PERANCANGAN APLIKASI DETEKSI KEMIRIPAN ISI DOKUMEN TEKS DENGAN MENGGUNAKAN METODE LEVENSHTEIN DISTANCE

Richard Junedy.S (0911505)

Mahasiswa Jurusan Teknik Informatika STMIK Budi Darma Medan
Jl. Sisingamangaraja No. 338 Simpang Limun Medan
www.stmik-budidarma.ac.id // Email :richard.junedy@yahoo.co.id

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. Algoritma Levenshtein Distance adalah salah satu algoritma pencocokanstring yang dapat digunakan untuk mendeteksi plagiarisme. Skripsi ini bertujuan untuk mendeteksi kemiripan dokumen teks menggunakan algoritma levenshtein distance sehingga dapat digunakan untuk membantu menentukan plagiarisme.Untuk meningkatkan kerja algoritma levenshtein distance maka sebelumnya dilakukan preprocessing terhadap dokumen teks yang akan diproses.

Dari ujicoba yang dilakukan, algoritma levenshtein distance mampu mendeteksi kemiripan dokumen teks, ini dibuktikan dengan nilai similarity yang tinggi untuk dokumen yang plagiat. Preprocessing terbukti dapat meningkatkan kinerja algoritma levenshtein distance meskipun menambah waktu untuk pemrosesan.

Kata kunci : Levenshtein Distance, pencocokan string, *preprocessing*

1. Pendahuluan

1.2. Latar Belakang Masalah

Banyak instansi atau budang yang memanfaatkan kelebihan Komputer dalam pengerjaan secara manual. Dengan adanya Komputer lebih memudahkan manusia untuk mengerjakan pekerjaan dengan efektif dan cepat. Dalam Dunia akademik khususnya bidang informatika , banyak Mahasiswa atau siswa yang mengerjakan Tugas atau Laporannya dengan hanya *Copy-paste* dengan temannya, maka dari itu instansi pendidikan harus cermat dalam mengatasi hal tersebut. Kemampuan untuk mendeteksi kemiripan suatu dokumen baik itu dalam bentuk tugas atau laporan dengan mengandalkan kemajuan teknologi yang terkomputerisasi sangat dibutuhkan memudahkan proses pengerjaan yang awalnya dikerjakan dengan cara manual, sehingga dapat menghasilkan data yang lebih akurat.

Tugas, Laporan atau penelitian yang diberikan dan dikerjakan dengan cara mengulang hasil kerja orang lain dapa mengurangi kreatifitas mahasiswa dan dapat menimbulkan kecenderungan sikap malas dan tidak mau berfikir, oleh karena itu, tindakan tersebut harus ditekan sejak dini. Dengan adannya pendeteksian dokumen yang digunakan untuk memeriksa hasil dari dokumen tersebut, maka akan diketahui tingkat kemiripan dari dokumen yang telah dikerjakan oleh dua dokumen yang telah dibandingkan tersebut. Pendeteksian secara manual sebenarnya mempunyai tingkat akurasi yang tinggi, hal ini dikarenakan kemampuan manusia dalam memahami makna dan maksud sangat terbatas, serta gaya bahasa dari kata atau kalimat. Hanya saja, dapat membutuhkan waktu dan tenaga yang banyak jika mendeteksi dokumen yang sangat banyak sehingga menjadi tidak efektif didalam proses pengerjaanya.

Melihat masalah tersebut, penulis tertarik merancang solusi untuk mendeteksi kemiripan suatu dokumen teks agar dapat membantu dalam proses pengerjaan untuk mendeteksi perbandingan duah buah dokumen dalam jumlah yang sangaat banyak dengan efektif dan efesien. Salah satu metode yang tepat dalam melakukan deteksi kemiripan dokumen teks adalah dengan melakukkan perhitungan dengan metode *levenshtein distance* dan membandingkan dua dokumen teks yang akan dibandingkan. Kemiripan dokumen diperiksa dengan menggunakan algoritma lenvenshtein distance dengan melakukan pengecekan dalam proses pengerjaanya.

1.2. Perumusan Masalah

Dari latar belakang di atas, terdapat beberapa permasalahan yang akan diangkat dalam skripsi ini, antara lain:

1. Bagaimana mengetahui proses pendeteksi perbandingan kemiripan Dokumen Teks?
2. Bagaimana menerapkan algoritma *Levenshtein distance* dalam mendeteksi kemiripan isi dokumen Teks?
3. Bagaimana merancang Aplikasi Deteksi Dokumen Teks?

1.3. Batasan Masalah

Agar pembahasan tidak menyimpang dari apa yang telah ditetapkan maka permasalahan dibatasi sebagai berikut :

1. Dokumen yang digunakan dalam penelitian ini adalah dokumen teks yang tidak meliputi dokumen yang berisi gambar
2. Dokumen yang digunakan untuk perbandingan teks ini adalah dokumen yang berbahasa Indonesia
3. Aplikasi ini mengabaikan adanya sinonim (persamaan kata).
4. Aplikasi ini tidak memperhatikan kesalahan penulis pada dokumen.

1.4. Tujuan dan Manfaat Penelitian

Tujuan yang diharapkan dari perancangan enkripsi pesan gambar adalah :

1. Mengetahui proses pendeteksi perbandingan dokumen teks
2. Menerapkan Algoritma *Levenshtein distance* dalam mendeteksi kemiripan isi dokumen teks
3. Merancang Aplikasi Levenhstein Disteance dengan menggunakan Bahasa Pemrograman Java yaitu Netbeans.

Manfaat yang dari pembuatan perancangan tersebut adalah :

1. Dapat membantu instansi pendidikan dalam mendeteksi hasil Laporan dan tugas yang diberikan.
2. Mengetahui tingkat kemiripan (*similarity*) dokumen teks antara dokumen yang satu dengan yang lain
3. Dengan mengetahui persentase kemiripan teks, sehingga dapat digunakan sebagai bahan pertimbangan untuk mendeteksi adanya tindakan meniru.
4. Memperepat proses pengoreksian dengan jumlah yang sangat besar dengan cepat dan efektif.

1.5. Metode Penelitian

Langkah-langkah yang ditempuh penulis dalam penulisan skripsi ini adalah sebagai berikut :

1. Studi literatur
   Studi literatur dilakukan dengan mempelajari tentang teks mining metode pencocokan string melalui berbagai macam media, antara lain : internet, buku-buku, dan jurnal-jurnal yang berkaitan dengan text processing.
2. Analisis dan perancangan sistem
   Setelah semua data terkumpul, selanjutnya adalah melakukan analisis terhadap system yang akan dibangun nanti seperti apa, Kemudian membuat sistem algoritma. Pada tahap ini juga dilakukan uji coba secara manual terhadap algoritma yang digunakan apakah sudah sesuai dengan yang diharapkan , sehingga nanti tidak menimbulkan masalah ketika sudah diimplementasikan kedalam aplikasi komputer.
3. implementasi sistem
   Imlementasi dilakukan dengtan membuat aplikasi kemiripan dokumen teks berdasarkan perancangan yang telah dibuat sebelumnya kedalam program computer.
4. uji coba program dan evaluasi
   Pada tahap ini dilakukan uji coba terhadap aplikasi yang telah dibuat kemudian dievaluasi untuk melihat kekurangan dan kesalahan yan selanjutnya dilakukan perbaikan jika masi ada kesalahan.

2. Landasan Teori

2.1. Dokumen

Kata dokumen berasal dari bahasa latin yaitu *docere* , yang berarti mengajar, pengertian dari dokumen ini menurut *Louis guttschalk (1986;38)*seringkali digunakan para ahli dalam dua pengertian yaitu pertama berarti sumber tertulis lagi informasi sebagai kebalikan daripada kesaksian lisan, dan peninggalan-peninggalan terlukis. Pengertian kedua diperuntukkan bagi surat-surat resmi dan surat-surat Negara seperti surat perjanjian, undang-undang,hibah,konsensi dan lainnya. Sedangkan menurut *Robert C. bogdan* seperti yang dikutip sugiyono (2005;82) dokumen merupakan catatan peristiwa yang telah berlalu, bisa berbentuk tulisan, gambar, karya-karya monumental dari seseorang.

2.2. Algoritma 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. Perhitungan jarak antara dua string ini ditentukan dari jumlah minimum operasi perubahan untuk membuat string A menjadi string B (Implementasi Algoritma Levenshtein Distance dan Metode Empiris untuk menampilkan saran perbaikan kesalahan pengetikan dokumen berbahasa Indonesia,Andriyani NM,2010).

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

1. 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”.
2. 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 di akhir kata, namun bias ditambahkan diawal maupun disisipkan di tengah string .
3. 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" (Andriyani NM,2010;20)..

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.

3. Analisa dan Perancangan

pada Sistem pendeteksian kemiripan dokumen teks ini mempunyai cara kerja yaitu yang pertama dengan user memasukkan dokumen teks yang asli dengan dokumen teks yang ingin diuji selanjutnya user memasukkan nilai *threshold*. Nilai *threshold* ini digunakan sebagai nilai pembanding terhadap nilai hasil proses kesamaan kata dari kalimat yang dibandingkan dan sebagai nilai ambang batas dokumen bisa dikatakan mengandung plagiat. Kemudian sistem akan menganalisa persentase kemiripan (*similarity*) dan waktunya dengan menggunakan Algoritma *Levenshtein Distance*.

Dokumen yang dibandingkan dalam sistem ini adalah dokumen teks. Dokumen dapat berupa file txt maupun doc. Pada sistem ini nantinya akan dilakukan pengujian atau pengukuran *similarity* menggunakan algoritma *levenshtein distance preprocessing* dan menggunakan algoritma *levenshtein distance standart* tanpa proses *filtering, stemming, dan sorting*. Dengan membandingkan hasil dari pengujian tersebut maka akan dapat diketahui pengaruh penggunaan *preprocessing* pada pendeteksian isi dokumen teks.

Penulis akan melakukan analisis bagaimana penerapan algoritma *levenshtein distance* dalam pencocokan string. Misalnya terdapat dua string yaitu string asli dan string pembanding String asli (CS) : **aku** String pembanding (ST): **abu**

Proses algoritma levenstein distance adalah sebagai berikut:

1. Jika panjang CS adalah 0 maka jarak CS ke ST adalah panjang ST dalam contoh diatas berarti bernilai 3.
2. jika panjang ST adalah 0 maka jarak CS ke ST adalah panjang CS dalam contoh diatas berarti bernilai 3.
3. Membuat matrik dengan ukuran (CS+1) x (ST+1)

		a	b	u
a	1			
k	2			
u	3			

4. Melakukan pencocokan dengan melakukan pembandingan dari setiap karakter CS dengan karakter ST.

Iterasi (1,1)

		a	b	u
	0	1	2	3
a	1	0		
k	2			
u	3			

5. Jika karakter 1 CS dengan karakter 1 ST sama maka nilai pada cell (1,1) diisi dengan 0 yang merupakan nilai matrik (x-1 , y-1)

		a	b	u
	0	1	2	3
a	1	0		
k	2	1		
u	3			

Iterasi (1,2)

Jika karakter 1 CS dengan karakter 2 ST berbeda maka nilai cell didapatkan dari nilai terkecil dari:

a. Nilai cell (x-1, y-1) + 1, pada contoh diatas nilainya adalah 2 .
b. Nilai cell (x-1, y) + 1, pada contoh diatas nilainya adalah 3
c. Nilai cell (x, y-1) + 1, pada contoh diatas nilainya adalah 1

Maka jarak antara karakter 1 CS dengan karakter 2 ST bernilai 1.

Setelah semua proses iterasi selesai maka didapatkan jarak string asli (CS) dengan string pembanding (ST) pada cell (3,3) adalah sebesar 1 atau *Diff* =1.

		a	b	u
	0	1	2	3
a	1	0	1	2
k	2	1	1	2
u	3	2	2	1

Dalam penerapannya pada program ini pencocokan string dilakukan kalimat per kalimat. Kalimat dari proses sorting yang dalam bentuk array diubah dulu menjadi bentuk string. Kemudian string tersebut akan dihitung nilai similarity-nya menggunakan rumus:

$$Plagiarized\ Value = \left\{1 - \frac{Diff}{Max(CS,ST)}\right\} * 100$$

Keterangan Rumus

Plagiarize Value = Kesamaan Kemiripan

Diff = Nilai dari Jarak string Asli ke String Pembanding

CS = String Asli

ST = String Pembanding
Max(CS,ST) = Jarak String terbesar dari string Pembanding dan string asli

Dari matriks diatas, maka dapat dihitung nilai Kesamaannya.

Dik : Diff = 1
Max(CS,ST) = 3

Penyelesaian :

$$= \left(1 - \frac{1}{3}\right) * 100$$

$$= \left(\frac{3}{3} - \frac{1}{3}\right) * 100$$

$$= \frac{2}{3} * 100$$

= 66,66

Maka nilai kesamaan dari contoh diatas adalah 66,66

4. Algoritma

Langkah pertama yang dilakukan setelah dokumen diinputkan adalah membaca file inputan, jika dokumen yang diinputkan adalah berupa dokumen yang berekstensi .docx maka program akan melakukan proses pengkonversian file docx tersebut menjadi bentuk file txt terlebih dahulu.

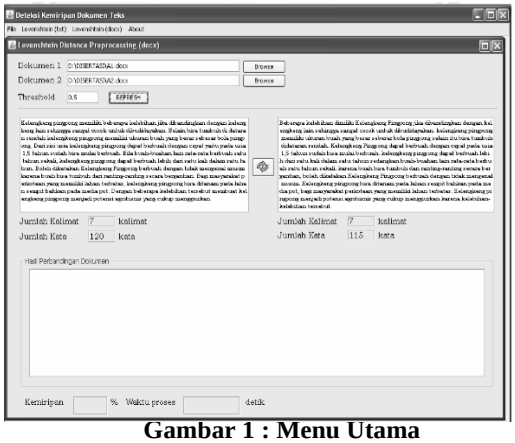
Algoritma untuk konversi docx menjadi txt adalah sebagai berikut :

Input : x (dokumen text)
C (keakurasian)
D (waktu)
Output : m
Proses :

if x (c^d) +c mod m
if x! =0 then
x=m
x= d + x! (c ^ d)
end if

5. Implementasi

Berdasarkan perancangan *user interface* pada bab 3, maka dihasilkan tampilan *user interface* seperti pada **Gambar 1**

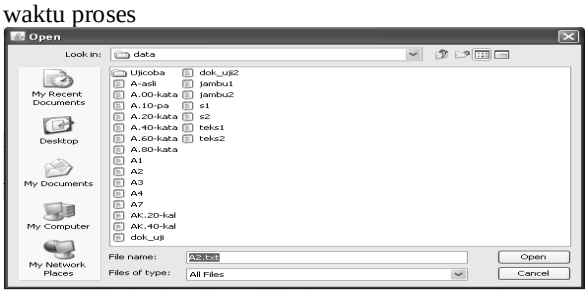


Gambar 1 : Menu Utama

b. Tampilan Upload File

Pada halaman utama ini *user* diharuskan memasukkan beberapa *field* yaitu *field* untuk *upload* dokumen asli dan dokumen pembanding seperti pada **2** Setelah itu *user* juga harus memasukkan nilai *threshold* sebagai nilai ambang batas perbandingan kalimat diproses lebih lanjut dan juga sebagai nilai batas dokumen berplagiat

Setelah proses pada gambar 1diatas maka akan didapatkan output berupa informasi dokumen (jumlah kata dan kalimat), perbandingan kalimat, persentase kemiripan dokumen (*similarity*) dan waktu proses



Gambar 2 :Upload file

c. Tampilan Proses Program

Hasil dari proses dapat dilihat pada **Gambar 4.3**

Gambar 3 : Hasil proses program

5. Kesimpulan dan Saran

5.1. Kesimpulan

Berdasarkan ujicoba yang dilakukan selama pembuatan skripsi ini dapat dibuat kesimpulan bahwa algoritma Levenshtein Distance dengan bantuan preprocessing mempunyai akurasi yang lebih baik dibanding dengan Algoritma Levenshtein Distance standart dalam mendeteksi kemiripan dua dokumen. Hal ini dibuktikan dengan nilai persentase kemiripan yang lebih tinggi untuk ujicoba menggunakan Data Uji 1 dan Data Uji 2.

1. Algoritma Levenshtein Distance standart memiliki waktu proses yang lebih cepat dibandingkan Algoritma Levenshtein Distance preprocessing tetapi pada penggujian Data Uji 2 nilai similarity yang dihasilkan masih rendah.

2. Penggunaan preprocessing terutama filtering stopword dan penggunaan stemming mempengaruhi nilai similarity dan waktu untuk proses.
3. Dengan menggunakan preprocessing membuat nilai similarity menjadi lebih baik tetapi juga memberikan efek terhadap lamanya proses pendeteksian.
4. Algoritma Levenshtein Distance akan bekerja lebih baik jika kedua dokumen yang dibandingkan mempunyai urutan kata yang sama, oleh karena itu pernggunaan sorting sangat membantu dalam mendeteksi kemiripan teks.
5. 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 %.

5.2. Saran

Untuk penelitian lebih lanjut tentang penelitian ini perlu ditambahkan beberapa pengembangan diantaranya:

1. Perlu ditambahkan proses untuk mendeteksi padanan kata yaitu kata-kata yang berbeda namun memiliki makna yang sama.
2. Program ini dapat dikembangkan menjadi program yang bisa mendeteksi kemiripan dokumen teks dalam jumlah banyak dokumen sekaligus tidak terbatas hanya membandingkan dua dokumen saja sehingga bisa menjadi lebih efektif.
3. 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.
4. Dalam skripsi ini program masih bisa mendeteksi dokumen berbentuk txt dan docx, untuk penelitian lebih lanjut bisa ditambahkan proses yang juga bisa membaca dokumen yang berbentuk doc dan pdf.

Daftar Pustaka

[1] Adi Nugroho, “Analisis dan Perancangan Sistem Informasi”, 2005.
[2] Azhar Susanto, *“Sistem Informasi Manajemen Konsep dan Pengembangannya”, 2004.*
[3] H.M Jogianto, “Analisis Sistem Informasi”, 2004
[4] M.Agus J.Alam, “Database Visual Basic dalam Sql Server”,2005
[5] Andriyani NM, “Implementasi Algoritma Levenshtein Distance dan Metode Empiris untuk menampilkan saran perbaikan kesalahan pengetikan dokumen berbahasa Indonesia “, 2010
[6] Edi Casnadi, **“Pemrograman Java Netbeans”,** 2012