

# PENERAPAN ALGORITMA DISTANCE UNTUK FITUR AUTOCOMPLETE PADA APLIKASI KATALOG PERPUSTAKAAN DI UNIVERSITAS ALMUSLIM

FARHAN<sup>a</sup>

<sup>a</sup>Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Almuslim  
Jln. Almuslim Tlp. (0644) 41384, Fax. 442166 Matanglumpangdua Bireuen

## ABSTRAK

Katalog perpustakaan merupakan suatu media yang dapat menampilkan informasi sejumlah koleksi perpustakaan, dengan mencari buku pada katalog perpustakaan maka informasi mengenai buku yang dicari dapat diperoleh dengan mudah. Namun, terkadang terdapat suatu kendala ketika ingin memperoleh informasi mengenai buku yang dicari. Kendala tersebut adalah kesalahan dalam pengetikan pada kotak pencarian, kesalahan tersebut dapat menyebabkan informasi mengenai buku yang dicari tidak dapat ditemukan, karena itu diperlukan suatu „alat bantu“ yang dapat membantu pengguna ketika mengetikkan judul buku pada kotak pencarian seperti *autocomplete*. *Autocomplete* merupakan suatu fitur atau layanan yang dapat menampilkan prediksi kata, jika kata yang diketikkan belum lengkap. Keluaran yang dihasilkan dari sistem ini berupa prediksi judul buku yang diketikkan oleh pengguna.

**Kata Kunci :** Katalog, Perpustakaan, Auto Complete

## 1. PENDAHULUAN

Mesin pencari merupakan program komputer yang dirancang agar mampu menemukan informasi yang dicari dari banyaknya kumpulan informasi yang tersedia (Haryanto, 2011). Dengan adanya mesin pencari setiap orang dapat dengan mudah memperoleh informasi yang diinginkan. Dengan mengetikkan kata yang ingin dicari pada mesin pencari maka seluruh informasi yang diinginkan akan ditampilkan. Namun sejumlah penelitian terhadap mesin pencari menyimpulkan bahwa rata-rata kesalahan dalam pengetikan kata yang dicari yang dilakukan oleh pengguna cukup tinggi (Benisius, 2010), kesalahan dalam pengetikan kata yang dicari oleh pengguna dapat menyebabkan informasi yang dicari tidak dapat ditemukan. Untuk itu dibutuhkan penambahan fitur yang dapat membantu pengguna ketika mengetik, dimana fitur tersebut dapat menampilkan prediksi kata seperti *autocomplete* yang dapat membantu pengguna untuk mengetikkan kata pada mesin pencari. *Autocomplete* merupakan fitur atau layanan yang dapat menampilkan prediksi kata jika kata yang diketikkan belum lengkap (Kusuma, 2012).

Beberapa penelitian yang berkaitan dengan masalah ini diantaranya yaitu penelitian yang pernah dilakukan oleh Chiquita (2011), pada penelitiannya Chiquita menerapkan algoritma Boyer-Moore untuk layanan *autocomplete* dan menggunakan algoritma Dynamic Programming untuk layanan *autocorrect* untuk mencari kata di dalam paragraf. Kemudian selanjutnya penelitian yang pernah dilakukan oleh Kusuma (2012),

Pada penelitian ini, penulis akan mensimulasikan algoritma *Levenshtein Distance* untuk

menghasilkan fitur *autocomplete*. Pada simulasinya fitur *autocomplete* digunakan untuk membantu pengetikan judul buku pada aplikasi katalog perpustakaan. Algoritma *Levenshtein Distance* merupakan salah satu algoritma *Approximate String Matching* yang digunakan dalam pencarian string berdasarkan pendekatan perkiraan (Adiwidya, 2009). Pada penelitian sebelumnya algoritma string matching seperti Brute Force, Boyer-Moore, dan Knuth-Morris Pratt melakukan pencocokan secara bertahap pada seluruh rangkaian string sehingga memiliki proses yang cenderung panjang dan rumit (Pradhana, 2012). Sedangkan algoritma *Levenshtein Distance* melakukan modifikasi dengan mengubah suatu string menjadi string yang lain sehingga prosesnya lebih sederhana (Ardiyanto, 2008).

Berdasarkan masalah diatas, maka penulis membuat sebuah sistem dengan menerapkan algoritma *Levenshtein Distance* dengan judul “Penerapan Algoritma Distance Untuk Fitur Autocomplete Pada Aplikasi Katalog Perpustakaan di Universitas Almuslim”

## 2. PEMBAHASAN

### Aplikasi

Aplikasi berasal dari kata *application* yang artinya penerapan; lamaran; penggunaan. Secara istilah aplikasi adalah: program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju.

Menurut Ali-Zaki (2007), Aplikasi adalah komponen yang berguna melakukan pengolahan data maupun kegiatan-kegiatan seperti pembuatan dokumen

atau pengolahan data, Aplikasi adalah bagian PC yang berinteraksi langsung dengan user. Aplikasi berjalan di atas sistem operasi, sehingga agar aplikasi bisa diaktifkan, kita perlu melakukan instalasi sistem operasi terlebih dahulu.

Menurut Pramana (2008), Aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti sistem perniagaan, game, pelayanan masyarakat, periklanan, atau semua proses yang hampir dilakukan manusia.

### Katalog Perpustakaan

Katalog perpustakaan merupakan suatu media yang dibutuhkan oleh perpustakaan agar dapat memudahkan pengunjung dalam memperoleh informasi mengenai koleksi apa saja yang dimiliki oleh perpustakaan.

Ada beberapa pengertian tentang katalog perpustakaan, antara lain yaitu :

- Gates (1989), menyatakan bahwa katalog perpustakaan adalah suatu daftar yang sistematis dari buku dan bahan-bahan lain dalam suatu perpustakaan, dengan informasi deskriptif mengenai pengarang, judul, penerbit, tahun terbit, bentuk fisik, subjek, dan ciri khas bahan.
- Taylor (1992), menyatakan bahwa katalog perpustakaan merupakan susunan yang sistematis dari seperangkat cantuman bibliografis yang merepresentasikan kumpulan dari suatu koleksi tertentu. Koleksi tersebut terdiri dari berbagai jenis bahan, seperti buku, terbitan berkala, peta, rekaman suara, gambar, notasi musik, dan sebagainya.

### Fitur Atau Layanan Autocomplete

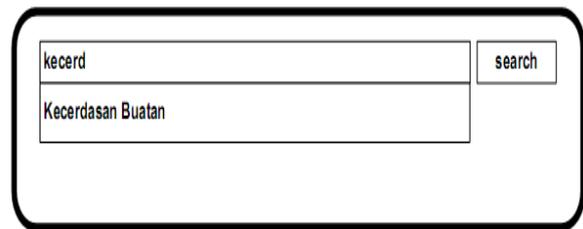
*Autocomplete* merupakan pola yang pertama kali muncul dalam bantuan fungsi aplikasi desktop, dimana pengguna mengetikkan teks ke dalam kotak kemudian saran pengetikan akan muncul secara otomatis (Morville & Callender, 2010). Autocomplete memecahkan beberapa masalah umum pada pengetikan (Morville & Callender, 2010) yaitu :

- Mengetik membutuhkan waktu.
- Pengguna tidak dapat mengeja kata dengan baik.
- Pengguna sering bingung ketika mengetikkan kata-kata, sulit mengingat istilah yang tepat.

Autocomplete melibatkan program yang dapat melakukan prediksi terhadap sebuah kata atau frasa yang pengguna ingin tulis tanpa harus menulis keseluruhan kata atau frasa secara lengkap (Kusuma, 2012). Autocomplete bekerja ketika pengguna menulis huruf pertama atau beberapa huruf/karakter dari sebuah kata, program yang melakukan prediksi akan mencari satu atau lebih kemungkinan kata sebagai pilihan. Jika kata yang dimaksud ada dalam pilihan kata prediksi maka kata yang dipilih tersebut akan disisipkan pada teks (Kusuma, 2012).

Saat ini autocomplete tidak hanya terdapat pada desktop, tetapi terdapat juga pada web browser, email-programs, search engine interface, source code editors, database query tools, word processor, dan

command line interpreters (Kusuma, 2012). Ilustrasi penggunaan layanan autocomplete dapat dilihat pada gambar 2.1.



Gambar Ilustrasi Penggunaan Autocomplete

### Algoritma Levenshtein Distance

Algoritma Levenshtein Distance ditemukan oleh Vladimir Levenshtein, seorang ilmuwan asal Rusia pada tahun 1965 (Janowski, 2010), algoritma ini sering juga disebut dengan Edit Distance (Husain, 2013). Yang dimaksud dengan distance adalah jumlah modifikasi yang dibutuhkan untuk mengubah suatu bentuk string ke bentuk string yang lain, 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 (Adiwiya, 2009). Algoritma Levenshtein Distance bekerja dengan menghitung jumlah minimum pentransformasian suatu string menjadi string lain yang meliputi penghapusan, penyisipan, dan penukaran (Husain, 2013).

Selisih perbedaan antar string dapat diperoleh dengan memeriksa apakah suatu string sumber sesuai dengan string target. Nilai selisih perbedaan ini disebut juga Edit distance/ jarak Levenshtein. 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  $s = s(1)s(2)s(3)...s(m)$  dan  $t = t(1)t(2)t(3)...t(n)$  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).

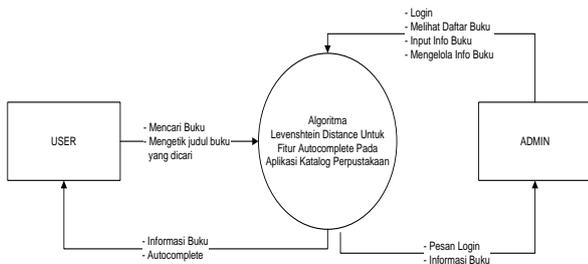
## 3. PERANCANGAN

### Diagram Konteks

Diagram Konteks berfungsi untuk menggambarkan sistem secara keseluruhan yang

diwakili oleh suatu lingkaran yang mewakili seluruh proses yang terjadi, juga menggambarkan bagaimana hubungan antara sistem dan entitas yang terlibat.

Diagram Konteks berfungsi untuk menggambarkan sistem secara keseluruhan yang diwakili oleh suatu lingkaran yang mewakili seluruh proses yang terjadi, juga menggambarkan bagaimana hubungan antara sistem dan entitas yang terlibat. Berikut adalah gambar dari Diagram Konteks Penerapan Algoritma *Distance* Untuk Fitur *Autocomplete* Pada Aplikasi Katalog Perpustakaan di Universitas Almuslim.



Gambar Diagram Konteks

Diagram konteks di atas terdiri dari dua entitas luar sebagai berikut :

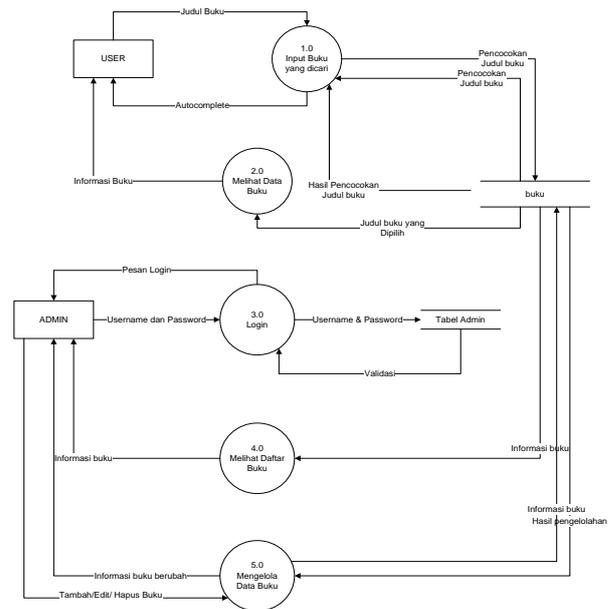
Tabel 3.1 Diagram Konteks Dua Entitas Luar

No	Entitas Luar	Keterangan
1	User	User dapat melakukan pencarian data buku dengan menyetikkan judul buku yang ingin dicari, selanjutnya user akan menerima keluaran berupa <i>autocomplete</i> dan informasi buku yang dicari
2	Admin	Admin melakukan proses login ke dalam sistem untuk dapat mengelola informasi buku berupa menambah, mengedit dan menghapus informasi buku

### Data Flow Diagram (DFD)

Data Flow Diagram adalah teknik yang menggambarkan komponen – komponen dari sebuah sistem dan aliran – aliran data di komponen tersebut asal, tujuan dan penyimpanan data. Berikut adalah *Data Flow Diagram* Penerapan Algoritma *Distance* Untuk Fitur *Autocomplete* Pada Aplikasi Katalog Perpustakaan di Universitas Almuslim.

1. DFD level 1 akan menggambarkan modul-modul yang ada di dalam sistem yang akan dibuat.

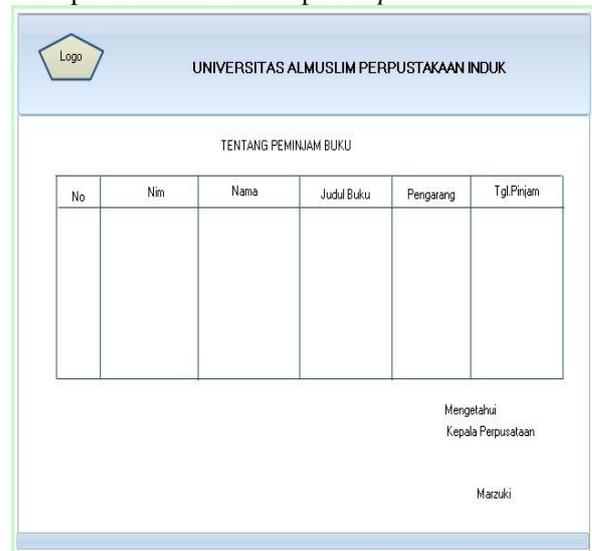


Gambar Data Flow Diagram Level 1

Pada DFD level 1 dapat dilihat modul-modul yang terdapat di dalam sistem, adapun modul ataupun bagian yang akan dilakukan oleh user yaitu input judul buku yang ingin dicari dan melihat data atau informasi buku. Sedangkan untuk modul atau bagian yang akan dilakukan oleh admin yaitu login, melihat daftar buku, input data buku dan mengelola data buku.

### Perancangan Output

*Output* adalah produk dari sistem informasi yang dapat dilihat. *Output* dapat berupa hasil di media kertas atau hasil di media lunak (berupa tampilan di layar video). Selain itu output dapat berupa hasil dari suatu proses yang akan digunakan oleh proses lain dan tersimpan di suatu media seperti *tape* atau *disk*.



Gambar Output Sistem

## Perancangan Algoritma

*Algoritma Levenshtein Distance* akan menampilkan layanan *autocomplete* dengan melakukan perbandingan berdasarkan judul buku yang diinputkan dengan judul buku yang terdapat di dalam database. Ketika judul buku diketikkan, sistem akan melakukan modifikasi terhadap judul buku yang terdapat di dalam database. Modifikasi tersebut meliputi proses penyisipan, penukaran dan penghapusan, jika proses modifikasi telah dilakukan selanjutnya akan dilakukan perhitungan nilai jarak *Levenshtein*.

Penentuan jarak *Levenshtein* didasarkan kepada nilai yang paling terkecil atau paling sedikit jumlah modifikasinya, sehingga string atau kata yang memiliki nilai modifikasi yang paling sedikit saat dibandingkan dengan string atau kata lain dianggap sebagai kata yang cocok atau paling mendekati.

Berikut adalah langkah-langkah proses pencarian jarak *Levenshtein* dengan kata depan dari judul buku yaitu:

### 1. Membandingkan string

String yang akan dibandingkan yaitu judul buku yang diinput sebagai string target dan judul buku yang terdapat di dalam database sebagai string sumber. Misalkan :

String Target : algoritma  
String Sumber : algoritma genetika, kecerdasan buatan.

Misalkan string target atau judul buku yang diinputkan adalah “algoritma” sedangkan string sumber atau judul buku yang terdapat di dalam database misalnya adalah “algoritma genetika,” kecerdasan buatan”. Judul buku yang terdapat di dalam database akan diubah menjadi string target atau judul buku yang diinputkan.

### 2. Melakukan modifikasi dan perhitungan jarak *Levenshtein*

Operasi *Levenshtein Distance* terdiri dari 3 yaitu, operasi penyisipan, operasi penukaran dan operasi penghapusan. Sehingga proses modifikasi yang dapat dilakukan yaitu :

a. Target = algoritma

Sumber = algoritma genetika

```

1 2 3 4 5 6 7 8 9 10 11 12 13
14 15 16 17 18
Target = a l g o r i t m a - -
- - - - -
Sumber = A l g o r i t m a -
G e n e t i k a
    
```

Pada proses modifikasi dapat dilihat terdapat 9 operasi penghapusan, yang dimulai dengan menghapus spasi pada indeks ke-10, kemudian menghapus karakter „g” pada indeks ke-11, „e” pada indeks ke-12, „n” pada indeks 13, „e” pada indeks ke-14, „t” pada indeks ke-15, „i” pada indeks ke-16, „k” ke-17 dan „a” pada indeks ke-18. Kemudian melakukan 1 operasi penukaran yaitu menukar karakter „A” menjadi „a” pada posisi indeks ke-1, sedangkan operasi penyisipan

tidak perlu dilakukan. Selanjutnya melakukan perhitungan jarak levenshtein dengan rumus yaitu:

$$D(s, t) = \sum_{i=1}^{18} d(s_i, t_i)$$

$$\begin{aligned}
 &= d(s_1, t_1) + d(s_2, t_2) + d(s_3, t_3) + d(s_4, t_4) + \\
 &d(s_5, t_5) + \\
 &d(s_6, t_6) + d(s_7, t_7) + d(s_8, t_8) + d(s_9, t_9) + \\
 &d(s_{10}, t_{10}) + \\
 &d(s_{11}, t_{11}) + d(s_{12}, t_{12}) + d(s_{13}, t_{13}) + d(s_{14}, \\
 &t_{14}) + \\
 &d(s_{15}, t_{15}) + d(s_{16}, t_{16}) + d(s_{17}, t_{17}) + d(s_{18}, \\
 &t_{18}) \\
 &= d(A, a) + d(l, l) + d(g, g) + d(o, o) + d(r, r) + \\
 &d(i, i) + d(t, t) + d(m, m) + d(a, a) + d(-, -) + \\
 &d(g, -) + d(e, -) + d(n, -) + d(e, -) + d(t, -) + d(i, -) \\
 &+ \\
 &d(g, -) + d(e, -) + d(n, -) + d(e, -) \\
 &+ d(t, -) + d(i, -) + d(k, -) + d(a, -) \\
 &= 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 1 + 1 + 1 + 1 + \\
 &1 + 1 + \\
 &1 + 1 + 1 \\
 &= 10
 \end{aligned}$$

Dari perhitungan yang dilakukan diperoleh nilai modifikasi atau jarak *Levenshtein* yang dibutuhkan adalah 10.

### 3. Menampilkan *autocomplete*

Dari perhitungan untuk mencari jarak *levenshtein distance* yang telah dilakukan maka diperoleh jarak *levenshtein* untuk string “Algoritma Genetika” menjadi “algoritma” adalah 10, untuk “Pemrograman C++” menjadi “algoritma” adalah 15 dan “Kecerdasan Buatan” menjadi “algoritma” adalah 15. Untuk dapat ditampilkan sebagai *autocomplete* maka diseleksi terlebih dahulu berdasarkan jarak *levenshtein* yang diperoleh, semakin kecil nilai jarak yang diperoleh maka kemungkinan untuk ditampilkan sebagai *autocomplete* akan semakin besar.

Prinsip kerja *algoritma levenshtein* yaitu berdasarkan jaraknya, semakin kecil nilai jarak yang diperoleh maka dianggap semakin mendekati dengan string target. Urutan string berdasarkan nilai jarak terkecil yang diperoleh yaitu judul buku “Algoritma Genetika”, dan “Kecerdasan Buatan”. Berdasarkan hal tersebut maka yang dapat ditampilkan sebagai *autocomplete* adalah “Algoritma Genetika” karena dianggap yang paling mendekati.

Selanjutnya akan dijelaskan langkah-langkah proses pencarian jarak *Levenshtein* dengan pola yang berbeda yaitu:

#### a. Membandingkan string

String yang akan dibandingkan yaitu judul buku yang diinput sebagai string target dan judul buku yang terdapat di dalam database sebagai string sumber. Misalkan :

String Target : cerdas  
String Sumber : Algoritma Genetika, Pemrograman c++, Kecerdasan Buatan.

Misalkan string target atau judul buku yang diinputkan adalah “cerdas” sedangkan string sumber atau judul buku yang terdapat di dalam database



Kesimpulan yang dapat diambil dari Penerapan Algoritma Distance Untuk Fitur Autocomplete Pada Aplikasi Katalog Perpustakaan di Universitas Almuslim adalah sebagai berikut:

1. Sistem ini dapat memberikan kemudahan kepada mahasiswa dalam meminjam buku pada perpustakaan Almuslim karena bersifat online
2. Dengan adanya penyajian data peta yang lebih interaktif diharapkan dapat memberikan bentuk penyajian informasi yang interaktif dan lebih mudah untuk dipahami.
3. Aplikasi Ini dibuat dengan menggunakan PHP, Mysql dan J-Query
4. Output yang dihasilkan dari sistem ini adalah informasi data anggota, data buku, data peminjaman buku, dan data pengembalian buku.

#### DAFTAR PUSTAKA

Teknologi

- Adiwidya, B. M. D. 2009. Algoritma levenshtein dalam pendekatan approximate string matching. Skripsi. Institut Teknologi Bandung.
- Adriyani, N. M. M. 2012. Implementasi algoritma levenshtein distance dan metode empiris untuk menampilkan saran perbaikan kesalahan pengetikan dokumen berbahasa Indonesia. Skripsi. Universitas Udayana.
- Andhika, F. R. 2010. Penerapan string suggestion dengan algoritma levenshtein distance dan alternatif algoritma lain dalam aplikasi. Skripsi. Institut Teknologi Bandung.
- Ardiyanto, R. I. 2008. Dynamic programming dalam levenshtein distance untuk mengetahui keterbedaan dua string. Skripsi. Institut Teknologi Bandung.
- Benisius. 2010. Sistem pengoreksian kata kunci dengan menggunakan metode levenshtein distance. Tesis. Universitas Gadjah Mada.
- Chiquita, C. 2011. Penerapan algoritma boyer moore-dynamic programming untuk layanan auto-complete dan auto-correct. Skripsi. Institut