semanTIK, Vol.2, No.1, Jan-Jun 2016, pp. 279-286
ISSN : 2502-8928 (Online)

APLIKASI PENDETEKSI KEMIRIPAN ISI TEKS DOKUMEN MENGGUNAKAN METODE *LEVENSHTEIN DISTANCE*

Na'firul Hasna Ariyani[*1], Sutardi[2], Rahmat Ramadhan[3]
[1,2,3]Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo, Kendari
e-mail: [*1]**nafirulhasna@gmail.com,** [2]sutardi_hapal@yahoo.com, [3]rahmat.ramadhan@innov-center.org

Abstrak

Teknologi informasi yang berkembang pesat membawa dampak positif dan negatif bagi kehidupan Salah satu dampak negatif yang ditimbulkan adalah plagiarisme. Plagiarisme adalah tindakan menjiplak karya orang lain dan mengakui sebagai hasil karya pribadinya. Oleh karena itu pendeteksian plagiarisme perlu dilakukan untuk mengurangi penjiplakan terhadap hasil karya orang lain.

Penelitian ini bertujuan untuk mendeteksi kemiripan dokumen teks menggunakan algoritma *Levenshtein Distance* sehingga dapat digunakan untuk membantu menentukan plagiarisme. Tipe dokumen yang diuji adalah .pdf .docx dan .txt. Dokumen yang digunakan untuk perbandingan teks ini adalah dokumen yang berbahasa Indonesia.ng, Tahapan dalam sistem adalah *preprocessing* yang teridiri dari *Case Folding, Tokenizing, Filtering, Stemming, Sorting*. Setelah proses preprocessing maka tahap selanjutnya adalah dilakukan perhitungan menggunakan metode *Levenshtein Distance* dan penggukuran nilai similarity sehingga mendapatkan nilai presentase kemiripan antara kedua dokumen.

Pada pengujian menggunakan data real yaitu data dokumen berplagiat dengan algoritma Levenshtein Distance menghasilkan nilai similarity yang tinggi yaitu diatas 77% sampai 100% untuk dokumen yang tingkat kemiripannya tinggi. Sedangkan untuk dokumen dengan tingkat kemiripan yang rendah atau tidak berplagiat maka menghasilkan nilai similarity dibawah 40%.

Kata kunci— Dokumen, *Levenshtein Distance*, *Preprocessing*, Kemiripan, Plagiat

Abstract

The rapidly evolving information technology brings positive and negative impacts to the lives One of the negative impacts is plagiarism. Plagiarism is the act of plagiarizing the work of others and recognize as his own handiwork. Therefore, detection of plagiarism needs to be done to reduce plagiarism against other people's work.

This thesis aims to detect text document similarity algorithm using Levenshtein Distance so that it can be used to help determine plagiarism. Type of document to be tested is .docx and .pdf .txt. Stages in the system is preprocessing that consist of Case Folding, tokenizing, Filtering, Stemming, Sorting. After the preprocessing the next step is to do the calculation using the method Levenshtein Distance and penggukuran value of similarity thus getting a percentage value of the similarity between the two documents.

In testing using real data ie data documents berplagiat with Levenshtein Distance algorithm produces a high similarity value is above 77% to 100% for the document that a high level of similarity. As for the document with a low degree of similarity or not berplagiat then generate similarity values below 40%.

Keywords— *Document, Levenshtein Distance, Preprocessing, Similarity, Plagiarism*

1. PENDAHULUAN

Pemanfaatan teknologi digital telah menjadi kebutuhan dalam era modern saat ini. Komponen yang ada di dalam dunia digital salah satunya adalah dokumen teks. Dokumen dalam bentuk digital memudahkan dalam hal penyimpanan, efisien, mudah dicari, bahkan mudah dalam hal penjiplakan. Praktek penjiplakan sering terjadi dalam dunia akademik, baik tingkat sekolah maupun perguruan tinggi. Demi menyelesaikan tugas-tugasnya dengan cepat, siswa maupun mahasiswa dapat melakukan teknik *copy-paste-modify* tanpa perlu mempelajari dan mengeksplorasi materi terlebih dahulu.

Penjiplakan atau plagiarisme berarti mencontoh atau meniru atau mencuri tulisan dan karya orang lain yang kemudian diakui sebagai karangannya sendiri dengan ataupun tanpa seizin penulisnya. Penjiplakan dokumen digital bukanlah hal yang susah, cukup dengan menggunakan teknik *copy-paste-modify* pada sebagian isi dokumen dan bahkan keseluruhan isi dokumen sudah bisa dikatakan bahwa dokumen tersebut merupakan hasil duplikasi dari dokumen lain[1].

Cara untuk mengatasi permasalahan terjadinya plagiarisme, yaitu dengan mencegah dan mendeteksi. Mencegah berarti menjaga atau menghalangi agar plagiarisme tidak dilakukan.Usaha ini harus dilakukan sedini mungkin terutama pada sistem pendidikan dan moral masyarakat. Cara mendekteksi dokumen yang tergolong dalam jenis plagiarisme dapat dilakukan secara manual yaitu bisa dengan cara membandingkan manual antara kedua jenis dokumen tersebut namun hal tersebut tentulah kurang efektif.

Salah satu metode yang tepat dalam melakukan deteksi kemiripan dokumen teks adalah dengan melakukkan perhitungan dengan metode *Levenshtein Distance*. *Levenshtein Distance* memperhatikan tiga operasi dalam menentukan jarak diff, yaitu (1) operasi penyisipan (*insertion*), (2) operasi penghapusan (*deletion*), (3) operasi penggatian (*subtitution*), sebuah huruf yang berdekatan. Hasil akhir yang diberikan dalam sistem ini adalah memberikan presentase nilai *similarity* antara kedua dokumen [2].

2. METODE PENELITIAN

2.1 Plagiarisme

Plagiarisme merupakan tindakan kriminal yang sering terjadi dalam dunia akademis. Plagiarisme itu sendiri berasal dari kata latin "*Plagiarus*" yang berarti penculik dan "*Plagiare*" yang berarti mencuri. Jadi, secara sederhana plagiat berarti mengambil ide, kata-kata, dan kalimat seseorang dan memposisikannya sebagai hasil karyanya sendiri atau mengunakan ide, kata-kata, dan kalimat tanpa mencantumkan sumber dimana seorang penulis mengutipnya[3].

Jenis plagiarisme berdasarkan klasifikasinya diantaranya:Jenis plagiarisme berdasarkan aspek yang dicuriyaitu kategori plagiarisme ide, plagiarisme isi, plagiarisme kata, kalimat, paragraf, dan plagiarisme total. Klasifikasi berdasarkan sengaja atau tidaknya plagiarism yaitu plagiarism engaja dan plagiarisme tidak sengaja.Berdasarkan pada pola plagiarismeyaitu plagiarisme kata demi kata (*word for word plagiarizing*) dan plagiarisme mosaik.Klasifikasi berdasarkan proporsi atau persentase kata, kalimat, paragraf yang dibajak yaitu:

a) Plagiarisme ringan, plagiarisme yang jumlah proporsi atau persentase kata, kalimat, paragraf yang dibajak tidak melebihi 30 persen (< 30%).
b) Plagiarisme sedang, plagiarisme yang jumlah proporsi atau persentase kata, kalimat, paragraf yang dibajak antara 30-70 persen.
c) Plagiarisme berat, plagiarisme yang jumlah proporsi atau persentase kata, kalimat, paragraf yang dibajak lebih dari 70 persen (>70%).

2.2 Dokumen

Dokumen adalah menurut bahasa Inggris berasal dari kata "*document*" yang mempunyai arti suatu yang tertulis atau yang tercetak dan segala benda yang memiliki berbagai keterangan dipilih untuk di disusun, di kumpulkan, di sediakan ataupun untuk disebarkan [4], sedangkan menurut Kamus Umum Bahasa Indonesia menyebutkan dokumen adalah sesuatu yang tertulis atau tercetak yang dapat dipergunakan sebagai bukti atau keterangan. Dokumen merupakan salah satu hal yang sangat penting karena merupakan sumber informasi yang diperlukan oleh suatu

instansi,organisasi, atau Negara. Tanpa dokumen kita akan kehilangan data-data yang diperlukan untuk kegiatan kantor/organisasi masa yang akan datang.

2.3 Teks Mining

Text mining memiliki definisi menambang data yang berupa teks dimana sumber data biasanya di dapatkan dari dokumen, dan tujuannya adalah mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen.

Text mining merupakan penerapan konsep dan teknik data mining untuk mencari pola dalam teks, yaitu proses penganalisisan teks guna menyarikan informasi yang bermanfaat untuk tujuan tertentu. Berdasarkan ketidakteraturan struktur data teks, maka proses text mining memerlukan beberapa tahap awal yang pada intinya adalah mempersiapkan agar teks dapat diubah menjadi lebih terstruktur[5].

Sistem yang diusulkan dalam pedeteksi isi teks dokumen adalah sistem yang dilakukan secara otomatis. Aplikasi proses pendeteksi kemiripan isi teks dokumen yang diusulkan ini menggunakan metode Leveinstein Distance. Dokumen yang diperiksa dalam sistem ini adalah dokumen berekstensi .doc, .pdf dan .txt. *User* dapat memasukan dokumen asli dan dokumen pembanding yang akan dihitung tingkat kemiripan isi teks dokumen tersebut. Keluaran dari sistem ini adalah presentase tingkat kemiripan isi teks dokumen. Setelah *user* memasukan kedua dokumen yang akan menjadi pembanding maka sistem akan melakukan tahap *preprocessing*dan tahap pencocokan*string.*

2.4 Analisa Kebutuhan Sistem

1. Kebutuhan Data Masukan

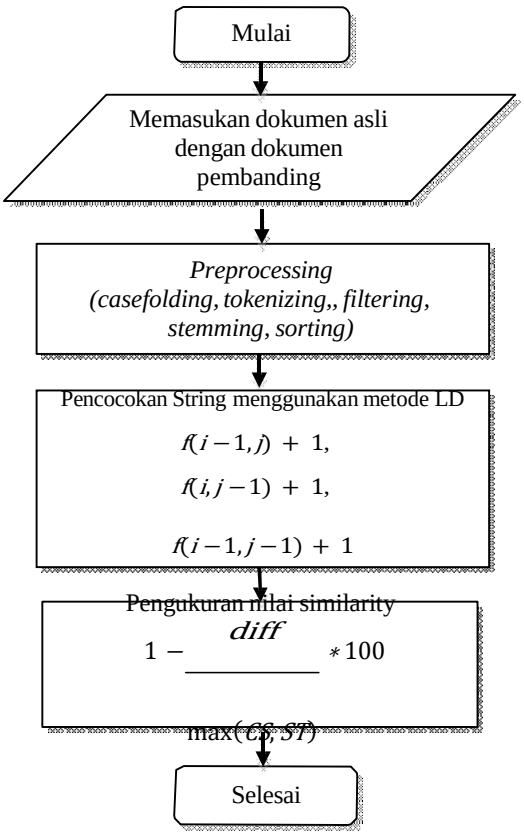
Untuk data masukan yang dibutuhkan dari sistem ini adalah dokumen yang berekstensi .doc, .txt, dan .pdf. Selanjutnya sistem akan menggunakan metode *Levenshtein Distance* dan Pengukuran Nilai *Similarity* untuk menghasilkan *output.*

2. Kebutuhan Data Keluaran

Data keluaran dari sistem yang telah diproses untuk kemudian ditampilkan kepada pengguna sistem yaitu presentase nilai kemiripan dan kesimpulan dari hasil presentase apakah dokumen tersebut merupakan plagiat atau bukan.

2.5 Arsitektur

Flowchart dari proses sistem yang dilakukan secara keseluruhan dapat dijelaskan seperti Gambar 1.



Gambar 1 *Flowchart* program

2.6 Ilustrasi Metode Levenhtein Distance Terhadap Perencanaan Aplikasi

1. Preprocessing

Fungsi *preprocessing* pada program ini adalah untuk mendapatkan *keyword* yang akan digunakan sebagai pencocokan *string* atau perbandingan dokumen.

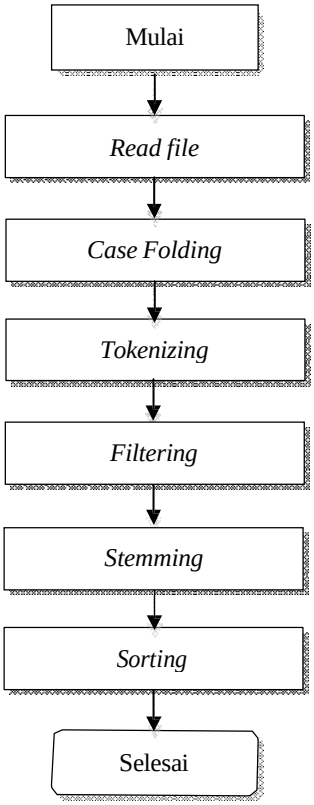
a. Case Folding

Proses *Case folding* adalah tahap mengubah semua huruf dalam dokumen menjadi huruf kecil hanya huruf a sampai z yang diterima.

b. Tokenizing

Pada dasarnya proses *Tokenizing* yaitu proses memisahkan setiap kata yang menyusun suatu

dokumen. Umumnya setiap kata teridentifikasi atau terpisahkan dengan kata lain oleh karakter spasi, sehingga proses *Tokenizing* mengandalkan karakter spasi pada dokumen untuk melakukan pemisahan kata



Gambar 2 *Flowchart Preprocessing*

c. Filtering

Tahap *Filtering* adalah tahap pengambilan kata yang penting dari hasil *Tokenizing*. Tahap *Filtering* ini dapat menggunakan algoritma *stoplist* dan *wordlist*.

d. *Stemming*

Stemming adalah proses mengubah kata menjadi kata dasarnya dengan menghilangkan imbuhan imbuhan pada kata dalam dokumen. Dalam tahap *stemming* diperlukan suatu algoritma yaitu Nazief dan Andriani.

Algoritma Nazief dan Andriani. Merupakan sebuah algoritma untuk mencari sebuah kata dasar atau lebih dikena dengan istilah *stemming*. Algoritma Nazief dan Adriani adalah algoritma *stemming* yang digunakan khusus untuk bahasa indonesia, walaupun ada banyak algoritma *stemming* lainnya untuk bahasa indonesia, akan tetapi Nazief dan Adriani lebih banyak digunakan oleh para praktisi maupun para pegiat akademik, karena memang sampai saat ini Nazief dan Adriani mempunyai akurat yang baik jika dibandingkan dengan yang lainnya.

Konjungsi dalam Algoritma *stemming* Nazief dan Adriani dikembangkan berdasarkan aturan morfologi Bahasa Indonesia yang mengelompokkan imbuhan menjadi awalan (prefix), sisipan (*infix*), akhiran (*suffix*) dan gabungan awalan akhiran (*confixes*) [6].

e. Sorting

Sorting teks digunakan untuk mengurutkan kata hasil dari *stemming* secara *ascending* atau menaik sehingga pencocokan string dokumen dilakukan pada data yang sudah terurut [5].

2. Penerapan Metode Levenshtein Distance

Levenshtein Distance dibuat oleh Vladimir Levenshtein pada tahun 1965. Perhitungan edit distance didapatkan dari matriks yang digunakan untuk menghitung jumlah perbedaan string antara dua string.

Ada 3 macam operasi utama yang dapat dilakukan oleh algoritma ini yaitu :

a) Operasi Pengubahan Karakter

Operasi pengubahan karakter merupakan operasi menukar sebuah karakter dengan karakter lain contohnya penulis menuliskan *string* "yamg" menjadi "yang". Dalam kasus ini karakter "m" diganti dengan huruf "n".

b) Operasi Penambahan Karakter

Operasi penambahan karakter berarti menambahkan karakter ke dalam suatu string. Contohnya string "kepad" menjadi string "kepada", dilakukan penambahan karakter "a" di akhir *string*. Penambahan karakter tidak hanya dilakukan diakhir kata, namun bias ditambahkan diawal maupun disisipkan di tengah *string*.

c) Operasi Penghapusan Karakter

Operasi penghapusan karakter dilakukan untuk menghilangkan karakter dari suatu *string*. Contohnya *string* "barur" karakter terakhir dihilangkan sehingga menjadi string "baru". Pada operasi ini dilakukan penghapusan karakter "r" [2].

Algoritma ini berjalan mulai dari pojok kiri atas sebuah array dua dimensi yang telah diisi sejumlah karakter sring awal dan string target dan diberikan nilai *cost*. Nilai *cost* pada ujung

kanan bawah menjadi nilai *edit-distance* yang menggambarkan jumlah perbedaan dua *string*.

$f(0,0) = 0$

$f(i,0) = i$

$f(0,j) = j$

$f(i,j) = min\{$

$f(i-1,j) + 1$; // *deletion*

$f(i,j-1) + 1$; // *insertion*

$f(i-1,j-1) + 1$ // *substitution*

3. Perhitungan Nilai Similarity

Setelah mendapatkan biaya *edit-distance* maka untuk menghitung nilai *Levenshtein Distance* atau perhitungan *similarity* menggunakan Persamaan (1).

$$Plagiarized\ Value = \{1 - \frac{diff}{max(CS,ST)}\} * 100 \quad (1)$$

Keterangan :

CS = *Source String*

ST = *Target String*

Similarity = Nilai kemiripan

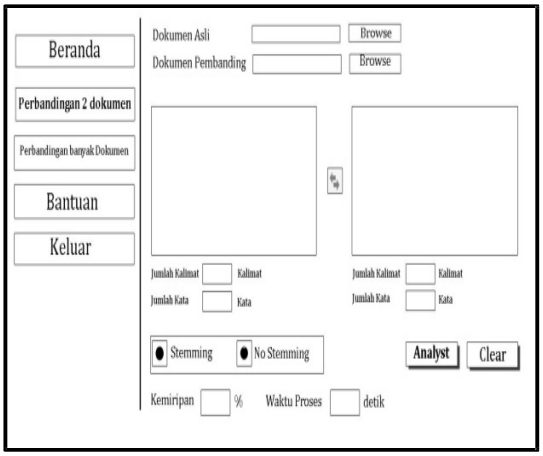
Diff = Jarak *Levenshtein*

Max(*CS*,*ST*) = Nilai *string* terpanjang [7].

3. HASIL DAN PEMBAHASAN

3. 1 Perancangan Interface

Perancangan antar muka (*interface*) merupakan perancangan untuk melihat desain awal dari sebuah sistem. Gambar 2 adalah perancangan sistem *interface* dari peracangan aplikasi pendeteksi kemiripan isi teks dokumen dengan menggunakan metode *Levenshtein Distance*.



Gambar 2. Desain *interface* halaman perbandingan dua dokumen

Pada menu ini *user* dapat memasukan dokumen asli dan dukumen pembanding. Kemudian dokumen yang dimasukan akan tampil pada kolom dan akan tampil info jumlah kalimat dan kata. *User* dapat memilih *stemming* atau *nostemming* dan memilih analis sehingga akan muncul nilai kemiripan dan

Gambar 3 Desain *interface* halaman perbandingan banyak dokumen

Pada desain perbandingan banyak dokumen *user* dapat memasukan dokumen kedalam tabel apabila dokumen tidak tersdeteksi plagiat maka dokumen akan masuk kedalam tabel, sedangkan dokumen yang terdeteksi plagiat dengan salah satu dari dokumen yang ada dalam tabel maka secara otomastis dokumen tidak dapat masuk ke dalam tabel.

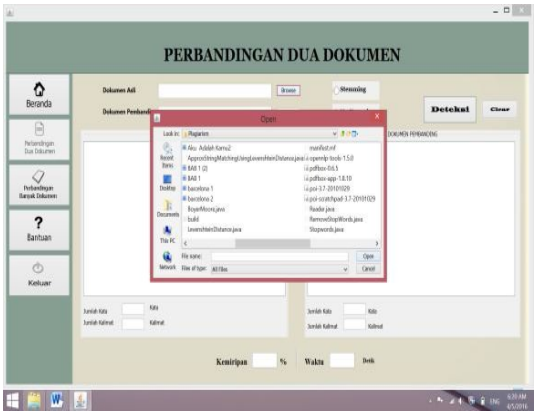
3.2 Pengujian Sistem

Pengujian ini dilakukan untuk memeriksa apakah semua menu dan submenu yang ada pada sistem dapat berfungsi dengan baik. Pengujian ini dilakukan dengan memasukkan dokumen asli dan dokumen pembanding, memilih proses *stemming* atau *no stemming*, melakukan deteksi kemiripan dokumen, menampilkan perbandingan banyak dokumen, melihat menu bantuan.

1. Tampilan Beranda pada Gambar 4 merupakan tampilan antarmuka yang muncul pertama ketika menjalankan program. Terdapat tombol yang memudahkan *user* untuk langsung memasukan dokumen dengan memilih *Browse* atau dapat memilih menu Perbandingan Dua Dokumen, Perbandingan banyak Dokumen, Bantuan, dan Exit.

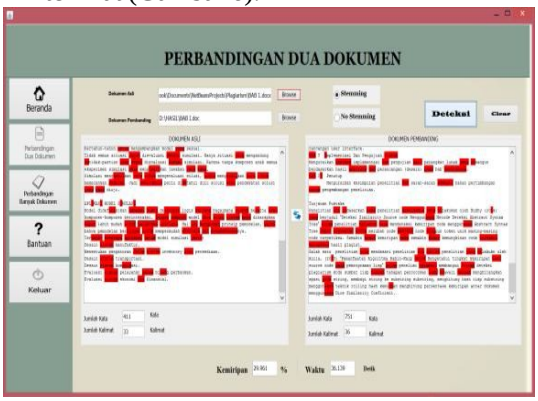
Gambar 4 Tampilan Beranda

2. Menu yang pertama adalah perbandingan dua dokumen (Gambar 5). Tahapan yang dapat dilakukan pada menu ini adalah memasukan dua dokumen yaitu dokumen asli dan dokumen pembanding yang akan diperiksa ke dalam *text editor*.



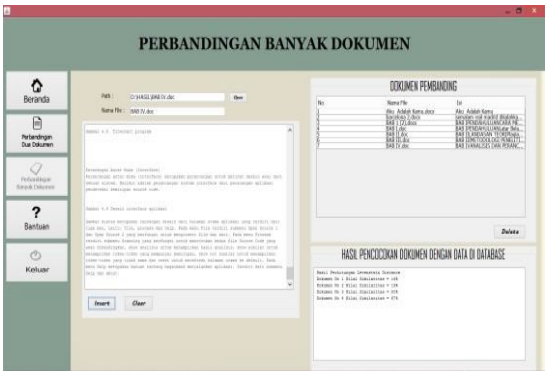
Gambar 5 Tampilan mengambil *file*

Setelah kedua dokumen tampil pada teks area maka langkah selanjutnya adalah memilih tombol deteksi sehingga hasil presentase kemiripan antar dokumen bisa terlihat (Gambar 6).



Gambar 6 Tampilan perbandingan dua dokumen

3. Pada menu Halaman Perbandingan banyak dokumen pengguna dapat melakukan perbandingan dengan banyak dokumen yang ada dalam *database* (Gambar 7). Proses perbandingan sama dengan proses perbandingan dua dokumen dalam menu yang sebelumnya. Pengguna dapat memasukan dokumen untuk dimasukan kedalam *database*. Tahap pertama adalah membaca *file* yang akan dimasukan kemudian setelah dokumen terbaca kedalam teks area maka akan dilakukan insert, apabila dokumen yang dimasukan terdeteksi plagiat dengan salah satu dokumen yang ada dalam database maka secara otomatis dokumen tidak akan ter-input kedalam *database*. Begitu sebaliknya apabila dokumen yang dimasukan tidak terdeteksi plagiat dengan salah satu dokumen yang ada dalam database maka dokumen akan secara otomatis masuk ke dalam *database*.



Gambar 7 Perbandingan Banyak Dokumen

4. Pada menu bantuan (Gambar 8) pengguna dalam melihat bantuan dalam menjalankan proses perbandingan. Terdapat dua menu dalam proses tersebut yaitu Perandingan Dua Dokumen dan Perbandingan banyak dokumen.

Gambar 8 Tampilan Menu Bantuan

3.3 Hasil Analisa

Pada pengujian sistem yang dilakukan, sistem memeriksa dua dokumen yaitu dokumen asli dan dokumen pembanding. Daftar dokumen asli yang di uji dapat ditunjukkan oleh Tabel 1 dan daftar dokumen pembanding yang di uji dapat ditunjukkan oleh Tabel 2.

Tabel 1 Daftar Dokumen Asli

No	Nama File	Isi	Tipe File	Ukuran file
1	Dok_Asli_1	Berisi makalah tentang pencemaran air	.doc	27 Kb
2	Dok_Asli_2	Berisi jurnal "sistem penilaian otomatis jawaban esai menggunakan Algoritma *Levenshtein Distance*"	.pdf	1196 Kb
3	Dok_Asli_3	Berisi tentang materi Cybercrime	.doc	24 Kb
4	Dok_Asli_4	Berisi tentang makalah Ekonomi islam	.doc	28 Kb
5	Dok_Asli_5	Berisi tentang materi jantung	.txt	13 Kb
6	Dok_Asli_6	Berisi tentang materi media sosial	.doc	8 Kb
7	Dok_Asli_7	Berisi tentang jurnal "TeknologiInformasi: Kesiapan Pustakawan Memanfaatkannya"	.doc	504 Kb
8	Dok_Asli_8	Berisi tentang jurnal "Tinjauan profesi di bidang teknologi Informasi"	.pdf	125 Kb
9	Dok_Asli_9	Berisi tentang "Pengertian Mea dan Ciri-ciri Masyarakat Ekonomi ASEAN"	.txt	6 Kb
10	Dok_Asli_10	Berisi tentang pembahasan "Pemanasan Global"	.doc	924 Kb

Tabel 2 Daftar Dokumen Pembanding

No	Nama File	Isi	Tipe file	Ukuran file
1	Dok_Uji_1	Berisi pembahasan dari Dok_Asli_1	.docx	28 Kb
2	Dok_Uji_2	Berisi pembahasan dari Dok_Asli_2	.docx	745 Kb
3	Dok_Uji_3	Berisi tentang materi cybercrime	.txt	19 Kb
4	Dok_Uji_4	Berisi makalah tentang "Makalah Ekonomi Islam Di Indonesia"	.docx	24 Kb
5	Dok_Uji_5	Berisi paper yang berjudul "skiat-kiat pola makan yang sehat"	.pdf	421 Kb
6	Dok_Uji_6	Berisi tentang "Media Sosial dan Interaksi Remaja"	.docx	44 Kb
7	Dok_Uji_7	Berisi tentang pendahuluan dari Dok_Asli_7	.docx	8 Kb
8	Dok_Uji_8	Berasal dari Dok_Asli_8	.docx	70 Kb
9	Dok_Uji_9	Berasal dari Dok_Asli_9	.pdf	48 Kb
10	Dok_Uji_10	Beris tentang "pemanasan global dan lubang ozon"	.pdf	467Kb

Dari hasil perbandingan antara dokumen asli dan dokumen pembanding maka diperoleh hasil perbandingan waktu dokumen dengan kemungkinan kemiripan yang besar , ditunjukkan oleh Tabel 3.

Tabel 3 Hasil Analisa Pengujian Sistem

No	Dokumen Asli	Dokumen Pembanding	*Stemming*		*No Stemming*	
			Kemiripan (%)	Waktu (detik)	Kemiripan (%)	Waktu (detik)
1	Dok_Asli_1	Dok_Uji_1	78,318	6,375	78.288	6.851
2	Dok_Asli_2	Dok_Uji_2	82,809	18,672	83,028	20,32
3	Dok_Asli_3	Dok_Uji_3	84,426	6,468	84.453	5.917
4	Dok_Asli_4	Dok_Uji_4	83,313	5,016	83.424	6.761
5	Dok_Asli_5	Dok_Uji_5	77,145	4,328	76.970	3.544
6	Dok_Asli_6	Dok_Uji_6	100	5,437	100	8.906
7	Dok_Asli_7	Dok_Uji_7	100	13,468	100	12,324
8	Dok_Asli_8	Dok_Uji_8	91,614	21,063	91,276	10,224
9	Dok_Asli_9	Dok_Uji_9	99,118	4,469	99,032	6,235
10	Dok_Asli_10	Dok_Uji_10	78,060	12,407	79,142	11,43

Hasil perbandingan dokumen tersebut menunjukan bahwa dokumen dengan kemungkinan kemiripan yang besar menghasilkan nilai similaritas yang tinggi yaitu diatas 77% sampai dengan 100%.

Tabel 3 Hasil Analisa Pengujian Sistem Dokumen yang tidak mirip

No	Dokumen Asli	Dokumen Pembanding	*Stemming*		*No Stemming*	
			Kemiripan (%)	Waktu (detik)	Kemiripan (%)	Waktu (detik)
1	Dok_Asli_1	Dok_Uji_3	35,013	5,609	5,609	10,625
2	Dok_Asli_2	Dok_Uji_5	19,670	5,297	5,297	6,813
3	Dok_Asli_3	Dok_Uji_4	28,196	13,516	13,516	6,39
4	Dok_Asli_4	Dok_Uji_7	27,985	9,469	9,469	5,86
5	Dok_Asli_5	Dok_Uji_8	25,724	5,969	5,969	5,641
6	Dok_Asli_6	Dok_Uji_9	27,605	3,109	3,109	5,64
7	Dok_Asli_7	Dok_Uji_10	30,851	9,563	9,563	7,127
8	Dok_Asli_8	Dok_Uji_6	26,295	12,484	12,484	11,23
9	Dok_Asli_9	Dok_Uji_2	14,581	7,891	7,891	8,83
10	Dok_Asli_10	Dok_Uji_1	36,565	11,188	11,188	11,224

Hasil perbandingan dokumen pada Tabel 3 menunjukan bahwa dokumen dengan kemungkinan kemiripan yang kecil menghasilkan nilai similaritas yang rendah yaitu dibawah 40%.

Dari kedua Hasil pengujian dokumen tersebut menunjukan bahwa yang paling mempengaruhi lama tidaknya perhitungan nilai *similarity* bukan ditentukan oleh jenis dokumen maupun ukuran dokumen, namun oleh jumlah substring yang dikandung oleh dokumen tersebut. Dimana semakin besar jumlah *substring* yang dikandung dokumen, maka waktu yang diperlukan untuk menghitung nilai kesamaan akan semakin besar.Hasil perbandingan diatas juga

menunjukan bahwa dengan melakukan proses *no stemming* maka waktu yang dibutuhkan akan lebih lama.

4. KESIMPULAN

Berdasarkan uraian dan hasil analisa yang telah dilakukan selama pengembangan Aplikasi Deteksi Kemiripan isi teks dokumen dengan menggunakan metode *Levenshtein Distance* ini, dapat diambil kesimpulan yaitu

1. Penggunaan *preprocessing* terutama *filtering stopword* dan penggunaan stemming mempengaruhi nilai *similarity* dan waktu untuk proses.
2. Dengan menggunakan *preprocessing* membuat nilai *similarity* menjadi lebih baik tetapi juga memberikan efek terhadap lamanya proses pendeteksian.
3. Pada pengujian menggunakan data real yaitu data dokumen berplagiat yang diambil dari artikel/berita lewat internet, algoritma *Levenshtein Distance* menghasilkan nilai *similarity* yang tinggi yaitu diatas 85 % sampai 100 % untuk dokumen yang tingkat kemiripannya tinggi. Sedangkan untuk dokumen dengan tingkat kemiripan yang rendah atau tidak berplagiat maka menghasilkan nilai similarity dibawah 40%.

5. SARAN

Dalam pembuatan Aplikasi pendeteksi kemiripan isi teks dokumen dengan menggunakan metode *Levenshtein Distance* ini masih banyak terdapat kekurangan dan jauh dari kata sempurna. Untuk itu masih perlu dilakukan sebuah penyempurnaan. Berikut beberapa saran untuk pengembangan lebih lanjut dari aplikasi ini :

1. Perlu ditambahkan proses untuk mendeteksi padanan kata yaitu kata-kata yang berbeda namun memiliki makna yang sama.
2. Perlu ditambahkan sebuah proses yang bisa mendeteksi adanya kesalahan ejaan karena kesalahan ejaan ini sangat mempengaruhi proses *filtering* dan stemming sehingga dapat mengurangi nilai *similarity*
3. Dalam program yang selanjutnya dapat dikembangkan untuk proses yang juga bisa membaca dokumen berbentuk .xlx

DAFTAR PUSTAKA

[1] Irianto, W.A., 2014, Penentuan Tingkat Plagiarisme Dokumen Penelitian Menggunakan Centroid Linkage Hierarchical Method (CLHM), *Jurnal*. Program Teknologi Informasi dan Ilmu Komputer. Universitas Brawijaya. Malang.

[2] Andriyani, N.M., 2010 , Implementasi Algoritma Levenshtein Distance dan Metode Empiris untuk menampilkan saran perbaikan kesalahan pengetikan dokumen berbahasa Indonesia, *Skripsi*, Teknik Informatika, Universitas Udayana, Bali.

[3] Sastroasmoro, S., 200, Beberapa Catatan Tentang Plagiarisme, *Majalah Kedokteran Indonesia*, Volume : 57, Nomor : 8, Agustus 2007.

[4] Sora, 2014, Mengetahui Pengertian Dokumen dan Dokumentasi, *http://www.pengertianku.net /2014/09/mengetahui-pengertian-dokumen-dan-dokumentasi.html*, Diakses pada 2 Juni 2015.

[5] Harlian, M., Text Mining, *http://iwanarif.lecturer.pens.ac.id/kuliah /dm/6Text%20 Mining.pdf*, Diakses pada 4 Juni 2015.

[6] Yufis, 2016, Stemming Bahsa Indonesia dengan Algoritma Nazief dan Adriani, *http://yufis.staff.umm.ac.id/2012/06/07stemming-bahasa-indonesia-dengan-algoritma-nazief-dan-adriani/*, Diakses pada 10 Juli 2015.

[7] Nafik, M.Z., 2013, Sistem Otomatis Jawaban Esai Menggunakan Algoritma Levenshtein Distance, *Skripsi*, Ilmu Komputer, Universitas Brawijaya, Malang.