and Hadriansa, "Implementasi Jaccard Index dan N-Gram pada Rekayasa Aplikasi Koreksi Kata Berbahasa Indonesia," *Sebatik* 1410-3737, pp. 95–101, 2018.
 - [3] M. J. Tannga, S. Rahman, T. Informatika, and A. L. Distance, "Analisis Perbandingan Algoritma Levenshtein Distance dan Jaro Winkler Untuk Aplikasi Deteksi Plagiarisme Dokumen Teks," *J. Technol. Res. Inf. Syst. Inf. Technol.*, vol. 4, no. 1, 2017.
 - [4] T. Tinaliah and T. Elizabeth, "Perbandingan Hasil Deteksi Plagiarisme Dokumen dengan Metode Jaro-Winkler Distance dan Metode Latent Semantic Analysis," *J. Teknol. dan Sist. Komput.*, vol. 6, no. 1, pp. 7–12, Jan. 2018, doi: 10.14710/jtsiskom.6.1.2018.7-12.
 - [5] A. Kurniawati, S. Puspitodjati, and S. Rahman, "Implementasi Algoritma Jaro-Winkler Distance untuk Membandingkan Kesamaan Dokumen Berbahasa Indonesia," *Proceeding, Semin. Ilm. Nas. Komput. dan Sist. Intelijen KOMMIT 2008, Depok, Indones.*, 2010.
 - [6] A. Librian, "Sastrawi." [Online]. Available: <https://github.com/sastrawi/sastrawi>.
 - [7] M. Adriani, J. Asian, B. Nazief, S. M. M. Tahaghoghi, and H. E. Williams, "Stemming Indonesian: A Confix-Stripping Approach," *ACM Trans. Asian Lang. Inf. Process.*, vol. 6, no. 4, pp. 1–33, Dec. 2007, doi: 10.1145/1316457.1316459.
 - [8] J. Asian, "Effective Techniques for Indonesian Text Retrieval," RMIT University, 2007.
 - [9] A. Z. Arifin, I. P. A. K. Mahendra, and H. T. Ciptaningtyas, "Enhanced Confix Stripping Stemmer and Ants Algorithm For Classifying News Document in Indonesian Language," in *International Conference on Information & Communication Technology and Systems*, 2009, pp. 149–157.
 - [10] W. E. Winkler, "Overview of Record Linkage and Current Research Directions," Washington, 2005.
-