# SISTEM DETEKSI PLAGIARISME DOKUMEN TEKS MENGGUNAKAN ALGORITMA JARO-WINKLER DISTANCE

**SKRIPSI**

Diajukan Untuk Memenuhi Sebagian Syarat Guna Memperoleh Gelar
Sarjana Komputer (S.Kom) Pada Program Studi Teknik Informatika
Fakultas Teknik Universitas Nusantara PGRI Kediri

OLEH :

**TEGUH WILLYANTO**

NPM : 11.1.03.02.0362

**FAKULTAS TEKNIK (FT)**

**UNIVERSITAS NUSANTARA PERSATUAN GURU REPUBLIK INDONESIA**

**UNP KEDIRI**

**2015**

Skripsi oleh :

**TEGUH WILLYANTO**

**NPM : 11.1.03.02.0362**

Judul :

**SISTEM DETEKSI PLAGIARISME DOKUMEN TEKS MENGGUNAKAN ALGORITMA JARO-WINKLER DISTANCE**

Telah disetujui untuk diajukan Kepada

Panitia Ujian/Sidang Skripsi Program Studi Teknik Informatika

Fakultas Teknik UNP Kediri

Tanggal : 28 Agustus 2015

| Pembimbing I | Pembimbing II |
|---|---|
| **Drs. Sigit Widiatmoko, M.Pd.** | **Rini Indriati, S.Kom., M.Kom** |
| **NIDN. 0717076301** | **NIY. 1081001070** |

ii

Skripsi oleh :

**TEGUH WILLYANTO**

NPM : 11.1.03.02.0362

Judul :

**SISTEM DETEKSI PLAGIARISME DOKUMEN TEKS MENGGUNAKAN ALGORITMA JARO-WINKLER DISTANCE**

Telah dipertahankan di depan Panitia Ujian/Sidang Skripsi

Program Studi Teknik Informatika Fakultas Teknik UNP Kediri

Pada tanggal : 28 Agustus 2015

**Dan Dinyatakan telah Memenuhi Persyaratan**

Panitia Penguji :

1. Ketua : Dr. Suryo Widodo, M.Pd
2. Penguji I : Dr. M. Anas, S.E.,M.M.,M.Si
3. Penguji II : Patmi Kasih, M.Kom.

Mengetahui,

Dekan Fakultas Teknik

UNIVERSITAS NUSANTARA FAKULTAS TEKNIK PGRI

**Dr. Suryo Widodo, M.Pd**
**NIP. 19640202 199103 1 002**

iii

# SISTEM DETEKSI PLAGIARISME DOKUMEN TEKS MENGGUNAKAN ALGORITMA JARO-WINKLER DISTANCE

Teguh Willyanto
11.1.03.02.0362
Teknik - Teknik Informatika
teguhwillyanto87@gmail.com
Drs. Sigit Widiatmoko, M.Pd. dan Rini Indriati, S.Kom., M.Kom
UNIVERSITAS NUSANTARA PGRI KEDIRI

## ABSTRAK

Plagiarisme atau biasa disebut plagiat adalah pengambilan karangan (pendapat dan sebagainya) orang lain dan menjadikannya seolah-olah karangan (pendapat) sendiri. Pada bidang akademis sering ditemukan aktivitas plagiarisme, khususnya dalam pengerjaan sebuah tugas dalam bentuk dokumen teks. Ada beberapa pendekatan yang bisa diambil yaitu algoritma Jaro-Winkler distance,hal ini dikarenakan selain dapat menghitung tingkat kemiripan antara dua string berdasarkan nilai dari penyusun string tersebut seperti algoritma cosine similarity, juga dipengaruhi oleh urutan dari penyusun string tersebut.

Tujuan penelitian ini adalah sebagai bahan pertimbangan dalam menentukan plagiarisme dalam ruang lingkup sekolah dan menentukan nilai kemiripan (similarity) antara dokumen yang diuji dengan dokumen asli oleh sistem.Data yang digunakan dalam penelitian ini diambil dengan melakukan wawancara secara langsung dengan pihak guru pada SMKN 1 Bagor serta lembar soal yang diujikan terhadap siswa-siswi.

Hasil analisis data diperoleh bahwa untuk soal uraian bertipe jawaban singkat nilai yang diberikan aplikasi cukup memuaskan dengan rata – rata selisih antara nilai hasil penilaian manual dan aplikasi sebesar 0,92 point. Sedangkan untuk soal uraian objektif selisih yang dihasilkan dari penilaian manual dan aplikasi adalah 1,44 point. Sehingga dapat disimpulkan bahwa algoritma ini lebih cocok digunakan pada soal uraian bertipe jawaban singkat.

**Kata Kunci :** Plagiarisme, Algoritma jaro-winkler distance,hitung kemiripan

## I. LATAR BELAKANG

Dokumen digital memang dapat diplagiat dengan sangat mudah. Praktik plagiat atau penjiplakan ini sudah sering terjadi khususnya pada kalangan akademisi baik lingkungan sekolah maupun perguruan tinggi. Sering kita temui pekerjaan teman yang dihargai dengan nilai "0" karena diketahui telah menyalin hasil pekerjaan temannya yang lain. Praktik menyalin beberapa bagian atau keseluruhan tulisan tanpa mencantumkan sumber menjadi hal biasa yang sering ditemukan pada tugas harian, makalah, bahkan skripsi.

Tindakan plagiat yang dilakukan siswa atau mahasiswa ini sangat tidak mencerminkan sikap kreatif dan terpelajar sebagai kaum intelektual. Bentuk plagiat ini dapat bermacam-macam seperti misalnya yang paling sering dilakukan yaitu copy-paste-edit suatu dokumen baik

merupakan pekerjaan teman maupun dokumen yang berasal dari website, sehingga menjadi suatu hasil yang berbeda dari sebelumnya. Selain itu plagiat dapat pula ditemukan dalam bentuk quote atau kutipan pada sebuah dokumen.

Dalam proses penilaian akan digunakan algoritma Jaro-Winkler distance, hal ini dikarenakan selain dapat menghitung tingkat kemiripan antara dua string berdasarkan nilai dari penyusun string tersebut seperti algoritma cosine similarity,
juga dipengaruhi oleh urutan dari penyusun string tersebut.

## II. METODE

### 1. Plagiarisme

"Plagiarisme" adalah "penjiplakan atau pengambilan karangan, pendapat, dan sebagainya dari orang lain dan menjadikannya seolah karangan dan pendapat sendiri". (KBBI, 1997: 775)
Plagiat adalah perbuatan sengaja atau tidak sengaja dalam memperoleh kredit atau nilai untuk suatu karya ilmiah, dengan mengutip sebagian atau seluruh karya dan atau karya ilmiah pihak lain yang diakui sebagai karya ilmiahnya, tanpa menyertakan sumber secara tepat dan memadai. Beberapa tipe plagiarism yaitu :

1. Plagiarisme Kata demi kata *(Word for word plagiarism)*. Penulis menggunakan kata-kata penulis lain (persis).

2. Plagiarisme atas sumber *(Plagiarism of sources)*. Penulis menggunakan gagasan orang lain tanpa memberikan pengakuan yang cukup (tanpa menyebutkan sumber secara jelas).

3. Plagiarisme Kepengarangan *(Plagiarism of authorship)*. Penulis mengakui sebagai pengarang karya tulis orang lain.

4. Self Plagiarism. "Penulis mempublikasikan satu artikel pada lebih dari satu redaksi publikasi". Dan mendaur ulang karya tulis/karya ilmiah. (Soelistyo, 2011).

**2. Teks Mining**

Teks mining dapat didefinisikan sebagai "*penemuan informasi baru dan tidak diketahui sebelumnya oleh komputer, dengan secara otomatis mengekstrak informasi dari sumber – sumber teks tak terstruktur yang berbeda"* (Tan,1999). Teks mining dapat dipandang sebagai perluasan dari data mining atau *knowledge-discovery in database (KDD)* yang mencoba menemukan pola menarik dari basis data berskala besar. Namun hal mendasar yang membedakan keduanya adalah sumber data yang diproses, dimana dalam data mining menggunakan data dari database yang relatif telah memiliki format yang terstruktur, sedangkan teks mining menggunakan data dalam format teks sebagai sumbernya yang tidak terstruktur karena berdasarkan tata bahasa manusia (*natural language*). Secara umum, basis data didesain untuk program dengan tujuan melakukan pemrosesan secara otomatis, sedangkan teks ditulis untuk dibaca langsung oleh manusia (Hearst, 2003).

**3. Ruang Lingkup Teks Mining**

"*Ada beberapa proses yang dilakukan dalam teks mining untuk mendapatkan informasi yang tersembunyi, diantaranya adalah pemrosesan awal terhadap teks (text preprocessing), transformasi teks (text transformation), pemilihan fitur (feature selection), dan penemuan pola (pattern discovery)."* (Even-Zohar, 2002).

a. *Text Preprocessing*

Tujuan dari pemrosesan awal adalah untuk mempersiapkan teks menjadi data yang akan mengalami pengolahan lebih lanjut. Dalam proses akan dilakukan *parse tree* untuk tiap – tiap kalimat, dan pembersihan teks.

b. *Text Transformation*

Pada proses ini bertujuan untuk mendapatkan representasi dokumen yang diharapkan. Transformasi teks sekaligus melakukan pengubahan kata – kata ke bentuk dasarnya dan pengurangan dimensi kata di dalam dokumen tersebut.

c. *Feature Selection*

Pemilihan fitur ( kata ) merupakan tahap lanjut dari proses pengurangan dimensi pada proses transformasi teks. Meskipun telah dilakukan penghapusan kata–kata yang dianggap tidak deskriptif (*stopwords*), namun tidak semua kata dalam dokumen memiliki arti yang penting. Oleh karena itu untuk mengurangi dimensi dari dokumen

dilakukan pemilihan terhadap kata–kata yang dianggap penting dan memiliki hubungan yang erat dengan isi dokumen. Ide dasar dari proses ini adalah menghapus kata–kata yang terlalu sering atau jarang muncul dalam dikumen tersebut.

d. *Pattern Discovery*

Proses ini merupakan tujuan dari teks mining yaitu menemukan informasi tersembunyi dari dokumen berupa pola. Masukan awal dari proses teks mining adalah suatu data teks dan menghasilkan keluaran berupa pola sebagai hasil dari interpretasi atau evaluasi. Apabila hasil keluaran dari penemuan pola belum sesuai untuk aplikasi, dilanjutkan evaluasi dengan melakukan iterasi ke satu atau beberapa tahap sebelumnya. Sebaliknya, hasil interpretasi merupakan tahap akhir proses teks mining dan akan disajikan ke pengguna dalam bentuk visual. (Even-Zohar, 2002).

**4. Ekstraksi Dokumen**

Pada umumnya data berupa teks memiliki dimensi yang tinggi, terdapat noise pada data, dan struktur kalimat yang tidak baik. Untuk itu dilakukan pemecahan dokumen menjadi kalimat, kata sehingga dapat dilakukan pemrosesan data dengan lebih baik. Sebenarnya ini merupakan bagian dari *text preprocessing.* Secara umum dalam tahap ini dilakukan proses – proses berikut ini, yaitu *case folding, tokeniszing, filtering, stemming, tagging* dan *analizing* (Triawati, 2009).

**a. Case Folding dan Tokenizing**

Case folding merupakan proses mengubah dokumen / kalimat menjadi huruf kecil, namun yang dirubah hanya huruf saja bukan angka maupun simbol. Sedangkan tokenizing adalah proses pemotongan kalimat berdasarkan pada kata – kata yang menyusunnya.

**b. Filtering**

Pada tahap ini dilakukan pemisahan antara kata – kata yang dianggap penting dalam sebuah kalimat dengan kata – kata yang dianggap tidak memiliki relevansi yang erat dengan isi dari dokumen akan dihapus. Dalam melakukan filtering dapat dilakukan dengan metode *wordlist* yaitu menyimpan kata – kata yang penting, atau dengan metode *stoplist* yaitu membuang kata – kata yang tidak deskriptif seperti kata dan, yang, supaya dan lain sebagainya.

**c. Stemming**

Proses stemming akan menghasilkan kata dasar dari setiap kata hasil tokenizing dan filtering. Dalam skripsi ini akan digunakan algoritma stemming-arifin untuk mendapatkan kata

dasar dalam bahasa Indonesia. Dalam proses ini dilakukan pemotongan terhadap imbuhan dan akhiran sehingga dihasilkan kata dasar dari kata tersebut.

**d. Algoritma Stemming Arifin**

Algoritma ini didahului dengan pembacaan tiap kata dari file sampel. Sehingga input dari algoritma ini adalah sebuah kata.

**5. Algoritma Jaro-Winkler Distance**

*Jaro-Winkler distance* adalah sebuah algoritma untuk mengukur kesamaan antara dua string, biasanya algoritma ini digunakan di dalam pendeteksian duplikat. Semakin tinggi *Jaro-Winkler distance* untuk dua string, semakin mirip dengan string tersebut. *Jaro-Winkler distance* terbaik dan cocok untuk digunakan dalam perbandingan string singkat seperti nama orang. Skor normalnya seperti 0 menandakan tidak ada kesamaan, dan 1 adalah sama persis.

Dasar dari algoritma ini memiliki tiga bagian:

1. Menghitung panjang string
2. Menemukan jumlah karakter yang sama di dalam dua string
3. Menemukan jumlah transposisi

Pada algoritma Jaro digunakan rumus untuk menghitung jarak (*dj*) antara dua string yaitu s1 dan s2

$$d_j = \frac{1}{3} x \left( \frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right)$$

.....(1)

dimana :

m = jumlah karakter yang sama persis

|s1 | = panjang string 1

|s2 | = panjang String 2

t = jumlah transposisi

Jarak teoritis dua buah karakter yang disamakan dapat dibenarkan jika tidak melebihi:

$$\left( \frac{max\left(|s_1|,|s_2|\right)}{s} \right) < -1$$

.....(2)

Akan tetapi bila mengacu kepada nilai yang akan dihasilkan oleh algoritma *Jaro-Winkler* maka nilai jarak maksimalnya adalah 1 yang menandakan kesamaan string yang dibandingkan mencapai seratus persen atau sama persis. Biasanya s1 digunakan sebagai acuan untuk urutan di dalam mencari transposisi. Yang dimaksud transposisi di sini adalah karakter yang sama dari string yang dibandingkan akan tetapi tertukar urutannya.

Sebagai contoh, dalam membandingkan kata CRATE dengan TRACE, bila dilihat seksama maka dapat dikatakan semua karakter yang ada di s1 ada dan sama dengan karakter yang ada di $s_2$ , tetapi dengan urutan yang berbeda. Dengan mengganti C dan T, dapat dilihat

perubahan kata CRATE menjadi TRACE. Pertukaran dua elemen string inilah adalah contoh nyata dari transposisi yang dijelaskan. Dalam pencocokkan DwAyNE dan DuANE memiliki urutan yang sama D-A-N-E, jadi tidak ada transposisi.

*Jaro-Winkler distance* menggunakan prefix scale (*p*) yang memberikan tingkat penilaian yang lebih, dan prefix length (*l*) yang menyatakan panjang awalan yaitu panjang karakter yang sama dari string yang dibandingkan sampai ditemukannya ketidaksamaan. Bila string $s_1$ dan $s_2$ yang diperbandingkan, maka *Jaro-Winkler distancenya* ($d_w$ ) adalah:

$$d_w = d_j + (lp(1 - d_j))$$

..... (3)

dimana :

$d_j$ = Jaro distance untuk strings $s_1$ dan $s_2$

l = panjang prefiks umum di awal string nilai maksimalnya 4 karakter(panjang karakter yg sama sebelum ditemukan ketidak samaan max 4). p = konstanta scaling factor. Nilai standar untuk konstanta ini menurut Winkler adalah p = 0,1.

## III. HASIL DAN KESIMPULAN

Berdasarkan latar belakang masalah diatas, maka dapat disimpulkan bahwa aplikasi yang dibangun harus memenuhi hal – hal berikut ini :

a) Membedakan hak akses antara guru (pihak yang mengadakan tes), admin dan siswa.
b) Memberikan prosentase kemiripan antar dokumen.
c) Menampilkan nilai siswa.

Pada tahap analisa sistem ada beberapa syarat yang harus disiapkan sebelumnya. Syarat-syarat tersebut meliputi perangkat keras (hardware) dan perangkat lunak (software).

a. Lingkungan Perangkat Keras.

Dalam perancangan dan pengembangan sistem deteksi anti-plagiarisme ini menggunakan laptop dengan spesifikasi:

1. Prosesor Intel Core2Duo T6600 @2.20GHz
2. VGA 512 ATI Radeon HD4300
3. Memory 2 GB
4. Harddisk 320 GB
5. Monitor 17”
6. Motherboard dan keyboard

b. Lingkungan Perangkat Lunak.

Perangkat lunak yang digunakan dalam pengembangan sistem deteksi anti-plagiarisme ini adalah:

1. Sistem Operasi Windows XP SP2
2. XAMPP 1.8.1
3. PHP Version 5.2.3
4. Apache 2.4.3

5. MySQL 5.4.7
6. Adobe Dreamweaver CS3
7. Google Chrome

Untuk menghitung tingkat kemiripan antara dua string selain menggunakan jaro - winckler distance dapat juga cosine similarity yang biasa digunakan untuk menghitung besarnya sudut dua vektor. Namun bila menggunakan cosine similarity, hanya menghitung kemiripan isi dokumen, tanpa memperhatikan urutan kata / kalimat yang menyusunnya. Hal ini kurang cocok bila digunakan untuk memeriksa jawaban dan kunci jawaban dimana susunan kata / kalimat diperhitungkan, misalnya untuk soal - soal yang meminta peserta didik untuk menjelaskan langkah - langkah sebuah proses dengan berurutan. Untuk itulah dalam penelitian ini digunakan algoritma jaro - winckler distance yang dapat menghitung tingkat kemiripan berdasarkan isi dan posisi kata penyusunnya. Berikut ini langkah - langkah yang akan digunakan untuk menghitung kemiripan dua dokumen setelah melalui preprocessing :

1. Masukkan array jawaban dan array kunci jawaban
2. Cari kata yang sama dalam kedua array, masukkan ke kata_sama
3. Cari kata yang tidak sama dari kedua array dan masukkan ke kunci_tidak_sama dan jawaban_tidak_sama
4. Lakukan perulangan langkah 5 sampai 7 sebanyak jumlah elemen jawaban_tidak_sama
5. Cari sinonim dari jawaban_tidak_sama[index], masukkan ke variabel sinonim
6. Jika tidak ditemukan sinonim dalam kunci_tidak_sama, kembali ke langkah 5
7. Bila ditemukan ganti nilai jawaban_tidak_sama[index] dengan sinonim dan tambahkan sinonim kedalam kata_sama, kembali ke langkah 5
8. Buat array kunci_sama dan jawaban_sama dengan index adalah posisi dimana kata_sama[index] ditemukan dalam kunci dan jawaban.
9. Urutkan berdasarkan index dari array kunci_sama dan jawaban_sama.
10. Inisialisasi bariabel i=t=l= 0, i untuk proses perulangan, t untuk mengetahui jumlah transpose dan l untuk mengetahui posisi sebelum ditemukan kata yang tidak sama.
11. Selama kata dalam kata_sama_kunci ( for a in

kata_sama_kunci ) belum diperiksa seluruhnya, lakukan langkah 11 samapai dengan 14

12. Periksa apakah kata_sama_kunci[a] == kata_sama_jawaban[i]
13. Jika tidak, increment bariabel t ( t++ ) dan kembali ke langkah 11
14. Jika ya, jika l == i ( untuk memeriksa posisi awal kata yang berbeda ) increment variabel l dan kembali ke langkah 11, jika l != i kembali ke langkah 11
15. Hitung tingkat kemiripan dengan rumus

$$d_j = \frac{1}{3} x \left( \frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right)$$

dan

$$d_w = d_j + (lp(1 - d_j))$$

Beberapa simpulan yang dapat ditarik berdasarkan penelitian yang telah dilakukan adalah sebagai berikut:

1. Sistem ini dapat menerapkan algoritma Jaro-Winkler distance dalam sebuah sistem pendeteksian terhadap teks berbahasa Indonesia.
2. Semakin tinggi Jaro-Winkler distance untuk dua string maka semakin mirip dengan string tersebut. Nilai normalnya ialah 0 menandakan tidak ada kesamaan dan 1 yang menandakan adanya Kesamaan.
3. Sistem ini berhasil membandingkan kemiripan dengan algoritma Jaro-Winkler Distance pada tugas siswa dengan inputan berupa teks.

## IV. DAFTAR PUSTAKA

Hatta, Ahmad., Ramadijanti, Nana., Helen, Afrida. 2010. Rancang Bangun Sistem Pengelolaan Dokumen-dokumen Penting Menggunakan Text Mining.(Online) Tersedia: http://digilib.its.ac.id/public/ITS-Undergraduate-14856-paperpdf.pdf, diunduh 6 Oktober 2014

Hamza, Syahriar., Sarosa,M. Budi, Purnomo. 2012. Sistem Koreksi Soal Essay Otomatis Dengan Menggunakan Metode Rabin Karp.(Online) Tersedia:http://download.portalgaruda.org/article.php?article=270353&val=7115title=Sistem%20Koreksi%20Soal%20Essay%20Otomatis%20%20Dengan%20%20Menggunakan%20Metode%20Rabin%20Karp, diunduh 23 Desember 2014

Ryan, Mudafiq. Budi, Eko., Indah, Gita. 2012. Aplikasi Pendeteksi Duplikasi Dokumen Teks Bahasa Indonesia Menggunakan Algoritma Winnowing Dengan Metode K-Gram Dan Synonym Recognition. (Online) Tersedia: *www.mudafiqriyan.net/2012/03/aplikasi-pendeteksi-duplikasi-dokumen-teks-bahasa-indonesia-menggunakan-algoritma-winnowing-dengan-metode-k-gram-dan-synonym-recognition/*, diunduh 6 Oktober 2014

Novian, Dian., Abdillah, Tajuddin., Tuloli, Moh.Syafri. 2012. Aplikasi Pendeteksian Plagiat Pada Karya Ilmiah menggunakan Algoritma Rabin Karp.(Online) Tersedia: *http://repository.ung.ac.id/get/simlit/1/526/2/Aplikasi-Pendeteksian-Plagiat.pdf*, diunduh 11 November 2014