



Implementasi Sistem Pendeteksian Kemiripan Judul Skripsi Dengan Algoritma Levenshtein Distance Pada Perpustakaan Universitas Pamulang

Adi Setiawan¹, Saprudin^{1*}

¹Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia
Email: adisetoawan124@gmail.com, dosen00845@unpam.ac.id

Abstrak – Penelitian ini membahas penerapan algoritma string matching Levenshtein distance dalam melakukan pencocokan sebuah string dengan metode pengembangan waterfall. Masalah yang dihadapi adalah sulitnya mendapatkan referensi terkait kemiripan judul skripsi yang sudah dan belum. Dalam penelitian ini, penulis mengembangkan sistem pendeteksian kemiripan judul skripsi dengan text preprocessing dan algoritma Levenshtein distance sebagai solusi untuk mengatasi masalah ini. Studi kasus dilakukan pada perpustakaan Universitas Pamulang. Hasil penelitian menunjukkan bahwa sistem yang dikembangkan dapat mengidentifikasi kesamaan antara judul skripsi dengan tingkat akurasi yang baik. Diharapkan sistem ini dapat membantu mahasiswa dalam mempercepat pengecekan judul skripsi yang diajukan dan menghindari judul yang hampir sama.

Kata Kunci : Levenshtein Distance, Text Preprocessing, Kemiripan Judul Skripsi

Abstract – This study discusses the application of the Levenshtein distance string matching algorithm in matching a string with the waterfall development method. The problem faced is the difficulty of getting references related to the closeness of the thesis titles that have been and have not. In this study, the authors developed a similarity detection system for thesis titles with text preprocessing and the Levenshtein distance algorithm as a solution to this problem. The case study was conducted at the Pamulang University library. The results showed that the developed system could identify convenience between thesis titles with a good degree of accuracy. It is hoped that this system can help students speed up checking the title of the proposed thesis and avoid having almost the same title.

Keywords: Levenshtein Distance, Text Preprocessing, Thesis Title Similarities

1. PENDAHULUAN

Teknologi adalah cara ilmiah untuk mencapai tujuan praktis dari ilmu terapan. Dengan bantuan teknologi, kita bisa menyediakan barang dan jasa yang sangat penting untuk kehidupan manusia. Teknologi berperan sebagai sarana untuk mengaplikasikan ilmu pengetahuan dalam kehidupan sehari-hari dengan cara yang lebih bermanfaat. Dalam perkembangan umat manusia, pemanfaatan teknologi telah sangat membantu dan berdampak pada munculnya nilai-nilai baru dalam kehidupan sosial.

Dalam dunia teknologi data, pencarian string adalah salah satu masalah yang paling umum dan untuk mengatasi masalah ini, penerapan algoritma string matching dapat berfungsi sebagai solusi yang layak. Untuk melakukan pencarian sebuah data atau informasi tidak terlepas dari pencocokan *string* dimana dari hasil pencocokan inilah akan ditemukan pola kalimat yang dicari (Prayoga, 2019). Algoritma String Matching, juga disebut sebagai algoritma pencocokan string, adalah algoritma pencarian untuk semua kejadian String Pendek dan Panjang. String Pendek disebut sebagai *string* pola dan String Panjang sebagai *string* teks. String Pattern adalah string dengan panjang m karakter ($m < n$). String Teks adalah String Panjang dengan panjang n karakter.

Masalah yang dihadapi mahasiswa saat pengajuan judul tugas akhir universitas pamulang ini adalah sulit mendapatkan referensi dari sebuah judul skripsi yang sudah ada atau belum. Saat ini belum ada sistem yang dapat mendeteksi kesamaan antar judul skripsi, sedangkan banyak mahasiswa yang memiliki judul skripsi yang hampir sama.



Dalam penelitian ini membahas tentang penerapan algoritma *string matching levenshtein distance* dalam melakukan pencocokan sebuah *string* dengan metode pengembangan *waterfall*.

2. METODE

2.1 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut:

- a. Observasi, pengumpulan data dengan secara langsung pada objek yang diteliti untuk menemukan informasi yang diperlukan berhubungan dengan penelitian.
- b. Wawancara, pengumpulan data dilakukan dengan cara mengadakan tanya jawab secara langsung kepada narasumber yang ada kaitanya dengan judul yang diambil.
- c. Studi Kepustakaan, Pengumpulan data dilakukan dengan cara *study* literatur, yaitu dengan memahami dan melakukan pengumpulan data dari artikel, buku-buku, dokumen dan bacaan-bacaan yang berkaitan dengan judul penelitian yang bersumber dari internet.

2.2. Teori

- a. Levenshtein Distance

Menurut Junedy(2014) pada Michael Julian Tannga, Syaiful Rahman dan Hasniati, 2017) [7], *Levenshtein Distance* merupakan algoritma yang digunakan untuk mengukur nilai kesamaan atau kemiripan antara dua buah kata (*string*).

Levenshtein Distance merupakan sebuah nilai numerik yang merepresentasikan jumlah minimum dari operasi pengeditan yang diperlukan untuk mengubah satu *string* menjadi *string* lainnya. Operasi pengeditan tersebut meliputi penyisipan karakter (*insertion*), penghapusan karakter (*deletion*), dan penggantian karakter (*substitution*).

Dengan memanfaatkan *Levenshtein Distance*, dapat membandingkan kedua *string* dan menentukan tingkat kemiripannya dengan cara mengukur seberapa jauh perbedaan antara keduanya. Algoritma ini sangat berguna dalam pengolahan teks dan analisis data, karena dapat membantu dalam pencarian dan pemrosesan informasi dengan lebih efektif dan akurat.

- b. Langkah-Langkah Algoritma *Levenshtein Distance*

1. $S = \text{pattern/pola}$, dan $T = \text{String Target}$
2. Berikut ini proses algoritma *levenshtein distance* dalam mendapatkan jarak karakter [8].
3. Langkah 1: Inisialisasi
 - a) Hitung panjang S dan T, misalkan m dan n
 - b) Buat matriks berukuran 0...m baris dan 0...n kolom
 - c) Inisialisasi baris pertama dengan 0...n d) Inisialisasi kolom pertama dengan 0...m
4. Langkah 2 : proses
 - a) Periksa $S[i]$ untuk $1 < i < n$
 - b) Periksa $T[j]$ untuk $1 < j < m$
 - c) Jika $S[i] = T[j]$, maka entrinya adalah nilai yang terletak pada tepat didiagonal atas sebelah kiri, yaitu $d[i,j] = d[i-1,j-1]$
 - d) Jika $S[i] \neq T[j]$, maka entrinya adalah $d[i,j]$ minimum dari: Nilai yang terletak tepat diatasnya, ditambah satu, yaitu $d[i,j-1]+1$ Nilai yang terletak tepat dikirinya, ditambah satu, yaitu $d[i-1,j]+1$ terletak pada tepat didiagonal atas sebelah kirinya, ditambah satu, yaitu

5. $d[i-1,j-1]+1$
6. Langkah 3: Hasil entri matriks pada baris ke-i dan kolom ke j, yaitu $d[i,j]$.
7. Langkah 2 diulang hingga entri $d[m,n]$ ditemukan.

c. *Entity-Relationship Diagram (ERD)*

Entity Relationship Diagram adalah suatu pemodelan berbasis pada persepsi dunia nyata yang mana terdiri dari kumpulan objek dasar yang disebut dengan entitas (*entity*) dan hubungan diantara objek-objek tersebut dengan menggunakan perangkat konseptual dalam bentuk diagram (Lubis, 2016: 38)

d. Logical Record Structure (LRS)

LRS terdiri dari *link* diantara tipe *record* lainnya, banyaknya *link* dari LRS yang diberi nama oleh *filed-filed* yang kelihatan pada kedua *link* tipe *record* (Sukmaindrayana & Sidik, 2017: 35)

e. Pengujian Black Box

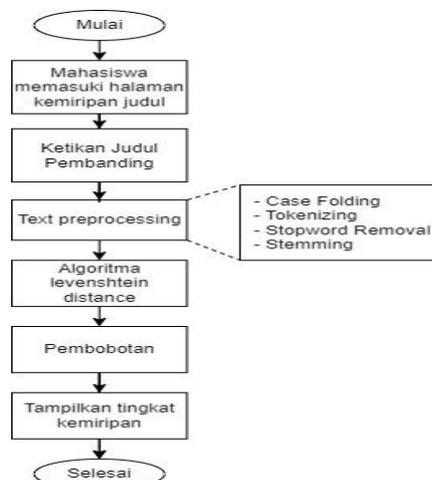
Pengujian kotak hiram dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan (Shalahuddin, 2018: 275).

3. ANALISA DAN PEMBAHASAN

3.1 Skema Sistem

Penjelasan berikut ini adalah mengenai skema sistem yang akan dibuat dalam pengembangan sebuah program. Program tersebut memiliki beberapa proses, diantaranya:

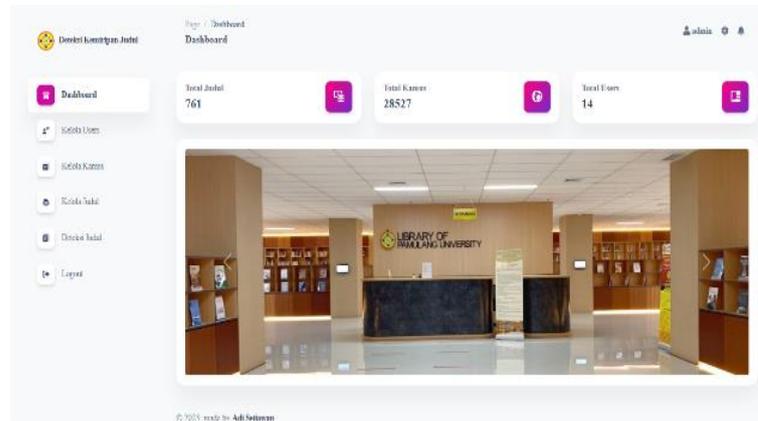
1. Pertama, terdapat proses awal, di mana sistem akan mulai berjalan.
2. Kedua, masukan berupa judul yang akan dicek kemiripannya.
3. Ketiga, setelah mendapatkan masukan judul, sistem akan melakukan proses Preprocessing dimulai dari : *case folding*, *tokenizing*, *stopword removal* dan *stemming*. Proses ini dilakukan untuk membersihkan teks yang tidak relevan, seperti tanda baca, simbol-simbol, kata umum yang tidak memiliki makna lalu merubahnya ke kata dasar. Hal ini dilakukan agar teks yang akan dianalisis lebih mudah diproses dan menghasilkan hasil yang lebih akurat.
4. Keempat, mencari jarak dari judul 1 dan judul lainnya dengan algoritma *levenshtein distance*.
5. Kelima, menghitung pembobotan dari hasil *levenshtein distance*.
6. Keenam, tampilkan tingkat kemiripan.



Gambar 1. Skema Sistem

3.2 Implementasi

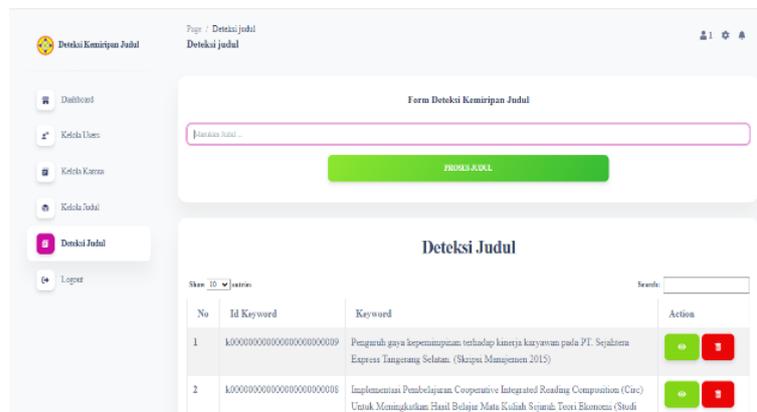
a. Implementasi dashboard admin



Gambar 2. Tampilan Halaman Dashboard Admin

Tampilan ini merupakan halaman utama yang dapat diakses pada saat akun yang digunakan adalah akun *admin*.

b. Implementasi deteksi judul



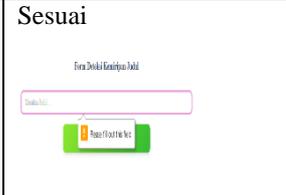
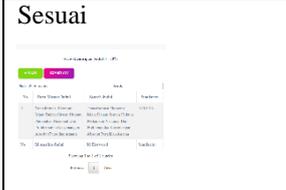
Gambar 3. Tampilan Halaman Deteksi Judul Skripsi

Tampilan halaman deteksi judul, pada halaman ini mahasiswa dapat melakukan pengelolaan data (menambah, *view* analisa, dan menghapus) *keyword*.

3.3 Pengujian Black Box Halaman Deteksi Judul

Tabel 1. Pengujian *Black Box* Halaman Deteksi Judul

No	Skenario Pengujian	Uji Kasus	Hasil Yang Diharapkan	Hasil Pengujian
1.	Isi <i>form</i> deteksi kemiripan judul lalu klik <i>button</i> proses judul	Isi dengan : Transformasi Ekonomi Islam Dalam Sistem Hukum Perbankan Nasional Dan Problematika Kewenangan Absolut Peradilan Agama	Sistem akan menampilkan data deteksi judul	Sesuai 

2.	Isi <i>form</i> deteksi kemiripan judul tanpa mengisi data	<i>Field</i> dikosongkan	Sistem akan menolak dan menampilkan pesan peringatan	Sesuai 
3.	Klik <i>button</i> view detail	Sistem akan menampilkan hasil tingkat kemiripan judul	Sistem akan menampilkan hasil tingkat kemiripan judul	Sesuai 
4.	Klik <i>button delete</i> pada kolom <i>action</i>	Klik <i>yes</i> pada modal hapus	Sistem akan menampilkan pesan peringatan "Proses <i>delete keyword</i> berhasil"	Sesuai 

4. KESIMPULAN

Kesimpulan Berdasarkan uraian sebelumnya, dapat diambil beberapa kesimpulan, yaitu:

- Dengan adanya sistem pendeteksian ini dapat mempermudah mahasiswa menemukan referensi judul yang belum ada dan sudah ada.
- Dengan adanya aplikasi ini dapat mengurangi terjadinya pengambilan judul skripsi yang hampir sama.

REFERENCES

- Prayoga, A. H. (2019). Algoritma Pencarian String Dengan Menggunakan Brute Force Pada Perpustakaan Univeristas Muhammadiyah Jember. *Teknik Informatika Fakultas Teknik*, 1–13. Diambil dari <http://repository.unmuhjember.ac.id/7110/>
- Tangga, M. J., Rahman, S., Informatika, T., & Hasniati. (2017). Analisis Perbandingan Algoritma Levenshtein Distance Dan Jaro Winkler Untuk Aplikasi Deteksi Plagiarisme Dokumen Teks. *JTRISTE*, 4(1), 44-54.
- Sukmaindrayana, A., & Sidik, R. (2017). Jurnal manajemen informatika. *JURNAL MANAJEMEN INFORMATIKA*, 4(2), 31–40.
- Lubis, A. (2016). *Basis Data Dasar*. Sleman: Deepublish.
- Shalahuddin, M., & Rosa. (2018). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: INFORMATIKA.