

# APLIKASI PENGOREKSIAN KESALAHAN BAHASA INDONESIA MENGUNAKAN METODE *JARO WINKLER DISTANCE* DAN *LEVENSHTEIN DISTANCE* BERBASIS WEB

**Sandra Nurva Agustin**

Program Studi S1 Teknik Informatika, Fakultas Teknologi Informasi, Universitas Hasyim Asy'ri  
Email : [sandranurvaagustin@gmail.com](mailto:sandranurvaagustin@gmail.com)

**Iftitaahul Mufarrihah**

Program Studi S1 Teknik Informatika, Fakultas Teknologi Informasi, Universitas Hasyim Asy'ri  
Email : [iftitaahulumufarrihah@unhasy.ac.id](mailto:iftitaahulumufarrihah@unhasy.ac.id)

**Dedy Rahman Prehanto**

Program Studi S1 Teknik Informatika, Fakultas Teknologi Informasi, Universitas Hasyim Asy'ri  
Email : [dedyrahman@unhasy.ac.id](mailto:dedyrahman@unhasy.ac.id)

## Abstrak

Kesalahan penulisan dalam pengetikan dokumen sangat sering terjadi dalam penulisan artikel, jurnal ilmiah, tugas kuliah atau juga tugas akhir. Kesalahan pengetikan tidak bisa dihindari ada beberapa faktor diantaranya yakni letak huruf yang berdekatan pada *keyboard*, Kesalahan dalam pengoreksian yang dilakukan oleh *word processor*, Kesalahan yang tidak disengaja seperti slip jari tangan. Untuk itu dibuatlah aplikasi pengoreksian kesalahan Bahasa Indonesia agar mempermudah pengecekan dokumen. Tujuan dari penelitian ini adalah merancang sistem pengoreksian menggunakan metode *Jaro Winkler Distance* dan *Levenshtein Distance*. Algoritma *Jaro winkler distance* pada dasarnya merupakan metode yang membandingkan kesamaan penyusun karakter dari dua buah *string* dan algoritma *Levenshtein Distance* adalah sebuah metrik *string* untuk mengukur perbedaan antara dua urutan. Hasil dari penelitian ini yakni berupa web pengoreksian kesalahan pengetikan dimana user dapat mengoreksi dokumen berformat *.doc/.docx*.

**Kata kunci** : jaro winkler distance, levenshtein distance.

## Abstract

Writing errors in typing documents very often occur in writing articles, scientific journals, lectures or final assignments. Typing errors cannot be avoided. There are several factors, including the location of the letters close to each other on the keyboard, errors in the corrections made by the word processor, accidental errors such as slip of a finger. For this reason, an application for error categorization in Indonesian was made to make it easier to check documents. The purpose of this study is to design a correction system using the Jaro Winkler Distance and Levenshtein Distance methods. The Jaro winkler distance algorithm is basically a method that compares the similarity of the constituent characters of two strings and the Levenshtein Distance algorithm is a string metric to measure the difference between two sequences. The results of this research are in the form of typing error correction web where the user can correct *.doc / .docx* formatted documents.

**Key words** : jaro winkler distance, levenshtein distance.

## PENDAHULUAN

Kesalahan penulisan dalam pengetikan dokumen sangat sering terjadi, terlebih lagi semakin banyak orang yang menyampaikan gagasannya kedalam artikel, jurnal ilmiah, tugas kuliah atau juga tugas akhir. Tentu dalam penulisan dokumen tersebut kesalahan pengetikan tidak bisa dihindari yang disebabkan oleh beberapa faktor diantaranya Letak huruf yang saling berdekatan pada *keyboard* sehingga menimbulkan slip jari tangan dan bisa juga karena pengoreksian otomatis

yang dilakukan *word processor*. Cara kerja fitur *word processor* yakni memindai kata dari berkas yang tidak ada pada kamus bahasa tertentu.

Beberapa algoritma dapat digunakan untuk mendapatkan kata perbaikan yang amat mendekati dari pengetikan ejaan yang salah. Salah satunya yakni dengan menggunakan algoritma *jaro winkler distance*. Cara kerja dari algoritma *jaro winkler distance* yakni dengan membandingkan kesamaan karakter penyusunan terhadap dua buah *string*. Kelemahan *jaro winkler* sendiri tidak bisa mengoreksi kata yang berdempet oleh karena itu dengan menambahkan metode *levenshtein distance* akan mempermudah pengoreksian bila terjadi kata berdempet.

Penelitian terhadap penerapan *jaro winkler* dalam perbaikan kata saran dengan judul “Penerapan *Jaro winkler distance* Dalam Aplikasi Pengoreksian Kesalahan Penulisan Bahasa Indonesia Berbasis *Web*” (Frando, dkk., 2019) pada penelitian ini *jaro winkler* dapat diterapkan dalam menganjurkan saran jika ada salah kata dalam pengetikan.

Penelitian berbeda perihal penerapan *levenshtein distance* dengan judul “Pengoreksian Ejaan Kata Berbahasa Indonesia Menggunakan Algoritma *Levenshtein distance*” (Braddley, dkk, 2017) pada penelitian ini aplikasi yang dibangun memanfaatkan metode *levenshtein distance* guna mencari saran pembenaran ejaan. Karena operasi pencarian jarak yang cukup lengkap. Selain itu aplikasi yang dibangun ini belum mendukung dalam pengoreksian berformat dokumen.

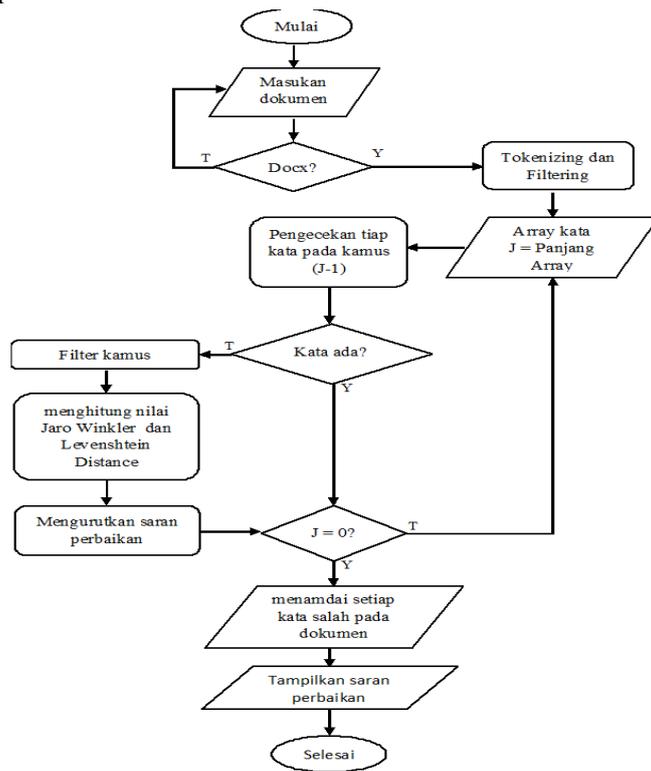
Berlandaskan penelitian yang sudah dilakukan terhadap pengoreksian kesalahan penulisan ejaan maka penulis memilih judul “Aplikasi Pengoreksian Kesalahan Bahasa Indonesia Menggunakan Metode *Jaro winkler distance* Dan *Levenshtein distance* Berbasis *Web*” dengan penggabungan dua algoritma pada penelitian ini diharapkan akan semakin menambah keakuratan proses pengoreksian otomatis dokumen jurnal atau tugas kuliah.

## **METODE**

Penelitian ini memiliki tujuan yakni merancang sistem pengoreksian kesalahan penulisan. Untuk metodologi penelitian sebagaimana berikut :

1. Pengumpulan data  
Dalam penelitian ini, metode pengumpulan data yang digunakan yakni Data yang didapat dari proses pengecekan kata dari dokumen proposal skripsi mahasiswa Fakultas Teknologi Informasi Universitas Hasyim Asy'ari.
2. Preprocessing
  - a. Tokenizing  
Tokenizing yaitu tahap pemotongan string atau kata yang diinputkan berdasarkan tiap kata yang menyusunnya. Proses ini bekerja untuk memecah satu kalimat menjadi kumpulan kata. (Prasetyo, dkk, 2017).
  - b. Filtering  
Filtering adalah tahap yang diambil dari hasil pada tahap tokenizing. Pada tahap ini kata yang ganda akan dibuang.

### 3. Flowchart Sistem



**Gambar 1** Flowchart Sistem

Pada flowchart yang ditentukan pada gambar 1 merupakan flowchart dari proses pengoreksian kata, dimana user memasukkan dokumen berupa .doc/ ,docx. Kemudian sistem akan masuk ketahap preprocessing yang terdiri dari tahap tokenizing dan tahap filtering. Setelah inputan kata selesai dalam proses preprocessing selanjutnya sistem akan masuk ketahap filtering kamus dimana kata yang tidak sesuai akan dikoreksi menggunakan perhitungan jaro winkler distance dan levenshtein distance untuk menemukan kata yang matcing. Selanjutnya sistem akan menampilkan saran perbaikan dari kata yang salah.

#### 4. Algoritma Jaro Winkler Distance

Algoritma *Jaro winkler distance* pada dasarnya merupakan metode untuk membandingkan kesamaan karakter penyusun dari dua buah *string*. Jaro menggunakan rumus untuk menghitung jarak ( $dj$ ) antara dua string yaitu  $S_1$  dan  $S_2$ , dengan  $m$  merupakan jumlah karakter yang sama dan  $t$  merupakan setengah dari jumlah karakter yang tertukar (transposisi). (Frando, dkk. 2019)

$$dj = \frac{1}{3} \times \left( \frac{m}{|S_1|} + \frac{m}{|S_2|} + \frac{m-t}{m} \right) \dots \dots \dots (1)$$

Keterangan :

$dj$  : nilai jaro distance

$m$  : jumlah karakter yang sama

$S_1$  : Panjang string 1

$S_2$  : Panjang string 2

$t$  : setengah dari jumlah karakter yang tertukar (transposisi)

Jika dua karakter terbilang cocok maka jarak teori dua buah karakter yang disamakan bisa dibenarkan apabila tidak melebihi :

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

Semakin tinggi nilai jaro maka kesamaan dari dua buah karakter memiliki kemiripan.

**Table 1** Contoh Kata

input string 1	t	e	m	p	a	t
input string 2	t	e	m	a	p	t

Perhitungan range uji  
 = (Banyaknya jumlah kata /2)-1  
 = (6 / 2) -1  
 = 3-1  
 = 2 (untuk indeks uji didepan dan belakang)

input string 1	t	e	m	p	a	t
input string 2	t	e	m	a	p	t

t	e	m	p	a	t
t	e	m	a	p	t

**Gambar 2** Contoh Kata

Jumlah karakter tidak sama = 2  
 Perhitungan jumlah transposisi (t)  
 $t = (2/2)$   
 = 1

Setelah mengetahui nilai transposisi (t) maka,  
 Diketahui :

$m = 6$   
 $S1 = 6$   
 $S2 = 6$

$t = 1$   
 $(dj) = 0.944$   
 $l = 3$   
 $p = 0.1$

Jaro distance

$$Jw = \frac{1}{3} \left( \frac{6}{6} + \frac{6}{6} + \frac{6-1}{6} \right)$$

$$= \frac{1}{3} \left( \frac{12}{6} + \frac{5}{6} \right)$$

$$= 0.944$$

$$Jw = 0.944 + (3 * 0.1 * (1 - 0.944))$$

$$= 0.944 + (3 * 0.1 * (0.056))$$

$$= 0.944 + (0.0168)$$

$$= 0.97$$

Teks Input

tempat

---

Teks Perbaikan

tempat

---

Metode Jaro Winkler

0.99761904761905

**Gambar 3** Nilai Jaro Winkler Distance

5. *Algoritma Levenstein Distance*

*Levenshtein distance* merupakan metrik *string* untuk mengukur perbedaan antara dua urutan. Pada algoritma *Levenshtein distance* semakin kecil nilai skor yang dimiliki maka kemiripan yang dimiliki semakin tinggi. (Yulianingsih, 2017).

$$lev_{a,b}(i,j) = \max(i,j) \\ \min \begin{matrix} lev_{a,b}(i-1,j) + 1 \\ lev_{a,b}(i,j-1) + 1 \\ lev_{a,b}(i-1,j-1) + (a_i \neq b_j) \dots \dots \dots (4) \end{matrix}$$

Dimana

$l(a_i \neq b_j)$  = fungsi indikator

$lev_{a,b}(i,j)$  = jarak pertama karakter dari a dan karakter pertama.

**Tabel 2** Contoh Kata

	Null	t	e	m	p	a	a	t
null	0	1	2	3	4	5	6	7
T	1	0	1	2	3	4	5	6
E	2	1	0	1	2	3	5	6
M	3	2	3	0	1	2	3	4
P	4	3	4	5	0	1	2	3
A	5	4	5	5	6	0	1	2
T	6	5	5	6	6	1	2	1

Dari matriks diatas maka diperoleh nilai dari *Algoritma Levenshtein Distance* yaitu nilai dari korom akhir dari baris terakhir yang bernilai 1. Setelah proses pengurutan matrik selanjutnya yakni menghitung kesamaan string sebagai berikut :

$$\begin{aligned} \text{Sim (tempat,tempaat)} &= 1 - \frac{LD(\text{tempat,tempaat})}{\max(|\text{tempat}|,|\text{tempaat}|)} \\ &= 1 - \frac{1}{\max(7,6)} \\ &= 1 - \frac{1}{7} = 0,85 \end{aligned}$$

$$\begin{aligned} \text{Sim (tempat,tempat)} &= 1 - \frac{LD(\text{tempat,tempat})}{\max(|\text{tempat}|,|\text{tempat}|)} \\ &= 1 - \frac{0}{\max(6,6)} \\ &= 1 - \frac{0}{6} = 1 \end{aligned}$$

Setelah mencari kesamaan kata tempat pada tiap string kalimat pada kamus maka selanjutnya dilakukan proses pemilihan rekomendasi kata sebagai berikut :

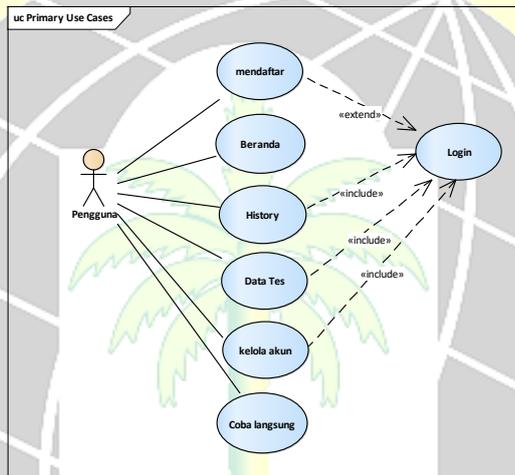
Perbandingan kata tempat- tempaat = 0,85

Perbandingan kata tempat – tempat = 1

Jadi hasil yang didapat dari perhitungan manual diatas bahwasanya dipilih kata tempat dengan nilai 1. Dikarenakan kata tempaat tidak ada dalam kamus maka tidak direkomendasikan menjadi kata perbaikan.

**Gambar 4 Nilai Levenshtein Distance**

6. Usecase Diagram



**Gambar 5 Usecase User**

**HASIL DAN PEMBAHASAN**

Berikut adalah hasil penelitian yang berjudul “Aplikasi Pengoreksian Kesalahan Bahasa Indonesia Menggunakan Metode *Jaro Winkler Distance* Dan *Levenshtein Distance* Berbasis Web” :

1. Halaman login



**Gambar 6 Login**

Tampilan awal dari halaman web pengoreksian kata yakni halaman login. Dimana pada halaman ini terdapat 4 menu yakni login, daftar, coba dan help.

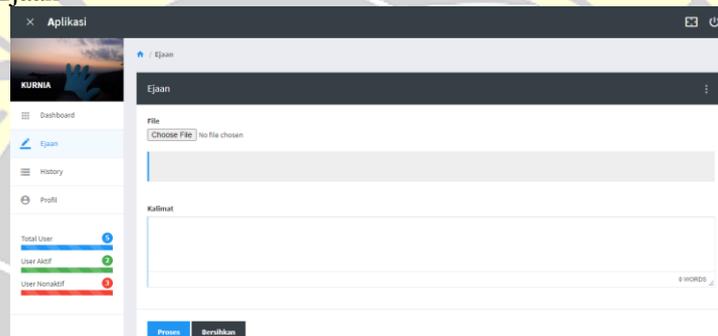
## 2. Halaman Dashboard



Gambar 7 Dashboard

Setelah user masuk sistem akan menampilkan halaman dashboard dimana terdapat 3 menu yakni Ejaan, History, Profil.

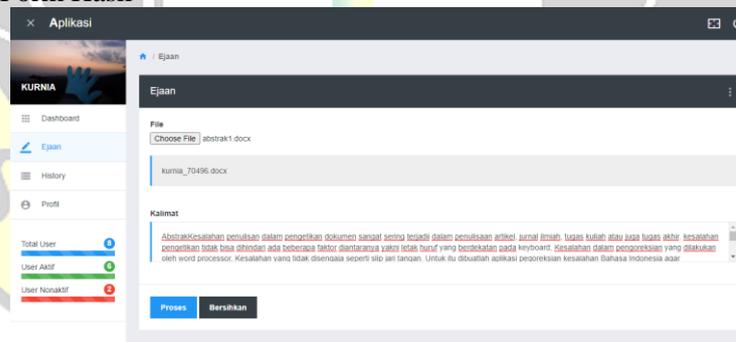
## 3. Halaman Ejaan



Gambar 8 Ejaan

Halaman ejaan merupakan halaman yang berfungsi untuk menginputkan text yang ingin dikoreksi yang nantinya akan diproses menggunakan *algoritma jaro winkler distance* dan *levenshtein distance*.

## 4. Tampilan Form Hasil



Gambar 9 Inputan



Gambar 10 Hasil

## **PENUTUP**

### **Kesimpulan**

Berdasarkan hasil dari penelitian, maka dapat ditarik kesimpulan sebagai berikut :

1. Perancangan aplikasi pada penelitian ini menggunakan Flowchart, Use Case Diagram, Dan Activity Diagram, dengan adanya rancangan tersebut sehingga mampu menggambarkan aplikasi yang akan diterapkan yakni hasil sistem pengoreksian kesalahan penulisan menggunakan Jaro Winkler Distance dan Levenshtein Distance berbasis web.
2. Penerapan sistem menggunakan Bahasa pemrograman PHP dengan metode jaro winkler distance dan levenshtein distance, pengguna menginputkan dokumen .doc/.docx selanjutnya sistem melakukan tahap tokenizing untuk memecah kata dan apabila dalam tahap tokenizing terdapat kata yang salah maka sistem akan melakukan perhitungan jaro winkler distance dan levenshtein distance untuk menemukan kata yang matcing.
3. Penerapan sistem dari pengujian blackbox sistem layak digunakan untuk pengoreksian kesalahan dalam pengetikan Bahasa Indonesia dan sistem bekerja dengan baik untuk mengoreksi kesalahan Dari 50 kata yang terdiri dari beberapa kata yang salah, aplikasi pengoreksian dapat mengoreksi 47 kata dengan benar. Sedangkan yang lainnya terdapat kesalahan dalam pemberian saran karena ketidak lengkapan kamus. Aplikasi hanya dapat mengoreksi 200 kata dengan optimal.

### **Saran**

Berikut ini Beberapa saran yang perlu disampaikan untuk penelitian selanjutnya:

1. Dapat menambahkan data kamus lain seperti KBBI Online agar tingkat keakurasian pengoreksian dapat meningkat.
2. Dapat menambahkan fitur format dokumen lain yang dapat diupload.
3. Dapat menambahkan metode untuk mengurai struktur kalimat. Karena pada penelitian ini bukanlah satu- satunya metode yang dapat diterapkan, karena masih terdapat metode lain yang lebih maksimal.

### **DAFTAR PUSTAKA**

- Frando, J., Ruslianto, I, Hidayati, R. 2019. "Penerapan *Jaro winkler distance* Dalam Aplikasi Pengoreksian Kesalahan Penulisan Bahasa Indonesia Berbasis *Web*". Jurnal Komputer Dan Aplikasi. Vol. 7 (3): hlm. 44-53.
- Braddley, M.O., Fachurrozi, M, Yusliani, N. 2017. Pengoreksian Ejaan Kata Berbahasa Indonesia Menggunakan Algoritma Levensthein Distance vol. 3, hlm 167-171.
- Pasetyo, A, Baihaqi, M.W., Had, S.I. 2018. "Algoritma *Jaro winkler distance* : Fitur Autocorrect Dan Spelling Suggestion Pada Penulisan Naskah Bahasa
- Yulianingsih. 2017. "*Implementasi Algoritma Jaro Winkler Dan Levenstein Distance* Dalam Pencarian Data Pada *Database*". Jurnal String. Vol. 2 (1): hlm. 18-27.