

PENERAPAN ALGORITMA *LEVENSHTEINDISTANCE* UNTUK PENCARIAN PADA SISTEM INFORMASI PERPUSTAKAAN FAKULTAS KEDOKTERAN UNIVERSITAS PALANGKA RAYA

Widiatry¹⁾, Nova Noor Kamala Sari²⁾, Viktor Handrianus Pranatawijaya³⁾, Putu Bagus Adidyana Anugrah Putra⁴⁾.

^{1,2,3,4)} Program Studi Teknik Informatika Universitas Palangka Raya

Kampus Unpar Tunjung Nyaho Jl. Yos Sudarso Palangka Raya

Email: widiatry@it.upr.ac.id¹⁾, novanoorks@it.upr.ac.id²⁾,

viktorhp@it.upr.ac.id³⁾, putubagus@it.upr.ac.id⁴⁾

Abstract

The library of Medicine Faculty of University of Palangka Raya regulates conventional management and has not utilized computerized-technology optimally. So that, it needs to improve its performance, particularly in delivering information of books or circulation services of book borrowing and returning transactions.

The methodology used in making information system is modified Waterfall software. This method consists of four stages; Analysis; Design; Implementation, and Testing. Errors in typing the words searched for by the users could cause the searching information cannot be found. Thus, levenshtein distance algorithm would display autocorrect or provide the word suggestions to improve the title of books that had the closest distance value to the word typed.

The results show that the single title search has 75% accuracy; the multi title with 2 words has 64.29% accuracy; the multi title with 3 word has 66.75% accuracy, and; the multi title with 4 words has 70.83% accuracy.

Keywords: *Levenshtein distance, Library, Information System*

PENDAHULUAN

Perpustakaan merupakan salah satu media yang menyediakan informasi serta ilmu bagi mahasiswa serta tenaga pengajar di suatu fakultas dari sebuah universitas sebagai sarana penyedia bacaan, bahan ajar dan bahan pustaka yang sesuai dengan kurikulum yang digunakan. Layanan perpustakaan

bertujuan untuk memberikan wawasan dan informasi untuk meningkatkan pengetahuan dan kemampuan dari pembaca.

Berdasarkan data perpustakaan Fakultas Kedokteran Universitas Palangka Raya, kurang lebih 106 jenis buku yang terdapat di perpustakaan tersebut. Banyaknya jenis buku yang

terdapat di perpustakaan, pencatatan transaksi peminjaman serta pengembalian buku, pencatatan laporan, pendataan anggota perpustakaan masih dilakukan secara manual.

Sebuah sistem informasi perpustakaan dapat diterapkan dalam media online berupa *website* untuk membantu dan memfasilitasi dalam melakukan transaksi peminjaman dan pengembalian buku dengan menggunakan *barcode*, membuat laporan-laporan, pendataan anggota, membantu mahasiswa apakah buku yang akan dipinjam tersedia atau tidak.

Website merupakan sarana untuk menampilkan, memberi dan mencari informasi dengan tujuan agar pengguna dapat berinteraksi dengan penyedia informasi dengan mudah dan cepat melalui jaringan internet. Teknologi informasi dapat diimplementasikan dalam mendukung proses peningkatan kualitas pelayanan terhadap masyarakat khususnya mahasiswa dan petugas perpustakaan untuk mengolah dan mengelola perpustakaan Fakultas Kedokteran Universitas Palangka Raya.

Sejumlah penelitian terhadap mesin pencari menyimpulkan bahwa rata-rata kesalahan dalam pengetikan kata yang dicari yang dilakukan oleh pengguna

cukup tinggi, kesalahan dalam pengetikan kata yang dicari oleh pengguna dapat menyebabkan informasi yang dicari tidak dapat ditemukan. Oleh sebab itu diperlukanlah suatu pendekatan yang dapat memberikan alternatif judul buku yang pengguna tuliskan. Algoritma *Levenshtein Distance* digunakan untuk memberikan saran atau sugesti kata yang mendekati dengan kata yang dimasukkan dalam pencarian pada Sistem Informasi Perpustakaan Fakultas Kedokteran Universitas Palangka Raya.

Peneliti sebelumnya juga telah melakukan penelitian diantaranya Rancang Bangun *Website* Sistem Informasi Praktikum Jurusan Teknik Informatika Universitas Palangka Raya (Widiatry, 2017) yang dapat menambah pemahaman dalam menganalisis dan mendesain suatu sistem di lingkungan Universitas Palangka Raya. Pengembangan perangkat lunak generatefile akun Uang Kuliah Tunggal (UKT) Universitas Palangka Raya (Pranatawijaya dkk., 2018) dapat menambah pemahaman mengenai cara kerja metode *waterfall* yang dimodifikasi bila diimplementasikan kedalam suatu sistem informasi. Analisis Dan Desain *Website*

Monitoring Konsultasi Bimbingan Kartu Rencana Studi (KRS) (Putra dkk.,2017), Rancang Bangun Sistem Informasi Pencarian Data Mahasiswa Dan Dosen pada Fakultas Hukum Universitas Palangka Raya (Putra dkk., 2017), dan Sistem Pendukung Keputusan untuk Seleksi Penerima Beasiswa BBP-PPA dengan Metode TOPSIS berbasis *Web* (Sari dkk., 2018) memberikan cara untuk pengelolaan data dan pemahaman dalam teknik searching data.

Berdasarkan penelitian sebelumnya tersebut akan melengkapi kajian tentang Penerapan Algoritma *Levenshtein Distance* untuk Pencarian pada Sistem Informasi Perpustakaan Fakultas Kedokteran Universitas Palangka Raya.

METODE

Metodologi pengembangan perangkat lunak yang digunakan dalam pembuatan website ini yaitu menggunakan metodologi *waterfall* yang dimodifikasi. Tahapan *waterfall* yang dilakukan yaitu analisis, desain, implementasi dan pengujian.

Berikut ini merupakan beberapa tahapan penelitian, yaitu:

1) Studi Pustaka dan Observasi

Metode ini melakukan studi pustaka terhadap buku, jurnal ilmiah nasional dan internasional sebagai pendukung dalam penelitian. Observasi dilakukan untuk mengambil data-data untuk titik dan berapa besar radius yang diperlukan untuk menandakan suatu ruangan yang nantinya akan dijadikan area untuk mengeluarkan informasi.

2) Analisis Sistem

Pada tahap ini akan dilakukan analisis kebutuhan yang diperlukan oleh *user* dalam pembuatan sistem informasi perpustakaan, dengan tujuan untuk mencari pilihan dan solusi apa saja yang akan dibangun dalam aplikasi ini. Proses pencarian kebutuhan difokuskan pada *software* untuk mengetahui sifat dari *website* yang akan dibuat, pemodelan proses bisnis menggunakan *Data FlowDiagram (DFD)* dan *Entity Relationship Diagram (ERD)*. *Data Flow Diagram* atau yang disingkat DFD merupakan diagram yang menggambarkan alir data dalam suatu entitas ke sistem atau sistem ke entitas. DFD juga dapat diartikan sebagai teknik grafis yang

menggambarkan alir data dan transformasi yang digunakan sebagai perjalanan data dari input atau masukan menuju keluaran atau output.

Entity Relationship Diagram (ERD) adalah diagram yang menghubungkan antarentitas di dalam penyusunan/perancangan basis data. *Entity Relationship Diagram* ini ditemukan oleh Chen tahun 1976. Tujuan dari *Entity Relationship Diagram* adalah untuk menunjukkan objek data dan *relationship* yang ada pada objek tersebut.

3) Desain Sistem

Proses ini digunakan untuk membuat “*blueprint*” *software* seperti, perancangan basis data dan perancangan antarmuka (*interface*).

4) Implementasi

Proses ini akan dilakukan pembuatan perangkat lunak sesuai dengan perancangan basis data dan antarmuka yang telah dilakukan pada tahap sebelumnya.

5) Pengujian Sistem

Proses pengujian akan dilakukan dengan *Blackbox Testing* yang disertai dengan melakukan simulasi data. Pengujian *Black Box* adalah

pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *Black Box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dieksekusi pada perangkat lunak dan kemudian keluar dari perangkat lunak dicek apakah telah sesuai yang diharapkan. Tujuan utama pengujian haruslah untuk mengekspos kesalahan dalam perangkat lunak dan untuk menghindari potensi kegagalan (B. Harsh dkk., 2014).

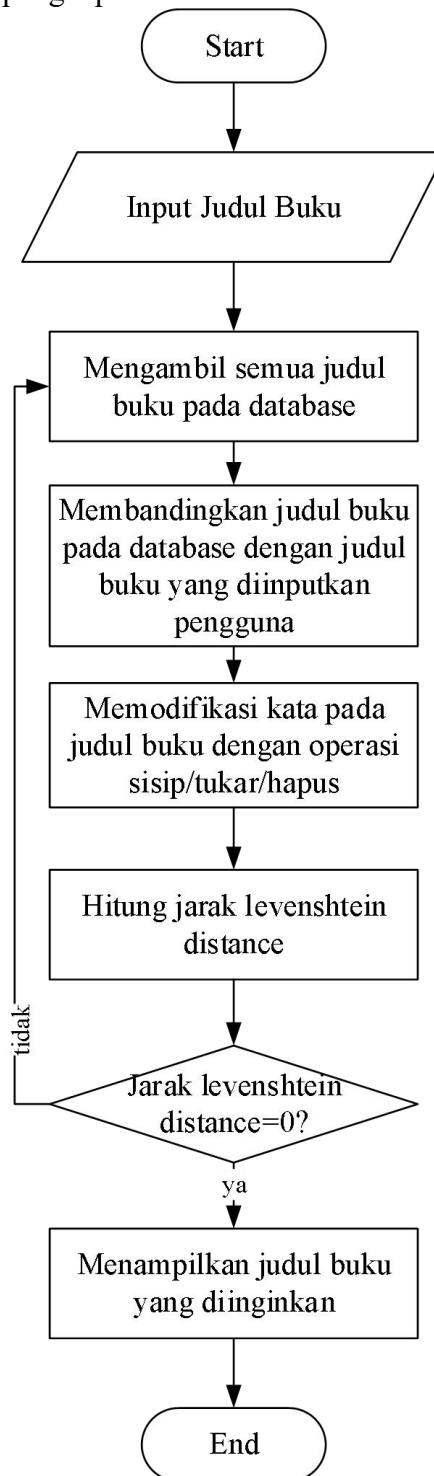
HASIL DAN PEMBAHASAN

Algoritma Levenshtein Distance

Pada *website* ini terdapat fitur pencarian judul buku, algoritma yang digunakan adalah *Levenshtein Distance* dimana algoritma ini digunakan untuk menentukan jarak *levenshtein* berdasarkan nilai yang paling terkecil atau paling sedikit jumlah modifikasinya. Sehingga string atau kata yang memiliki nilai modifikasi paling sedikit saat dibandingkan dengan

string atau kata lain dianggap sebagai kata yang cocok atau paling mendekati. Modifikasi tersebut meliputi proses penyisipan, penukaran dan penghapusan.

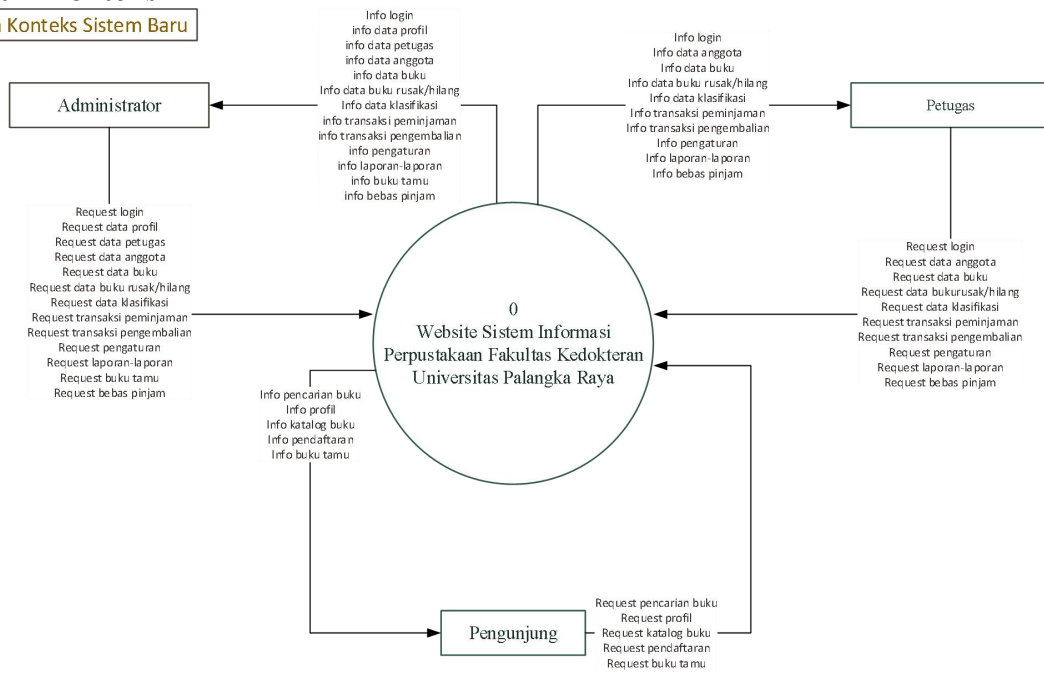
Berikut ini adalah alur kerja yang terdapat pada algoritma *Levenshtein Distance*:



Gambar 1. Flowchart Algoritma *Levenshtein Distance*

Diagram Konteks

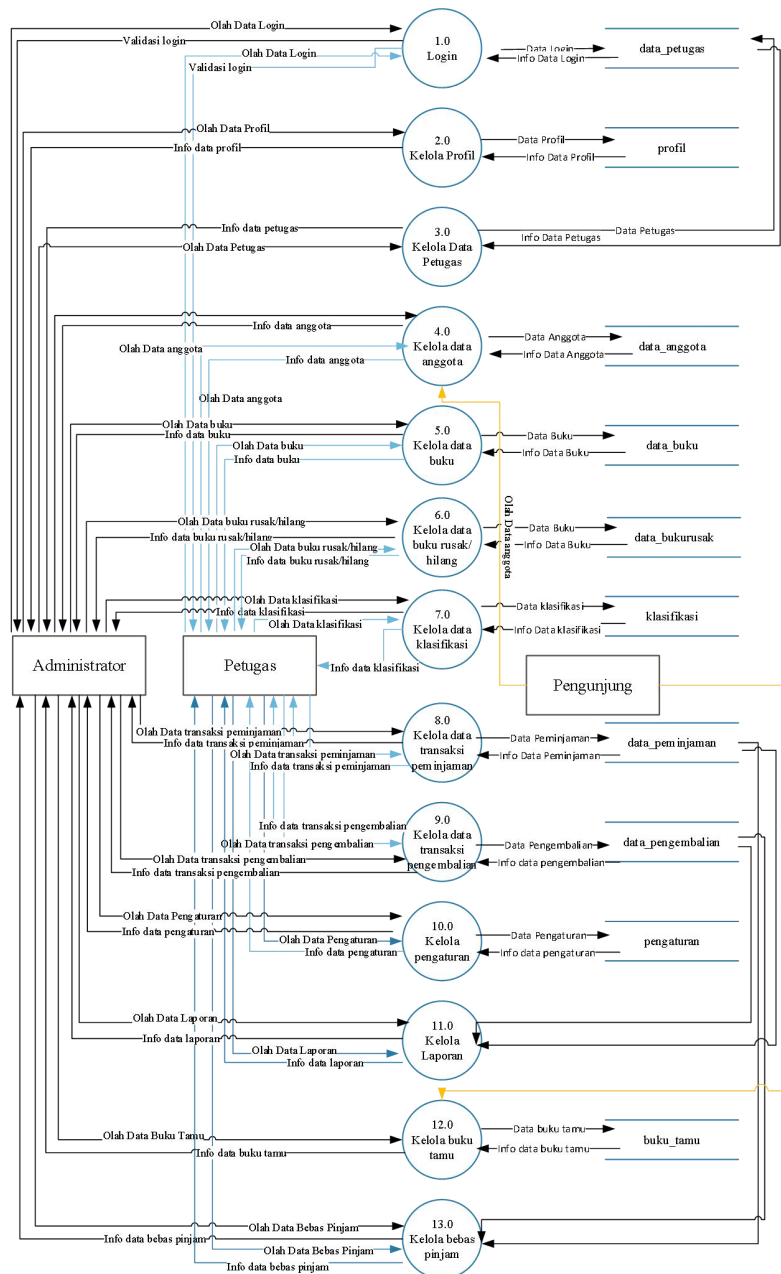
Diagram Konteks Sistem Baru



Gambar 2. Diagram Konteks

Gambar 2 merupakan diagram konteks atau DFD level 0 Sistem informasi perpustakaan Fakultas Kedokteran Universitas Palangka Raya, dimana dalam diagram ini terdapat tiga pihak terkait yang digambarkan secara umum. Kesatuan luar yang terlibat secara langsung adalah

Admin, Petugas dan Pengunjung. Admin dan petugas berfungsi sebagai pengelola *Website* Sistem Informasi Perpustakaan Fakultas Kedokteran Universitas Palangka Raya dan Pengunjung sebagai pihak yang mengakses profil dan informasi pada *website*.



Gambar 3. DFD Level 1

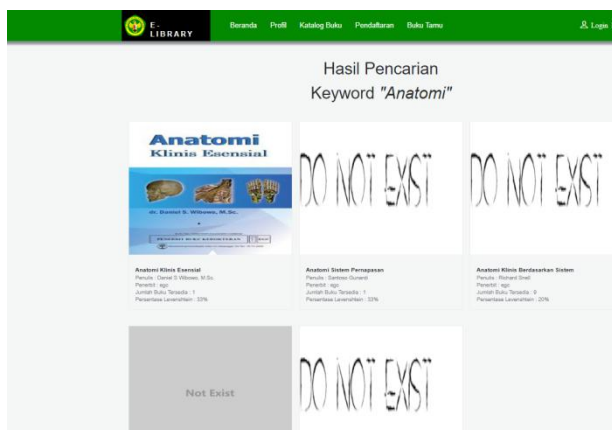
Gambar 3 merupakan DFD Level 1 atau yang disebut dengan *overviewed diagram* merupakan sebuah penggabungan secara keseluruhan yang melibatkan semua kesatuan luar secara lengkap. DFD Level 1

ini merupakan penjabaran dari diagram konteks.

perhitungan jarak *levenshtein* dengan rumus:

$$\begin{aligned}
 D(s,t) &= \sum_{i=1}^7 d(s_i, t_i) \quad (1) \\
 &= d(s_1, t_1) + d(s_2, t_2) + d(s_3, t_3) \\
 &+ d(s_4, t_4) + d(s_5, t_5) + d(s_6, t_6) \\
 &+ d(s_7, t_7) \\
 &= d(A, A) + d(n, n) + d(a, a) \\
 &+ d(t, t) + d(o, o) + d(m, m) \\
 &+ d(i, i) \\
 &= 0 + 0 + 0 + 0 + 0 + 0 + 0 \\
 &= 0
 \end{aligned}$$

Dari hasil perhitungan yang dilakukan diperoleh hasil nilai modifikasi atau jarak *levenshtein* yaitu 0, sehingga data anatomi akan ditampilkan. Adapun pengujian perhitungan manual diatas terhadap sistem yang telah dibuat adalah sebagai berikut:



Gambar 5. Hasil Pengujian Single Target dengan nilai distance sama dengan nol

Contoh 2:

1. Membandingkan *String*

Sumber : A n a t o m o
 Target : A n a t o m i

2. Melakukan modifikasi dan perhitungan jarak *levenshtein*

1 2 3 4 5 6 7

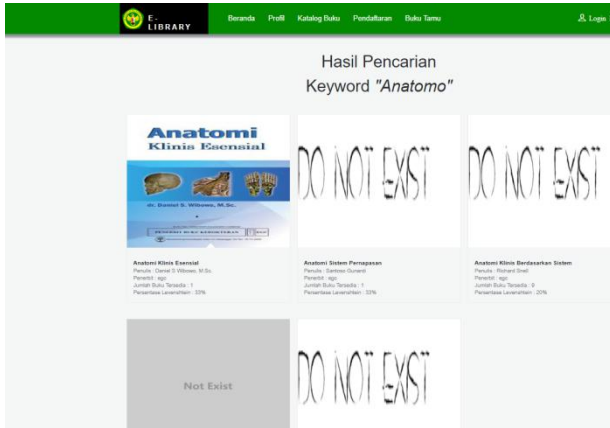
Sumber : A n a t o m o
 Target : A n a t o m i

Pada proses modifikasi dapat dilihat terdapat 1 operasi penukaran pada indeks ke 7 yaitu o menjadi i, selanjutnya akan dilakukan perhitungan jarak *levenshtein* dengan rumus:

$$\begin{aligned}
 D(s,t) &= \sum_{i=1}^7 d(s_i, t_i) \quad (2) \\
 &= d(s_1, t_1) + d(s_2, t_2) + d(s_3, t_3) \\
 &+ d(s_4, t_4) + d(s_5, t_5) + d(s_6, t_6) \\
 &+ d(s_7, t_7) \\
 &= d(A, A) + d(n, n) + d(a, a) \\
 &+ d(t, t) + d(o, o) + d(m, m) \\
 &+ d(o, i) \\
 &= 0 + 0 + 0 + 0 + 0 + 0 + 1 \\
 &= 1
 \end{aligned}$$

Dari hasil perhitungan yang dilakukan diperoleh hasil nilai modifikasi atau jarak *levenshtein* yaitu 1. Karena *threshold*=3 maka apabila pengguna salah mengetikkan

satu huruf data anatomi tetap akan ditampilkan. Adapun pengujian perhitungan manual diatas terhadap sistem yang telah dibuat adalah sebagai berikut:



Gambar 6. Hasil Pengujian Single Target dengan nilai distance sama dengan satu

Contoh 3:

1. Membandingkan *String*
 Sumber : A n a t i n i
 Target : A n a t o m i
2. Melakukan modifikasi dan perhitungan jarak *levenshtein*

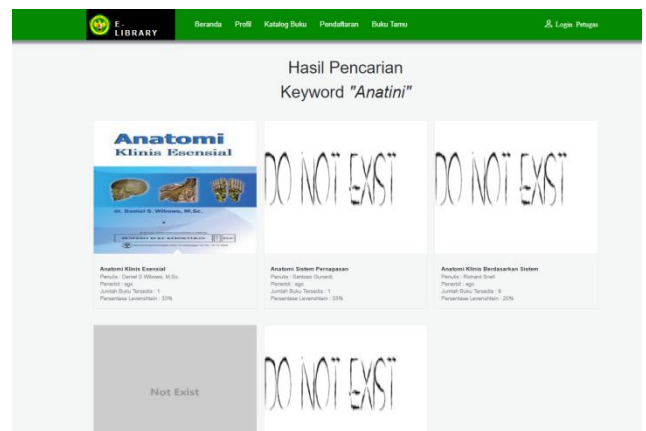
1 2 3 4 5 6 7

Sumber : A n a t i n i
 Target : A n a t o m i

Pada proses modifikasi dapat dilihat terdapat 2 operasi penukaran pada indeks ke 5 yaitu i menjadi o, pada indeks ke 6 yaitu operasi penukaran n menjadi m, maka selanjutnya akan dilakukan perhitungan jarak *levenshtein* dengan rumus:

$$\begin{aligned}
 D(s,t) &= \sum_{i=1}^7 d(s_i, t_i) \quad (3) \\
 &= d(s_1, t_1) + d(s_2, t_2) + d(s_3, t_3) \\
 &+ d(s_4, t_4) + d(s_5, t_5) + d(s_6, t_6) \\
 &+ d(s_7, t_7) \\
 &= d(A, A) + d(n, n) + d(a, a) \\
 &+ d(t, t) + d(i, o) + d(n, m) \\
 &+ d(i, i) \\
 &= 0 + 0 + 0 + 0 + 1 + 1 + 0 \\
 &= 2
 \end{aligned}$$

Dari hasil perhitungan yang dilakukan diperoleh hasil nilai modifikasi atau jarak *levenshtein* yaitu 2. Karena *threshold*=3 maka apabila pengguna salah mengetikkan dua huruf data anatomi tetap akan ditampilkan. Adapun pengujian perhitungan manual diatas terhadap sistem yang telah dibuat adalah sebagai berikut:



Gambar 7. Hasil Pengujian Single Target dengan nilai distance sama dengan dua

Contoh 4:

1. Membandingkan *String*

Sumber : A n a b i n i
 Target : A n a t o m i

2. Melakukan modifikasi dan perhitungan jarak *levenshtein*

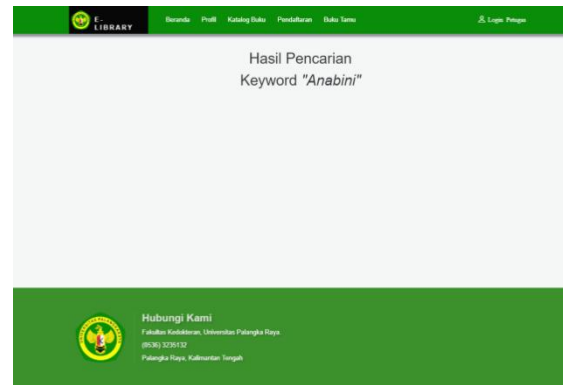
1 2 3 4 5 6 7
 Sumber : A n a b i n i
 Target : A n a t o m i

Pada proses modifikasi dapat dilihat terdapat 3 operasi penukaran pada indeks ke 4 yaitu b menjadi t, pada indeks ke 5 i menjadi o, dan pada indeks ke 6 n menjadi m, maka selanjutnya akan dilakukan perhitungan jarak *levenshtein* dengan rumus:

$$\begin{aligned}
 D(s,t) &= \sum_{i=1}^7 d(s_i, t_i) \quad (4) \\
 &= d(s_1, t_1) + d(s_2, t_2) + d(s_3, t_3) \\
 &+ d(s_4, t_4) + d(s_5, t_5) + d(s_6, t_6) \\
 &+ d(s_7, t_7) \\
 &= d(A, A) + d(n, n) + d(a, a) \\
 &+ d(b, t) + d(i, o) + d(n, m) \\
 &+ d(i, i) \\
 &= 0 + 0 + 0 + 1 + 1 + 1 + 0 \\
 &= 3
 \end{aligned}$$

Dari hasil perhitungan yang dilakukan diperoleh hasil nilai modifikasi atau jarak

levenshtein yaitu 3. Karena *threshold=3* maka apabila pengguna salah mengetikkan tiga huruf maka data anatomi tidak akan ditampilkan.



Gambar 8. Hasil Pengujian *Single Target* dengan nilai *distance* sama dengan tiga

Perhitungan manual untuk pencarian judul buku *multi target*

Untuk perhitungan pencarian judul buku *multi target* (pencarian lebih dari satu kata) akan dilakukan perhitungan nilai *distance* per-kata terlebih dahulu, kemudian akan dilanjutkan dengan menghitung persentase *levenshtein* tersebut sehingga akan menampilkan hasil yang diinginkan. Apabila nilai persentase adalah 0% maka data target tidak akan muncul

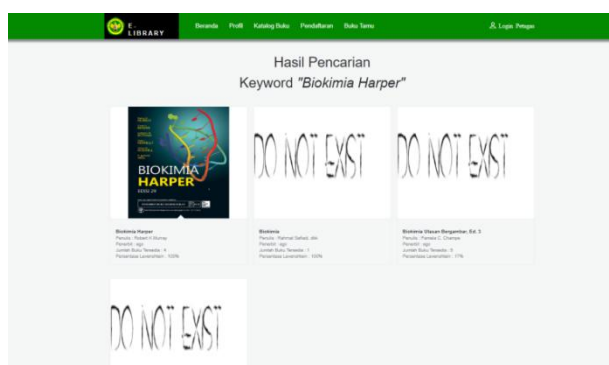
Perhitungan *multi target* untuk judul dengan 2 kata

Contoh 5:

Sumber : Biokimia Harper
 Target : Biokimia Harper
 Distance : 0 0

Dari contoh 5 untuk perhitungan *multi target* untuk judul buku dengan 2 kata, dapat dilihat bahwa tidak terdapat perbedaan *string* antar kata pertama pada sumber dan target, begitu pula tidak terdapat perbedaan pula pada kata kedua pada sumber dan target. Sehingga *distance* adalah 0 dan masih termasuk kedalam batas *threshold*. Selanjutnya, akan dilakukan perhitungan persentase *levenshtein* dengan cara (jumlah kata yang sama/jumlah kata target)x100%.

$$\begin{aligned}
 \text{Persentase Levenshtein} &= \frac{\text{jumlah kata yang sama}}{\text{jumlah kata target}} \times 100\% \\
 &= \frac{2}{2} \times 100\% \\
 &= 100\%
 \end{aligned}
 \tag{5}$$



Gambar 9. Hasil Pengujian *Multi Target* untuk Judul 2 kata *distance* sama dengan 0

Gambar 9 menunjukkan hasil pengujian multi target untuk judul 2 kata pada sistem, dimana kata sumber yang diinputkan oleh pengunjung adalah

Biokimia Harper dan kata target adalah Biokimia Harper, dengan persentase *levenshtein* adalah sebesar 100%.

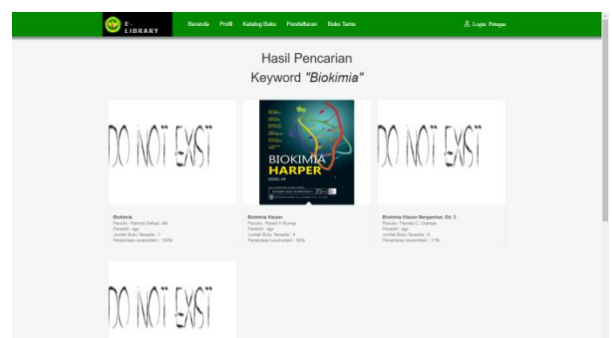
Contoh 6:

Sumber : Biokimia
 Target : Biokimia Harper
 Distance : 0 6

Dari contoh 6 untuk perhitungan *multi target* untuk judul buku dengan 2 kata, dapat dilihat bahwa terdapat perbedaan string antar kata pertama pada sumber dan target sebanyak 0, dan terdapat perbedaan pula pada kata kedua pada sumber dan target yaitu sebanyak 6. Selanjutnya, akan dilakukan perhitungan persentase *levenshtein* dengan cara (jumlah kata yang sama/jumlah kata target)x100%.

$$\text{Persentase Levenshtein} = \frac{\text{jumlah kata yang sama}}{\text{jumlah kata target}} \times 100\%$$

$$\begin{aligned}
 &= \frac{1}{2} \times 100\% \\
 &= 50\%
 \end{aligned}
 \tag{6}$$



Gambar 10. Hasil Pengujian *Multi Target* untuk Judul 2 kata

Gambar 10 menunjukkan hasil pengujian multi target untuk judul 2 kata pada sistem, dimana kata sumber yang diinputkan oleh pengunjung adalah Biokimia dan kata target adalah Biokimia Harper, dengan persentase *levenshtein* adalah sebesar 50%.

Perhitungan *multi target* untuk judul dengan 3 kata

Contoh 7:

Sumber : amatomi Klinik Esensial
 Target : Anatomi Klinis Esensial
 Distance : 2 1 1

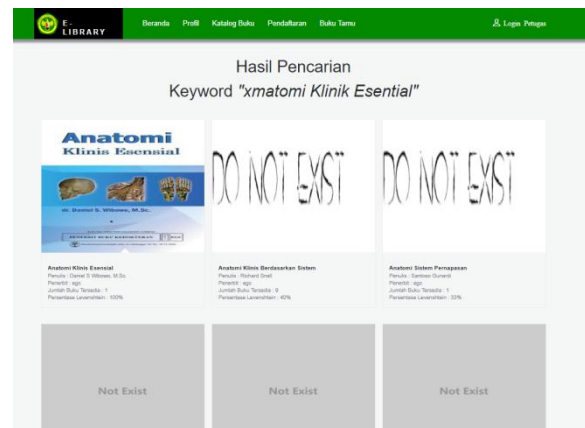
Dari contoh 7 untuk perhitungan *multi target* untuk judul buku dengan 3 kata, dapat dilihat bahwa terdapat perbedaan *string* antar kata pertama pada sumber dan target yaitu sebanyak 2, begitu pula terdapat perbedaan pada kata kedua sebanyak 1 dan pada kata ketiga pada sumber dan target sebanyak 1. Namun *distance* atau perbedaan *string* pada kata pertama, kedua dan ketiga masih termasuk ke dalam *threshold*. Selanjutnya, akan dilakukan perhitungan persentase *levenshtein* dengan cara (jumlah kata yang sama/jumlah kata target)x100%.

$$\text{Persentase Levenshtein} = \frac{\text{jumlah kata yang sama}}{\text{jumlah kata target}} \times 100\%$$

(7)

$$= \frac{3}{3} \times 100\%$$

$$= 100\%$$



Gambar 11. Hasil Pengujian *Multi Target* untuk Judul 3 kata

Gambar 11 menunjukkan hasil pengujian multi target untuk judul 3 kata pada sistem, dimana kata sumber yang diinputkan oleh pengunjung adalah xmatomi Klinik Esensial dan kata target adalah Anatomi Klinis Esensial, dengan persentase *levenshtein* adalah sebesar 100%.

Perhitungan *multi target* untuk judul dengan 4 kata

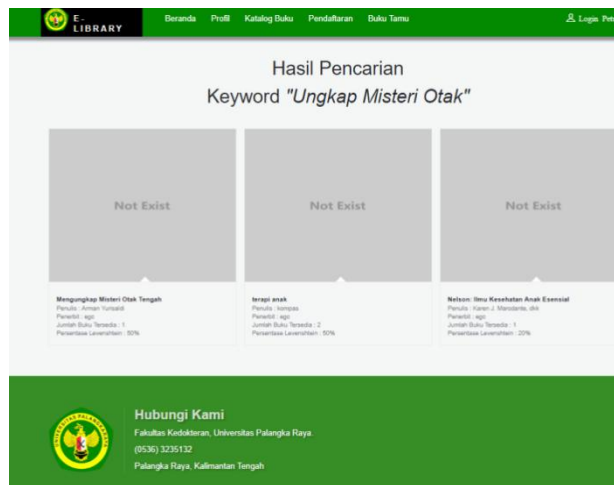
Contoh 8:

Sumber Ungkap Misteri Otak
 Target Mengungkap Misteri Otak Tengah
 Distance 4 0 0 6

Dari contoh 8 untuk perhitungan *multi target* untuk judul buku dengan 4 kata, dapat dilihat bahwa tidak terdapat perbedaan *string* antara kata kedua dan kata ketiga pada sumber dan target, namun pada kata pertama terdapat *distance*

sebanyak 4 dan kata keempat terdapat perbedaan distance sebanyak 6. Distance sebanyak 4 dan 6 tersebut tidak masuk kedalam *threshold*, sehingga dianggap sebagai kata yang berbeda. Selanjutnya, akan dilakukan perhitungan persentase *levenshtein* dengan cara (jumlah kata yang sama/jumlah kata target)x100%.

$$\begin{aligned}
 \text{Persentase Levenshtein} &= \frac{\text{jumlah kata yang sama}}{\text{jumlah kata target}} \times 100\% \\
 &= \frac{2}{4} \times 100\% \\
 &= 50\%
 \end{aligned}$$



Gambar 12. Hasil Pengujian Multi Target untuk Judul 4 kata

Gambar 12 menunjukkan hasil pengujian multi target untuk judul 4 kata pada sistem, dimana kata sumber yang diinputkan oleh pengunjung adalah Ungkap Misteri Otak dan kata target adalah Mengungkap Misteri Otak Tengah, dengan persentase *levenshtein* adalah sebesar 50%.



Gambar 13. Implementasi Halaman Pengunjung

Implementasi halaman utama pengunjung dapat dilihat pada Gambar 13, dimana pada halaman ini, terdapat fitur beranda, profil, katalog buku, pendaftaran, dan buku tamu. Pada halaman beranda, pengunjung dapat melakukan pencarian buku, pada pencarian buku inilah akan dilakukan penerapan algoritma *levenshteindistance*. Selain itu juga terdapat fitur login petugas, dimana fitur ini dapat diakses oleh petugas perpustakaan yang memiliki hak akses untuk masuk dan mengelola sistem informasi didalamnya.



Gambar 14. Implementasi Halaman Administrator

Implementasi halaman utama halaman administrator dapat dilihat pada Gambar 14, dimana seorang administrator memiliki

hak akses untuk mengelola data utama *website* ini, diantaranya adalah kelola data petugas, kelola data anggota, kelola data buku, kelola data buku rusak/hilang, kelola data klasifikasi, kelola transaksi (transaksi peminjaman, dan transaksi pengembalian), kelola pengaturan, kelola laporan-laporan (laporan data anggota, laporan data buku, laporan transaksi peminjaman, laporan transaksi pengembalian), kelola buku tamu, serta kelola bebas pinjam.

KESIMPULAN

Dalam membangun, merancang dan menerapkan algoritma *Levenshtein Distance* untuk pencarian judul buku pada Sistem Informasi Perpustakaan Fakultas Kedokteran Universitas Palangka Raya ini digunakan metode pengembangan perangkat lunak *Waterfall* yang telah dimodifikasi. Tahapan yang dilakukan antara lain: Analisis sistem seperti mendapatkan kebutuhan data dan teknologi yang digunakan pada proses bisnis dan rencana kegiatan, serta pembuatan *Data Flow Diagram* dan *Entity Relationship Diagram*. Pada tahap desain dilakukan desain tabel, desain navigasi menggunakan *sitemap*, dan desain antarmuka dan *Implementasi* dilakukan dengan bahasa pemrograman PHP. Tahap pengujian menggunakan *Blackbox Testing* dimana

dengan tahapan tersebut menghasilkan *website* yang dapat berjalan dengan baik.

Sistem Informasi Perpustakaan Fakultas Kedokteran Universitas Palangka Raya ini berjalan dengan baik pada *web browser Google Chrome*. Pengujian aplikasi dilakukan dengan melakukan *Blacbox Testing*. Berdasarkan hasil analisis dan pengujian, maka dapat disimpulkan bahwa algoritma *Levenshtein Distance* mampu menampilkan *autocorrect* untuk perbaikan judul buku yang memiliki nilai *distance* terdekat dengan kata yang diketikkan. Berdasarkan hasil perhitungan pada bagian implementasi algoritma *Levenshtein Distance* untuk pencarian judul buku *single target* memiliki keakuratan sebesar 75%, untuk pencarian judul buku *multi target* dengan 2 kata memiliki keakuratan sebesar 64,29%, untuk pencarian judul buku *multi target* dengan 3 kata memiliki keakuratan sebesar 66,75%, dan untuk pencarian judul buku *multi target* dengan 4 kata memiliki keakuratan sebesar 70,83%. Semua judul buku yang ditampilkan pada *autocorrect* merupakan judul buku yang diperoleh berdasarkan nilai/jarak *levenshtein* terkecil, semakin kecil nilai jarak *levenshtein* yang diperoleh dari proses modifikasi maka dianggap semakin mendekati atau mirip dengan judul buku yang diinputkan.

Semakin lengkap judul buku yang diketikkan maka nilai *Levenshtein* juga semakin kecil.

SARAN

Untuk meningkatkan performa sistem yang lebih baik lagi, maka sistem ini dapat dikembangkan dengan melakukan penambahan metode lain seperti semantik atau algoritma pencocokan *string* yang lain, misalnya algoritma *Boyer Moore* untuk menampilkan *autocomplete*. *Website* sistem informasi perpustakaan dapat dikembangkan dengan menggunakan alat *fingerprint* untuk kedepannya.

DAFTAR PUSTAKA

- A.S, Rosa & Shalahuddin, M. 2013. *Rekayasa Perangkat Lunak: Terstruktur dan Berorientasi Objek*. Bandung. Informatika.
- Bin Ladjamudin, Al-Bahra. 2013. *Analisis dan Desain Sistem Informasi*. Yogyakarta. Graha Ilmu.
- Davis, Gordon B. 2003. *Sistem Informasi Manajemen*. Yogyakarta. Andi
- Harsh, B., Khanna, E dan Sudha. 2014. *Black Box Testing based on Requirement Analysis and Design Specifications. International Journal of Computer Applications (0975 – 8887). Volume 87–No.18, (February)*.
- Pranatawijaya, V. H., Putra, P. B. A. A., Widiatry., Sari, N. N. K. 2018. *Pengembangan perangkat lunak generate File Akun Uang Kuliah Tunggal (UKT) Universitas Palangka Raya*. Jurnal Saintekom, 8(2), 166-178.
- Primadani, Yuli. 2014. *Simulasi Algoritma Levenshtein Distance untuk Fitur Autocomplete pada Aplikasi Katalog Perpustakaan*. Medan. Program Studi Teknologi Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara.
- Putra, P. B. A. A., Sari, N. N. K., & Pranatawijaya, V. H. 2017. *Analisis Dan Desain Website Monitoring Konsultasi Bimbingan Kartu Rencana Studi (KRS)*. Jurnal Teknologi Informasi, 11(1).
- Putra, P. B. A. A., Pranatawijaya, V. H., Widiatry, W., & Lisa, L. 2017. *Rancang Bangun Sistem Informasi Pencarian Data Mahasiswa dan Dosen pada Fakultas Hukum Universitas Palangka Raya*. Jurnal Teknologi Informasi, 11(2).
- Sari, N. N. K., Widiatry, W., & Chitayae, N. 2018. *Sistem Pendukung Keputusan untuk Seleksi Penerima Beasiswa BBP-PPA dengan Metode TOPSIS berbasis Web*. Anterior Jurnal, 18(1), 86-91.
- Simarmata, Janner dan Iman Paryudi. 2005. *Basis Data*. Yogyakarta. Penerbit Andi.
- Simarmata, Janner. 2010. *Rekayasa WEB*. Yogyakarta. Penerbit Andi.
- Sugiarto. 2014. *Implementasi Sistem Informasi Perpustakaan untuk Mengelola Koleksi Fisik dan Koleksi Digital pada Perpustakaan Binaan Yayasan Trampil*. Surabaya.

Program Studi Teknik Informatika,
Fakultas Teknologi Industri,
Universitas Kristen Petra, Surabaya.

Sutabri, Tata. 2012. *Analisis Sistem Informasi*. Yogyakarta. Andi

Widiarty, W. 2017. *Rancang Bangun Website Sistem Informasi Praktikum Jurusan Teknik Informatika Univeritas Palangka Raya*. Jurnal SAINTEKOM, 6(2), 12-24.