

PENERAPAN *FUZZY STRING MATCHING* PADA APLIKASI PENCARIAN TUGAS AKHIR MAHASISWA JURUSAN SISTEM INFORMASI BERBASIS WEB (Studi Kasus: Fakultas Sains dan Teknologi UIN Suska Riau)

¹Ardi Isbad Amar Gurning, ²Zarnelly, ³Arabiatal Adawiyah

^{1,2,3}Jurusan Sistem Informasi Fakultas Sains dan Teknologi UIN Suska Riau

Jl. HR.Soebrantas KM.18 Panam Pekanbaru-Riau

Email: ¹gurningardi@gmail.com, ²zarnelly@gmail.com, ³adawiyaharabiatal@gmail.com

ABSTRAK

Pada saat ini, untuk mengetahui judul tugas akhir yang telah ada dan pernah diajukan di Jurusan Sistem Informasi, mahasiswa perlu mencari dan membaca tugas akhir yang ada di perpustakaan. Tentu ini tidak efektif dikarenakan tugas akhir yang masuk akan terus bertambah setiap semesternya. Selain itu, mahasiswa membutuhkan waktu lebih dalam peminjaman tugas akhir di perpustakaan, karena harus mengikuti prosedur peminjaman di perpustakaan. Tugas akhir tersebut perlu bagi mahasiswa sebagai materi pelajaran atau sebagai referensi bagi penelitian mereka. Selain itu, dosen koordinator tugas akhir dalam melakukan pencarian data tugas akhir, harus melakukan pencarian dari dokumen tertulis dan dokumen di dalam komputer. Dokumen yang terdapat dalam komputer tersebut berupa kelengkapan administrasi tugas akhir seperti file surat menyurat dan form nilai tugas akhir. Untuk itu perlu dibuat sebuah aplikasi yang dapat mengakomodir masalah diatas yaitu sebuah aplikasi yang dapat mengelola data tugas akhir dan juga melakukan pencarian didalamnya. Untuk mencari informasi yang tepat digunakan pula teknik pencarian yang tepat. Salah satu teknik yang digunakan adalah Fuzzy String Matching. Teknik ini akan mencari kata yang sesuai dan juga string yang mendekati dengan string yang dicari. Salah satu algoritma untuk melakukan pencarian tersebut adalah Levenshtein Distance. Algoritma tersebut dijalankan dengan melakukan tiga operasi yaitu penghapusan, penyisipan dan penukaran. Dengan menggunakan aplikasi berbasis web, pengguna dapat mengakses saat terhubung dengan koneksi internet. Dari implementasi teknik dan algoritma diatas, aplikasi ini dapat membantu koordinator tugas akhir dalam melakukan pengelolaan data-data tugas akhir mahasiswa. Selain itu, mahasiswa sebagai pengguna, dapat terbantu dalam mencari judul yang pernah diterima sebelumnya, yang dapat digunakan sebagai referensi penelitian.

Kata kunci: *fuzzy string matching, levenshtein distance, pencarian, tugas akhir, web*

I. PENDAHULUAN

Perkembangan teknologi sudah dirasakan penting oleh manusia dalam era globalisasi saat ini. Hal itu terjadi karena kemajuan teknologi yang ada tidak dapat dipisahkan dari kehidupan sehari-hari. Salah satunya adalah pemanfaatan teknologi dalam pencarian informasi.

Informasi menjadi salah satu kebutuhan dalam sebuah institusi. Informasi dimanfaatkan untuk menambah pengetahuan, mengurangi resiko kegagalan dan membantu pengguna dalam pengambilan keputusan yang diinginkan. Salah satu alat yang dapat digunakan dalam mencari informasi adalah dengan menggunakan mesin pencari.

Mesin pencari merupakan program komputer yang dirancang agar mampu menemukan informasi yang dicari yang dicari dari banyaknya kumpulan informasi yang tersedia [5]. Mesin pencari dibutuhkan dalam mencari informasi yang diinginkan. Dengan mengetikkan informasi pada mesin pencari, informasi yang dibutuhkan akan didapatkan. Namun dalam sebuah penelitian

terhadap mesin pencari menyimpulkan bahwa rata-rata kesalahan dalam pengetikan kata yang dicari yang dilakukan oleh pengguna cukup tinggi [3], akibat kesalahan dalam pengetikan kata yang dicari, pengguna tidak dapat menemukan informasi yang diinginkan.

Informasi yang dibutuhkan tentu merupakan informasi yang lengkap dan akurat. Informasi yang lengkap dan akurat dapat dicari dengan pemanfaatan teknologi dan teknik yang tepat. Untuk itu diperlukan teknik dalam mencocokkan dua string yang berbeda, yaitu antara string inputan dan string sumber yang ada pada database.

Teknik yang digunakan penulis adalah dengan menggunakan metode pencarian pencocokan string (String Matching). Metode pencocokan string (String Matching) yang merupakan bagian dalam proses pencarian string digunakan untuk mendapatkan data yang sesuai dengan kebutuhan informasi. Fuzzy String Matching (inexact String Matching) merupakan pencocokan string yang melakukan pencarian terhadap string yang sama dan juga string yang

mendekati dengan string lain yang terkumpul dalam sebuah database. Pencocokan string ini masih dapat dibedakan menjadi dua yaitu berdasarkan kemiripan penulisan (approximate String Matching) dan berdasarkan kemiripan ucapan (phonetic String Matching).

Pada saat ini, untuk mengetahui judul tugas akhir yang telah ada dan pernah diajukan di Jurusan Sistem Informasi, mahasiswa perlu mencari dan membaca tugas akhir yang ada di perpustakaan. Tentu ini tidak efektif dikarenakan tugas akhir yang masuk akan terus bertambah setiap semesternya. Selain itu, mahasiswa membutuhkan waktu lebih dalam peminjaman tugas akhir di perpustakaan, karena harus mengikuti prosedur peminjaman di perpustakaan. Tugas akhir tersebut perlu bagi mahasiswa sebagai materi pelajaran atau sebagai referensi bagi penelitian mereka. Selain itu, dosen koordinator tugas akhir dalam melakukan pencarian data tugas akhir, harus melakukan pencarian dari dokumen tertulis dan dokumen di dalam komputer. Dokumen yang terdapat dalam komputer tersebut berupa kelengkapan administrasi tugas akhir seperti file surat menyurat dan form nilai tugas akhir.

Pada sebuah penelitian dengan judul Rancang Bangun Prototype Mesin Pencari String Menggunakan Metode Fuzzy String Matching [5], peneliti tersebut menggunakan algoritma Knuth Morris Pratt dalam pencarian string dengan hasil pencarian berupa informasi URL dan metadata (informasi situs) yang sesuai dengan keyword yang dimasukkan. Dilanjutkan dengan penelitian dengan judul Perbandingan Algoritma String Searching Brute Force, Knuth Morris Pratt, Booyer Moore, dan Karp Rabin Pada Teks Alkitab Bahasa Indonesia [9], penelitian tersebut berkaitan dengan hubungan antara panjang pola yang dicari dengan waktu penemuannya. Sedangkan algoritma Levenshtein distance melakukan modifikasi dengan mengubah suatu string menjadi string yang lain sehingga prosesnya lebih sederhana [2]. Dalam penelitian lain disebutkan salah satu algoritma Approximate String Matching dengan menggunakan algoritma Levenshtein distance yang digunakan dalam pencarian string berdasarkan pendekatan perkiraan [1]. Pada penelitian itu, algoritma Levenshtein distance digunakan untuk menambah fitur autocomplete, yang dapat membantu pengetikan. Sehingga pada penelitian ini, penulis akan menggunakan algoritma Levenshtein Distance untuk pencarian string yang mendekati dengan kata yang dicari .

II. LANDASAN TEORI

Metode Fuzzy String Matching diarahkan untuk mencari nilai dari beberapa string yang mendekati dan tidak hanya mengasilkan cocok atau tidak cocok [8]. Konsep Fuzzy String Matching [4]:

A. Fuzzy String Matching

Fuzzy String Matching adalah salah satu metode pencarian string yang menggunakan proses pendekatan terhadap pola dari string yang dicari. Melakukan pencarian terhadap string yang sama dan juga string yang mendekati dengan string lain yang terkumpul dalam sebuah penampung atau kamus.

Kunci dari konsep pencarian ini adalah bagaimana memutuskan bahwa sebuah string yang dicari memiliki kesamaan dengan string tertampung di kamus, meskipun tidak sama persis dalam susunan karakternya. Untuk memutuskan 'kesamaan' ini dipergunakan sebuah fungsi yang diistilahkan sebagai similarity function. Fungsi ini akan bertugas memutuskan string hasil pencarian jika ditemukan string hasil pendekatan (aproksimasi).

B. Approximate String Matching

Approximate String Matching merupakan pencocokan string dengan dasar kemiripan dari segi penulisannya (jumlah karakter dan susunan karakter), tingkat kemiripan ditentukan dengan jauh tidaknya beda penulisan dua buah string yang dibandingkan tersebut [5].

Operasi mengubah string tersebut dapat berupa mengubah satu huruf ke huruf yang lain, menghapus satu huruf dari string, atau memasukkan satu huruf ke dalam string. Operasi-operasi ini digunakan untuk menghitung jumlah perbedaan yang diperlukan untuk pertimbangan kecocokan suatu string dengan string sumber, jumlah perbedaan tersebut diperoleh dari penjumlahan semua perubahan yang terjadi dari masing-masing operasi.

Operasi untuk mengubah string tersebut ada 3 operasi [1], yaitu:

1. Operasi Penghapusan
Operasi ini dilakukan dengan menghapus karakter pada suatu kata agar menyesuaikan string sumber (S) dengan string target (T).
2. Operasi Penyisipan
Operasi ini dilakukan dengan menyisipkan karakter pada indeks tertentu untuk menyamakan string sumber (S) dengan string target (T).
3. Operasi Penukaran
Operasi ini dilakukan dengan menukar suatu karakter untuk menyamakan string sumber (S) dengan string target (T)

C. Algoritma Levenshtein Distance

Algoritma Levenshtein distance ditemukan oleh ilmuwan asal Rusia bernama Vladimir Levenshtein pada tahun 1963 [7], algoritma ini juga disebut dengan algoritma Edit Distance. Perhitungan edit distance didapatkan dari matriks yang digunakan untuk menghitung jumlah perbedaan string antara dua string, sebagai contoh

hasil penggunaan algoritma ini, string “komputer” dan “computer” memiliki distance 1 karena hanya perlu dilakukan satu operasi saja untuk mengubah satu string ke string yang lain. Dalam kasus dua string di atas, string “computer” dapat menjadi “komputer” hanya dengan melakukan satu penukaran karakter “c” menjadi “k” (Andhika, 2010).

Algoritma Levenshtein distance digunakan secara luas dalam berbagai bidang, misalnya mesin pencari, pengecek ejaan (spell checking), pengenalan pembicaraan (speech recognition), pengucapan dialek, analisis DNA, pendeteksi pemalsuan, dan lain-lain. Algoritma ini menghitung jumlah operasi string paling sedikit yang diperlukan untuk mentransformasikan suatu string menjadi string yang lain [1].

Algoritma Levenshtein distance bekerja dengan menghitung jumlah minimum penranformasian suatu string menjadi string lain yang meliputi penghapusan, penyisipan, dan penukaran.

Selisih perbedaan antar string dapat diperoleh dengan memeriksa apakah suatu string sumber sesuai dengan string target. Nilai selisih perbedaan ini disebut juga edit distance atau jarak Levenhstein. Jarak Levenshtein antar string “s” dan string “t” tersebut adalah fungsi D yang memetakan (s,t) ke suatu bilangan real nonnegatif, sebagai contoh diberikan dua buah string $s = s(1)s(2),s(3),\dots,s(m)$ dan $t = t(1),t(2),t(3),\dots,t(n)$ dengan $|s| = m$ dan $|t| = n$ sepanjang alfabet V berukuran r sehingga “s” dan “t” anggota dari V^* . $s(j)$ adalah karakter pada posisi ke-j pada string “s” dan $t(i)$ adalah karakter pada posisi ke-i pada string “t”. Sehingga jarak Levenshtein dapat didefinisikan sebagai (Harahap, 2013).

$$D(s, t) = d(s_1, t_1) + d(s_2, t_2) + \dots + d(s_i, t_i)$$

$$D(s, t) = \sum_{i=1}^l d(s_i, t_i) \quad (2.1)$$

dimana : $s_i, t_i \in V$ untuk $i = 1, 2, \dots, l$

$$d(s_i, t_i) = 0 \quad \text{jika } s_i = t_i \text{ dan}$$

$$d(s_i, t_i) = 1 \quad \text{jika } s_i \neq t_i$$

$D(s,t)$ adalah banyaknya operasi minimum dari operasi penghapusan, penyisipan dan penukaran untuk menyamakan string s dan t. Pada implementasi pencocokan antar string, ketiga operasi tersebut dapat dilakukan sekaligus untuk menyamakan string sumber dengan string target.

III. ANALISA DAN PERANCANGAN

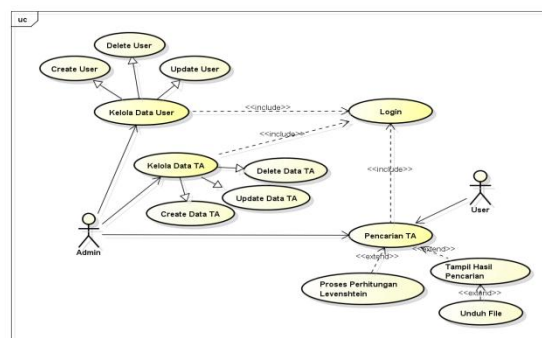
A. Analisis Sistem yang Berjalan

Proses sistem yang berjalan dimulai dari proses persetujuan laporan tugas akhir. Dimana tugas akhir yang telah disidangkan dan dinyatakan lulus, laporan tersebut terlebih dahulu diperiksa oleh pembimbing dan penguji disesuaikan dengan berita acara sidang tugas akhir. Laporan yang telah

disetujui oleh pembimbing akan diserahkan kepada Ketua Jurusan untuk diperiksa format penulisan. Setelah itu lalu diserahkan kepada Dekan untuk disahkan dan ditandatangani. Laporan yang telah disahkan lalu dijilid keras sesuai ketentuan Pedoman Sistem dan Prosedur Tugas Akhir. Laporan tersebut diserahkan kepada Fakultas dan Universitas berupa hardcopy dan softcopy (dalam bentuk CD). Hardcopy dalam bentuk jilid keras yang diserahkan kepada Fakultas dan Universitas akan menjadi arsip di perpustakaan, baik perpustakaan Fakultas maupun perpustakaan Universitas. Sehingga mahasiswa yang ingin melihat tugas akhir yang telah selesai, dapat membaca di perpustakaan Fakultas atau perpustakaan Universitas.

B. Perancangan Model Sistem Usulan

B.1. Use Case Diagram

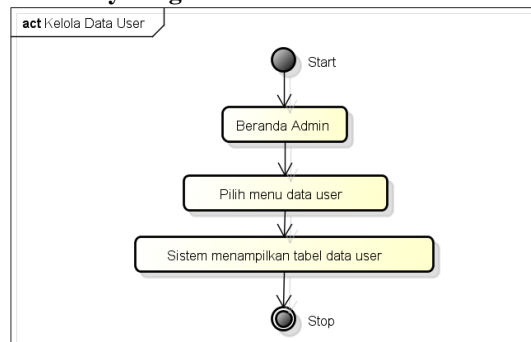


Gambar 1. Use case diagram

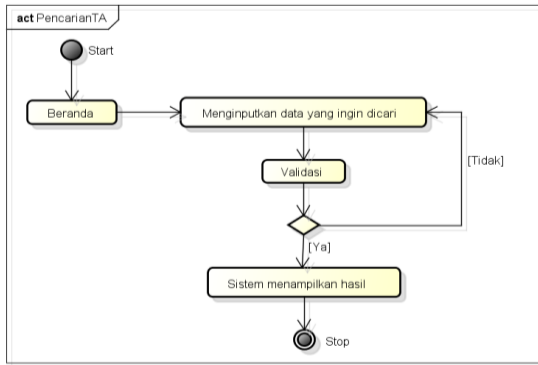
Penjelasan aktor dari use case diagram diatas:

Admin yaitu koordinator tugas akhir memiliki hak akses untuk mengelola data tugas akhir dan juga melakukan pencarian. User yaitu dosen dan mahasiswa, dimana mereka dapat melakukan pencarian dan mengubah password pribadi.

B.2. Activity Diagram

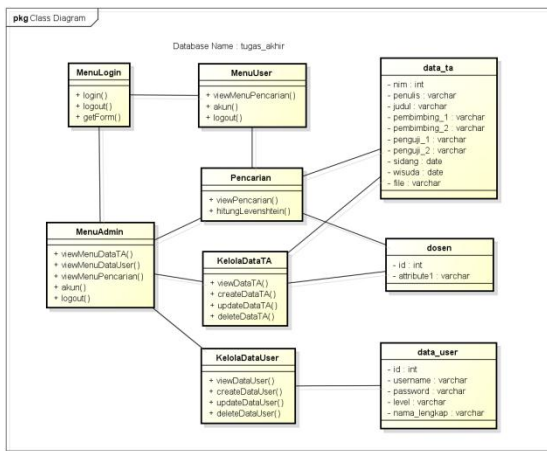


Gambar 2. Activity diagram kelola data user



Gambar 3. Activity diagram pencarian TA

B.3. Class Diagram



Gambar 4. Class diagram

C. Perancangan Database

C.1. Tabel Data User

Tabel 1. Tabel data user

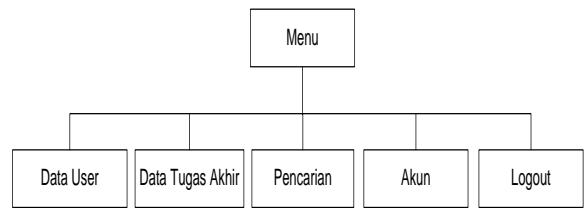
Fields	Type	Length
id_user*	Int	30
Username	varchar	50
Password	varchar	50
Level	int	5

C.2. Tabel Data Tugas Akhir

Tabel 2. Tabel data tugas akhir

Fields	Type	Length
nim*	int	13
Penulis	varchar	50
Judul	varchar	255
pembimbing_1	varchar	50
pembimbing_2	varchar	50
penguji_1	varchar	50
penguji_2	varchar	50
Sidang	date	-
Wisuda	varchar	20
File	varchar	100

D. Rancangan Menu



Gambar 5. Rancangan menu

E. Rancangan Antarmuka

E.1. Rancangan Halaman Pencarian

No.	Judul	Penulis	NIM	File
1				lihat file
2				lihat file
3				lihat file

Gambar 6. Rancangan halaman pencarian

E.2 Rancangan Halaman Lihat Data TA

Gambar 7. Rancangan halaman lihat data TA

IV. IMPLEMENTASI DAN PENGUJIAN

A. Batasan Implementasi

Adapun batasan implementasi dalam pembuatan sistem ini antara lain:

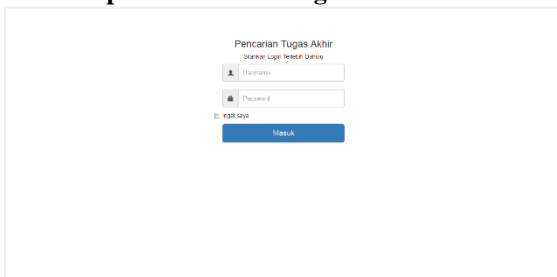
- Sistem yang dibangun merupakan sistem yang berbasis web.
- Sistem ini menampilkan data informasi user, data informasi tugas akhir dan pencarian tugas akhir.
- Terdapat hak akses pengguna terhadap sistem, yaitu administrator dan user dimana masing-masing pengguna dapat menggunakan sistem sesuai hak akses masing-masing.
- Sistem diimplementasikan untuk mempermudah pengguna dalam pencarian informasi mengenai tugas akhir yang ada pada jurusan Sistem Informasi. Menggunakan framework Twitter Bootstrap dan MySQL sebagai database serta XAMPP sebagai localhost server.

- e. Menggunakan algoritma Levenshtein distance sebagai algoritma pencarian, dengan jarak levenshtein sebesar 20.

B. Implementasi Tampilan Antarmuka

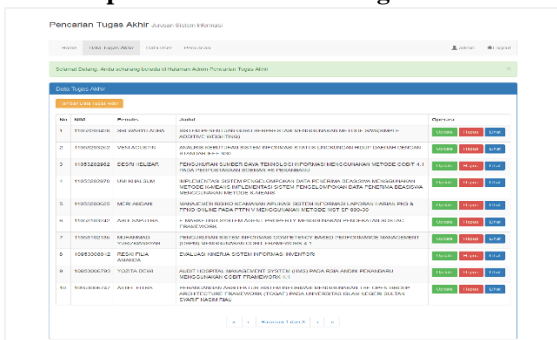
Untuk memulai aplikasi, pastikan XAMPP telah terinstall di PC sehingga dapat dijalankan pada browser. Kemudian meng-copy file program ke dalam folder htdocs, kemudian agar database dapat terbuka maka lakukan import database (tugas_akhir.sql) ke dalam phpmyadmin. Kemudian memasukkan fungsi levenshtein ke dalam database.

B.1. Tampilan Halaman Login



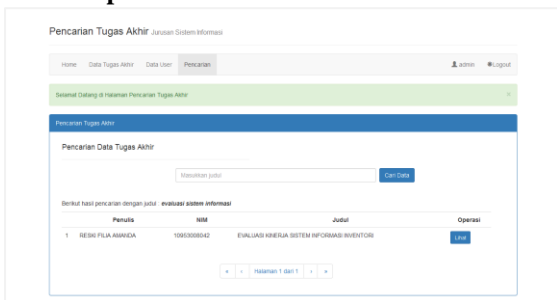
Gambar 8. Halaman login

B.2. Tampilan Halaman Data Tugas Akhir



Gambar 9. Halaman data tugas akhir

B.3. Tampilan Halaman Pencarian



Gambar 10. Halaman pencarian

C. Pengujian Sistem

C.1. Pengujian Blackbox

Berikut merupakan detail pengujian yang akan dilakukan:

Tabel 3. Tabel detail pengujian

Kelas Uji	Detail Pengujian
Login	Verifikasi data login user dan admin dengan memasukkan username dan password
Pengolahan data user	Create data user Update data user Delete data user
Pengolahan data tugasakhir	Create data tugasakhir Update data tugasakhir Delete data tugasakhir Lihatprofil data tugasakhir
Pengujian Proses Pencarian	Proses pencarian data tugasakhir
PengaturanAkun	Proses mengaturakunpribadi

Setelah dilakukan pengujian terhadap aplikasi berdasarkan detail pengujian diatas, didapatkan hasil setiap detail pengujian berjalan dengan sukses.

C.2. Pengujian User Acceptance Test

Proses pengujian oleh pengguna dan menghasilkan dokumen untuk dijadikan bukti bahwa aplikasi yang dikembangkan dapat diterima user dan hasil pengujiannya dianggap memenuhi kebutuhan. Dalam pengujian user acceptance test ini, dibagi atas tiga aspek kategori yaitu:

- a. Aspek Rekayasa Perangkat Lunak
- b. Aspek Fungsionalitas
- c. Aspek Komunikasi Visual

Dari hasil pengujian berdasarkan tiga aspek diatas dan dengan lima skala pengujian dari mulai dari buruk hingga sangat baik, didapat 54,17% pengguna menilai bahwa aplikasi ini bernilai baik seperti yang terlihat pada Tabel 4.

Tabel 4. Hasil pengujian user acceptance test

No	Hasil Pengujian	Persentase (%)
1	Sangat Buruk	0
2	Buruk	8,33
3	Cukup Baik	37,5
4	Baik	54,17
5	Sangat Baik	0

IV. PENUTUP

A. Kesimpulan

Berdasarkan implementasi dan pengujian sistem yang telah dilakukan maka dapat ditarik beberapa kesimpulan yaitu:

- 1. Aplikasi ini dapat membantu dosen dan mahasiswa melakukan pencarian tugas akhir, yang dapat dijadikan sebagai referensi penelitian.
- 2. Aplikasi ini dapat membantu koordinator dalam mengelola data tugas akhir yang ada.
- 3. Algoritma Levenshtein distance dapat menampilkan hasil pencarian untuk judul tugas akhir yang mendekati dengan judul tugas akhir yang diketikkan, namun jika nilai jarak levenshtein yang diperoleh terlalu besar

maka kemungkinan judul tugas akhir yang berkaitan dengan judul tugas akhir yang dicari tidak dapat ditampilkan.

B. Saran

Saran yang dapat diberikan dari hasil penelitian untuk pengembangan sistem ini ke depan adalah sebagai berikut:

1. Untuk mendapatkan hasil pencarian yang maksimal, maka aplikasi ini dapat dikembangkan dengan melakukan penambahan metode seperti algoritma pencocokan string yang lain.
2. Menggunakan metode perankingan dan pengindeksan dalam penyajian hasil pencarian sehingga hasil yang paling relevan disajikan pada urutan pertama.

REFERENSI

- [1] Adiwidya, B. M. . Algoritma Levenshtein Dalam Pendekatan Approximate String Matching. Institut Teknologi Bandung , Bandung. 2009
- [2] Ardiyanto, R. I., "Dynamic Programming Dalam Levenshtein Distance Untuk Mengetahui Keterbedaan Dua String". Institut Teknologi Bandung, Bandung. 2008
- [3] Benisius. "Sistem Pengoreksian Kata Kunci dengan Menggunakan Metode Levenshtein Distance". Tesis. 2010
- [4] Dewanto, R. A., & Aradea. "Aplikasi SMS Gateway dengan Koreksi Kesalahan Menggunakan Fuzzy String Matching". Universitas Siliwangi. 2009
- [5] Haryanto, B. " Sistem Manajemen Basis Data". Informatika, Bandung. 2004
- [6] Haryanto, E. V. "Rancang Bangun Prototype Mesin Pencari String Menggunakan Metode Fuzzy String Matching". Konferensi Nasional Sistem dan Informatika,halaman 76. STMIK Potensi Utama, Bali, 2011
- [7] Junaedy S., R. "Perancangan Aplikasi Deteksi Kemiripan Isi Dokumen Teks Dengan Menggunakan Metode Levenshtein Distance". Pelita Informatika Budi Darma, hal. 161. 2014
- [8] Syaroni, M., & Munir, R. "Pencocokan String Berdasarkan Kemiripan Ucapan (Phonetic String Matching)". Jurnal. 2011
- [9] Utomo, D., Harjo, E. W., & Handoko. "Perbandingan Algoritma String Searching Brute Force, Knuthmorris Pratt, Boyer Moore, Dan Karp Rabin Pada Teks Alkitab Bahasa Indonesia". Jurnal. 2008