



Testing The Accuracy Of Fingerprint Recognition Using Levenshtein Distance And Hamming Distance Methods

Uji Ketepatan Pengenalan Sidik Jari Dengan Metode Levenshtein Distance Dan Hamming Distance

Gregorius Sakti Ginantaka^{1*}, Laurentius Kuncoro P.S.², Sri Suwarno³

Informatika, Universitas Kristen Duta Wacana, Indonesia

E-mail addresses: gregorius.sakti@ti.ukdw.ac.id

Abstract. The presence or evidence of attendance is crucial in monitoring the presence of every individual working in a particular field. Developing an employee attendance system using fingerprints can expedite the processing of data of employees who have or have not attended. One brand of machine used as a fingerprint attendance tool is Fingerspot Flexcode. The data obtained from the machine comes in the form of bitmap images that are converted into strings using encoding. Although the resulting string sequences are different, there is a possibility of similarity in fingerprint data among employees because the system cannot distinguish data precisely. Therefore, the comparison between the Levenshtein Distance and Hamming Distance methods is used to determine which method has the highest accuracy in processing the system's calculation. The method with the highest accuracy will determine the level of compatibility of the method with the tested tool. For example, 6 fingerprint data are taken from each of the 7 different employees, resulting in a total of 42 data as test data. The calculation results show that the accuracy of the Levenshtein Distance method is 80,76 % with a precision of 46,43 %, while the Hamming Distance method is 78,34 % with a precision of 30,50 % in processing string similarity in fingerprint data. Based on these results, it can be concluded that the Levenshtein Distance method is better in calculating similarity in fingerprint data compared to the Hamming Distance method because it has a higher level of accuracy and precision compared to the Hamming Distance method.

Keywords - Presence, Fingerprint, Levenshtein, Hamming, Accuracy.

Abstrak. Keberadaan atau bukti hadir sangat penting dalam memantau kehadiran setiap orang yang bekerja di bidang tertentu. Pembangunan sistem presensi karyawan dengan menggunakan sidik jari atau fingerprint dapat mempercepat proses pengolahan data karyawan yang telah hadir atau belum hadir. Salah satu merek mesin yang digunakan sebagai alat presensi sidik jari adalah Fingerspot Flexcode. Data yang diperoleh dari mesin tersebut berupa gambar bitmap yang diubah menjadi string menggunakan encoding. Meskipun deretan string yang dihasilkan berbeda, ada kemungkinan terdapat kesamaan data sidik jari antara karyawan karena sistem tidak dapat membedakan data dengan tepat. Oleh karena itu, perbandingan antara metode Levenshtein Distance dan Hamming Distance digunakan untuk mengetahui metode mana yang memiliki akurasi paling tinggi dalam pemrosesan perhitungan sistem. Metode dengan akurasi tertinggi akan menentukan tingkat kecocokan metode terhadap alat yang diuji. Sebagai contoh, 6 data sidik jari diambil dari masing-masing 7 karyawan yang berbeda sehingga terkumpul sebanyak 42 data sebagai data uji. Hasil perhitungan menunjukkan bahwa akurasi metode Levenshtein Distance adalah 80,76 % dengan presisi sebesar 46,43 % dan metode Hamming Distance adalah 78,34 % dengan presisi 30,50 % dalam memproses kemiripan string pada data sidik jari. Berdasarkan hasil tersebut, dapat disimpulkan bahwa metode Levenshtein Distance lebih baik dalam menghitung kemiripan pada data sidik jari dibandingkan dengan metode Hamming Distance karena mempunyai tingkat keakuratan dan juga presisi yang lebih tinggi dibandingkan dengan metode hamming distance.

Kata kunci - Presensi, Sidik Jari, Levenshtein, Hamming, Akurasi.

PENDAHULUAN

Gaji kepada karyawannya dan presensi ini juga digunakan sebagai pertimbangan dalam pemberian gaji karyawan. Seperti halnya jika karyawan lembur atau presensi atau bukti kehadiran merupakan hal yang tidak hadir, hal tersebut tentu akan memberi pengaruh penting untuk mengetahui dan memantau kehadiran dari terhadap gaji yang diterima. setiap orang yang bekerja disuatu bidang tertentu. Salah

Bukti dari cepatnya perkembangan dalam bidang satunya adalah jenis pekerjaan seperti karyawan

teknologi dan informasi yang ada pada zaman modern perusahaan. Pentingnya presensi atau bukti kehadiran ini ini salah satunya adalah biometrik, sidik jari atau dikarenakan setiap perusahaan pasti akan memberikan fingerprint yang merupakan salah satu identifikasi pribadi manusia dapat dimanfaatkan untuk mempermudah sistem dalam membedakan data tiap karyawan yang ada.

Pembangunan sistem presensi karyawan dengan menggunakan sidik jari atau fingerprint ini tentunya dapat mempercepat proses pengolahan data karyawan yang sudah tercatat hadir dan yang belum hadir. Pengembangan sistem ini tentu akan memudahkan

perusahaan yang mempunyai karyawan dalam jumlah banyak untuk melakukan proses presensi kehadiran secara digital atau melalui sistem sehingga tidak menggunakan cara lama atau cara konvensional secara satu persatu.

Salah satu merk dari mesin yang digunakan sebagai alat presensi sidik jari atau fingerprint ini adalah Fingerspot Flexcode. Data yang didapatkan dari mesin tersebut adalah berupa gambar berformat bitmap yang kemudian akan diubah dengan encoding menjadi bentuk string. Hasil dari masing-masing deret string tersebut tentunya akan berbeda, tetapi tidak menutup kemungkinan bahwa akan ada kesamaan data sidik jari antar karyawan.

Hal itu terjadi karena sistem tidak dapat membedakan data secara tepat, oleh karena itu perbandingan penggunaan metode Levenshtein Distance dan Hamming Distance digunakan untuk mengetahui manakah dari kedua metode tersebut yang memiliki nilai keakuratan paling tinggi dalam pemrosesan perhitungan yang ada pada sistem. Hal ini akan menghasilkan nilai akurasi tertinggi dan menentukan tingkat kecocokan metode terhadap alat yang akan diuji. Sebagai sampel dari pengujian ini penulis akan mengambil sebanyak 6 data sidik jari dari masing-masing 7 orang karyawan yang berbeda sehingga akan terkumpul sebanyak 42 data untuk dijadikan sebagai data uji.

Penelitian mengenai biometric fusion system yang dimana tujuan dilakukannya penelitian adalah untuk mengetahui apakah terjadi peningkatan akurasi fungsi dari biometric yang menggunakan pengenalan terhadap wajah dan juga sidik jari atau fingerprint. Dengan menggunakan teknik pendekatan Whale Optimized Algorithm (WOA) bersama dengan Maximally Stable Extremal Region (MSER) untuk mengekstraksi point fitur dari gambar mereka melakukan pendeteksian pola menggunakan SVM untuk mendapatkan tingkat akurasi yang diukur berdasarkan gambar yang digunakan sebagai data penelitian. Dari penelitian yang dilakukan didapatkan bahwa tingkat akurasi yang dihasilkan oleh biometric fusion system sebesar 91,6 % yang dimana tingkat presisinya sebesar 12,3 %.[1]

Pengembangan sebuah sistem deteksi gender dan perhitungannya berdasarkan dari sidik jari atau fingerprint ini jarang dijumpai. Dalam melakukan penelitian dimana mereka membuat sebuah klasifikasi pada fingerprint atau sidik jari untuk mendeteksi gender menggunakan metode fingerprint global feature. Dari penelitian yang dilakukan oleh mereka, hasil yang didapatkan yaitu sebesar 74,5 % pendeteksian klasifikasi pada gender laki-laki dan perempuan berhasil dilakukan. Berdasarkan dari penelitian ini pula metode yang digunakan sangat direkomendasikan untuk digunakan kembali apabila akan melakukan penelitian serupa yang berupa klasifikasi.[2]

Penelitian untuk pengenalan error yang mungkin saja terjadi dalam pengejaan dalam Bahasa Arab. Pengenalan error terhadap pengejaan Bahasa Arab ini berfokus pada inserting, deleting, dan permutation. Penelitian ini dilakukan karena terinspirasi dari algoritma levenshtein dimana digunakan untuk pendeteksian error yang serupa pada sebuah kata atau rangkaian huruf. Hasil yang diperoleh berdasarkan penelitian yang dilakukan berupa penurunan tingkat error dari percobaan pertama yang sebesar 10 % menjadi 1,57 %.[3]

Membangun sebuah sistem yang mampu mendeteksi tingkat kemiripan antar dokumen teks menggunakan

algoritma Levenshtein distance dengan menambahkan proses seperti case folding, tokenization, stop word deletion, stemming dan sorting. Proses pencocokan string dari algoritma ini dapat menghasilkan nilai jarak yang menentukan persen kemiripan terbobot. Analisis stop word removal, derivasi, dan pengurutan yang digunakan dilakukan untuk melihat pengaruhnya terhadap kinerja algoritma Levenshtein distance. Simulasi algoritma ini dilakukan dengan dua dataset dan satu dataset real. Hasil simulasi menunjukkan bahwa sorting berpengaruh besar terhadap algoritma Levenshtein distance. Hasil terbaik pada dataset 1 diperoleh untuk proses yang menggunakan stopwords deletion, forwarding dan sorting secara bersamaan. Hasil terbaik pada dataset 2 terlihat pada proses yang menggunakan stopwords dan wire yang dikombinasikan dengan sorting. Hasil terbaik dari data real diperoleh dalam proses stemming-sorting.[4]

Pengujian algoritma dengan menggunakan aplikasi pendeteksi plagiarisme untuk menghitung skor kemiripan dan waktu proses yang dihasilkan oleh kedua algoritma tersebut. Tujuan dari penelitian ini adalah untuk mengukur perbandingan kinerja dari algoritma Levenshtein Distance dan algoritma Jaro Winkler untuk mendeteksi plagiarisme dalam dokumen teks. Data uji yang digunakan dalam penelitian ini terdiri dari dua data uji, yaitu data uji untuk mengukur kesamaan algoritma dan data uji untuk mengukur waktu pemrosesan algoritma. Setelah dilakukan pengujian, hasil dari kedua pengujian tersebut dirata-ratakan kemudian dianalisis perbandingannya.

Hasil analisis perbandingan rata-rata kesamaan algoritma Jaro-Winkler adalah 80,92%, sedangkan nilai kesamaan rata-rata algoritma Levenshtein-Distance adalah 49,43%. Kemudian dilakukan analisis perbandingan waktu proses rata-rata algoritma JaroWinkler adalah 0,054 detik, sedangkan waktu proses rata-rata algoritma Levenshtein-Distance adalah 0,138 detik. Berdasarkan analisis komparatif, algoritme Jaro Winkler terbukti memiliki akurasi kesamaan yang lebih tinggi dan waktu pemrosesan yang lebih cepat daripada algoritme jarak Levenshtein dalam deteksi plagiarisme dokumen.[5]

Berdasarkan penelitian didapatkan bahwa proses daur ulang teks yang telah didefinisikan sebagai penggunaan sumber tulisan yang ada merujuk untuk menulis teks baru. Oleh karena itu, diperlukan sistem deteksi kemiripan teks untuk mengenali teks mana yang tumpang tindih dan mana yang hampir tumpang tindih. Oleh karena itu, penelitian ini berfokus pada pendeteksian kesamaan teks menggunakan algoritma simhash. Algoritma ini membuat sidik jari dokumen yang bertindak sebagai fitur dokumen untuk membandingkan kesamaan tekstual dengan pengukuran jarak menggunakan metode hamming. Dalam pengujian yang berfokus pada dokumen duplikat dan hampir duplikat, tingkat pemulihan dokumen cukup tinggi yaitu sebesar 80%.[6]

A. Biometrik

Biometric merupakan sebuah alat yang berkembang saat ini dibidang teknologi dan informasi, yang mana kegunaan dari alat ini adalah sebagai alat untuk mengetahui ukuran dan juga melakukan perhitungan pada tubuh manusia yang setelah itu digunakan sebagai patokan dalam menunjukkan dan menentukan karakteristik dari manusia itu. Dalam penggunaannya

biometric membutuhkan fase verifikasi dan juga identifikasi.

Sidik jari berarti suatu pemeriksaan dengan suatu alat untuk mengetahui garis rekaman dari jari. Biasanya jari yang digunakan adalah ibu jari. Sidik jari dapat ditinggalkan pada suatu benda yang pernah tersentuh dengan telapak tangan atau pun telapak kaki.[7]

Untuk mengetahui pemilik dari sidik jari bisa digunakan sebuah alat yaitu fingerprint. Fingerprint ini adalah suatu alat yang memiliki sensor yang bisa mengetahui sidik jari seseorang. Hal tersebut sering digunakan sebagai verifikasi identitas. Fingerprint sudah banyak digunakan di banyak tempat bahkan di alat elektronik seperti smartphone, pintu masuk, dan juga alat absensi karyawan, dsb.

B. Approximate String Matching

Approximate String Matching ialah sebuah algoritma proses untuk menemukan kemiripan atau persamaan yang ada pada dua buah string berbeda yang berdasarkan pada segi penulisan baik dalam susunan atau jumlah dari karakternya. Persentase kemiripan string ini dilihat dari jauh tidaknya perbedaan antara string satu dengan yang lainnya.[8][9]

Levenshtein Distance adalah metrik string yang digunakan melakukan pengukuran perbedaan antara dua buah barisan kata (string). Secara informal, Levenshtein Distance yang ada antara dua buah string merupakan jumlah minimum dalam melakukan perubahan sebuah karakter baik insert, delete, maupun substitution yang dibutuhkan dalam melakukan perubahan sebuah string menjadi string lain.[10][11]

Hamming Distance merupakan sebuah metode yang digunakan untuk melakukan perhitungan pada dua buah string yang memiliki panjang sama, dimana setiap daripada string tersebut dikodekan terlebih dahulu dalam bentuk biner lalu dihitung nilai minimum substitusi yang dibutuhkan untuk mengubah sebuah string menjadi string yang lainnya.[6]

C. Rumus Nilai Kemiripan (Similarity)

Rumus nomor 1 merupakan rumus nilai kemiripan (Similarity).

$$Similarity = \left(1 - \left(\frac{Distance}{MaxLenghtString}\right)\right) * 100 \% \{1\}$$

Similarity = Nilai kemiripan string.
Distance = Jarak perbedaan antar string.
MaxLenghtString = Panjang Maksimal string.

D. Rumus Nilai Rerata (Threshold)

Rumus nomor 2 merupakan rumus nilai rerata (Threshold).

$$hreshold = \left(\frac{1 + x_2 + x_3 + x_4 + x_5}{x \text{ Total Similarity}}\right) * 100 \% \{2\}$$

Threshold = Nilai batas similarity.
X = Nilai similarity.
Total Similarity = Banyaknya nilai similarity.

E. Rumus Nilai Presisi

Rumus nomor 3 merupakan rumus nilai presisi.

$$Precision = \left(\frac{TP}{TP + FP}\right) * 100 \% \{3\}$$

Precision = Nilai presisi.
TP = True positif.
FP = False positif.

F. Rumus Nilai Akurasi

Rumus nomor 4 merupakan rumus nilai akurasi.

$$Accuration = \left(\frac{TP}{TP + TN + FP + FN}\right) * 100 \% \{4\}$$

Accuration = Nilai akurasi.
TP = True positif.
TN = True Negatif.
Total Dataset = Total jumlah data uji.

METODE PENELITIAN

A. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem yang dilakukan oleh penulis pada penelitian kali ini berguna untuk mengetahui apa saja yang diperlukan dalam melakukan penelitian khususnya kebutuhan akan perangkat keras maupun perangkat lunak yang ada dan mendukung dalam berjalannya proses penelitian. Analisis kebutuhan sistem ini terbagi menjadi 2 (dua) yaitu kebutuhan non fungsional dan kebutuhan fungsional.

Untuk melakukan proses penelitian ini penulis menggunakan bantuan perangkat keras laptop merk MSI Seri GF63 11SC dengan spesifikasi processor Tiger Lake i5 11400H dan perangkat keras fingerprint merk Flexcode seri 4500 dengan spesifikasi support sistem U.are.U 4500. Penulis juga menggunakan beberapa aplikasi perangkat lunak seperti Microsoft Visual Studio, Heidi Sql, My SQL Workbench, bahasa dan library pemrograman VB.Net dan SQL.

B. Perancangan Penelitian

Pada penelitian ini penulis melakukan beberapa penatalaksanaan yang sudah diurutkan berdasarkan pada kebutuhan dalam pelaksanaan penelitian seperti berikut :

a. Studi Literatur

Pada tahap ini penulis melakukan pengumpulan pustaka atau informasi secara teoritis yang berkaitan dengan penelitian yang dilakukan oleh penulis. Pustaka atau informasi yang sudah terkumpul tersebut kemudian dikaji berdasarkan pada sumber-sumber yang valid seperti tesis, artikel, maupun jurnal yang resmi.

b. Pengumpulan Data

Data sampel yang digunakan oleh penulis sebanyak jumlah populasi yang ada pada sebuah kantor di Yogyakarta, dimana karyawan yang berkerja di kantor tersebut sebanyak 7 orang dengan rincian pengambilan data sidik jari pada masing-masing karyawan adalah sebanyak 6 kali dengan pengambilan pertama digunakan sebagai acuan.

Berdasarkan pengambilan data tersebut didapatkan sebanyak 42 data sidik jari yang terkumpul dan akan digunakan sebagai sampel pada penelitian yang dilakukan oleh penulis. Data yang diambil oleh penulis merupakan data mentah atau raw data yang berupa

gambar sidik jari dengan format bitmap.

c. Pengolahan Data

Setelah data selesai dikumpulkan maka dilakukan perhitungan berapa jarak perbedaan antar data sidik jari yang ada untuk kemudian setelah ditemukan jarak yang dihitung, dilakukan perhitungan kembali untuk menentukan nilai kemiripan pada setiap data yang telah dibandingkan dengan levenshtein distance dan juga hamming distance.

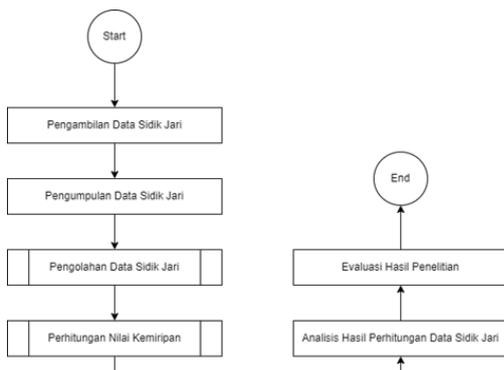
d. Evaluasi

Pada evaluasi kali ini, dari setiap hasil yang telah didapatkan setelah proses perhitungan yang dilakukan maka hasil tersebut akan dievaluasi kembali untuk mengetahui dan menemukan apakah terdapat kesalahan dalam proses perhitungan dan pengolahan data.

Apabila ditemukan adanya kesalahan dalam melakukan proses pengolahan data dan perhitungan maka akan dilakukan perhitungan kembali dan apabila tidak ditemukan adanya kesalahan yang berarti pada proses perhitungan maka data yang diperoleh dapat dinyatakan sebagai data yang valid atas dasar perhitungan yang dilakukan.

C. Blok Diagram Penelitian

Blok diagram penelitian dapat dilihat pada gambar 1 berikut :



Gambar 1. Blok Diagram Penelitian

HASIL DAN PEMBAHASAN

Pada penelitian ini, yang dilakukan oleh penulis di tahap implementasi awal adalah pengambilan serta pengumpulan data untuk penelitian. Data yang diambil merupakan data gambar yang memiliki format bitmap. Penggunaan format bitmap dengan alasan bahwa format bitmap tersebut mempunyai keunggulan apabila gambar yang disimpan mempunyai ukuran yang sama, maka setelah dilakukan proses encoding hasil string yang didapatkan akan mempunyai jumlah karakter yang sama dan tidak akan berbeda panjang atau jumlah stringnya.

A. Implementasi Sistem

Setelah data yang digunakan untuk penelitian terkumpul, maka langkah selanjutnya yang dilakukan oleh penulis adalah sebagai berikut : a. Pengambilan dan Pengumpulan Dataset .

Kegiatan penelitian dimulai dengan melakukan pengumpulan data sidik jari pada sebuah kantor di Yogyakarta dimana jumlah populasi karyawan berjumlah 7 orang dengan rincian pengambilan data sidik jari sebanyak 6x untuk setiap orangnya yang ditunjukkan oleh Gambar 2.



Gambar 2. Contoh Dataset

a. Pengubahan Ukuran Dataset (Resize)

Pada saat pengambilan dan pengumpulan data didapatkan bahwa data gambar yang terkumpul mempunyai ukuran sebesar 411 KB yang dimana ukuran tersebut terlalu besar untuk diolah ke dalam bentuk string. Oleh karena itu penulis melakukan proses pengubahan ukuran data atau resize data sebesar 50 %, yang dimana keputusan melakukan pengubahan gambar sebanyak 50 % dilakukan untuk menghindari rusaknya data.

Proses pengubahan data menggunakan bantuan website resize dengan domain aspose.app serta menggunakan ketentuan bahwa data yang diubah ukurannya tetap berformat bitmap. Data yang telah diubah ukurannya atau resize melalui website tersebut menghasilkan data bitmap berukuran 137 KB.

b. Pemampatan Dataset (Compress)

Setelah dilakukan proses pengubahan ukuran data, tahapan selanjutnya yang dilakukan penulis dalam pengolahan data adalah pemampatan data atau compress data menggunakan bantuan website dengan domain yang sama yaitu aspose.app yang dimana ketentuan dalam pemampatan data adalah menggunakan compression type RGB dan palette bits sebesar 1 seperti yang terlihat pada Gambar 3.



Gambar 3. Hasil Compress Dataset

c. Pengubahan Dataset Menjadi String (Encode)

Proses pengubahan data gambar menjadi data string melalui proses encoding. Pada tahap ini dipilihnya proses

encoding sebagai jembatan untuk mengubah data yang berupa gambar menjadi data berbentuk string untuk memudahkan proses perhitungan menggunakan metode levenshtein dan juga metode hamming.

Dalam proses pengubahannya penulis menggunakan bantuan website dengan domain codebeautify.org.

d. Entry Dataset ke Database

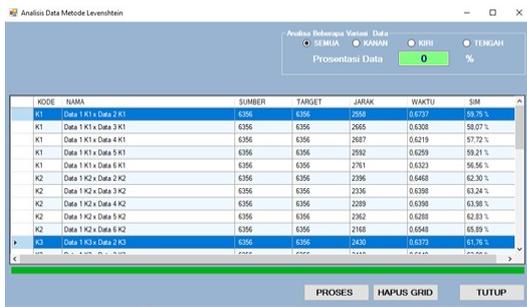
Pada tahap ini, setelah proses perubahan dataset menjadi string selesai maka yang dilakukan penulis adalah proses penginputan data ke dalam database untuk membantu proses perhitungan dengan masing-masing metode yang ada. Proses penginputan data pada database menggunakan bantuan dari program yang telah penulis buat.

Proses penginputan data sidik jari mempunyai ketentuan sebagai berikut:

1. Kode data digunakan sebagai inisial pada proses penginputan data, contohnya penginputan data untuk karyawan 1, maka kode data yang digunakan adalah "K1" apabila penginputan data digunakan untuk karyawan 1 dan karyawan 2 maka kode data yang digunakan adalah "K1K2" dan seterusnya.
2. Nama data digunakan sebagai penanda dari data acuan dan data pembanding yang diinputkan, contohnya data yang diinputkan adalah data 1 sidik jari dan data 2 sidik jari dari karyawan 1, maka nama data yang digunakan adalah "Data 1 K1 x Data 2 K1".
3. Data 1 digunakan sebagai data acuan.
4. Data 2, 3, 4, 5, dan 6 digunakan sebagai data pembanding.

B. Perhitungan Metode Levenshtein

Pada tahap proses perhitungan dengan metode levenshtein distance penulis menggunakan program pada menu "Analisi Data Metode Levenshtein" yang telah dibuat untuk memudahkan proses perhitungan seperti yang ditunjukkan oleh Gambar 4.



Gambar 4. Tampilan Menu Metode Levenshtein

Hasil perhitungan metode levenshtein dapat dilihat pada tabel 1.

Tabel 1. Hasil Perhitungan Metode Levenshtein

Kode	Total Hasil Similarity (%)	Threshold (%)
K1	291,31	58,26
K2	318,24	63,65
K3	309,63	61,93
K4	344,12	68,82
K5	319,06	63,81
K6	285,87	57,17
K7	300,23	60,05

Setelah perhitungan dilakukan didapatkan data hasil yang tertampil dalam Tabel 1. Berdasarkan pada tabel tersebut dilakukan pula proses perhitungan treshold dengan hasil yang tertampil pada tabel 2.

Tabel 2. Hasil Perhitungan Threshold Metode Levenshtein

Kode	Similarity (%)				
	Hasil 1	Hasil 2	Hasil 3	Hasil 4	Hasil 5
K1	59,75	58,07	57,72	59,21	56,56
K2	62,30	63,24	63,98	62,83	65,89
K3	61,76	62,08	61,65	60,46	63,68
K4	68,09	70,89	66,67	69,80	68,67
K5	64,91	65,37	62,02	64,82	61,94
K6	55,41	58,51	57,28	57,04	57,63
K7	60,99	58,99	59,92	60,38	59,95

Perhitungan yang dilakukan selanjutnya adalah mencari akurasi dari metode levenshtein distance yang digunakan dengan melakukan perbandingan data menggunakan Data 1 sebagai acuan dan data 2, 3, 4, 5, dan 6 sebagai pembanding. Apabila hasil perhitungan yang didapatkan melebihi threshold yang ditentukan maka data dinyatakan mirip dan apabila hasil yang didapatkan kurang dari threshold yang ditentukan maka data dinyatakan berbeda seperti yang ditunjukkan oleh Tabel 3.

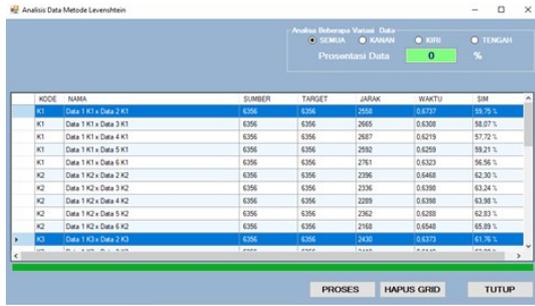
Tabel 3. Hasil Analisis Perhitungan Metode Levenshtein

Data Uji	Data Acuan							Rata-rata
	K 1	K 2	K 3	K 4	K 5	K 6	K 7	
K1	2	1	0	0	0	0	1	
K2	0	2	0	0	0	0	4	
K3	1	0	2	0	0	0	2	
K4	2	0	5	2	0	0	5	
K5	5	0	3	0	3	0	3	
K6	0	0	0	0	0	3	0	
K7	1	0	0	0	0	0	2	
Presisi	5	3	4	1	2	1	6	46
	0	3	0	4	1	0	7	%
	%	%	%	%	%	0	%	%
Akurasi	7	9	7	7	7	1	6	81
	8	0	8	6	8	0	7	%
	%	%	%	%	%	0	%	%

C. Perhitungan Metode Hamming

Pada tahap proses perhitungan dengan metode hamming distance penulis menggunakan program pada menu "Analisi Data Metode Hamming" yang telah dibuat untuk memudahkan proses perhitungan yang ditunjukkan

oleh Gambar 5.



Gambar 5. Tampilan Menu Metode Hamming

Hasil perhitungan metode hamming bisa dilihat pada tabel 4 :

Tabel 4. Hasil Perhitungan Metode Hamming Similarity (%)

Kode	Similarity (%)				
	Hasil 1	Hasil 2	Hasil 3	Hasil 4	Hasil 5
K1	58,38	56,51	56,40	57,72	55,01
K2	60,76	61,86	62,38	61,35	64,34
K3	60,55	60,88	60,16	59,15	62,08
K4	67,40	70,37	65,84	69,24	67,90
K5	63,81	64,26	59,78	63,26	60,39
K6	54,12	55,63	54,73	54,75	56,12
K7	60,28	57,94	58,82	59,50	58,82
K5	63,81	64,26	59,78	63,26	60,39
K6	54,12	55,63	54,73	54,75	56,12
K7	60,28	57,94	58,82	59,50	58,82

Setelah perhitungan dilakukan didapatkan data hasil yang tertampil dalam Tabel IV. Berdasarkan pada tabel tersebut dilakukan pula proses perhitungan treshold dengan hasil yang tertampil pada Tabel 5.

Tabel 5. Hasil Perhitungan Threshold Metode Hamming

Kode	Total Hasil Similarity (%)	Threshold (%)
K1	284,02	56,80
K2	310,69	62,14
K3	302,82	60,56
K4	340,75	68,15
K5	311,50	62,30
K6	275,35	55,07
K7	295,36	59,07

Perhitungan yang dilakukan selanjutnya adalah mencari akurasi dari metode hamming distance yang digunakan dengan melakukan perbandingan data menggunakan Data 1 sebagai acuan dan data 2, 3, 4, 5, dan 6 sebagai pembanding. Apabila hasil perhitungan yang didapatkan melebihi threshold yang ditentukan maka data dinyatakan mirip dan apabila hasil yang didapatkan kurang dari threshold yang ditentukan maka data dinyatakan berbeda.

Berikut ini adalah tabel hasil perhitungan menggunakan metode hamming, yang ditunjukkan oleh

Tabel 6.

Tabel 6. Hasil Analisis Perhitungan Metode Hamming

	Data Acuan							Rata-rata
	K 1	K 2	K 3	K 4	K 5	K 6	K 7	
Data Uji	K 1	2	1	1	0	0	0	1
K 2	1	2	0	0	0	0	0	4
K 3	2	0	2	0	0	1	2	
K 4	5	1	5	2	1	2	5	
K 5	0	0	3	0	3	2	3	
K 6	2	0	0	0	0	2	0	
K 7	4	0	0	0	0	1	2	
Presisi	4	2	2	1	2	5	2	31
	0	9	9	0	7	0	9	%
	%	%	%	%	%	%	%	%
Akurasi	7	8	7	6	8	8	6	78
	3	9	7	9	5	7	8	%
	%	%	%	%	%	%	%	%

KESIMPULAN DAN SARAN

Pada hasil dari perhitungan yang telah dilakukan didapatkan bahwa akurasi dari metode levenshtein distance dalam mengukur kemiripan string pada data sidik jari yang ada dan telah diolah ke dalam bentuk string adalah sebesar 80,76 % dengan rata-rata presisi data sebanyak 46,43 % dan untuk metode hamming distance dalam proses perhitungan kemiripan antara data string sidik jari satu dan sidik jari lainnya mempunyai nilai akurasi yang sebesar 78,34 % dengan rata-rata presisi data sebanyak 30,50 %. Berdasarkan pada hasil yang didapatkan, dapat disimpulkan bahwa metode levenshtein distance merupakan metode yang lebih baik dalam perhitungan nilai kemiripan pada data sidik jari dibandingkan dengan metode hamming distance dikarenakan untuk deteksi kemiripan string yang dilakukan oleh keduanya, metode levenshtein distance mempunyai tingkat keakuratan dan juga presisi yang lebih tinggi dibandingkan dengan metode hamming distance.

Berdasarkan dari penelitian yang telah dilakukan oleh penulis dan dengan menimbang beberapa kekurangan yang terdapat pada penelitian ini, maka berikut beberapa saran serta usulan dari penulis bagi peneliti berikutnya yang ingin melakukan penelitian serupa dengan metode yang berbeda untuk mendapatkan hasil yang lebih maksimal:

- Perbanyak pengambilan data dan coba untuk melakukan clening data atau preprosesing data dengan cara atau proses yang lebih baik dan maksimal tanpa ada kecacatan pada data uji.

- b. Mencari dan menambahkan lebih banyak referensi mengenai penggunaan metode yang tepat dalam proses perhitungan data yang ingin diuji.
- c. Usahakan melakukan testing terlebih dahulu terhadap data real dan bukan data dummy dalam program yang digunakan sebagai alat bantu agar terhindar dari errornya program.

REFERENSI

- [1] Kumar, T., Bhushan, S., & Jangra, S. (2021). An Improved Biometric Fusion System of Fingerprint and Face using Whale Optimization. *International Journal of Advanced Computer Science and Applications*, 12(1), 664–671. <https://doi.org/10.14569/IJACSA.2021.0120176>
- [2] Abdullah, S. F., Abas, Z. A., Rahman, A. F. N. A., & Saad, W. H. M. (2016). Development of a Fingerprint Gender Classification Algorithm Using Fingerprint Global Features. *International Journal of Advanced Computer Science and Applications*, 7(6), 275–279. <https://doi.org/10.14569/ijacsa.2016.070635M>.
- [3] Rianti, S., & Supono, R. A. (2019). Perbandingan Algoritma Edit Distance, Levenshtein Distance, Hamming Distance, Jaccard Similarity Dalam Mendeteksi String Matching. 10(2), 71–76.
- [4] S. R. M. H. K. Mufti Ari Bianto, "Perancangan Sistem Pendeteksi Plagiarisme Terhadap Topik Penelitian Menggunakan Metode K-Means Clustering dan Model Bayesian," *Seminar Nasional Teknologi Informasi dan Multimedia 2018*, p. 19, 2018.
- [5] M. Julian Tanga, S. Rahman, and Hasniati, "Analisis Perbandingan Algoritma Levenshtein Distance dan Jaro Winkler Untuk Aplikasi deteksi Plagiarisme Dokumen Teks," *JTRISTE*, vol. 4, no. 1, pp. 44–54, 2017.
- [6] Anggelina, L. Dwi Krisnawati, and D. Sebastian, "Penerapan Simhash dan Hamming distance dalam Deteksi kemiripan Teks Berita," *J. Terap. Teknol. Inf.*, vol. 6, no. 2, pp. 131–141, 2022, doi: 10.21460/jutei.2022.62.216.
- [7] A. Iskandar, Muhajirin, and Lisah, "Sistem Keamanan Pintu Berbasis Arduino Mega," *J. Inform. Upgris*, vol. 3, no. 2, pp. 99–104, 2017, doi: 10.26877/jiu.v3i2.1803.
- [8] T. Kopelowitz and E. Porat, "A Simple Algorithm for Approximating the Text-To-Pattern Hamming Distance," *Open Access Ser. Informatics*, vol. 61, no. 683064, pp. 1–5, 2018, doi: 10.4230/OASICS.SOSA.2018.10.
- [9] Hakak, S. et al. (2019) 'Exact String Matching Algorithms ', *IEEE Access. IEEE*, PP(c), p. 1. doi: 10.1109/ACCESS.2019.2914071.
- [10] M. F. Azhri, D. Swanjaya, and R. K. Niswatin, "Penerapan Algoritma Levenshtein Distance pada Aplikasi Asisten Guru Bahasa Inggris," *Semin. Nas. Inov. Teknol.*, pp. 155–160, 2019.
- [11] Levenshtein Distance," *Jurnal Pinter* Vol. 1 No. 1 Juni 2017, p. 24, 2017.
- [12] Krstinić, D., Braović, M., Šerić, L., & Božić-Štulić, D. (2020). Multi-label Classifier Performance Evaluation with Confusion Matrix. 01–14. <https://doi.org/10.5121/csit.2020.100801>
- [13] Fiarni, C., Gunawan, A. S., & Anthony, I. (2022). Sistem Deteksi Penyalahgunaan Promosi Menggunakan Metode Similarity dan Penilaian Risiko. 11(3), 168–175.
- [14] S. Adi, "Penerapan Algoritma Rabin Karp Untuk Medeteksi Kemiripan Dua Dokumen Teks," *J. Mantik Penusa*, vol. 22, no. 1, pp. 125–130, 2018, [Online]. Available: <https://e-jurnal.pelitanusantara.ac.id/index.php/mantik/article/view/472/277>
- [15] Markoulidakis, I., Kopsiaftis, G., Rallis, I., & Georgoulas, I. (2021). Multi-Class Confusion Matrix Reduction method and its application on Net Promoter Score classification problem. *ACM International Conference Proceeding Series*, Cx, 412–419. <https://doi.org/10.1145/3453892.3461323>

Conflict of Interest Statement:

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Article History:

Received: 29 January 2023 | Accepted: 01 March 2023 | Published: 30 April 2023