

# Penerapan Algoritma Levenshtein Distance Untuk Pencarian Judul Pada Aplikasi Lagu-Lagu Nasional

**Hyasinta Oktavia Nahampun**

Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia

Email: remiaremia14@gmail.com

Email Penulis Korespondensi: remiaremia14@gmail.com

**Abstrak-**Algoritma Levenshtein Distance adalah operasi penghapusan, penyisipan dan penukaran. Berdasarkan hasil pengujian yang telah dilakukan algoritma Levenshtein Distance mampu menghitung edit distance yang di dapatkan dari matriks yang digunakan untuk menghitung jumlah perbedaan string antara dua string. Perhitungan jarak antara dua string ini ditentukan dari jumlah minimum operasi perubahan untuk membuat string A menjadi string B. Ada 3 macam operasi utama yang dapat dilakukan oleh algoritma ini: 1. pengubahan karakter 2. penambahan karakter 3. penghapusan karakter Algoritma ini berjalan mulai dari pojok kiri atas sebuah array dua dimensi yang telah diisi sejumlah karakter string awal dan string target dan diberikan nilai cost. Nilai cost pada ujung kanan bawah menjadi nilai edit distance yang menggambarkan jumlah perbedaan dua string.

**Kata Kunci:** String Matching; Pencarian; Lagu; Algoritma Levenshtein Distance

**Abstract-**The Levenshtein Distance algorithm is a deletion, insertion and swap operation. Based on the test results that have been carried out, the Levenshtein Distance algorithm is able to calculate the edit distance obtained from the matrix used to calculate the number of strings differences between two strings. The calculation of the distance between these two strings is determined from the minimum number of change operations to make string A into string B. There are 3 main types of operations that can be performed by this algorithm: 1. change characters 2. add characters 3. delete characters This algorithm runs from the corner. top left of a two-dimensional array that has been filled with a number of characters from the initial string and the string target and given the cost value. The cost value in the lower right corner becomes the edit distance value which represents the difference between the two strings.

**Keywords:** String Matching; Search; Song; Levenshtein Distance Algorithm

## 1. PENDAHULUAN

Seiring dengan perkembangan teknologi, maka kegunaan komputer dirasa makin besar. Komputer berperan penting dalam mempermudah pekerjaan sehari-hari. Salah satu manfaat utama komputer adalah sebagai alat bantu untuk mencari lagu-lagu nasional. Berbagai aplikasi seperti joox, 9mm sudah ditanamkan pada komputer untuk mempermudah pengguna. Meskipun berbagai aplikasi sudah tersedia, bukan berarti penggunaannya tanpa masalah. Masalah yang dimaksud adalah adanya kesalahan dalam pencarian judul lagu yang sering kali disebabkan karena kesamaan pencipta lagu. Kesalahan tersebut bisa disebabkan oleh ketidaktahuan pencipta, kesalahan yang berhubungan terjadi karena nama pencipta lagu sama dan syair isi dari lagu tersebut. Kesalahan-kesalahan yang umumnya terjadi antara lain: kemiripan syair, kemiripan melodi.

Untuk permasalahan yang dialami untuk melakukan pencarian judul lagu, maka diperlukan suatu metode pendekatan pencarian string agar hasil pencarian dapat maksimal. Ada beberapa algoritma yang dapat diimplementasikan dalam memberikan kata saran yang paling mendekati dari kata yang salah penetikannya. Salah satu algoritma yang dapat digunakan adalah Levenshtein yang dapat menghitung jarak keterbedaan antara dua string[1].

Pada penelitian ini, dibangun sebuah sistem yang mampu mendeteksi tingkat kemiripan antar dokumen teks menggunakan algoritma Levenshtein distance dengan menambahkan proses case folding, tokenizing, stopword, removal, stemming, dan sorting. Proses pencocokan string pada algoritma ini dapat menghasilkan nilai distance yang menjadi penentu persentase bobot similarity. Analisa penggunaan stopword removal, stemming, dan sorting dilakukan untuk melihat pengaruhnya terhadap kinerja algoritma Levenshtein distance.[2]

## 2. METODOLOGI PENELITIAN

### 2.1 Penerapan

Penerapan adalah suatu aktifitas, aksi, tindakan adanya suatu sistem. Penerapan bukan sekedar aktivitas tetapi suatu kegiatan yang terencana yang harus dicapai untuk memenuhi tujuan kegiatan, penerapan merupakan perluasan aktifitas yang saling menyesuaikan proses interaksi antara tujuan dan tindakan untuk mencapai tujuan dalam memenuhi aktivitas kegiatan.[3]

### 2.2 Algoritma Levenshtein

Algoritma Levenshtein Distance adalah sebuah matriks string yang digunakan untuk mengukur perbedaan atau jarak (distance) antara dua string. Algoritma Levenshtein ditemukan oleh Vladimir Iosifovich Levenshtein yang merupakan ilmuwan asal Rusia pada tahun 1965.[5] Algoritma ini lebih dikenal dengan Levenshtein Distance yaitu algoritma yang mengukur kesamaan antara 2 string, Jarak Levenshtein distance diperoleh dengan mencari cara termudah untuk mengubah suatu string. Secara umum, operasi mengubah yang diperbolehkan untuk keperluan ini adalah :

1. Memasukkan karakter kedalam string,

2. Menghapus sebuah karakter dari suatu string,
3. Mengganti karakter string dengan karakter lain.

Untuk menghitung jarak, digunakan matriks  $(n+1) \times (m+1)$  dimana  $n$  adalah panjang string  $S_1$  dan  $m$  adalah panjang string  $S_2$ .

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Analisa Masalah

Kegiatan plagiat judul skripsi adalah suatu hal yang penting untuk diperhatikan. Berbagai macam cara telah ditawarkan sebagai solusi dari permasalahan ini, baik yang berefek jangka panjang maupun jangka pendek. Salah satunya adalah sistem deteksi plagiarisme. Pada beberapa string matching, dilakukan untuk menguji suatu algoritma untuk mendeteksi suatu judul skripsi yang akan dibandingkan tingkat plagiarismenya. Masalah yang akan diselesaikan dengan menggunakan sistem ini adalah bagaimana melakukan pendeteksian judul skripsi yang memiliki kesamaan identik. Metode yang digunakan untuk perbaikan citra pada penelitian ini adalah Levenshtein Distance.

Sistem deteksi plagiat secara umum dirancang untuk dapat mendeteksi kemiripan judul skripsi yang dimungkinkan kemiripan ini adalah hasil plagiat. Inputan sistem diperoleh dari judul skripsi diupload oleh user. Selanjutnya judul akan melewati tahap preprocessing menggunakan text mining sebelum dibandingkan dengan judul-judul yang ada didalam database sistem. Tahap ini terdiri dari tokenizing (memecah kalimat kedalam potongan kata sekaligus mengubah huruf menjadi kecil/lowercase), filtering (membuang stopword/ kata yang tidak deskriptif), dan stemming (mengembalikan setiap kata ke bentuk dasarnya). Setelah itu judul hasil processing akan melewati tahap processing menggunakan algoritma Levenshtein Distance. Tahap ini adalah tahap pencocokan kata. Tahap ini dimulai dari memasukkan karakter kedalam string. Kemudian menghapus sebuah karakter dari suatu string, setelah itu mengganti karakter string dengan karakter lain. Akhirnya setiap kata pada judul skripsi yang telah diinput akan memungkinkan mendapat kesamaan kata identik yang diambil dari dalam database sistem yang telah dirancang. Dan kemudian kita dapat melihat seberapa banyak kata-kata pada judul skripsi yang diinput yang memiliki banyak kesamaan identik.

#### 3.1.1 Penerapan Algoritma Levenshtein

Terdapat dua buah kata dimana dua string itu adalah LEGENDARIS dengan LENEGDI. Maka untuk mengubah string LEGENDARIS menjadi LENEGDI diperlukan 6 operasi. Secara umum, operasi mengubah yang diperbolehkan untuk keperluan ini adalah :

1. Memasukkan karakter kedalam string,
2. Menghapus sebuah karakter dari suatu string,
3. Mengganti karakter string dengan karakter lain. Maka penyelesaian untuk mengubah string LEGENDARIS menjadi LENEGDI diperlukan beberapa operasi, yaitu:
4. Mensubstitusikan G dengan N ( LEGENDARIS  $\rightarrow$  LENENDARIS)
5. Mensubstitusikan N dengan G (LENENDARIS  $\rightarrow$  LENEGDARIS)
6. Mensubstitusikan A dengan I (LENEGDARIS  $\rightarrow$  LENEGDIRIS)
7. Menghapus S (LENEKDIRIS  $\rightarrow$  LENEGDIRI)
8. Menghapus I (LENEGDIRI  $\rightarrow$  LENEGDIR)
9. Menghapus R (LENEGDIR  $\rightarrow$  LENEGDI)

Dengan menggunakan representasi matriks dapat dilihat pada tabel dibawah dan dengan ketentuan sebagai berikut

1. Jika  $S_1 = S_2 \rightarrow$  nilai yang diinput mengikuti nilai yang sebelumnya.
2. Jika  $S_1 \neq S_2 \rightarrow$  nilai yang diinput ditambahkan 1 dari nilai yang sebelumnya

Tabel 1. representasi matriks

	S1	L	E	G	E	N	D	A	R	I	S
S2	0	1	2	3	4	5	6	7	8	9	10
L	1										
E	2										
N	3										
E	4										
G	5										
D	6										
I	7										

Maka untuk mengisi titik temu pertama diantara karakter pertama pada  $S_1$  dan karakter 1 pada  $S_2$  yaitu (L-L) dengan syarat, jika kedua karakter itu mirip maka nilainya 0. Dapat dilihat pada tabel di bawah ini

Tabel 2. representasi matriks langkah 1

	S1	L	E	G	E	N	D	A	R	I	S
S2	0	1	2	3	4	5	6	7	8	9	10

	S1	L	E	G	E	N	D	A	R	I	S
L	1	0									
E	2										
N	3										
E	4										
G	5										
D	6										
I	7										

Selanjutnya untuk mengisi titik temu diantara karakter 2 pada S1 dan karakter 2 pada S2 yaitu (E-E) dengan syarat, jika kedua karakter tersebut mirip maka nilainya sama dengan nilai yang ada pada sebelumnya yaitu 0. Dapat dilihat pada tabel di bawah ini

**Tabel 3.** representasi matriks langkah 2

	S1	L	E	G	E	N	D	A	R	I	S
S2	0	1	2	3	4	5	6	7	8	9	10
L	1	0									
E	2		0								
N	3										
E	4										
G	5										
D	6										
I	7										

Tahapan berikutnya untuk mengisi titik temu diantara karakter 3 pada S1 dan karakter 3 pada S2 yaitu (G-N) dan memiliki perbedaan karakter maka nilai yang harus diisi adalah 1. Karena nilai sebelumnya adalah 0 maka ditambah dengan 1. Dapat dilihat pada tabel dibawah ini.

**Tabel 4.** representasi matriks langkah 3

	S1	L	E	G	E	N	D	A	R	I	S
S2	0	1	2	3	4	5	6	7	8	9	10
L	1	0									
E	2		0								
N	3			1							
E	4										
G	5										
D	6										
I	7										

Selanjutnya untuk mengisi titik temu antara karakter 4 pada S1 dan karakter 4 pada S2 yaitu (E-E) dengan syarat, jika kedua karakter tersebut mirip maka nilainya sama dengan nilai yang ada pada sebelumnya yaitu 1 maka nilainya sama dengan 1. Dapat dilihat pada tabel di bawah ini

**Tabel 5.** representasi matriks langkah 4

	S1	L	E	G	E	N	D	A	R	I	S
S2	0	1	2	3	4	5	6	7	8	9	10
L	1	0									
E	2		0								
N	3			1							
E	4				1						
G	5										
D	6										
I	7										

Tahapan selanjutnya untuk mengisi titik temu antara karakter 5 pada S1 dan karakter 5 pada S2 yaitu (N-G) maka nilainya adalah 2, karena kedua karakter berbeda oleh karena itu nilai yang di input adalah nilai sebelumnya (1) ditambah dengan 1 maka (1 + 1 = 2). Dapat dilihat pada tabel dibawah ini

**Tabel 6.** representasi matriks langkah 5

	S1	L	E	G	E	N	D	A	R	I	S
S2	0	1	2	3	4	5	6	7	8	9	10
L	1	0									
E	2		0								

	S1	L	E	G	E	N	D	A	R	I	S
N	3			1							
E	4				1						
G	5					2					
D	6										
I	7										

Untuk mengisi titik temu antara karakter 6 pada S1 dan karakter 6 pada S2 yaitu (D-D) maka nilainya adalah 2. Karena kedua karakter mirip maka nilai yang diinputkan sama dengan nilai yang sebelum nya. Dapat dilihat pada tabel dibawah ini

**Tabel 7.** representasi matriks langkah 6

	S1	L	E	G	E	N	D	A	R	I	S
S2	0	1	2	3	4	5	6	7	8	9	10
L	1	0									
E	2		0								
N	3			1							
E	4				1						
G	5					2					
D	6						2				
I	7										

Tahapan selanjutnya untuk mengisi titik temu antara karakter 7 pada S1 dan karakter 7 pada S2 yaitu (A-I) maka nilainya adalah 3 karena kedua karakter berbeda oleh karena itu nilai yang di input adalah nilai sebelumnya (2) ditambah dengan 1 maka (2 + 1 = 3). Dapat dilihat pada tabel dibawah ini

**Tabel 8.** representasi matriks langkah 7

	1	L	E	G	E	N	D	A	R	I	S
S2	0	1	2	3	4	5	6	7	8	9	10
L	1	0									
E	2		0								
N	3			1							
E	4				1						
G	5					2					
D	6						2				
I	7							3			

Setelah perbandingan antara karakter-karakter S1 dan S2 telah didapat, maka kolom tersebut terbagi menjadi 2 bagian yang kosong antara kiri dan kanan. Untuk mengisinya dengan cara, pada bagian kanan kolom diisi sesuai urutan nilai hasil perbandingan secara horizontal. Namun pada bagian kiri kolom diisi secara vertical sesuai urutan nilai hasil perbandingan. Dapat dilihat pada tabel dibawah ini

**Tabel 9.** representasi matriks langkah 8

	S1	L	E	G	E	N	D	A	R	I	S
S2	0	1	2	3	4	5	6	7	8	9	10
L	1	0	1	2	3	4	5	6	7	8	9
E	2	1	0	1	2	3	4	5	6	7	8
N	3	2	1	1	2	3	4	5	6	7	8
E	4	3	2	2	1	2	3	4	5	6	7
G	5	4	3	3	2	2	3	4	5	6	7
D	6	5	4	4	3	3	2	3	4	5	6
I	7	6	5	5	4	4	3	3	4	5	6

Lalu pada tabel tersebut, elemen terakhir yang diarsir merah (kanan bawah) adalah elemen yang nilainya menyatakan jarak kedua String yang dibandingkan.

Lalu untuk menghitung nilai kemiripan menggunakan rumus :

$$\text{Sim} = \left(1 - \frac{\text{Dis}}{\text{MaxLength}}\right) \times 100\% \tag{1}$$

$$\begin{aligned} \text{Sim} &= \left(1 - \frac{6}{10}\right) \times 100\% \\ &= 0,4 \times 100\% \\ &= 40\% \end{aligned}$$

Keterangan :

1. Sim = similarity/nilai kemiripan
2. Dis = jarak kedua string
3. MaxLenght = nilai string tertinggi

Maka hasil perkalian tersebut telah didapat nilai similarity/nilai kemiripan pada kedua string adalah 40%. Algoritma ini telah digunakan untuk Spell checking, Speech Recognition, DNA Analysis, Plagiarism Detection.

#### **4. KESIMPULAN**

Hasil akhir dari penelitian merupakan kesimpulan, dimana kesimpulan didapatkan dari hasil analisa yang telah dilakukan pada bab – bab sebelumnya pada penelitian ini, adapun kesimplan yang didapat dari penelitian ini yaitu Perancangan pencari lagu – lagu nasional sebagai refrensi untuk memudahkan pencarian terhadap lagu – lagu nasional. Algoritma Levenshtein Distance dapat diterapkan pada aplikasi pencarian lagu – lagu nasional dan memudahkan aplikasi untuk melakukan pencarian.

#### **REFERENCES**

- [1] I. B. K. S. Arnawa, “Implementasi Algoritma Levenshtein Pada Sistem Pencarian Judul Skripsi/Tugas Akhir,” Sist. Dan Inform., 2017.
- [2] S. A. P. B.P.Pratama, “Analisi kinerja algoritma levenshtein distance dalam medeteksi kemiripan dokumen teks,” Sist. Dan Inform., vol. 6, 2016.
- [3] Situmorang, “Pengertian Skripsi,” Sist. dan Inform., vol. 3, 2013.
- [4] Gun Gun Maulana, “Pembelajaran Dasar Algoritma Dan Pemrograman Menggunakan EL-Goritma Berbasis Web,” Tek. Mesin, 2017.
- [5] R. L. B. Muhammad Bahari Ilmy, Nitia rahmi, “Penerapan Algoritma Levenshtein Distance untuk Mengoreksi Kesalahan Pengejaan Pada Editor Teks.”
- [6] M. K. Drs. Lamhot Sitorus, Pengertian Flowchart. 2015.
- [7] D. Anika, “Simbol-simbol Flowchart,” Www.It-jurnal.com, 2016.
- [8] E. Kusnadi, “Pengertian Unified Modeling Language (UML),” www.mypctutorel.co.id, 2013.
- [9] Andre, “Mengenai Personal Home Page (PHP),” Www.duniaikom.com, 2014.
- [10] Anisya, “Aplikasi sistem Database Rumah Sakit Terpusat Pada rumah Sakit Umum AISYIYAH,” 2013.
- [11] D. N. Sari and D. P. Utomo, “Implementasi Algoritma Rabin-Karp Pada Pencarian Quotes Tokoh Terkenal,” Pelita Inform. Inf. dan Inform., vol. 9, no. 1, pp. 43–55, 2020.
- [12] A. A. B. Ginting and D. P. Utomo, “PERANCANGAN APLIKASI CATALOG WISATA DI SUMATERA UTARA MENGGUNAKAN ALGORITMA RABIN-KARP,” KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer), vol. 3, no. 1, pp. 57–63, 2019.