

## Penerapan Algoritma *Levenshtein Distance* pada Aplikasi Asisten Guru Bahasa Inggris

M. Fahrur Azhri<sup>1</sup>, Daniel Swanjaya<sup>2</sup>, Ratih Kumalasari Niswatin<sup>3</sup>

<sup>1,2</sup>Teknik Informatika, Fakultas Teknik, Universitas Nusantara PGRI Kediri

E-mail: <sup>1</sup>\*1amfahrur@gmail.com, <sup>2</sup>daniel@unpkediri.ac.id, <sup>3</sup>ratih.workmail@gmail.com

**Abstract** – Many plagiarism acts among high school students when submitting assignments and lacking systems which are able to detect similarities in the contents of students' tasks. In a nutshell, the teacher finds it difficult to get the authenticity of the tasks done by students. This study aims at creating an English teacher assistant application, to make it easier to correct the authenticity of students' essay assignments so that the teacher can get the task that is the original work of the students themselves without any act of plagiarism. This application is divided into two stages, namely the stages of the text preprocessing and then the similarity calculation is done with the *Levenshtein Distance* algorithm. Stages of text preprocessing consist of tokenizing, purifying, stopword removal, stemming, and sorting. To get text similarity scores between students, calculations are made with the *Levenshtein Distance* algorithm by applying the limit of the text similarity value (threshold) to 70%, if the students' text similarity is less than 70%, the document will be received by the system and entered into the database. Text similarity exceeds 70%, then the document will be rejected. Therefore, the results of this study can help English teachers, especially educational institutions, to obtain original documents and reduce plagiarism from an early age at the high school level.

**Keywords**— Text similarity, plagiarism, *Levenshtein Distance*.

### 1. PENDAHULUAN

Pengajaran menulis di kalangan siswa kelas menengah atas saat ini sedang gencar dilakukan sebagai persiapan siswa ke jenjang kerja atau perguruan tinggi. Bersamaan dengan itu perkembangan teknologi komunikasi melalui internet baik dari komputer, laptop, atau *gadget* siswa juga mengalami banyak perkembangan dan pemberdayaan. Guru dengan menggunakan *blended learning* di era digital ini juga memiliki tugas penting untuk tetap menjaga integritas asli siswa dalam menulis terutama masalah plagiarisme.

Penjiplakan atau plagiarisme berarti mencontoh atau meniru atau mencuri tulisan dan karya orang lain yang kemudian diakui sebagai karangannya sendiri dengan ataupun tanpa seizin penulisnya [1]. Penjiplakan dokumen digital bukanlah hal yang susah, cukup dengan menggunakan teknik *copy-paste-modify* pada sebagian isi dokumen dan bahkan keseluruhan isi dokumen sudah bisa dikatakan bahwa dokumen tersebut merupakan hasil duplikasi dari dokumen lain.

Salah satu contoh hasil observasi penulis di SMA Negeri 4 Blitar yakni pada pelajaran Bahasa Inggris yang sering menggunakan dokumen teks dalam pengerjaan tugas menunjukkan adanya kecenderungan siswa melakukan penjiplakan hasil menulis mereka baik dari teman, *website*, atau dari buku. Dari 34 siswa dalam satu kelas terdapat 24 siswa yang menjiplak dengan berbagai jenis penjiplakan seperti mengambil ide yang sudah ada tanpa menyebut sumber dengan jelas, mengambil data penelitian orang lain, menggunakan kata, kalimat, paragraf yang sama dan bahkan ada yang melakukan penjiplakan keseluruhan.

Sebagai pendidik, seorang guru harus mampu menepis kebiasaan buruk yang dapat melanggar hukum di kemudian hari di lingkup perguruan tinggi dan lingkungan kerja berkenaan dengan Undang – Undang Hak Cipta dan Plagiat. Dalam studi ini, peneliti menggunakan rekayasa perangkat untuk mengurangi tindak plagiarisme di lingkungan sekolah. Dengan menggunakan aplikasi pendeteksi kemiripan isi teks tugas siswa menggunakan algoritma *Levenshtein Distance*, tindak plagiarisme secara perlahan dapat ditekan dan untuk mendukung penekanan tindak plagiarisme ini di lingkungan sekolah. Oleh karena itu, dibutuhkan adanya sistem yang memudahkan dalam mendeteksi dan mengukur kemiripan dokumen. Selain dapat mengetahui tindak plagiarisme, pengukuran kemiripan dokumen ini dapat membantu dalam pengelompokan dokumen. Sebagian besar kasus plagiarisme ditemukan di kalangan siswa, berupa mengambil ide yang sudah ada tanpa menyebut sumber dengan jelas, mengambil data penelitian orang lain, menggunakan kata, kalimat, paragraf yang sama dan bahkan ada yang melakukan penjiplakan keseluruhan, dan sebagainya. Deteksi plagiarisme ini dilakukan dengan membandingkan sebuah dokumen dengan dokumen lainnya. Tingkat kesamaan dokumen tersebut akan menjadi dasar pendeteksian plagiarisme dari tugas yang diberikan oleh guru.

Penggunaan metode *Levenshtein Distance* menjadi sebuah matriks *string* yang digunakan untuk mengukur perbedaan atau jarak (*distance*) antara dua *string* [2]. Nilai *distance* antara dua *string* ini ditentukan oleh jumlah minimum dari operasi - operasi perubahan yang diperlukan untuk melakukan transformasi dari suatu *string* menjadi *string* lainnya. Operasi-operasi tersebut adalah penyisipan (*insertion*),

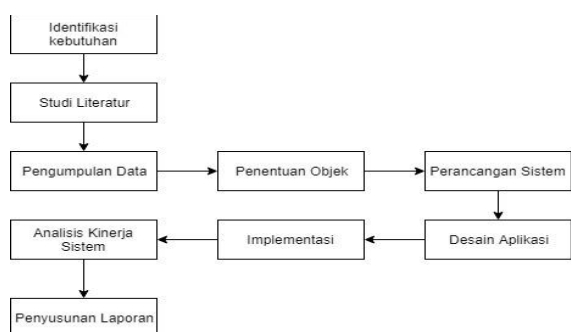
penghapusan (*deletion*), atau penukaran (*substitution*). *Levenshtein Distance* merupakan salah satu algoritma yang dapat digunakan dalam mendeteksi kemiripan antara dua *string* yang berpotensi melakukan tindakplagiarisme.

Berdasarkan latar belakang di atas, maka diperlukan suatu sistem untuk mengenali kemiripan isi teks tugas siswa, sehingga integritas suatu data tersebut terjamin dan tidak terkena tindak plagiarisme. Rumusan masalah pada penelitian ini adalah ‘Bagaimana merancang dan membangun suatu sistem untuk mendeteksi kemiripan isi teks tugas siswa?’ dan ‘Bagaimana menerapkan Algoritma *Levenshtein Distance* dalam membangun aplikasi kemiripan isi teks tugasiswa?’ Tujuan penelitian ini adalah untuk merancang dan membangun suatu sistem yang dapat mendeteksi kemiripan isi teks tugasiswa dan menerapkan *Levenshtein Distance* dalam membangun aplikasi kemiripan isi teks tugasiswa.

## 2. METODE PENELITIAN

Pada penelitian ini metode penelitian yang digunakan yaitu metode penelitian deskriptif kuantitatif, yaitu meliputi pengambilan sampel, identifikasi sampel, pengumpulan data dan analisis data. Teknik yang digunakan adalah penelitian pengembangan, yaitu meliputi rencana, rancangan, konstruksi dari model, terapan atau objek hasil penelitian yang harus teruji berdasarkan metode formal, komputasi, maupun pengujian matematis.

Prosedur penelitian yang digunakan pada penelitian ini meliputi identifikasi kebutuhan, studi literatur, pengumpulan data, penentuan objek, perancangan sistem, desain aplikasi, implementasi, analisis kinerja sistem, dan penyusunan laporan seperti gambar 1 di bawah ini :



Gambar 1. Prosedur Penelitian

### 2.1 Identifikasi Kebutuhan

Menentukan objek instansi pendidikan untuk mendapatkan data tugas siswa serta informasi pendukung.

### 2.2 Studi Literatur

Mengumpulkan referensi baik dari internet, buku maupun dari sumber-sumber lain, diantaranya : a) mempelajari tentang plagiarisme, b) mempelajari tentang caramengetahui tindak plagiarisme dan macam-macam teknik plagiarisme, dan c) mempelajari tentang

metode-metode yang akan digunakan pada perancangan sistem aplikasi yang akan dirancang yaitu Aplikasi Asisten Guru Bahasa Inggris.

### 2.3 Pengumpulan Data

Pengumpulan data yaitu tugas siswa yang di *upload* kedalam sistem yang telah dibuat. Kemudian dilakukan tahapan *text preprocessing* dan perhitungan algoritma *Levenshtein Distance*.

### 2.4 Penentuan Objek

Objek dalam penelitian ini adalah data tugas siswa dalam format dokumen \*.pdf.

### 2.5 Perancangan Sistem

Merancang algoritma sistem, masukan dan keluaran sistem, desain tampilan, serta menentukan perangkat-perangkat yang dibutuhkan dalam sistem. Kemudian merealisasikan rancangan kedalam bahasa pemrograman.

### 2.6 Desain Aplikasi

Tahap ini merupakan tahapan setelah proses perancangan selesai agar aplikasi dapat berjalan sesuai kebutuhan dan siap dioperasikan.

### 2.7 Implementasi

Tahap ini merupakan menerjemahkan kebutuhan kedalam bahasa pemrograman .

### 2.8 Analisis Kinerja Sistem

Menganalisis hasil uji coba program dan menarik kesimpulan, sehingga mengetahui kinerja dari proses analisa pada sistem.

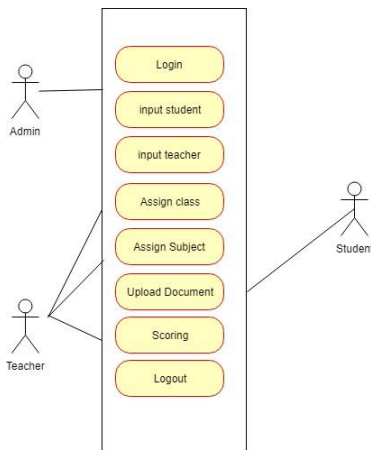
### 2.9 Penyusunan Laporan

Yaitu membuat laporan kegiatan penelitian serta penjelasan singkat mengenai teori yang digunakan. Laporan disusun berdasarkan data yang diperoleh, pembelajaran materi, perancangan dan pembuatan sistem, serta implementasi dan data pengujian sistem yang telah dilakukan sebelumnya.

## 3. HASIL DAN PEMBAHASAN

Aplikasi Asisten Guru Bahasa Inggris ini menggunakan tahapan *text preprocessing* dan algoritma *Levenshtein Distance*. Dimana tahapan *text preprocessing* perlu dilakukan agar susunan kalimat suatu dokumen tetap urut sesuai abjad, hal ini dilakukan untuk menghindari tindak plagiarisme yang dilakukan baik di awal kalimat, tengah kalimat, maupun pada akhir kalimat. Setelah melewati tahapan ini, kemudian dilakukan perhitungan dengan algoritma *Levenshtein Distance* untuk mengukur jarak antar string dan menghitung nilai kemiripan teks antar dokumen. Berikut merupakan rancang bangun dari Aplikasi Asisten Guru Bahasa Inggris :

### 3.1 Use Case Diagram

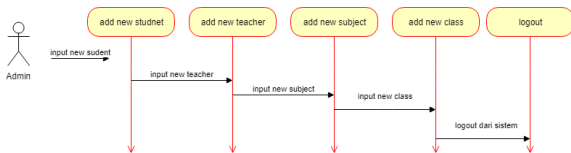


Gambar 2. Use Case Diagram Aplikasi Asisten Guru Bahasa Inggris

Keterangan :

- Admin**  
Admin bertindak sebagai *superuser* dimana mempunyai hak akses *create, read, update, delete, edit* didalam sistem yang telah dibuat.
- Teacher**  
Teacher bertindak untuk memberikan tugas dan memberikan nilai atas tugas yang dikerjakan oleh siswa. *Teacher* juga memiliki tanggung jawab terhadap kelas yang diampunya. Hal ini dibuat agar ketika terjadi kesalahan *input* nilai, *admin* akan menghubungi *teacher* yang bersangkutan.
- Student**  
*Student* bertindak sebagai client didalam sistem yang telah dibuat, seorang siswa bertugas untuk melakukan *upload* tugas kedalam sistem. Ketika melakukan *upload* tugas mendapatkan nilai kemiripan teks kurang dari 70%, maka dokumen siswa akan diterima oleh sistem. Akan tetapi, ketika nilai kemiripan teks lebih dari 70%, maka dokumen siswa tersebut akan ditolak (*rejected*) oleh sistem.

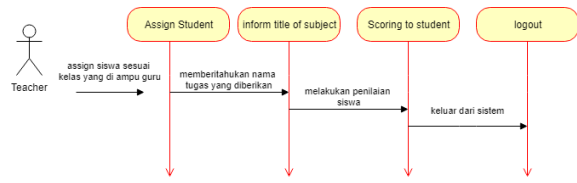
### 3.2 Sequence Diagram Admin



Gambar 3. Sequence Diagram Admin Aplikasi Asisten Guru Bahasa Inggris

Gambar 2 menunjukkan alur tugas yang dilakukan oleh *admin* yaitu, *add new student, add new teacher, add new subject, add new class*, dan *logout* dari sistem.

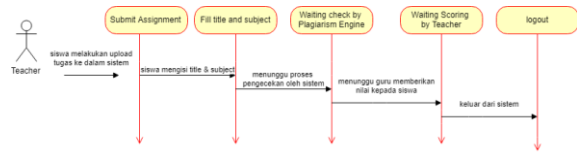
### 3.3 Sequence Diagram Teacher



Gambar 4. Sequence Diagram Teacher Aplikasi Asisten Guru Bahasa Inggris

Gambar 3. menunjukkan alur tugas yang dilakukan oleh *teacher* yaitu, *assign student, inform title of subject, scoring to student*, dan *logout* dari sistem.

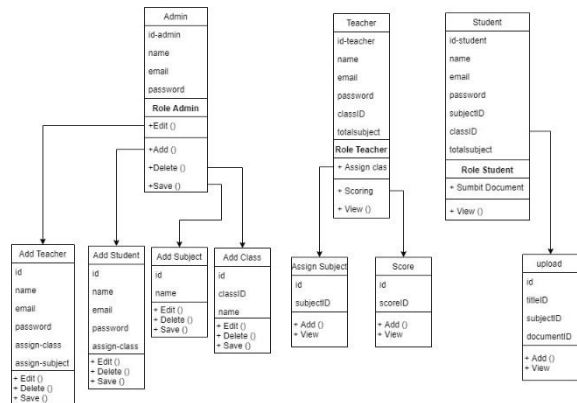
### 3.4 Sequence Diagram Student



Gambar 5. Sequence Diagram Student Aplikasi Asisten Guru Bahasa Inggris

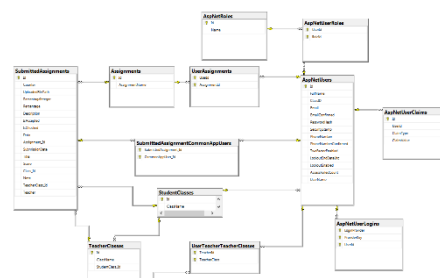
Gambar 4 menunjukkan alur tugas yang dilakukan oleh *student* yaitu, *submit assignment, fill title and subject, waiting check by plagiarism engine, waiting score form teacher*, dan *logout* dari sistem.

### 3.5 Class Diagram



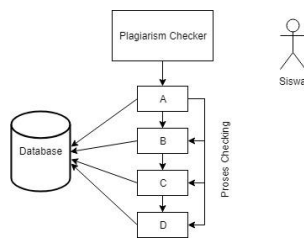
Gambar 6. Class Diagram Aplikasi Asisten Guru Bahasa Inggris

### 3.6 Relationship Diagram



Gambar 7. Relationship Diagram Aplikasi Asisten Bahasa Inggris

### 3.7 Proses Checking



Gambar 8. Gambaran Proses *Checking* Dokumen Aplikasi Asisten Bahasa Inggris

Gambar 8 menampilkan gambaran proses ketika dilakukan proses perbandingan antara dokumen 1 dengan dokumen lainnya. Siswa yang pertama melakukan *upload* dokumen akan di inialisasi dengan huruf 'A'. Kemudian jika ada siswa yang kedua melakukan *upload* dokumen ke sistem dalam hal ini di inialisasi dengan huruf 'B' kemudian dokumen B akan di dibandingkan dengan dokumen A, Sistem menerapkan batasan nilai kemiripan (*threshold*) sebesar 70%. Jadi, ketika nilai kemiripan teks siswa melebihi 70% ketika proses pengecekan berlangsung, dokumen akan ditolak oleh sistem dan tidak masuk kedalam *database*. Ketika dokumen B berhasil diterima, dan ada siswa ketiga ingin melakukan *upload* dokumen dalam hal ini diinialisasi dengan huruf 'C', maka proses pengecekannya adalah dokumen 'C' akan dibandingkan dengan huruf 'A', jika nilai kemiripan teks kurang dari 70%, akan dilanjutkan dibandingkan dengan dokumen 'B'. Jika ketika dibandingkan dengan dokumen 'B' nilai kemiripan teks lebih dari 70%, maka dokumen ditolak sistem, ketika kurang dari 70% maka akan diterima oleh sistem dan masuk kedalam *database*. Proses selanjutnya akan sama jika terdapat siswa baru yang berhasil melakukan *upload* kedalam sistem.

### 3.8 Implementasi Algoritma *Levenshtein Distance*

*Levenshtein Distance* adalah sebuah *matriks string* yang digunakan untuk mengukur perbedaan atau jarak (*distance*) antara dua *string*. Nilai *distance* antara dua *string* ini ditentukan oleh jumlah minimum dari operasi-operasi perubahan yang diperlukan untuk melakukan transformasi dari suatu *string* menjadi *string* lainnya. Operasi-operasi tersebut adalah penyisipan (*insertion*), penghapusan (*deletion*), atau penukaran (*substitution*). *Levenshtein Distance* merupakan salah satu algoritma yang dapat digunakan dalam mendeteksi kemiripan antara dua *string* yang berpotensi melakukan tindak plagiarisme [3].

#### Operasi-Operasi pada *Levenshtein Distance*

Pada *Levenshtein Distance*, ada 3 macam operasi yang dapat dilakukan yaitu:

##### 1. Operasi Penyisipan Karakter (*Insertion*)

Operasi ini adalah penyisipan sebuah karakter baru kedalam *string*. Sebagai contoh kata 'diskrit' di

lakukan penyisipan huruf 'k' menjadi *string* baru 'diskrit' seperti Tabel 1 dibawah ini :

Tabel 1. Operasi Penyisipan Karakter

<i>String 1</i>	d	i	s	-	r	i	t
<i>String 2</i>	d	i	s	k	r	i	t
<i>Insertion</i>				k			

Dalam proses penyisipan karakter, dapat dilakukan pada awal kalimat, tengah kalimat, maupun akhir kalimat.

##### 2. Operasi Penghapusan Karakter (*Deletion*)

Tabel 2. Operasi Penghapusan Karakter

<i>String 1</i>	m	a	t	e	m	a	t	I	k	a	n
<i>String 2</i>	m	a	t	e	m	a	t	I	k	a	n
<i>Insertion</i>											n

Dalam Tabel 2 diatas, dilakukan penghapusan karakter 'n' di akhir kalimat. Awal *string* adalah 'matematikan' *string* menjadi 'matematika'.

##### 3. Operasi Pertukaran Karakter (*Substitution*)

Tabel 3. Operasi Pertukaran Karakter

<i>String 1</i>	H	i	m	p	u	n	a	N
<i>String 2</i>	G	i	m	p	u	n	a	N
<i>Insertion</i>	G							

Dalam Tabel 3 diatas, dilakukan pertukaran karakter 'h' dengan karakter 'g'. Awal *string* adalah 'Himpunan' menjadi *string* Gimpunan.

Langkah – langkah algoritma *Levenshtein Distance* sebagai berikut :

##### 1) Langkah pertama

- Set variabel N yang menyimpan panjang *string source* (S)
- Set variabel M yang menyimpan panjang *string target* (T)
- Jika N = 0 atau M=0 maka exit
- Buat matriks ber-ordo [0 – N][0 – M]

##### 2) Langkah kedua

- Inialisasi baris 0 – N
- Inialisasi kolom 0 – M

##### 3) Langkah ketiga

Periksa setiap karakter dari *string* S (looping dari 1 ke N -> variabel i)

##### 4) Langkah keempat

Periksa setiap karakter dari *string* T (looping dari 1 ke M -> variabel j)

##### 5) Langkah kelima

- jika  $S[i] = T[j] \rightarrow \text{cost} = 0$

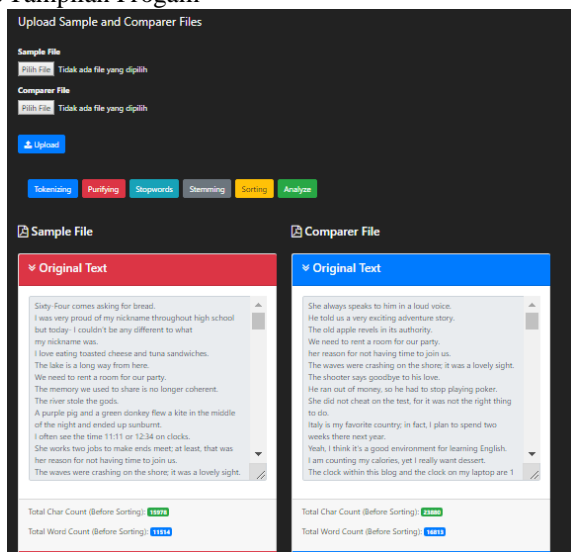
- b) jika  $S[i] \neq T[i] \rightarrow \text{cost} = 1$
- 6) Langkah keenam  
Set value  $d[i,j]$  yang diambil dari minimum jumlah:  
 $d[i-1,j] + 1$   
 $d[i,j-1] + 1$   
 $d[i-1,j-1] + \text{cost}$
- 7) Langkah ketujuh  
Setelah langkah 3,4,5,6 selesai (tidak ada looping lagi), hasilnya akan ketemu di element  $d[N,M]$   
Untuk menghitung nilai kemiripan teks dilakukan rumus sebagai berikut :

$$\text{Sim} = 1 - \frac{\text{Distance Value}}{\text{Max Lenght}} \times 100 \%$$

Keterangan :

- a. *Sim* = nilai kemiripan teks  
 b. *Distance Value* = nilai jarak *string*  
 c. *Max Lenght* = nilai terbesar diantara 2 *string*

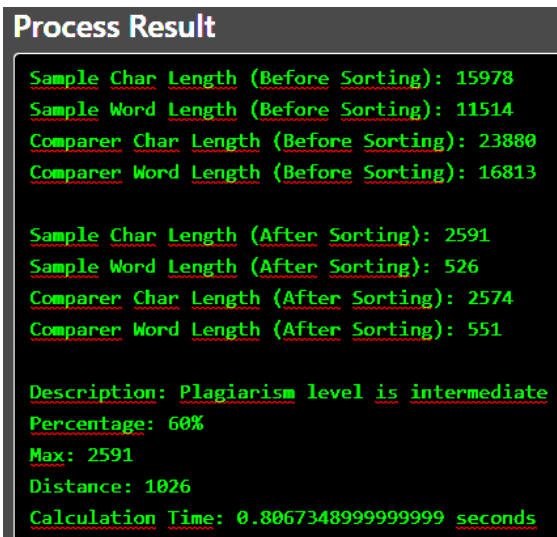
### 3.8 Tampilan Program



Gambar 9. Gambaran simulasi kerja Aplikasi Asisten Bahasa Inggris

Dalam langkah simulation page ini dokumen siswa dilakukan tahapan *text preprocessing* lalu dilakukan perhitungan dengan algoritma *levenshtein distance* untuk mencari nilai kemiripan dokumen.

### 3.9 Hasil Perhitungan



Gambar 10. Hasil perhitungan setelah melewati tahapan *text preprocessing* dan perhitungan *levenshtein distance*

## 4. SIMPULAN

Aplikasi Asisten Bahasa Inggris memberikan dampak positif kepada guru khususnya pihak instansi pendidikan dalam mendapatkan tugas siswa yang *original*. Hal ini berguna untuk menciptakan generasi muda yang jujur dan mampu berdaya saing.

## 5. SARAN

Aplikasi asisten Bahasa Inggris dapat dikembangkan lebih lanjut dengan menggabungkan algoritma yang mampu mengolah data gambar serta mengenali sinonim huruf agar tercipta sistem yang lebih baik lagi. Secara praktis, guru Bahasa Inggris dapat menggunakan aplikasi ini untuk membantu siswa mengurangi kebiasaan melakukan tindakan plagiarisme.

## DAFTAR PUSTAKA

- [1] Irianto, WA. 2014. Penentuan Tingkat Plagiarisme Dokumen Penelitian Menggunakan Centroid Linkage Hierarchical Method (CLHM). Jurnal Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya. Malang.
- [2] Pratama, B. P. & Pamungkas, S. A. 2016. Analisis Kinerja Algoritma Levenshtein Distance dalam Mendeteksi Kemiripan Dokumen Teks. Jurnal Logika, Jilid 6, No. 2, 2016, Hal. 131-143I.
- [3] Zhan Su, Byung-Ryul Ahn, Ki-yol Eom, Min-koo Kang, Jin-Pyung Kim, dan MoonKyun Kim. 2008. *Plagiarism Detection Using the Levenshtein Distance and Smith Waterman Algorithm*. The 3rd International Conference on Innovative Computing Informaion and Control, Department of Artificial Intelligence, University of Sungkyunkwan, Cheoncheon dong, Jangan-gu, Suwon, Korea.

*Halaman ini sengaja dikosongkan*