Prosiding Seminar Nasional Teknologi Informasi dan Kedirgantaraan : Peran Teknologi untuk Revitalisasi Bandara dan Transportasi Udara, Yogyakarta, 10 Desember 2019
SENATIK 2019, Vol. V, ISBN 978-602-52742-1-3
DOI: 10.28989/senatik.v5i0.344

KEYWORDS SEARCH CORRECTION USING DAMERAU LEVENSHTEIN DISTANCE ALGORITHM

Enny Dwi Oktaviyani[1)], Sherly Christina[2)], Deddy Ronaldo[3)]
Universitas Palangka Raya
Kampus UPR,Tunjung Nyaho, Jl. Yos Sudarso
Email : [1]enny@it.upr.ac.id, [2]sherly.christina.upr@gmail.com, [3]deddy.ronaldo@gmail.com

Abstract

Searching is one of the important features on the website, but it is not uncommon for users to make typos when typing keywords. Typing errors of these keywords is usually referred to as typo. This study aims to build a system by providing suggestions for correcting typos in the search feature. Keywords search correction are obtained using the Damerau-Levenshtein Distance Approximate String Matching algorithm by to calculate the editing distance of each word in a keywords with each word in the Indonesian word dictionary. Testing was carried out as many as 40 experiments, with 10 keywords and 250 articles taken randomly. The test results show the Damerau-Levenshtein Distance algorithm is able to provide precision and recall values of 91.24% and 89.58% in providing keyword improvement suggestions. With the improvement of the system, each trial increases with precision value of 0.80 and recall value of 0.98.

Keywords*: Damerau-Levenshtein Distance, searching, keywords, correction*

1. Pendahuluan

Searching adalah salah satu fitur penting dalam *website*. Fitur ini mempermudah pengguna untuk menemukan apa yang diinginkan tanpa perlu menelusuri setiap halaman yang ada pada suatu website. Fitur pencarian ini berkerja dengan cara menerima masukan dari pengguna berupa kata kunci, kemudian kata kunci tersebut diproses dan dicocokan dengan data *searchable* yang ada di *database* website. Jika ditemukan data yang memiliki kecocokan dengan kata kunci maka akan ditampilkan ke pengguna sebagai kembaliannya, namun jika tidak ada data yang cocok maka akan ditampilkan hasil kosong atau pemberitahuan bahwa data yang dicari tidak ada. Suatu kata kunci bisa terdiri dari sebuah kata saja atau beberapa kata (frasa).

Fitur pencarian ini memang sangat memudahkan sekali bagi para pengguna website karena dapat menghemat banyak waktu dalam pencarian suatu informasi atau data. Namun, tidak jarang seorang pengguna melakukan kesalahan pengetikan suatu kata pada saat mengetikan kata kunci. Kesalahan pengetikan kata ini biasanya disebut dengan istilah *typo*. *Typo* ini bisa disebabkan oleh berbagai macam hal, seperti media pengetikan yang tidak sesuai dengan fisik pengguna (misalnya seperti pengguna berjari besar dengan layar *touchscreen* yang kecil), kerusakan pada media pengetikan (misalnya seperti kerusakan pada *keyboard* atau *touchscreen*), minimnya kosa kata pengguna dan masih banyak lagi. Kesalahan pengetikan suatu kata atau *typo* ini tentu akan berdampak pada hasil pencarian data. Misalnya, data *searchable* yang terdapat pada suatu *database* website adalah "Levenshtein algorithm" kemudian kata kunci pencarian yang dimasukan pengguna adalah "Levensten algoritm". Hasil proses pencarian tentu saja tidak ditemukan karena tidak ada kecocokan antara data yang ada dengan kata kunci pengguna.

Penelitian [1] mengusulkan sebuah sistem untuk mendeteksi dan mengoreksi kesalahan ejaan kata pada dokumen jurnal menggunakan metode *Dictionary Lookup* dan *Damerau-Levenshtein Distance.* Penelitian [2] melakukan survey pada beberapa teknik-teknik

untuk mendeteksi dan mengoreksi ejaan, dan penelitian [3] mengusulkan sistem untuk mengoreksi ejaan *query* pada kata kunci pencarian menggunakan *vector space model* (VSM) untuk pembobotan *query* dan algoritma *Damerau-Levenshtein Distance*. Penelitian ini mencoba mengusulkan suatu sistem untuk melakukan koreksi pada kata kunci pencarian seperti pada penelitian [3] dengan menggunakan algoritma *Damerau-Levenshtein Distance*, yaitu setiap kata dari *kata kunci* yang dimasukan oleh pengguna dibandingkan dengan kata – kata yang tepat yang disimpan sebagai kamus di *database*, namun menggunakan desain database kamus kata bahasa Indonesia, dataset pengujian, serta tahapan model sistem yang berbeda. Algoritma *Damerau-Levenshtein Distance* bekerja dengan menghitung jarak antara *StringSumber* dengan *StringTarget*. Yang dimaksud dengan jarak adalah berapa kali sebuah *StringSumber* harus diubah hingga sama persis dengan *StringTarget*. Jika perbandingan antara *StringSumber* dan *StringTarget1* memiliki jarak sama dengan 2, sementara perbandingan antara *StringSumber* dan *StringTarget2* memiliki jarak sama dengan 3, maka dapat disimpulkan bahwa *StringTarget1* lah kata yang mungkin ingin diketik oleh pengguna. Dengan implementasi algoritma *Damerau-Levenshtein Distance* untuk memperbaiki kesalahan pengetikan *kata kunci* pencarian ini diharapkan suatu website dapat menampilkan hasil pencarian dan hasil pengujian yang optimal.

2. Metodologi Penelitian

Metodologi penelitian yang digunakan pada penelitian ini terdiri dari studi literature, pengumpulan data, analisis dan desain, pembuatan program, dan pengujian [4].

A. Studi Literatur

Tahapan ini diperlukan untuk memahami konsep-konsep atau dasar-dasar teori yang digunakan dalam tiap tahapan penelitian. Literatur-literatur yang dipelajari adalah literatur yang berkaitan dengan implementasi algoritma *Damerau-Levenshtein Distance* dan pengoreksian kata yang salah atau *spell checking*. Algoritma *Damerau-Levenshtein Distance* yang diimplementasikan tidak berbeda dengan versi aslinya yang mendukung empat operasi *edit*, yaitu pengurangan, penambahan, subtitusi dan transposisi seperti pada gambar berikut.

```
algorithm damerau-levenshtein distance is
    input: strings a[1..length(a)], b[1..length(b)]
    output: distance, integer
    let d[0..length(a), 0..length(b)] be a 2-d array of integers, dimensions length(a)+1, length(b)+1
    for i := 0 to length(a) inclusive do
        d[i, 0] := i
    for j := 0 to length(b) inclusive do
        d[0, j] := j
    for i := 1 to length(a) inclusive do
        for j := 1 to length(b) inclusive do
            if a[i] = b[j] then
                cost := 0
            else
                cost := 1
            d[i, j] := minimum(d[i-1, j] + 1,     // deletion
                               d[i, j-1] + 1,     // insertion
                               d[i-1, j-1] + cost)  // substitution
            if i > 1 and j > 1 and a[i] = b[j-1] and a[i-1] = b[j] then
                d[i, j] := minimum(d[i, j],
                                   d[i-2, j-2] + cost)  // transposition
    return d[length(a), length(b)]
```

Gambar 2. Algoritma Damerau-Levenshtein Distance

Algoritma *Damerau-Levenshtein Distance* di atas akan menghasilkan jarak edit antara *string* target dan *string* sumber. Jarak edit inilah yang dibutuhkan oleh system untuk menentukan apakah terdapat kesalahan pengetikan pada suatu kata.

B. Pengumpulan Data

Tahapan ini merupakan tahap mengumpulkan data-data berupa kata – kata dalam bahasa Indonesia yang dianggap valid. Kata-kata tersebut adalah kata-kata yang biasa

digunakan oleh pengguna sebagai *kata kunci* pencarian. Kata – kata yang dijadikan sebagai pembanding disimpan sebagai kamus di *database*. Kata – kata pembanding ini terbatas pada bahasa Indonesia yang sesuai dengan KBBI (Kamus Besar Bahasa Indonesia) saja dan beberapa kata lainnya seperti nama orang.

Tabel 1. Struktur Tabel Kamus

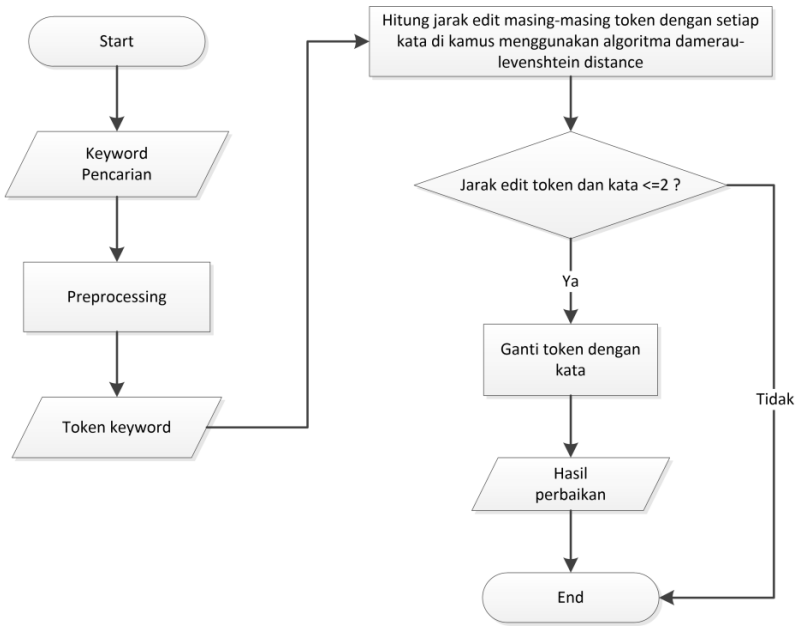
word (kata)	**first_l (huruf depan)**
a	a
aba	a
abad	a
dst	

Jumlah kata yang disimpan untuk pengujian adalah sebanyak 20.000 kata dan 250 artikel. Setiap kata yang ada pada *kata kunci* pencarian pengguna akan dihitung jarak *edit*nya dengan setiap kata yang ada pada kamus.

C. Analisis dan Desain

1. Analisis

Pada tahap ini akan dianalisis bagaimana struktur tabel kamus, metode pengecekan dan perbaikan *kata kunci* pencarian yang dimasukan, dan analisis sistem dengan menggunakan UML [5]. Tahapan pengecekan dan perbaikan *kata kunci* pencarian yang dimasukan digambarkan dalam flowchart berikut ini :



Gambar 3. Metode pengecekan dan perbaikan *kata kunci*

Berikut ini merupakan penjelasan setiap proses dalam flowchart pengecekan dan perbaikan *kata kunci* pencarian :

a. Menerima *kata kunci* pencarian

b. Melakukan *preprocessing kata kunci* pencarian dengan langkah – langkah sebagai berikut :

1) *Case folding*, yaitu tahapan merubah semua huruf di *kata kunci* pencarian menjadi *lowercase*.

2) *Tokenization*, yaitu tahapan menghapus semua tanda baca dari *kata kunci* jika terdapat tanda baca dan kemudian memisahkan deretan kata dalam *kata kunci* menjadi *token* atau potongan kata tunggal jika *kata kunci* berbentuk frase.

c. *Token – token kata kunci* dihasilkan dari tahap *preprocessing*

d. Menghitung jarak edit masing – masing *token* dengan setiap kata yang ada di kamus menggunakan algoritma *Damerau-Levenshtein Distance*.

e. Menentukan apakah jarak edit antara satu *token* dengan kata pembanding yang ada di kamus berjumlah <= 2. Jika jarak edit <= 2 maka proses akan berlanjut ke langkah berikutnya, jika tidak maka proses berakhir.

f. Mengganti *token* yang memiliki jarak edit <= 2 dengan kata pembanding.

g. Hasil perbaikan didapatkan. Selesai

2. Desain

Tahap desain merupakan tahap pembuatan desain tabel dan desain antarmuka yang nantinya akan digunakan pada tahap pembuatan program.

D. Pembuatan Program

Tahap ini merupakan tahap *coding* menggunakan bahasa pemrograman PHP. dan pembuatan database dengan menggunakan MySQL.

E. Pengujian

Pada tahap ini akan dilakukan pengujian keakuratan dan relevansi hasil pencarian menggunakan *precision* dan *recall*.

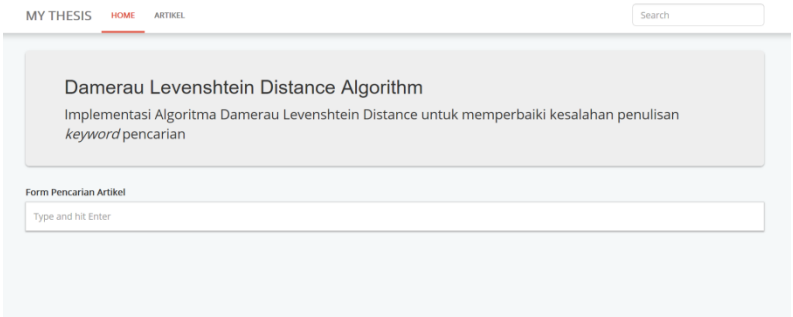
3. Hasil dan Pembahasan

Hasil dari penelitian ini adalah sistem untuk mengatasi masalah kesalahan pengetikan *kata kunci* pencarian agar fitur pencarian mendapatkan hasil yang optimal dan relevan terhadap *kata kunci* pengguna. Berikut adalah *interface* sistem yang dibangun.

A. Tampilan Halaman Website

a. Halaman Beranda

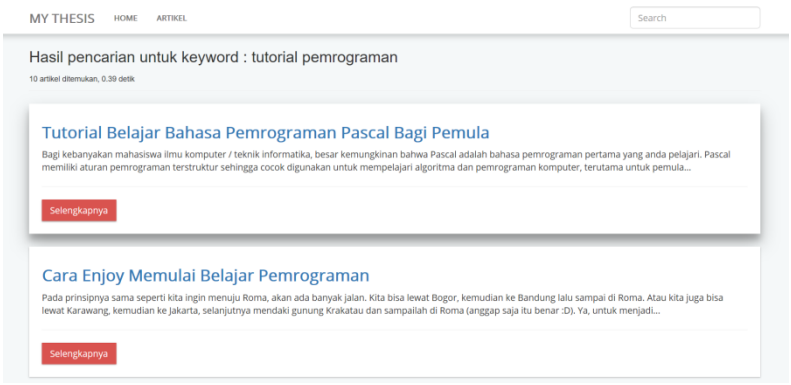
Gambar 4 menampilkan halaman beranda untuk pengunjung. Seperti halnya halaman beranda untuk pengelola, halaman ini hanya berisi deskripsi system dan *form* pencarian artikel.



Gambar 4. Halaman beranda pengunjung

b. Halaman Hasil Pencarian Artikel

Gambar 5 menampilkan halaman hasil pencarian artikel yang dilakukan oleh pengunjung. Hasil artikel yang didapatkan tergantung dengan *kata kunci* yang dinputkan.



Gambar 5 Halaman hasil pencarian artikel untuk pengunjung

B. Pengujian Perbaikan Pada *Kata kunci*

Jenis - jenis pengujian didasarkan pada empat operasi edit yang diperbolehkan dalam algoritma *Damerau-Levenshtein Distance*, yaitu: pengujian perbaikan *typo* karena kekurangan huruf dalam kata (operasi *insertion* atau penyisipan), pengujian perbaikan *typo* karena kelebihan huruf dalam kata (operasi *deletion* atau penghapusan), pengujian perbaikan *typo* karena posisi huruf tertukar dalam kata (operasi *transposition* atau transposisi), dan pengujian perbaikan *typo* karena terdapatnya huruf yang tidak benar dalam kata (operasi *substitution* atau subtitusi).

Skema pengujian yang dilakukan adalah dengan cara memasukan *kata kunci* pencarian dalam bentuk frasa yang terdiri dari dua sampai empat kata yang tidak ditulis secara benar (sesuai dengan jenis pengujian) ke *form* pencarian sistem, kemudian hasil perbaikan *kata kunci* yang dihasilkan oleh sistem akan dibandingkan dengan hasil perbaikan yang dilakukan oleh pakar. Saran perbaikan yang diberikan oleh pakar akan dijadikan acuan untuk mengukur nilai *precision* dan *recall* sistem dalam memberikan saran perbaikan yang benar dalam persentase seperti contoh berikut [6] :

Kata kunci : pnemu boal lampo
Saran perbaikan dari pakar : penemu bola lampu
Saran perbaikan dari sistem : penemu bual lampu
True Positive (Kata relevan tersugesti) = 2
False Positive (Kata tidak relevan tersugesti) = 1
True Negative (Kata relevan tidak tersugesti) = 1

$$Precision = \frac{TP}{TP + FP} \times 100 = \frac{2}{2+1} \times 100 = 66.66\%$$

$$Recall = \frac{TP}{TP + TN} \times 100 = \frac{2}{2+1} \times 100 = 66.66\%$$

Masing – masing jenis pengujian menggunakan sepuluh *kata kunci* yang sama dengan jenis *typo* yang berbeda tergantung dengan jenis pengujian. Kesepuluh *kata kunci* ini masing – masing memiliki sepuluh artikel yang relevan dari total 250 artikel di dalam sistem untuk dapat diukur perbandingan hasil pencarian artikel yang didapat dengan *kata kunci* sebelum dan sesudah diperbaiki oleh sistem dengan menghitung nilai *precision* dan *recall*-nya. Sepuluh *kata kunci* yang digunakan dalam pengujian diambil secara acak. 10 kata kunci tersebut adalah sebagai berikut :

1. Tutorial pemrograman

2. Jadwal rilis film
3. Pemain sepak bola terbaik
4. Penyanyi populer Indonesia
5. Daftar nama pahlawan nasional
6. Obat sakit perut
7. Jumlah utang Negara
8. Toko online termurah
9. Biaya hidup di Jakarta
10. Bahaya narkoba

Contoh perhitungan nilai *precision* dan *recall* artikel hasil pencarian adalah sebagai berikut :
Kata kunci : obat sakit perut
Hasil pencarian / artikel yang didapat : 11 artikel
True Positive (Artikel relevan yang didapat) = 10
False Positive (Artikel tidak relevan yang didapat) = 1
True Negative (Artikel relevan tidak didapat) = 0

$$Precision = \frac{TP}{TP + FP} = \frac{10}{10 + 1} = 0.90$$

$$Recall = \frac{TP}{TP + TN} = \frac{10}{10 + 0} = 1$$

Asumsi – asumsi dalam pengujian ini adalah :
1. Jumlah karakter atau huruf yang salah atau tidak benar dalam suatu kata tidak lebih dari dua buah karakter.
2. Kesalahan pengetikan kata dapat terjadi di pertengahan hingga akhir kata, bukan di karakter atau huruf awal kata.
3. Perbaikan kata tidak memperhatikan konteks atau makna dari *kata kunci*

Tabel 1 sampai dengan tabel 9 menunjukkan tabel – tabel hasil pengujian yang telah dilakukan.

Tabel 1. Perbandingan hasil perbaikan *typo* karena kekurangan huruf dalam kata oleh sistem dan pakar

No.	Kata kunci	Sistem	Pakar	*Precision*	*Recall*
1.	Tutrial pemroraman	Tutorial pemrograman	Tutorial pemrograman	100	100
2.	Jadal rlis flm	Jadwal rilis film	Jadwal rilis film	100	100
3.	Peman spak bla terbik	Pemain suka bola terbaik	Pemain sepak bola terbaik	75	75
4.	Penyayi popler Indonesa	Penyanyi populer Indonesia	Penyanyi populer Indonesia	100	100
5.	Daftr nma pahlwn nasonal	Daftar nama pailan nasional	Daftar nama pahlawan nasional	75	75
6.	Obt sait prut	Obat sakit perut	Obat sakit perut	100	100
7.	Jmlh utng negra	Jumlah utang netra	Jumlah utang negara	66.66	66.66
8.	Tok oline termrh	Toko online termurah	Toko online termurah	100	100
9.	Biay hdup di jakrta	Biaya hidup di Jakarta	Biaya hidup di Jakarta	100	100
10.	Bahya Nrkoba	Bahaya narkoba	Bahaya narkoba	100	100
Rata – rata akurasi (%)				91.66	91.66

Tabel 2. Perbandingan hasil pencarian artikel yang didapat sebelum dan sesudah perbaikan *typo* karena kekurangan huruf dalam kata

No.	Sebelum Diperbaiki	Sesudah Diperbaiki	*Precision*	*Recall*
1.	0 artikel	10 artikel (10 relevan, 0 tidak)	1	1
2.	0 artikel	11 artikel (10 relevan, 1 tidak)	0.90	1
3.	0 artikel	18 artikel (10 relevan, 8 tidak)	0.55	1
4.	0 artikel	36 artikel (10 relevan, 26 tidak)	0.27	1
5.	0 artikel	15 artikel (10 relevan, 5 tidak)	0.66	1
6.	0 artikel	10 artikel (10 relevan, 0 tidak)	1	1
7.	0 artikel	10 artikel (10 relevan, 0 tidak)	1	1
8.	0 artikel	12 artikel (10 relevan, 2 tidak)	0.83	1
9.	0 artikel	11 artikel (10 relevan, 1 tidak)	0.90	1
10.	0 artikel	10 artikel (10 relevan, 0 tidak)	1	1
Rata – rata			0.81	1

Tabel 3. Perbandingan hasil perbaikan *typo* karena kelebihan huruf dalam kata oleh sistem dan pakar

No.	Kata kunci	Sistem	Pakar	*Precision*	*Recall*
1.	Tuttorial pemrogramman	Tutorial pemrograman	Tutorial pemrograman	100	100
2.	Jaduwal rillis filem	Jadwal rilis film	Jadwal rilis film	100	100
3.	Pemaiin seppak bolla terbaikk	Pemain sepepak bola terbaik	Pemain sepak bola terbaik	75	75
4.	Penyannyi poppuler Indonessia	Penyanyi populer Indonesia	Penyanyi populer Indonesia	100	100
5.	Dafftar namma pahlawwan nassional	Daftar nama pahlawan nasional	Daftar nama pahlawan nasional	100	100
6.	Obbat sakiit perutt	Obat sakit perut	Obat sakit perut	100	100
7.	Jummlah utaang negarra	Jumlah utang negara	Jumlah utang negara	100	100
8.	Tokko onnline termmurah	Toko online termurah	Toko online termurah	100	100
9.	Biayya hiduup di Jakkarta	Biaya hidup di Jakarta	Biaya hidup di Jakarta	100	100
10.	Bahaiya narkboba	Bahaya narkoba	Bahaya narkoba	100	100
Rata – rata akurasi (%)				97.5	97.5

Tabel 4. Perbandingan hasil pencarian artikel yang didapat sebelum dan sesudah perbaikan *typo* karena kelebihan huruf dalam kata

No.	Sebelum Diperbaiki	Sesudah Diperbaiki	*Precision*	*Recall*
1.	0 artikel	10 artikel (10 relevan, 0 tidak)	1	1
2.	0 artikel	11 artikel (10 relevan, 1 tidak)	0.90	1
3.	0 artikel	18 artikel (10 relevan, 8 tidak)	0.55	1
4.	0 artikel	36 artikel (10 relevan, 26 tidak)	0.27	1
5.	0 artikel	16 artikel (10 relevan, 6 tidak)	0.62	1
6.	0 artikel	10 artikel (10 relevan, 0 tidak)	1	1
7.	0 artikel	10 artikel (10 relevan, 0 tidak)	1	1
8.	0 artikel	12 artikel (10 relevan, 2 tidak)	0.83	1
9.	0 artikel	11 artikel (10 relevan, 1 tidak)	0.90	1
10.	0 artikel	10 artikel (10 relevan, 0 tidak)	1	1

Rata – rata	0.80	1

Tabel 5. Perbandingan hasil perbaikan *typo* karena posisi huruf tertukar dalam kata oleh sistem dan pakar

No.	Kata kunci	Sistem	Pakar	*Precision*	*Recall*
1.	Tutorail pemorgraman	Tutorial pemrograman	Tutorial pemrograman	100	100
2.	Jawdal riils flim	Jadwal rilis film	Jadwal rilis film	100	100
3.	Pemian seapk boal terbiak	Pemain sepak bola terbaik	Pemain sepak bola terbaik	100	100
4.	Penyaniy populre Indonesai	Penyanyi populer Indonesia	Penyanyi populer Indonesia	100	100
5.	Datfar naam pahalwan nasoinal	Daftar nama pahlawan nasional	Daftar nama pahlawan nasional	100	100
6.	Oabt skait pertu	Obat sekait perut	Obat sakit perut	66.66	66.66
7.	Julmah utagn negraa	Jumlah utang negara	Jumlah utang negara	100	100
8.	Took onlnie terumrah	Toko online termurah	Toko online termurah	100	100
9.	Biaay hidpu di jakrata	Biaya hidup di jakarta	Biaya hidup di Jakarta	100	100
10.	Baahya nakroba	Bahaya narkoba	Bahaya narkoba	100	100
Rata – rata akurasi (%)				96.66	96.66

Tabel 6. Perbandingan hasil pencarian artikel yang didapat sebelum dan sesudah perbaikan *typo* karena posisi huruf tertukar dalam kata

No.	Sebelum Diperbaiki	Sesudah Diperbaiki	*Precision*	*Recall*
1.	0 artikel	10 artikel (10 relevan, 0 tidak)	1	1
2.	0 artikel	11 artikel (10 relevan, 1 tidak)	0.90	1
3.	0 artikel	18 artikel (10 relevan, 8 tidak)	0.55	1
4.	0 artikel	36 artikel (10 relevan, 26 tidak)	0.27	1
5.	0 artikel	16 artikel (10 relevan, 6 tidak)	0.62	1
6.	0 artikel	10 artikel (10 relevan, 0 tidak)	1	1
7.	0 artikel	10 artikel (10 relevan, 0 tidak)	1	1
8.	0 artikel	12 artikel (10 relevan, 2 tidak)	0.83	1
9.	0 artikel	11 artikel (10 relevan, 1 tidak)	0.90	1
10.	0 artikel	10 artikel (10 relevan, 0 tidak)	1	1
Rata – rata			0.80	1

Tabel 7. Perbandingan perbaikan *typo* karena adanya huruf yang tidak tepat dalam kata oleh sistem dan pakar

No.	Kata kunci	Sistem	Pakar	*Precision*	*Recall*
1.	Tutprial pemeogranan	Tutorial pemrograman	Tutorial pemrograman	100	100
2.	Jaswsl rolis fikn	Jaswsl rilis fikn	Jadwal rilis film	100	33.33
3.	Penain swpsk bils twrbaik	Pemain sepak bis terbaik	Pemain sepak bola terbaik	66.66	66.66
4.	Pwnyamyi popiker Imdonwsia	Penyanyi populer Indonesia	Penyanyi populer Indonesia	100	100
5.	Dafysr nams pajlswan nadional	Daftar nas pahlawan nasional	Daftar nama pahlawan nasional	75	75

6.	Ovat salit peeut	Oval sulit perut	Obat sakit perut	33.33	33.33
7.	Jumkah utwmg nwgaea	Jumlah utang negara	Jumlah utang negara	100	100
8.	Tokp onlime twrmirah	Toko online tersirah	Toko online termurah	66.66	66.66
9.	Biaua hodup di Jakaeta	Biaya hidup di Jakarta	Biaya hidup di Jakarta	100	100
10.	Bajwya narjiba	Baja narkoba	Bahaya narkoba	50	50
Rata – rata akurasi (%)				79.16	72.49

Tabel 8. Perbandingan hasil pencarian artikel yang didapat sebelum dan sesudah perbaikan *typo* karena adanya huruf yang tidak tepat dalam kata

No.	**Sebelum Diperbaiki**	**Sesudah Diperbaiki**	***Precision***	***Recall***
1.	0 artikel	10 artikel (10 relevan, 0 tidak)	1	1
2.	0 artikel	5 artikel (5 relevan, 0 tidak)	1	0.5
3.	0 artikel	18 artikel (10 relevan, 8 tidak)	0.55	1
4.	0 artikel	36 artikel (10 relevan, 26 tidak)	0.27	1
5.	0 artikel	17 artikel (10 relevan, 7 tidak)	0.58	1
6.	0 artikel	10 artikel (10 relevan, 0 tidak)	1	1
7.	0 artikel	10 artikel (10 relevan, 0 tidak)	1	1
8.	0 artikel	12 artikel (10 relevan, 2 tidak)	0.83	1
9.	0 artikel	11 artikel (10 relevan, 1 tidak)	0.90	1
10.	0 artikel	10 artikel (10 relevan, 0 tidak)	1	1
Rata – rata			0.81	0.95

Tabel 9. Nilai rata – rata *precision* dan *recall* hasil pengujian secara menyeluruh

***Precision* Perbaikan *Kata kunci* (%)**	***Recall* Perbaikan *Kata kunci* (%)**	***Precision* Pencarian Artikel**	***Recall* Pencarian Artikel**
91.24%	89.58%	0.80	0.98

Hasil serangkaian pengujian perbaikan kesalahan pengetikan kata kunci pencarian atau *typo* menunjukan bahwa algoritma *Damerau-levenshtein distance* mampu memberikan saran perbaikan dengan nilai *precision* mencapai 91.24% dan nilai *recall* mencapai 89.58% secara keseluruhan, dengan skor *precision* dan *recall* tertinggi diperoleh pada pengujian perbaikan *typo* karena kelebihan huruf dalam kata yaitu mencapai 97.5% (tabel 3), sementara nilai *precision* dan *recall* terendah diperoleh pada pengujian perbaikan *typo* karena adanya huruf yang tidak tepat dalam kata dengan nilai 79.16% untuk *precision* dan 72.49 % untuk *recall*.

Hasil uji coba 3, 5, 7 pada tabel 1, uji coba 3 pada tabel 3, uji coba 6 pada tabel 5, dan uji coba 2, 5, 6, 8, 10 pada tabel 7 menunjukan kelemahan dari algoritma *Damerau-levenshtein distance* yaitu risiko kemungkinan keadaan *false positive* yang tinggi jika digunakan untuk mengoreksi frasa yang memiliki makna seperti *kata kunci* pencarian. Contoh pada uji coba 5 tabel 7, kata *typo* "nams" seharusnya dikoreksi menjadi "nama" tetapi dikoreksi menjadi "nas" karena jarak edit antara kata "nama" dan "nas" dengan kata *typo* "nams" sama yaitu 1, namun kata "nas" terpilih karena kata tersebut merupakan kandidat terakhir yang memenuhi syarat sebagai kata pengganti (jarak edit <= 2) setelah proses *looping* algoritma selesai. Jika memperhatikan kata sebelumnya yaitu "daftar" dan kata sesudahnya yaitu "pahlawan", maka kata pengganti yang lebih baik adalah kata "nama" dibandingkan kata "nas". Berdasarkan hasil pengujian terhadap dataset dan tahapan model sistem yang berbeda dengan penelitian [3], Algoritma *Damerau-levenshtein distance* mampu bekerja optimal, dimana hasil pencarian setelah *kata kunci* diperbaiki menunjukan hasil

pencarian dengan rata – rata nilai *precision* mencapai 0.80 dan nilai *recall* mencapai 0.98 pada data yang diuji, dan pada penelitian [3] peningkatan rata-rata *precision* 44,82%.

4. Kesimpulan

Algoritma *Damerau-levenshtein distance* untuk perbaikan kesalahan pengetikan kata kunci pencarian mampu memberikan nilai rata – rata 91.24% untuk *precision* dan 89.58% untuk *recall* berdasarkan pengujian yang telah dilakukan. Meskipun demikian, algoritma ini memiliki kelemahan jika digunakan untuk memperbaiki kata dalam sebuah frasa yang memiliki makna, karena kemungkinan keadaan *false positive* yang tinggi sehingga saran perbaikan yang muncul tidak sesuai dengan makna atau konteks dari *keyword* yang diperbaiki

Ucapan Terima Kasih

Kami berterimakasih kepada Jurusan Teknik Informatika, Unversitas Palangka Raya dan rekan-rekan sejawat yang telah mendukung penelitian kami.

Daftar Pustaka

[1] Maghfira, T. N., Cholissodin, I., & Widodo, A. W. (2017). Deteksi Kesalahan Ejaan dan Penentuan Rekomendasi Koreksi Kata yang Tepat Pada Dokumen Jurnal JTIIK Menggunakan Dictionary Lookup dan Damerau-Levenshtein Distance. Malang. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 1(6), 498-506.

[2] Mishra, R., & Kaur, N. (2013). A survey of spelling error detection and correction techniques. International Journal of Computer Trends and Technology, 4(3), 372-374.

[3] Sutisna, U., & Adisantoso, J. (2010). Koreksi Ejaan Query Bahasa Indonesia Menggunakan Algoritme Damerau Levenshtein. Jurnal Ilmiah Ilmu Komputer, 8(2).

[4] Jogiyanto, H. M. (2008). Metodologi penelitian sistem informasi. Yogyakarta: Andi Offset.

[5] Nugroho, A. (2009). rekayasa perangkat lunak menggunakan UML dan JAVA. Penerbit Andi.

[6] Wibowo, A. (2011). Pengujian Kerelevanan Sistem Temu Kembali Informasi. In Seminar Nasional Ilmu Komputer.