

## Implementasi Algoritma *Levenshtein Distance* Untuk Pencarian Judul Skripsi Dan Menambahkan Fitur *Autocomplete* Pada *Website Repository Skripsi Jurusan Teknik Informatika Universitas Palangka Raya*

Elsa Apriliansi<sup>1)</sup>, Ariesta Lestari<sup>2\*)</sup>, Abertun Sagit Sahay<sup>3)</sup>

<sup>1) 2) 3)</sup> Teknik Informatika, Fakultas Teknik, Universitas Palangka Raya  
Jl. Hendrik Timang Kampus UPR, Kota Palangka Raya

\*corresponding author

<sup>1)</sup> elsapriliansi@gmail.com

<sup>2)</sup> aries22ta@gmail.com

<sup>3)</sup> abertun@gmail.com

### Abstrak

Skripsi merupakan suatu karya tulis ilmiah berupa paparan tulisan hasil penelitian. Dalam menyimpan skripsi yang cukup banyak memerlukan waktu yang lama untuk mencari data skripsi satu persatu, sehingga memerlukan media penyimpanan yang memfasilitasi kecepatan dalam mencari skripsi seperti sebuah repository. Repository berbasis website merupakan salah satu media penyimpanan yang sangat dibutuhkan untuk menyimpan data-data skripsi tersebut. Dalam membuat website repository sendiri menggunakan bahasa pemrograman PHP, database MySQL dengan model perancangan waterfall melalui tahapan analisis, perancangan, pemrograman, pengujian dan pemeliharaan. Dalam website ini juga ditambahkan algoritma Levenshtein Distance untuk mengoptimalkan pencarian agar waktu pencarian menjadi lebih efisien dan menambahkan fitur autocomplete yang berfungsi untuk memberikan prediksi atau saran kata sesuai dengan karakter yang diketikkan oleh pengguna sehingga membantu pengguna agar lebih mencari kata yang ingin dicari..

Berdasarkan hasil analisis dan pengujian, penerapan algoritma Levenshtein Distance dalam mencari judul skripsi mampu menampilkan judul skripsi yang memiliki nilai Distance yang terkecil atau terdekat dengan kata yang diinputkan oleh pengguna. Dan untuk autocomplete sendiri juga mampu menampilkan prediksi atau saran kata yang sesuai dengan kata yang di ketik dan hasil perhitungan untuk pencarian single target memiliki keakuratan 75% dan multi target dengan 2 kata memiliki keakuratan 50%. sesuai dengan konsep algoritma Levenshtein sendiri semakin kecil nilai jarak yang diperoleh maka dianggap semakin mendekati judul skripsi di input.

Kata kunci : *Website, Algoritma, Levenshtein Distance, Autocomplete*

### Abstract

*Thesis is a scientific paper in the form of an exposition of research results. In saving a large number of theses, it takes a long time to search for thesis data one by one, so it requires a storage media that facilitates the speed in finding theses such as a repository. A website-based repository is one of the storage media that is needed to store the thesis data. In making their own repository website using the PHP programming language, MySQL database with a waterfall design model through the stages of analysis, design, programming, testing and maintenance. This website also adds the Levenshtein Distance algorithm to optimize searches so that search times become more efficient and adds an autocomplete feature that functions to provide word predictions or suggestions according to the characters typed by the user so as to help users search more for the words they want to search for.*

*Based on the results of analysis and testing, the application of the Levenshtein Distance algorithm in finding the thesis title is able to display the thesis title which has the smallest or*

---

closest Distance value to the word inputted by the user. And for autocomplete itself, it is also able to display predictions or word suggestions that match the typed words and the calculation results for single target searches have an accuracy of 75% and multi-targets with 2 words have an accuracy of 50%. in accordance with the concept of the Levenshtein algorithm itself, the smaller the distance value obtained, the closer it is to the input thesis title.

*Keywords: Website, Algorithm, Levenshtein Distance, Autocomplete*

## **1. PENDAHULUAN**

### **1.1 Latar Belakang**

Di era saat ini internet memiliki peran sangat penting terutama dalam mengakses dan mencari informasi. Dengan keberadaan informasi manusia dapat mengetahui apapun dan kapanpun hal yang terjadi di kehidupan mulai dari masa lampau hingga saat ini. Lembaga pendidikan adalah salah satu organisasi yang mengakses banyak informasi yang bersifat ilmu pengetahuan, hiburan dan rekreasi. Perpustakaan di Lembaga pendidikan menjadi wadah untuk mengakses informasi dalam format apapun. Perpustakaan merupakan upaya untuk memelihara dan meningkatkan efisiensi dan efektivitas proses belajar-mengajar. Perpustakaan yang terorganisasi secara baik dan sistematis, secara langsung ataupun tidak langsung dapat memberikan kemudahan bagi proses belajar mengajar di sekolah tempat perpustakaan tersebut berada. Hal ini, terkait dengan kemajuan bidang pendidikan dan dengan adanya perbaikan metode belajar-mengajar yang dirasakan tidak bisa dipisahkan dari masalah penyediaan fasilitas dan sarana pendidikan.

Pada perguruan tinggi perpustakaan berperan penting dalam menyimpan karya tugas akhir atau skripsi mahasiswa. Contohnya di Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya. Skripsi yang cukup banyak sangat memakan banyak tempat. Selain itu dengan penyusunan skripsi yang secara acak membuat mahasiswa cukup kesulitan dalam mencari skripsi yang dibutuhkan sebagai referensi bagi mahasiswa yang ingin dan sedang menyusun skripsi. Pada saat mencari skripsi yang sesuai dengan bidang yang ingin diambil, tahun maupun dosen-dosen pembimbingnya yang diinginkan, mahasiswa harus mencari dan mengecek satu persatu skripsi untuk melihat detailnya. Selain itu, mahasiswa harus membuka lembar per lembar untuk mendapatkan bagian informasi yang mereka inginkan seperti abstrak dan setiap babnya. Hal ini memakan banyak waktu terlebih jika mahasiswa ingin mencari referensi skripsi lebih dari satu. Oleh karena itu, dibutuhkan suatu sistem dimana mahasiswa dapat mencari skripsi yang diinginkan dengan cepat seperti repository.

Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya merupakan salah satu Jurusan yang sudah memiliki aplikasi repository skripsi. Aplikasi repository skripsi yang berbasis website pada Jurusan Teknik Informatika merupakan aplikasi yang berfungsi untuk menyimpan karya tugas akhir atau skripsi. Aplikasi ini dapat digunakan untuk melihat judul skripsi mahasiswa dengan filter yang telah tersedia yaitu berdasarkan dosen pembimbing, bidang keahlian dan tahun lulus mahasiswa. Setiap tahun mahasiswa yang lulus akan semakin bertambah banyak jumlahnya maka judul skripsi mahasiswa juga beragam dan bertambah banyak, melihat skripsi berdasarkan dosen pembimbing, bidang keahlian dan tahun lulus saja masih belum cukup untuk mendapatkan informasi judul skripsi yang diinginkan. Oleh karena itu meskipun telah terdapat filter, aplikasi repository ini masih memiliki kekurangan yaitu tidak adanya fitur pencarian dan jika pengguna menggunakan filter maka hasilnya hanya akan menampilkan skripsi sesuai dengan kategori filter yang tersedia. Selain itu, karena tidak adanya fitur pencarian pengguna juga harus mencari dan membaca satu persatu judul skripsi sampai habis yang memakan banyak waktu. Untuk mengatasi hal tersebut perlu ditambahkan fitur pencarian judul skripsi yang menerapkan Algoritma Levenshtein Distance pada aplikasi ini agar dapat memudahkan pengguna untuk mencari informasi secara cepat dan sesuai dengan judul skripsi yang diinginkan.

Azka Mutia (2017) mengimplemetasikan permasalahan tentang pencarian teks dengan judul "Penerapan Algoritma Levenshtein Distance untuk Pencarian Nama Buku pada Sistem Informasi Perpustakaan Fakultas Kedokteran Universitas Palangka Raya". Pada penelitian tersebut

permasalahan yang terjadi adalah pengguna seringkali salah mengetikkan kata yang ingin dicari sehingga menimbulkan informasi yang tampil tidak sesuai dengan yang diinginkan bahkan tidak ditemukan, Karena hal tersebut peneliti memerlukan suatu pendekatan yang dapat memberikan alternatif nama buku sesuai dengan yang dituliskan oleh pengguna. Sesuai dengan kebutuhan tersebut maka Algoritma Levenshtein Distance yang merupakan salah satu algoritma pencarian digunakan untuk memberikan saran atau sugesti kata yang mendekati dengan kata yang dimasukkan untuk single target (pencarian dengan satu kata saja) dan multi target (pencarian dengan lebih dari satu kata). Dari hasil penelitian yang pernah dilakukan dapat disimpulkan bahwa Algoritma Levenshtein Distance mampu menampilkan autocorrect untuk perbaikan nama buku dengan hasil keakuratan untuk pencarian single target 75%, multi target dengan 2 kata 64,29% , multi target dengan 3 kata 66,75%, dan multi target dengan 4 kata 70,83%.

Azka Mutia (2017) mengimplementasikan algoritma Levenshtein hanya untuk menghitung jarak guna membandingkan dua string dan memodifikasi string tersebut yang meliputi perubahan/penukaran, penambahan dan penghapusan karakter padahal algoritma Levenshtein juga dapat disimulasikan untuk penerapan fitur autocomplete. Fitur autocomplete dapat dipergunakan untuk mengatasi kesalahan penulisan. Selain itu, keuntungan penggunaan autocomplete adalah membantu pengguna untuk mengingat kata yang akan dimasukkan. Kesalahan pada proses memasukkan kata dapat diminimalisir dengan memberikan prediksi/saran kata yang akan diinputkan dengan menampilkan kata atau kalimat yang diambil dari data yang tersedia atau data yang sudah ditentukan. Prediksi kata yang ditampilkan berkaitan dengan kata yang dimasukkan sehingga pengguna tidak perlu mengetikkan secara lengkap kalimat yang ingin dicari dengan demikian informasi yang didapatkan masih saling berkaitan dan kesalahan penulisan dapat lebih terminimalisir serta dengan memanfaatkan fitur autocomplete pencarian menjadi lebih praktis. Untuk melakukan autocomplete sendiri yaitu dengan menggunakan plugin jQuery Autocomplete dimana data saran atau sugeti kata yang akan ditampilkan nanti diambil dari data yang sudah ada pada basis data. Berkaitan dengan hasil penelitian dan kekurangannya tersebut maka pada penelitian yang akan dilakukan ini juga menerapkan Algoritma Levenshtein Distance dengan multiple keyword (pencarian banyak kunci atau lebih dari satu kunci) agar pencarian lebih akurat sekaligus juga pada penelitian ini akan menambahkan fitur autocomplete.

Berdasarkan latar belakang dari permasalahan diatas, maka pada penelitian ini diangkatlah judul “Implementasi Algoritma Levenshtein Distance untuk Pencarian Judul Skripsi dan Menambahkan Fitur Autocomplete pada Website Repository Skripsi Jurusan Teknik Informatika Univeristas Palangka Raya” yang bertujuan untuk mengatasi permasalahan pengetikan kata yang tidak sesuai dengan ejaan atau kata baku agar informasi yang dicari dapat sesuai dengan yang diinginkan dengan dilengkapi fitur autocomplete yang memberikan prediksi pilihan kata yang mirip dengan kata yang dimasukkan oleh pengguna sehingga kesalahan pada proses memasukkan kata dapat diminimalisir.

## 1.2 Tujuan Penelitian

Tujuan dalam penelitian ini adalah untuk :

- Menerapkan algoritma levenshtein dalam pencarian judul skripsi di website repository skripsi jurusan teknik informatika universitas palangka raya.
- Bagi mahasiswa memperluas wawasan mengenai penerapan algoritma levenshtein distance pada pencarian terlebih untuk pencarian multiple keyword.

## 2. TINJAUAN PUSTAKA

Dalam melakukan penelitian ini, terdapat beberapa penelitian-penelitian terkait yang telah dilakukan sebagai referensi sebagai berikut :

Penelitian yang menerapkan algoritma Levenshtein distance yaitu penelitian yang pernah dilakukan oleh Azka Mutia (2017) Jurusan Teknik Informatika Universitas Palangka Raya yang juga mengangkat permasalahan tentang pencarian yaitu dengan judul “Penerapan Algoritma Levenshtein distance untuk Pencarian Nama Buku pada Sistem Informasi Perpustakaan Fakultas

Kedokteran Universitas Palangka Raya”. Pada penelitian tersebut permasalahan yang terjadi adalah pengguna seringkali salah mengetikkan kata yang ingin dicari sehingga menimbulkan informasi yang tampil tidak sesuai dengan yang diinginkan bahkan tidak ditemukan, Karena hal tersebut peneliti memerlukan suatu pendekatan yang dapat memberikan alternatif judul buku sesuai dengan yang dituliskan oleh pengguna.

Penelitian lain selanjutnya menerapkan algoritma Levenshtein distance yang dilakukan oleh Ida Bagus Ketut Surya Arnawa (2017), STIKOM Bali dengan judul “Implementasi Algoritma Levenshtein distance pada Sistem Pencarian Judul Skripsi/Tugas Akhir”. Penelitian ini bertujuan untuk mengatasi permasalahan yang dialami mahasiswa dalam melakukan pencarian judul Skripsi/Tugas Akhir dimana seringkali dalam melakukan pencarian mahasiswa menginputkan kata kunci yang bukan merupakan ejaan yang baku atau salah ketik sehingga hal tersebut membuat mahasiswa dalam memperoleh informasi menjadi kurang lengkap dan bahkan mahasiswa bisa gagal dalam mendapatkan informasi yang sesuai dengan kata kunci yang dimasukkan. Proses penelitian ini dilakukan dengan menerapkan algoritma Levenshtein distance yang dapat menghitung jarak keterbedaan antara dua string.

Penelitian lain yang menerapkan Levenshtein Distance yaitu penelitian yang dilakukan oleh Kristien Margi S, Henny Hartono, dan Agus Toni (2016) Teknik Informatika, Universitas Bunda Mulia dengan judul “Implementasi Metode Levenshtein Distance untuk Pencarian Keyword pada Bahan Pustaka”. Penelitian ini bertujuan untuk membantu user suatu sistem melakukan pencarian judul buku, yang tidak hanya sesuai kecocokan “keyword” yang dimasukkan, tetapi juga berdasarkan hubungan padanan kata yang ada pada judul pustaka yang dicari melalui kemiripan keyword. Penelitian ini menggunakan metodologi pengembangan sistem waterfall dengan menerapkan metode Levenshtein distance dimana jika Keyword yang diinputkan oleh user terjadi kesalahan pengetikan ataupun terdapat kata yang mendekati kata kunci yang diketikkan, maka hasilnya adalah suggestion word seperti keyword yang dimaksud.

### 3. METODE PENELITIAN

Metode yang digunakan dalam pengembangan perangkat lunak ini adalah menggunakan metode pengembangan *waterfall* yang digunakan yaitu metode *Waterfall* (Ian Sommerville, 2011). Dalam pengembangannya metode waterfall memiliki beberapa tahapan yang berurutan yaitu: *requirement* (analisis kebutuhan), *design system* (desain sistem), *Coding* (pengkodean) & *Testing* (pengujian), Penerapan Program, pemeliharaan. Tahapan tahapan dari metode *waterfall* adalah sebagai berikut :

#### a. *Analysis*

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi ini biasanya dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

#### b. *System Design*

Spesifikasi kebutuhan dari tahap sebelumnya akan dipelajari dalam fase ini dan desain sistem disiapkan. Desain Sistem membantu dalam menentukan perangkat keras (hardware) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

#### c. *Implementation*

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit testing.

#### d. *Integration & Testing*

Seluruh unit yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing unit. Setelah integrasi seluruh sistem diuji untuk mengecek setiap kegagalan maupun kesalahan.

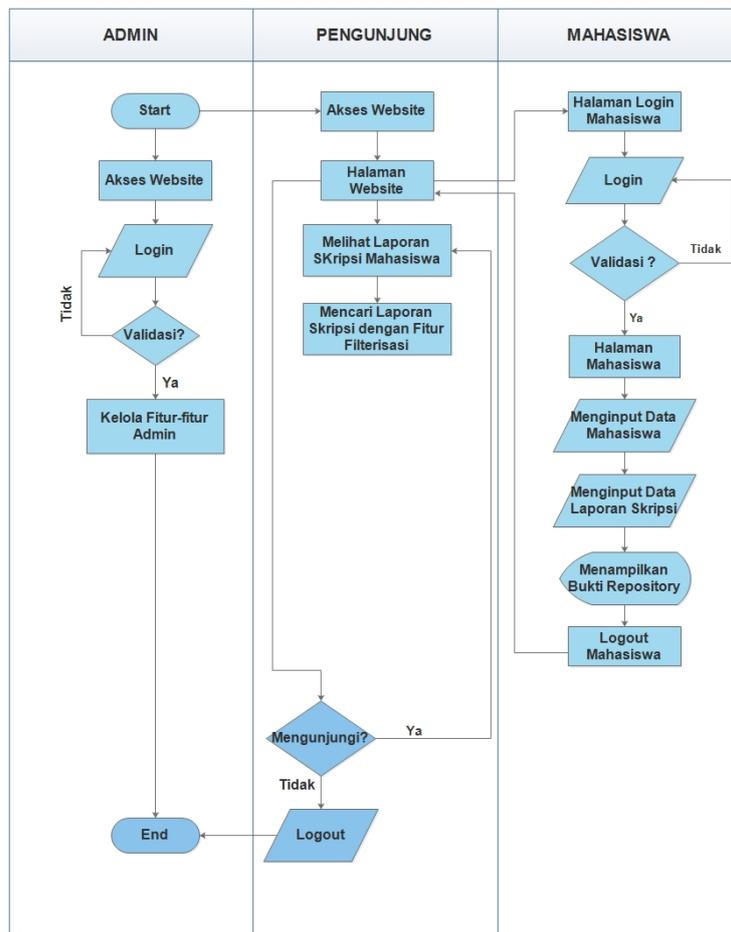
#### e. *Operation & Maintenance*

Tahap akhir dalam model waterfall. Perangkat lunak yang sudah jadi, dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaiki implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

#### 4. PEMBAHASAN

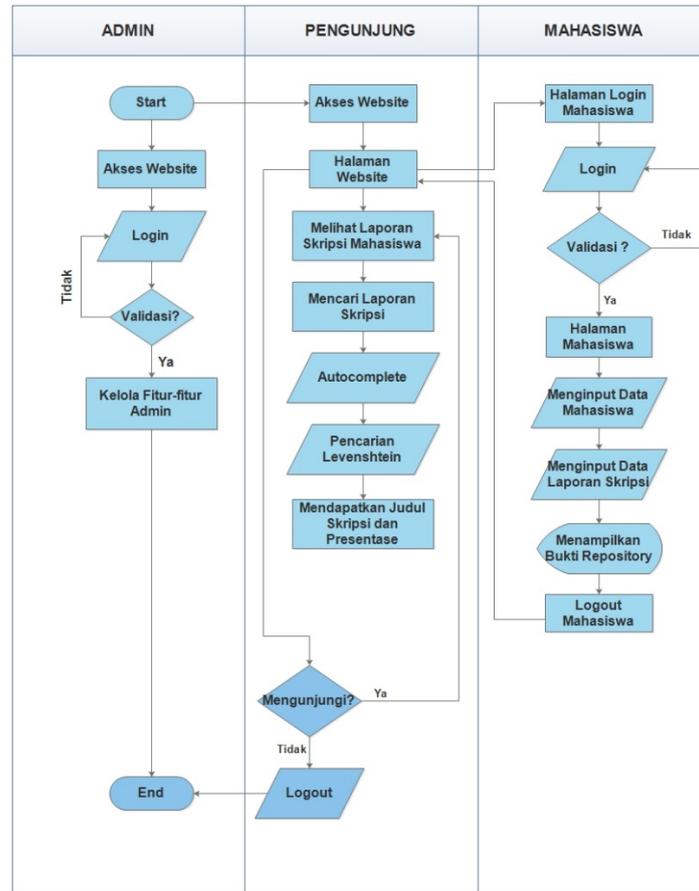
##### 4.1. Analisis Sistem

Berdasarkan hasil observasi yang telah dilakukan terhadap website repository skripsi Jurusan Teknik Informatika Universitas Palangka Raya, terdapat sistem yang sudah berjalan yaitu pada website tersebut memiliki tiga hak akses yaitu admin, pengunjung dan mahasiswa. Sistem digambarkan pada *flowchart* berikut ini :



Gambar 1. *Flowchart* Sistem Berjalan

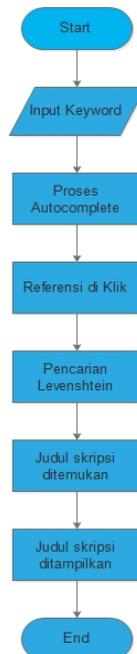
Berdasarkan analisis sitem berjalan yang telah dilakukan pada rekomendasi sistem ini akan dilakukan penambahan fitur pencarian dimana pada fitur pencarian ini akan menerapkan algoritma Levenshtein Distance sebagai suatu pendekatan yang dapat memberikan alternatif judul skripsi sesuai dengan yang diketik dan dicari oleh mahasiswa. Pencarian juga dilengkapi dengan autocomplete agar memudahkan mahasiswa dalam mencari informasi judul skripsi yang sesuai sekaligus mengatasi kesalahan penulisan kata. Berdasarkan analisis rekomendasi sistem baru maka flowchart sistem yang baru adalah sebagai berikut ini :



Gambar 2. Flowchart Sistem Baru

#### 4.2. Analisis Autocomplete

Autocomplete merupakan suatu fitur untuk memberikan sugesti/saran data yang akan diinputkan ke dalam sistem. Data berupa sugesti /saran ini diambil dari data lain yang telah tersedia atau data yang sudah ditetapkan. Berikut ini adalah proses alur kerja autocomplete :



Gambar 3. Flowchart Alur Kerja Autocomplete

### 4.3. Analisis Algoritma Levenshtein Distance

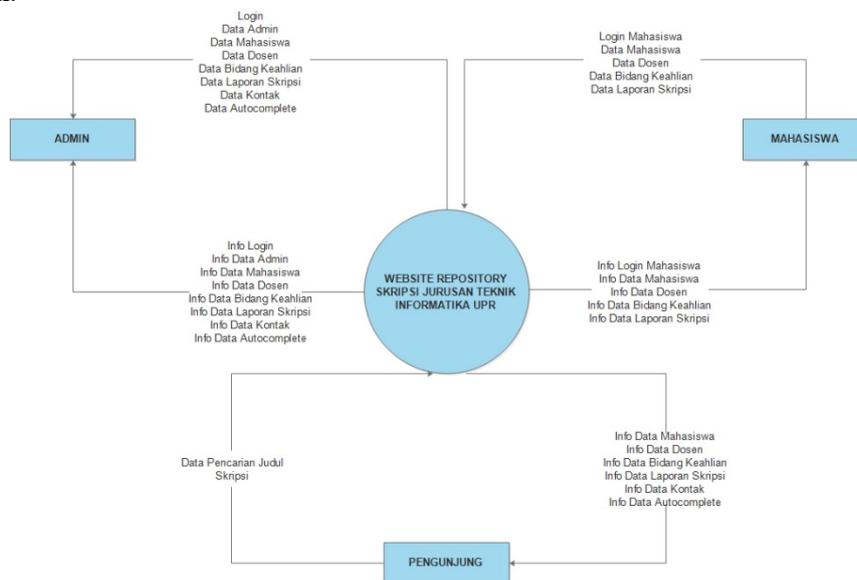
Algoritma ini digunakan sebagaimana fungsinya untuk menentukan jarak atau jumlah perbedaan dua string. Jarak yang dimaksud adalah jumlah perubahan yang diperlukan untuk mengubah suatu bentuk string ke string lainnya yang ditentukan berdasarkan nilai yang paling terkecil atau yang paling sedikit jumlah perubahannya. String atau kata yang memiliki nilai terkecil atau yang paling sedikit jumlah modifikasinya saat dibandingkan dengan string atau kata lain dianggap kata yang cocok. Alur kerja algoritma Levenshtein diatas dapat digambarkan pada flowchart berikut ini :



Gambar 4. Flowchart Algoritma Levenshtein Distance

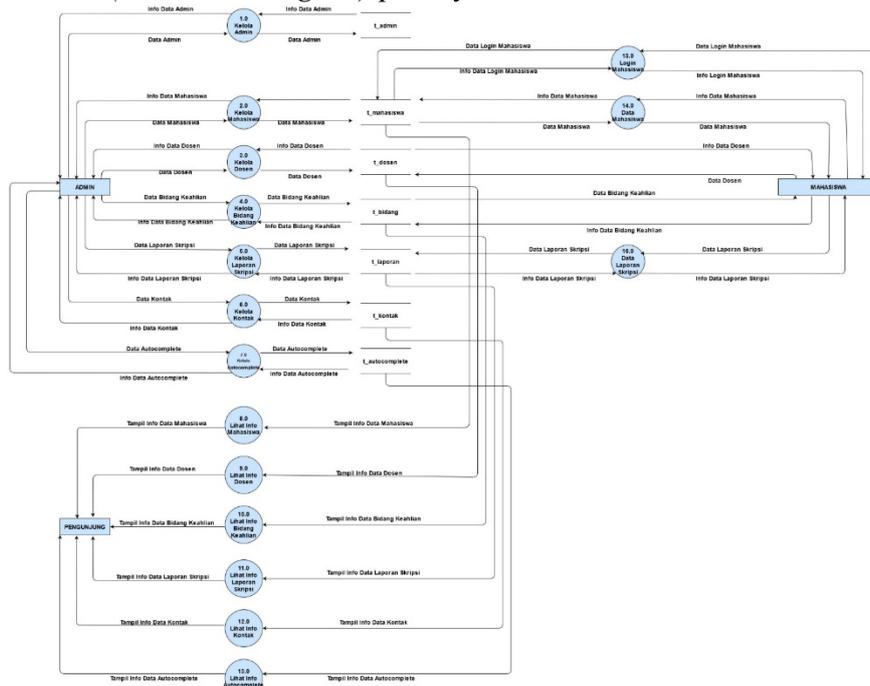
### 4.4. Perancangan Sistem

Tahapan perancangan sistem ini akan mendesain rekomendasi sistem baru, adapun desain yang akan dibangun berupa DFD (Data Flow Diagram), ERD (Entity Relationship Diagram) dan Desain Interface. Berikut ini adalah diagram konteks merupakan diagram yang mencakup masukan-masukan dasar, sistem umum dan keluaran, diagram ini merupakan tingkatan tertinggi dalam diagram aliran data dan hanya memuat satu proses dan menunjukkan sistem secara keseluruhan.



Gambar 5. Diagram Konteks

Berikut ini DFD (*Data Flow Diagram*) pada system ini :



Gambar 6. DFD (*Data Flow Diagram*)

Tabel atau tempat penyimpanan (storage) yang digunakan pada pembuatan website ini terdapat sebanyak 7 tabel yakni tabel tb\_admin, tb\_mahasiswa, tb\_dosen, tb\_bidang, tb\_laporan, tb\_autocomplete dan tb\_kontak.

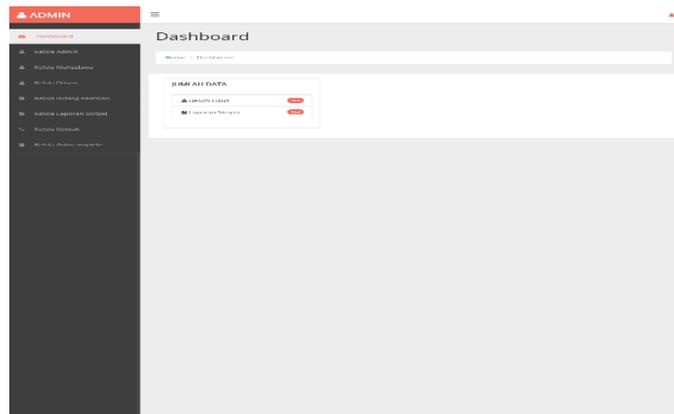
#### 4.5. Implementasi

Implementasi aplikasi dilakukan sesuai dengan perancangan yang telah dilakukan pada tahap desain interface sebelumnya.

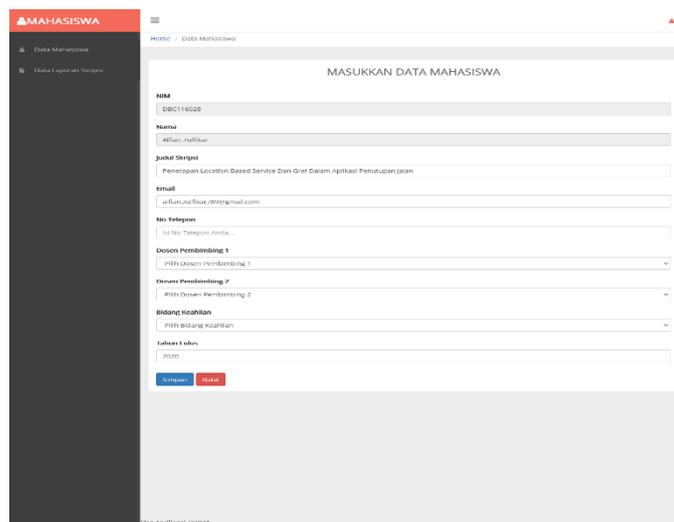


Gambar 7. Halaman Beranda Pengunjung

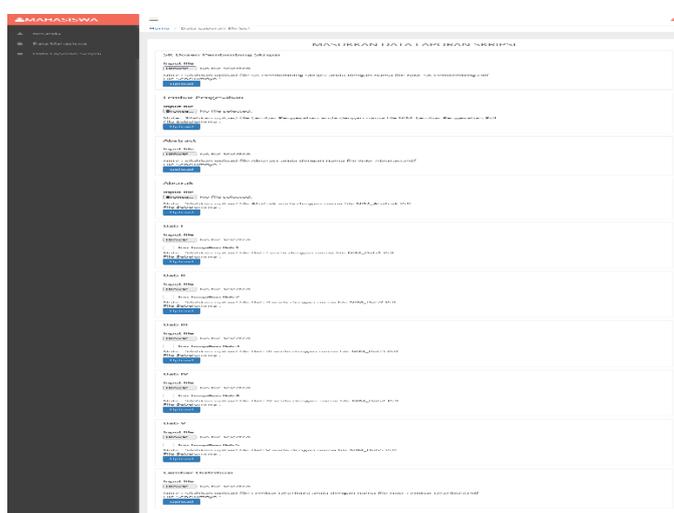
Selanjutnya pada halaman admin dan mahasiswa terdapat beberapa fitur seperti yang terlihat pada gambar menu dibawah ini.



Gambar 8. Halaman Beranda Admin



Gambar 9. Halaman Data Mahasiswa



Gambar 10. Halaman Data Laporan Mahasiswa

#### 4.6. Pengujian

Algoritma Levenshtein Distance merupakan sebuah algoritma yang dapat mengukur kesamaan antara string. Pengujian algoritma Levenshtein Distance pada website repository skripsi ini

dibatasi hanya dengan menggunakan threshold atau ambang batas kurang dari 3 string saja, sehingga kata yang diinputkan memiliki kesalahan pengetikan sebanyak 3 karakter maka hasil yang diinginkan tidak akan muncul. Pengujian pada website Repository Skripsi Jurusan Teknik Informatika ini menggunakan pengujian Blackbox. Pengujian dilakukan oleh 3 pihak yakni admin yang mengelola website, pengunjung dan mahasiswa Jurusan Teknik Informatika yang telah mengerjakan skripsi dapat login dengan username dan password berupa NIM masing-masing karena akun mahasiswa tersebut akan aktif apabila mengerjakan skripsi dan mengumpulkan laporan. Pada tahap pengujian website ini menggunakan Blackbox Testing dimana merupakan strategi yang memperhatikan atau memfokuskan kepada faktor fungsionalitasnya.

## 5. KESIMPULAN

Pada aplikasi Repository Skripsi Jurusan Teknik Informatika menerapkan algoritma Levenshtein Distance pada fitur pencarian serta menambahkan autocomplete. Berdasarkan hasil analisis dan pengujian, penerapan algoritma Levenshtein Distance dalam mencari judul skripsi mampu menampilkan judul skripsi yang memiliki nilai Distance yang terkecil atau terdekat dengan kata yang diinputkan oleh pengguna. Dan untuk autocomplete sendiri juga mampu menampilkan prediksi atau saran kata yang sesuai dengan kata yang di ketik. Dari pengujian sistem dapat di lihat semua judul skripsi yang tampil merupakan judul skripsi yang diperoleh berdasarkan nilai atau jarak Levenstein terkecil. Berdasarkan hasil perhitungan untuk pencarian single target memiliki keakuratan 75% dan multi target dengan 2 kata memiliki keakuratan 50%. sesuai dengan konsep algoritma Levenshtein sendiri semakin kecil nilai jarak yang diperoleh maka dianggap semakin mendekati judul skripsi di input

## DAFTAR PUSTAKA

- [1] Aditiays.2019.Metode Waterfall pada Pengembangan Sistem. <https://aditiays.wordpress.com/2019/02/18/metode-waterfall-dan-spiral-pada-pengembangan-sistem/>
- [2] Arwana, Ida Bagus Ketut Surya. 2017. Implementasi Algoritma Levenshtein Distance pada Sistem Pencarian Judul Skripsi/Tugas Akhir. Jurnal Sistem dan Informatika, Vol. 11, No. 2, Hal. 46-53.
- [3] Fadillah, Sarah Farrah.2019.Pengertian DFD. <https://www.nesabamedia.com/pengertian-dfd/>
- [4] Kosngosan.2019.Pengertian Repository. <https://www.kosngosan.com/2019/08/pengertian-repository-adalah.html>
- [5] Margi S, Kristien dkk. 2016. Implementasi Metode Levenshtein Distance untuk Pencarian Keyword pada Bahan Pustaka. Seminar Nasional APTIKOM (SEMNASTIKOM), Hal. 488-493.
- [6] Mecha, Satria.2014.Pengertian ERD (Entity Relationship Diagram). <http://satriamecha.blogspot.com/2014/08/pengertian-erd-entity-relationship.html>
- [7]] Mutia, Azka. 2017. Penerapan Algoritma Levenshtein Distance untuk Pencarian Nama Buku pada Sistem Informasi Perpustakaan Fakultas Kedokteran Universitas Palangka Raya.
- [8] NN.2019.Pengertian Flowchart. <https://www.romadecade.org/pengertian-flowchart/#!>
- [9] NN.2020.Pengertian Database. <https://www.termasmedia.com/lainnya/software/69-pengertian-database.html>