

# Stemming Kata Berimbuhan Tidak Baku Bahasa Indonesia Menggunakan Algoritma Jaro-Winkler Distance

## *Stemming Non-Formal Affix Word Indonesian Language Using Jaro-Winkler Distance Algorithm*

**Mudawil Qulub<sup>\*1</sup>, Ema Utami<sup>2</sup>, Andi Sunyoto<sup>3</sup>**

<sup>1,2,3</sup>Magister Informatika, Universitas AMIKOM Yogyakarta

E-mail: <sup>\*1</sup>mudawil.qulub@students.amikom.ac.id, <sup>2</sup>ema.u@amikom.ac.id,  
<sup>3</sup>andi@amikom.ac.id

### **Abstrak**

Bahasa merupakan media untuk mengekspresikan keinginan, gagasan, dan perasaan. Dalam kaitannya dengan bahasa Indonesia, bahasa dapat mencerminkan perilaku dalam sebuah masyarakat. Bahasa Indonesia dapat digunakan secara formal maupun non-formal. Bahasa non-formal berbentuk kata tidak baku yang biasanya digunakan dalam komentar atau posting dalam media sosial. Salah satu contoh kata tidak baku adalah "nemenin" yang kata formalnya "menemani" dan kata dasarnya adalah "teman". Dalam penelitian ini akan dilakukan pengujian akurasi algoritma Jaro-Winkler distance dalam mengubah kata imbuhan tidak baku menjadi bentuk dasarnya, proses ini disebut dengan stemming. Penelitian ini menggunakan data sebanyak 60 kata berimbuhan tidak baku. Hasil pengujian menunjukkan tingkat akurasi algoritma Jaro-Winkler sebesar 85% atau 51 kata berhasil di-stemming (3 overstemming, 6 unstemming, 0 understemming).

**Kata Kunci** — Kata Imbuhan Tidak Baku, Stemming, Jaro-Winkler Distance

### **Abstract**

Language that represents the media to attract interest, regulate, and feel. In reversing it with Indonesian, language can reflect behavior in a society. Indonesian can be used formally or informally. Non-formal language in the form of non-formal words used in comments or posts on social media. One example of a non-formal word is "accompanying" the formal word "accompany" and the basic word is "friend". In this research, the Jaro-Winkler algorithm will be tested the distance in changing non-formal affix words into a form of interaction, this process is called stemming. This research uses data as many as 60 words that are not standardized. The test results show the level of testing of the Jaro-Winkler algorithm is 85% or 51 words successfully stemmed (3 overstemming, 6 unstemming, 0 understemming).

**Keywords** — Non-Formal Affix, Stemming, Jaro-Winkler Distance

## 1. PENDAHULUAN

Bahasa digunakan oleh seseorang untuk mengekspresikan keinginan, gagasan, dan perasaan. Selain sebagai media ekspresi personal, bahasa juga bisa menjadi media ekspresi masyarakat. Dalam kaitannya dengan bahasa Indonesia, pemakaian bahasa oleh seseorang atau sekelompok orang dapat mencerminkan perilaku kebahasaan masyarakat Indonesia. Pemakaian bahasa Indonesia yang baik dan benar ditentukan oleh ragam pemakaian dengan situasi dan kondisi serta bidang kehidupan tertentu [1]. Bahasa Indonesia dapat digunakan secara formal dan non-formal. Bahasa non-formal biasanya digunakan disaat kondisi sedang santai seperti saat berkomentar di sosial media, *posting*, dan *chatting*. Salah satu contohnya adalah “Lu udah nemenin emak belanja”, pada kalimat ini ditemukan kata “nemenin” yang bentuk formalnya adalah “menemani” dengan kata dasar “teman”. Twitter merupakan media sosial yang menyediakan banyak sekali data yang mudah didapat, dimana masyarakat Indonesia kebanyakan menggunakan kata tidak baku dalam setiap komentar dan *post* mereka [2].

*Stemming* adalah proses mengubah kata menjadi kata dasarnya. *Stemming* biasanya digunakan pada IR (*Information Retrieval*), peningkatan performa pencarian, dan deteksi *plagiarism*, dan sistem pemrosesan bahasa alami untuk meningkatkan kinerjanya. *Stemming* dapat meningkatkan teks karena proses ini akan menggabungkan kata-kata yang memiliki kata dasar yang sama. Jika kata-kata memiliki kata dasar yang sama, maka mereka dianggap memiliki kemiripan makna [3]. *Stemming* pada kata tidak baku dapat digunakan pada *chat bot*, *sentiment analysis*, dan pemrosesan teks yang lainnya [4].

Penelitian tentang *stemming* bahasa Indonesia dengan menggunakan algoritma *Flexibility Affix Classification* dan membandingkannya dengan algoritma *stemming* Nazief & Adriani. Penelitian ini melakukan percobaan dengan menggunakan data bahasa Indonesia sebanyak 1.704 teks dokumen dengan 255.182 tokens, dan 3648 kata stemmed. Penelitian ini menghasilkan pendekatan kinerja yang lebih baik dari *confix-stripping* dari Nazief & Adriani. Kelebihan dari penelitian ini adalah cara mudah dalam menambahkan imbuhan dengan algoritma *Flexibility Affix Classification* dalam *stemming* bahasa Indonesia. Kekurangan penelitian algoritma ini yaitu tidak bisa melakukan *stemming* kata reduplikasi bahasa Indonesia [5].

Penelitian lainnya tentang *stemming* kata tidak baku bahasa Indonesia dengan memodifikasi algoritma *Non-Formal Affix* menggunakan pendekatan *Levenstein distance*. Pada penelitian ini menggunakan data kata tidak baku sebanyak 60 kata, menghasilkan tingkat akurasi sebesar 96,6% atau 58 kata berhasil di-*stemming* [2].

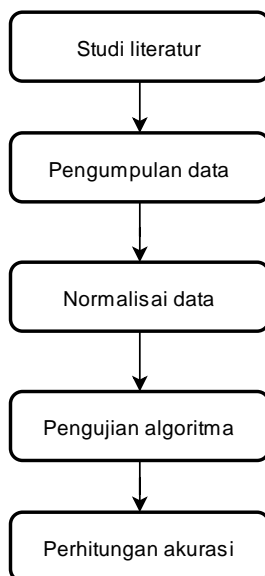
Penelitian ketiga yang relevan yaitu penelitian algoritma Nazief & Adriani digunakan sebagai *retrieval system* dalam Al Hadith dalam bahasa Indonesia oleh Atqia Aulia, Dewi Khairani, dan Nashruel Hakim [6]. Pada penelitian ini *stemming* dengan menerapkan algoritma Nazief & Adriani pada bahasa pemrograman PHP untuk menampilkan hasil pencarian hadith. Analisis hasil uji menunjukkan bahwa repositori menggunakan *stemming* menghasilkan penarikan dengan skor sempurna 1 dan presisi 0,961. Ini menunjukkan bahwa repositori dapat mengembalikan sejumlah besar dokumen yang relevan.

Berdasarkan uraian diatas, pada penelitian ini peneliti mengusulkan penelitian dengan menggunakan algoritma Jaro-Winkler distance untuk proses *stemming* kata berimbuhan tidak baku dalam bahasa Indonesia.

## 2. METODE PENELITIAN

### 2.1. Tahapan Penelitian

Penelitian ini berfokus pada pengujian akurasi algoritma Jaro-Winkler distance dalam melakukan stemming kata tidak baku pada bahasa Indonesia. Tahapan penelitian dapat dilihat pada Gambar 1.

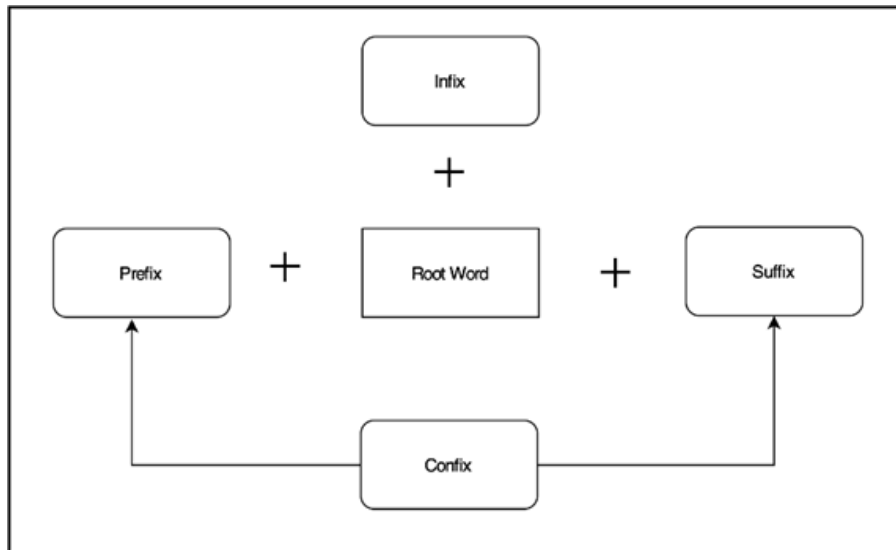


Gambar 1. Tahapan Penelitian

- Studi literature dilakukan untuk memperdalam pengetahuan serta mencari landasan teori mengenai struktur imbuhan bahasa Indonesia, imbuhan tidak baku bahasa Indonesia, *stemming*, dan algoritma *Jaro-Winkler distance*.
- Data pada penelitian jurnal ini merupakan data yang sekunder yang diambil dari penelitian sebelumnya yang berjudul “Optimalisasi *stemming* Kata Berimbuhan Tidak Baku Pada Bahasa Indonesia Dengan *Levenshtein Distance*” [2].
- Normalisasi data bertujuan untuk membuat data lebih kompetibel ketika diterapkan pada bahasa pemrograman.
- Pengujian dilakukan dengan data yang sudah didapatkan pada aplikasi *stemming* yang sudah diterapkan algoritma *Jaro-Winkler distance*.
- Perhitungan akurasi dilakukan dengan menggunakan metode *confusion matrix*.

### 2.2. Struktur Imbuhan Bahasa Indonesia

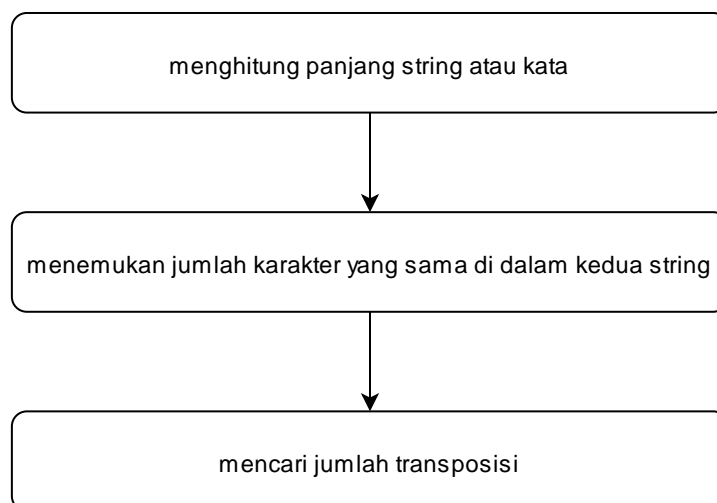
Imbuhan bahasa Indonesia terdiri dari prefix (awalan), infiks, suffix (akhiran), dan confix. Imbuhan (affix) adalah morfeme yang ditambahkan ke kata untuk membuat kata baru. Prefix (awalan) adalah imbuhan yang ditambahkan di awal kata. Infix adalah imbuhan yang dimasukkan ke dalam kata. Suffix (akhiran) adalah imbuhan yang ditambahkan di akhiran kata. Sementara itu, confix adalah kombinasi di awalan dan akhiran kata. Pada bahasa Indonesia, imbuhan dapat ditambahkan pada kata kerja, kata sifat, atau kata benda [5]. Gambar 2 mengilustrasikan konsep imbuhan dalam bahasa Indonesia.



Gambar 2. Konsep Imbuhan Bahasa Indonesia

- a. Prefix  
Prefix adalah imbuhan yang ditambahkan ke awal kata untuk membentuk kata yang baru. Menurut S. Takdir Alisjahbana dalam Khotimah awalnya *di-*, *ke-*, *ter-* memiliki fungsi dalam menyatakan tempat dalam bentuk pasif [7]. Contoh *ke-* (kedepan), *ter-* (terdeteksi), *di-* (diakses), dan lain-lain.
- b. Suffix  
Suffix adalah imbuhan yang diletakkan pada belakang kata dasar [7]. Ada tiga jenis suffix yaitu *-kan*, *-an*, *-i*. Contoh kata yaitu *-kan* (tawarkan), *-an* (makanan), *-i* (manusiawi), akhiran *wi* dalam kata manusiawi merupakan bunyi dari suffix *-i*.
- c. Infix  
Infix merupakan imbuhan yang diletakkan pada kata dasar [7]. *-em* (gemetar), *-er* (gerigi), *-el* (gelembung).
- d. Confix  
Confix adalah imbuhan yang diletakkan pada awal dan akhir sebuah kata. Contoh *me-kan* (merahasiakan), *per-an* (perkuliahan), dan *ber-an* (berpakaian).

### 2.3. Algoritma Jaro-Winkler Distance



Gambar 3. Komponen Dasar Algoritma Jaro-Winkler Distance

Algoritma Jaro-Winkler distance adalah algoritma yang menghitung kesamaan antara dua string [8]. Semakin tinggi nilai Jaro-Winkler distance untuk kedua string maka semakin tinggi tingkat kemiripan kedua string tersebut. Nilai normalnya adalah 0 menandakan tidak ada persamaan dan nilai 1 untuk menandakan adanya kesamaan [9]. Algoritma Jaro-Winkler distance memiliki tiga komponen dasar ditunjukkan oleh Gambar 3.

Pada algoritma Jaro-Winkler distance, jarak antara dua kata dihitung dengan menggunakan persamaan (1).

$$Jaro(s1, s2) = \frac{1}{3} \left( \frac{m}{|s1|} + \frac{m}{|s2|} + \frac{m-t}{m} \right) \quad (1)$$

Nilai  $m$  merupakan jumlah karakter yang dari kedua *string*. Suatu karakter dapat dikatakan memiliki kesamaan jika memiliki jarak teoritis yang tidak melebihi persamaan (2).  $s1$  merupakan panjang *string* pertama, sedangkan  $s2$  merupakan panjang *string* kedua dan  $t$  merupakan jumlah trasposisi atau jumlah *string* yang tertukar posisi.

$$\left[ \frac{\max(|s1|, |s2|)}{2} \right] - 1 \quad (2)$$

Jaro-Winkler distance menggunakan *prefix scale* ( $p$ ) yang memberikan tingkat penilaian yang lebih, dan *prefix length* ( $l$ ) yang menyatakan panjang awalan yaitu panjang karakter *string* yang dibandingkan sampai ditemukannya ketidaksamaan. Apabila *string*  $s1$  dan  $s2$  dibandingkan maka Jaro-Winkler distance-nya ( $d_w$ ) dirumuskan persamaan (3).

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

Dimana  $d_w$  merupakan nilai Jaro-Winkler distance  $d_j$  adalah nilai Jaro-Winkler distance antara kedua. *String*  $l$  menunjukkan panjang *prefix* umum di awal *string*, dimana nilai maksimalnya adalah 4 karakter atau panjang karakter yang sama sebelum ditemukan perbedaan. Sedangkan  $p$  adalah konstanta *scaling factor*. Nilai standar pada konstanta ini menurut winkler adalah 0,1 [10].

### 2.3.1. Contoh Perhitungan Algoritma Jaro-Winkler Distance

Pada contoh perhitungan berikut ini diambil dari penelitin Prasetyo [11], misalkan pengguna ingin menulis kata “tidur”, namun terjadi kesalahan ketik sehingga tertulis kata “tdyr”. Pada kasus ini akan dilakukan perhitungan kedekatan jarak kata “tdyr” dengan semua kata yang jumlah karakternya empat sampai enam karakter pada database. Contoh kata “tidak”, “tiban”, “tidur”.

Perhitungan jarak kedekatan kata antara kata “tdyr” dengan “tiban” menggunakan algoritma Jaro-Winkler distance.

$$Jaro(s1, s2) = \frac{1}{3} \left( \frac{m}{|s1|} + \frac{m}{|s2|} + \frac{m-t}{m} \right)$$

$$string1 (s1) = tidyr$$

$$string2 (s2) = tiban$$

$$Jumlah\ karakter\ string\ 1\ (s1) = 5$$

$$Jumlah\ karakter\ string\ 2\ (s2) = 5$$

$$Jumlah\ karakter\ s1\ cocok\ dengan\ s2\ (m) = 2$$

$$Jumlah\ transposisi\ (t) = 0$$

$$Jaro(s_1)(s_2) = \frac{1}{3} \left( \frac{2}{5} + \frac{2}{5} + \frac{2-0}{2} \right)$$

$$Jaro(s_1)(s_2) = 0,6$$

$$Jaro - Winkler(s_1)(s_2) = Jaro(s_1, s_2) + (L * p(1 - jaro(s_1, s_2)))$$

$$Kesamaan awal (L) = 2$$

$$Faktor Penskalaan (p) = 0,1$$

$$Jaro - Winklar(s_1, s_2) = 0,6 + (2 * 0,1(1 - 0,6))$$

$$Jaro - Winklar(s_1, s_2) = 0,68$$

Sehingga dapat diketahui hasil perhitungan pada tabel 1.

Tabel 1. Analisis Komputasi

No	String 1	String 2	Nilai Jaro-Winkler
1	Tidyr	tiban	0.68
2	Tidyr	tidur	0.906
3	Tidyr	tidak	0.813

Dari hasil perhitungan di atas “tidyr” memiliki kemiripan karakter paling dekat dengan kata “tidur” dengan nilai Jaro-Winkler 0.906.

#### 2.4. Data Kata Berimbuhan Tidak Baku

Bahasa Indonesia terdiri dari dua jenis kata, yaitu kata baku dan tidak baku. Kata baku biasanya digunakan pada sesuatu yang bersifat formal seperti penulisan jurnal, laporan penelitian, dan lain sebagainya. Sedangkan kata tidak baku digunakan pada suasana tidak formal, misalnya berbicara santai dengan teman dekat, chatting, komentar atau posting di media sosial, dan yang lainnya. Pada penelitian ini akan digunakan data kata imbuhan tidak baku bahasa Indonesia yang diambil dari penelitian Rahardyan [2] dengan jumlah data yaitu 60 kata imbuhan tidak baku yang dapat dilihat pada Tabel 2.

Tabel 2. Kata Imbuhan Tidak Baku Bahasa Indonesia

No	Kata Dasar	Bentuk Tidak Baku
1	Terjang	nerjang
2	Tuduh	nuduh
3	Terima	nerima
4	Tegur	negur
5	pukul	mukul
6	pimpin	mimpin
7	coba	nyoba
8	siram	nyiram
9	suruh	nyuruh
10	simpan	nyimpen
11	sebrang	nyebrang
12	anggap	nganggap
13	amuk	ngamuk
14	ambil	ngambil
15	buka	ngebuka
16	bantu	ngebantu

No	Kata Dasar	Bentuk Tidak Baku
17	lepas	ngelepas
18	bayang	kebayang
19	injak	keinjek
20	kabul	kekabul
21	pergok	kepergok
22	tipu	ketipu
23	ulang	keulang
24	wujud	kewujud
25	cerita	critain
26	betul	betulin
27	manja	manjain
28	ganggu	gangguin
29	ganti	gantian
30	ikut	ikutan
31	musuh	musuhan
32	sabun	sabunan
33	teman	temenan
34	tukar	tukeran
35	tanya	nanyain
36	tunjuk	nunjukin
37	penting	mentingin
38	pegang	megangin
39	selamat	nyelametin
40	sempat	nyempetin
41	korban	ngorbanin
42	hadap	ngadepin
43	Bukti	ngebuktiin
44	warna	ngewarnain
45	dengar	kedengeran
46	ketemu	ketemuan
47	benar	beneran
48	begini	ginian
49	kawin	kawinan
50	main	mainan
51	parkir	parkiran
52	dulu	duluan
53	gendut	gendutan
54	karat	karatan
55	paling	palingan
56	sabar	sabaran
57	bagus	kebagusan
58	sana	sanaan
59	cepat	cepatan
60	pagi	sepagian

### 3. HASIL DAN PEMBAHASAN

Pada penelitian ini dilakukan pengukuran tingkat akurasi algoritma Jaro-Winkler distance dalam melakukan stemming kata imbuhan tidak baku bahasa Indonesia. Algoritma Jaro-Winkler distance diterapkan pada bahasa pemrograman PHP. Kata imbuhan tidak baku di input ke aplikasi, kemudian setiap kata akan di stemming satu persatu. Hasil stemming akan masukkan ke Tabel 3 berikut ini.

Tabel 3. Hasil *Stemming* Algoritma Jaro-Winkler distance

No	Kata Dasar	Bentuk Tidak Baku	<i>Stemming</i> Jaro-Winkler distance
1	terjang	nerjang	<b>terjang</b>
2	tuduh	nuduh	<b>tuduh</b>
3	terima	nerima	<b>terima</b>
4	tegur	negur	<b>tegur</b>
5	pukul	mukul	<b>pukul</b>
6	pimpin	mimpin	<b>pimpin</b>
7	coba	nyoba	<b>coba</b>
8	siram	nyiram	<b>siram</b>
9	suruh	nyuruh	<b>suruh</b>
10	simpan	nyimpen	pimpin
11	sebrang	nyebrang	<b>sebrang</b>
12	anggap	nganggep	pegang
13	amuk	ngamuk	<b>amuk</b>
14	ambil	ngambil	<b>ambil</b>
15	buka	ngebuka	ngebuka
16	bantu	ngebantu	ngebantu
17	lepas	ngelepas	ngelepas
18	bayang	kebayang	<b>bayang</b>
19	injak	keinjek	<b>injak</b>
20	kabul	kekabul	<b>kabul</b>
21	pergok	kepergok	<b>pergok</b>
22	tipu	ketipu	<b>tipu</b>
23	ulang	keulang	<b>ulan</b>
24	wujud	kewujud	<b>wujud</b>
25	cerita	critain	<b>cerita</b>
26	betul	betulin	<b>betul</b>
27	manja	manjain	<b>manja</b>
28	ganggu	gangguin	<b>ganggu</b>
29	ganti	gantian	<b>ganti</b>
30	ikut	ikutan	<b>ikut</b>
31	musuh	musuhan	<b>musuh</b>
32	sabun	sabunan	<b>sabun</b>
33	teman	temenan	<b>teman</b>
34	tukar	tukeran	<b>tukar</b>
35	tanya	nanyain	<b>tanya</b>
36	tunjuk	nunjukin	<b>tunjuk</b>
37	penting	mentingin	<b>penting</b>
38	pegang	megangin	<b>pegang</b>
39	selamat	nyelametin	<b>selamat</b>
40	sempat	nyempetin	pimpin
41	korban	ngorbanin	<b>korban</b>



No	Kata Dasar	Bentuk Tidak Baku	Stemming Jaro-Winkler distance
42	hadap	ngadepin	ngadepin
43	bukti	ngebuktiin	ngebuktiin
44	warna	ngewarnain	ngewarnain
45	dengar	kedengeran	<b>dengar</b>
46	ketemu	ketemuan	<b>ketemu</b>
47	benar	beneran	<b>beneran</b>
48	begini	ginian	<b>begini</b>
49	kawin	kawinan	<b>kawin</b>
50	main	mainan	<b>main</b>
51	parkir	parkiran	<b>parkir</b>
52	dulu	duluan	<b>dulu</b>
53	gendut	gendutan	<b>gendut</b>
54	karat	karatan	<b>karat</b>
55	paling	palingan	<b>paling</b>
56	sabar	sabaran	<b>sabar</b>
57	bagus	kebagusan	<b>bagus</b>
58	sana	sanaan	<b>sana</b>
59	cepat	cepatan	<b>cepat</b>
60	pagi	sepagian	<b>pagi</b>

Berdasarkan hasil pengujian pada Tabel 3 dengan pendekatan *similarity* menggunakan algoritma Jaro-Winkler Distance diketahui bahwa tingkat akurasi 85% atau 51 kata berhasil di-*stemming* (3 overstemming, 6 unstemming, 0 understemming).

#### 4. KESIMPULAN

Berdasarkan hasil analisis dan pengujian maka dapat ditarik kesimpulan bahwa *stemming* kata berimbuhan tidak baku bahasa Indonesia menggunakan algoritma Jaro-Winkler distance, menghasilkan tingkat akurasi yang cukup baik yaitu sebesar 85% atau berhasil melakukan *stemming* sebanyak 51 kata (3 overstemming, 6 unstemming, 0 understemming) dari 60 kata yang digunakan sebagai data pengujian.

#### 5. SARAN

Dalam penelitian ini ditemukan 9 kata yang tidak berhasil di-*stemming* menggunakan algoritma Jaro-Winkler distance. Oleh karena itu untuk penelitian selanjutnya penulis menyarankan untuk melakukan kombinasi beberapa algoritma seperti algoritma Nazief & Adriani untuk menghasilkan tingkat akurasi yang lebih baik.

#### 6. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada dosen pembimbing yang telah memberi bimbingan dan terima kasih kepada teman saya Azi karena sudah membantu terhadap penelitian ini.

## DAFTAR PUSTAKA

- [1] Nursyamsi, 2016, Kesalahan Penulisan Kata Bahasa Indonesia, *Jurnal Ilmiah Sains dan Teknologi*, No. 2, Vol. 15.
- [2] Putra, R. B. S., Utami, E., Raharjo, S., 2018, Optimalisasi Stemming Kata Berimbuhan Tidak Baku Pada Bahasa Indonesia Dengan Levenshtein Distance, *Jurnal. Pengemb. IT*, No. 2, Vol. 03, Hal. 200–205.
- [3] Rizki, A. S., Tjahyanto, A., Trialih, R., 2019, Comparison of stemming algorithms on Indonesian text processing, *TELKOMNIKA*, No. 1, Vol. 17, Hal. 95–103.
- [4] Putra, R. B. S., Utami, E., 2018, Non-formal Affixed Word Stemming in Indonesian Language, *ICOIACT 2018*, Yogyakarta, 6-7 Maret.
- [5] Setiawan, R., Kurniawan, A., Budiharto, W., Kartowisastro, I. H., Prabowo, H., 2016, Flexible Affix Classification for Stemming Indonesian Language, *13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Chiang Mai, 28 Juni-1 Juli.
- [6] Aulia, A., Khairani, D., Hakiem, N., 2017, Development of a Retrieval System for Al Hadith in Bahasa (Case Study: Hadith Bukhari), *5th International Conference on Cyber and IT Service Management (CITSM)*, Bali, 8-10 Agustus.
- [7] Khotimah, K., 2012, Analysis of Indonesia Affixes in English Words Found in Mobile Guide Edition, *LANTERN Jurnal*, No. 2, Vol. 1, 54-59.
- [8] Novantara, P., Pasruli, O., 2017, Implementasi Algoritma Jaro-Winkler Distance Untuk Sistem Pendeteksi Plagiarisme Pada Dokumen Skripsi, *Jurnal Buffer Informatika*, No 2, Vol. 3.
- [9] A. Kornain, F. Yansen, and Tinaliah, 2014, Penerapan Algoritma Jaro-Winkler Distance untuk Sistem Pendeteksi Plagiarisme pada Dokumen Teks Berbahasa Indonesia, *SPHP-ILKOM*, pp. 1–9.
- [10] A. P. Sari, R. Saptono, and E. Suryani, 2018, The Implementation of Jaro-Winkler Distance and Naive Bayes Classifier for Identification System of Pests and Diseases on Paddy, vol. 7, no. 1, hal 1–7.
- [11] A. Prasetyo, W. M. Baihaqi, and I. S. Had, 2018, Algoritma Jaro-Winkler Distance: Fitur Autocorrect dan Spelling Suggestion Pada Penulisan Naskah Bahasa Indonesia di BMS TV, *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 5, hal 435–444.