JURNAL PINTER
VOL. 1 NO. 1 JUNI 2017
e-ISSN: 2597- 4475
http://doi.org/10.21009/pinter.1.1.4

Aplikasi Pendeteksi Dugaan Awal Plagiarisme Pada Tugas Siswa Dan Mahasiswa Berdasarkan Kemiripan Isi Teks Menggunakan Algoritma Levenshtein Distance

Hamidillah Ajie ,Agung Surya Bangsa

Abstrak

Tujuan dari penelitian ini adalah menerapkan algoritma Levenshtein distance pada aplikasi komputer yang dapat menghitung persentase kemiripan isi teks dokumen tugas siswa/mahasiswa sehingga dapat membantu guru/dosen dalam melakukan pendeteksian dugaan awal plagiarisme. Penelitian dilakukan di laboraturium Multimedia Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Jakarta dari bulan Agustus 2013 hingga April 2014. Metode yang digunakan adalah salah satu metode pengembangan perangkat lunak, yaitu metode Waterfall. Hasil uji coba menunjukkan bahwa setelah melalui tahapan-tahapan preprocessing, algoritma Levenshtein distance dapat diimplementasi pada aplikasi dan cocok untuk mendeteksi dugaan awal plagiarisme kata demi kata.

Kata kunci: *Levenshtein distance, plagiarisme dokumen, kemiripan isi teks*

1. Pendahuluan

Menurut Sastroasmoro (2006: 239), plagiarisme adalah penggunaan ide, pikiran, data, kalimat orang lain seolah-olah sebagai miliknya tanpa menyebutkan sumbernya. Plagiarisme merupakan salah satu pelanggaran serius yang bersifat universal terhadap etika akademis (scientific misconduct). Tindakan plagiarisme ini merupakan sebuah tindakan tercela, karena merupakan ketidakjujuran alias kebohongan.

Perkembangan pesat pada dunia teknologi dan informasi memudahkan kita untuk memperoleh berbagai macam informasi. Internet sangat berguna sebagai sumber informasi dalam mencari berita, serta mencari referensi untuk membantu pembuatan tugas dan karya tulis. Namun, hal tersebut terkadang dimanfaatkan oleh sebagian orang untuk kepentingannya sendiri, seperti tindakan plagiarisme dalam pengerjaan tugas dan karya tulis.

Secara garis besar, plagiarisme dapat dikelompokkan menjadi 2, yaitu plagiarisme ide dan plagiarisme kata demi kata (word for word plagiarizing). Derajat plagiarisme yang paling berat adalah word for word plagiarizing yakni pencurian kata demi kata, yang dapat mencakup kalimat, paragraf, atau seluruh tulisan (Sastroasmoro, 2006: 243). Sedangkan, dalam karya tulis ilmiah, plagiarisme ide sering dihubungkan dengan laporan hasil penelitian replikatif, yaitu penelitian yang secara garis besar mengulang penelitian orang lain, dengan maksud untuk menambah data, menguji hasil hipotesis (Sastroasmoro, 2006: 240).

Besarnya persentase kata atau kalimat yang dicuri dapat diperoleh melalui penghitungan menggunakan algoritma tertentu.

Plagiarisme umum dilakukan oleh kalangan pelajar, baik siswa/siswi usia sekolah hingga tingkat mahasiswa. Menurut hasil survei Pew Research Center, salah satu lembaga survey Amerika Serikat yang juga bekerja sama dengan The Chronicle of Higher Education, dari survei terhadap 1055 mahasiswa (baik universitas negeri maupun universitas swasta) didapat data sebanyak 55 persen mahasiswa melakukan plagiat skripsi sepanjang sepuluh tahun terakhir. 89 persen dari pelaku plagiat tersebut mengatakan bahwa komputer dan internet sangat berperan dalam tindak plagiarisme yang mereka lakukan.

Di Indonesia pun banyak pelajar maupun mahasiswa yang menjadi pelaku plagiat, meskipun belum ada data survei di Indonesia yang mendukung. Kebanyakan dari mereka sering kali menyalin kalimat secara langsung dari internet dalam rangka pengerjaan tugas maupun karya tulis karena kebanyakan dari mereka malas untuk membaca buku dan menganggap jika mencari informasi dari internet itu lebih mudah, cepat dan instan tanpa mereka perlu membuka-buka dan membacanya satu per satu untuk mendapatkan suatu informasi.

Selain sumber dari internet, pelajar pun sering melakukan plagiat terhadap tugas maupun karya tulis pelajar lain. Hal ini sangatlah memprihatinkan. Perlu dilakukan aksi nyata oleh para pengajar dan pendidik untuk mencegah agar aksi plagiarisme tersebut tidak semakin membudaya dalam dunia pendidikan. Misalnya, memberikan hukuman kepada pelajar yang

e-ISSN: 2597- 4475
http://doi.org/10.21009/pinter.1.1.4

ketahuan melakukan tindak plagiarisme pada pengerjaan tugas maupun karya tulis mereka. Dengan demikian, untuk setiap subjek tugas yang diberikan, para pengajar dan pendidik harus membandingkan

tugas atau karya tulis seorang pelajar dengan pelajar lainnya satu per satu untuk mengidentifikasi tindak plagiarisme. Namun, para pengajar dan pendidik seringkali kewalahan karena banyaknya jumlah tugas atau karya tulis yang harus diperiksa. Hal ini membuat proses identifikasi plagiarisme menjadi tidak efektif. Diperlukan sebuah alat untuk membantu mendeteksi dugaan awal plagiarisme sehingga proses identifikasi dapat dilakukan lebih cepat dan efektif karena dengan begitu para pengajar atau pendidik tidak perlu memeriksa semua tugas atau karya tulis satu per satu. Para pengajar atau pendidik cukup memeriksa tugas atau karya tulis yang memiliki tingkat persentase kemiripan yang tinggi atau yang melebihi batas toleransi yang ditentukan oleh para pengajar atau pendidik itu sendiri.

Oleh karena itu, hal yang menjadi perhatian dalam skripsi ini adalah implementasi algoritma Levenshtein distance pada aplikasi komputer untuk menghasilkan persentase kemiripan isi teks pada dokumen tugas peserta didik sehingga dapat membantu pendidik dalam melakukan pendugaan awal terjadinya tindak plagiarisme diantara peserta didiknya.

Algoritma Levenshtein distance sendiri merupakan salah satu algoritma untuk text similarity, yaitu algoritma untuk menghitung kemiripan antar dua string input yang dibandingkan. Algoritma ini memiliki kompleksitas yang kecil, yaitu O(m*n), dimana m dan n adalah panjang dari string input 1 dan string input 2 sehingga lebih efektif dalam melakukan penghitungan dibanding algoritma text similarity lain, terutama untuk kasus string yang panjang.

Berdasarkan latar belakang permasalahan yang ada, penulis bermaksud untuk mengimplementasi algoritma Levenshtein distance pada sebuah aplikasi yang dapat menghitung persentase kemiripan isi teks pada tugas siswa dan mahasiswa.

Adapun perumusan masalahnya adalah sebagai berikut: ”Bagaimana mengimplementasi algoritma Levenshtein distance pada aplikasi komputer untuk menghitung persentase kemiripan antara isi teks sebuah dokumen tugas siswa/mahasiswa lain sehingga dapat digunakan oleh guru/dosen dalam membantu menduga terjadinya tindak plagiarisme?

Dalam penelitian ini, masalah akan dibatasi pada:

a. Aplikasi yang mengimplementasi algoritma Levenshtein distance pada penelitian ini hanya memproses isi teks
b. Aplikasi yang mengimplementasi algoritma Levenshtein distance pada penelitian ini tidak memperhatikan kesalahan eja/penulisan maupun sinonim/persamaan kata pada isi teks dokumen yang dibandingkan.
c. Output dari aplikasi yang mengimplementasi algoritma Levenshtein distance pada penelitian ini berupa nilai persentase kemiripan isi teks dari pasangan dokumen yang dibandingkan, namun output tidak memberikan kepastian apakah terjadi plagiarisme diantara dokumen teks yang dibandingkan atau tidak.

Tujuan dari penelitian ini adalah untuk mengimplementasi algoritma Levenshtein distance pada aplikasi yang dapat menghitung kemiripan isi teks antar dokumen tugas siswa/mahasiswa sehingga dapat membantu pendidik dalam melakukan pendeteksian dugaan awal plagiarisme, terutama plagiarisme kata demi kata (*word for word plagiarizing*).

Manfaat dari penelitian ini adalah dapat membantu instansi pendidikan dalam mendeteksi kemiripan isi teks antar dokumen teks tugas siswa/mahasiswa. Output aplikasi yang berupa persentase kemiripan isi teks antar dokumen teks tugas siswa/mahasiswa yang dibandingkan dapat dijadikan sebagai bahan pertimbangan untuk mengidentifikasi terjadinya tindak plagiarisme, terutama plagiarisme kata demi kata (*word for word plagiarizing*).

2. Metodologi

2.1 Algoritma Levenshtein Distance

Dalam teori informasi dan ilmu komputer, algoritma Leventshtein *distance* merupakan matriks yang digunakan untuk mengukur perbedaan jarak antara dua sekuens (Janowski dan Mohanty, 2010: 259).

Operasi yang dilakukan dan diperbolehkan digunakan dalam menentukan Levenshtein *distance* ini ada 3 macam operasi (Andhika, 2010: 1), yaitu:

a. Insertion (penyisipan)
   Operasi penyisipan sebuah karakter kedalam sebuah string.
b. Deletion (penghapusan)
   Operasi penghapusan sebuah karakter dari sebuah string.
c. Substitution (penukaran)
   Operasi penukaran sebuah karakter pada sebuah string dengan karakter lain.

Misalnya terdapat dua string, yaitu CS = “ayu” dan ST = “adu”. Proses perhitungan algoritma Levenshtein *distance* adalah sebagai berikut:

a. Membuat matriks berukuran (CS+1) kali (ST+1).

e-ISSN: 2597- 4475
http://doi.org/10.21009/pinter.1.1.4

		a	d	u
	0	1	2	3
a	1			
y	2			
u	3			

b. Melakukan pencocokkan dengan melakukan pembandingan dari setiap karakter CS dengan karakter ST satu per satu secara iteratif.

Jika karakter ke-x CS berbeda dengan karakter ke-y ST, maka nilai cell diperoleh dari nilai terkecil diantara:

a. Nilai cell(x, y-1) + 1.
b. Nilai cell(x, y) + 1.
c. Nilai cell (x-1, y) + 1.

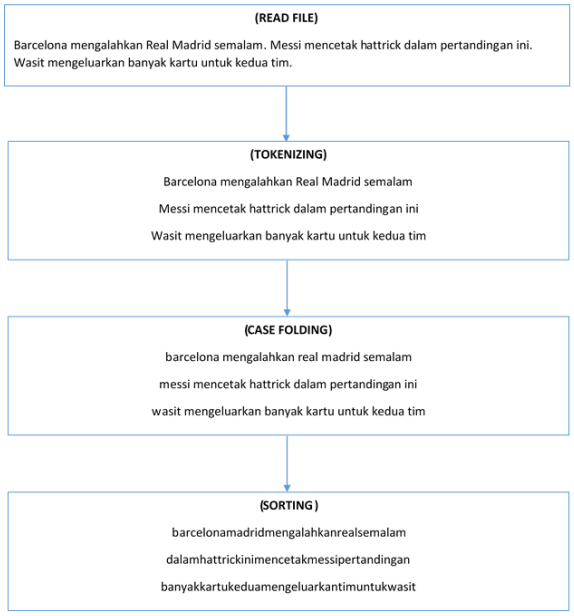
Setelah setiap karakter selesai dibandingkan, nilai terakhir yang didapatkan adalah 1. Hal ini berarti, jarak/distance dari string CS dan ST adalah 1.

		a	d	u
	0	1	2	3
a	1	0	1	2
y	2	1	1	2
u	3	2	2	1

2.2 Preprocessing

Dalam buku The Text Mining Handbook, text mining didefinisikan sebagai proses mendapatkan informasi secara intensif dimana pengguna berinteraksi dengan koleksi-koleksi dokumen menggunakan tools analisis. Secara umum, proses-proses pada text mining mengadopsi proses data mining. Tujuan dari text mining adalah untuk mendapatkan informasi yang berguna dari sekumpulan dokumen. Jadi, sumber data yang digunakan pada text mining adalah kumpulan teks yang memiliki format yang tidak terstruktur atau minimal semi terstruktur.

Pada text mining, informasi yang akan digali berisi informasi-informasi yang strukturnya sembarang. Oleh karena itu, diperlukan proses pengubahan bentuk menjadi data yang terstruktur sesuai kebutuhan untuk proses selanjutnya. Salah satu tahap implementasi dari text mining adalah tahap preprocessing. Tahap preprocessing adalah tahap dimana aplikasi melakukan seleksi data yang akan diproses pada setiap dokumen. Proses preprocessing ini meliputi read file, tokenizing, case folding, filtering, stemming, dan sorting. Namun pada penelitian ini proses filtering dan stemming tidak disertakan karena alasan tertentu. Gambar berikut merupakan contoh dari tahap preprocessing:



Gambar 1. Tahap preprocessing

2.3 Plagiarized Value

Tahap preprocessing menghasilkan token-token keyword string masing-masing dokumen yang dibandingkan. Tahap berikutnya adalah menghitung kemiripan dengan algoritma Levenshtein distance. Setelah itu, dicari nilai persentasenya. Nilai ini disebut juga plagiarized value. Langkah-langkah dalam menentukan plagiarized value dugaan awal kemiripan isi teks adalah (Liaqat dan Ahmad, 2011: 11):

a. Setelah dilakukan penghitungan dari kedua string tersebut menggunakan algoritma Levenshtein *distance*, maka algoritma ini akan memberikan angka d*istance* yang merupakan nilai perbedaan dari kedua string. Misalkan: *String* sumber = *CS*, *String* target = *ST*, *Diff* = *distance*
b. Setelah didapatkan *distance* dari kedua string tersebut, maka formula untuk menghitung dugaan awal kemiripan (*plagiarized value*) kedua string dapat ditentukan sebagai berikut:

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

Keterangan:
Max(CS, ST) merupakan nilai yang paling panjang yang diberikan dari perbandingan CS dan ST. Sedangkan plagiarized value adalah persentase dugaan awal kemiripan antara string yang dibandingkan.

e-ISSN: 2597- 4475
http://doi.org/10.21009/pinter.1.1.4

2.4 Persentase Akhir Kemiripan Dokumen

Output yang dihasilkan oleh aplikasi adalah nilai persentase dugaan awal kemiripan isi teks antar dokumen yang dibandingkan. Persentase akhir diperoleh dengan mengakumulasi plagiarized value semua pasangan token kalimat yang memiliki plagiarized value lebih dari *threshold*, kemudian membaginya dengan nilai terbanyak antara total kalimat terbanyak diantara dokumen yang dibandingkan dengan banyaknya pasangan token kalimat yang lolos *threshold*. Penjabarannya adalah sebagai berikut:

$$Persentase\ Akhir = \left\{\frac{Akumulasi\ PV}{Max(Max(kd1, kd2),\ Lolos)}\right\}$$

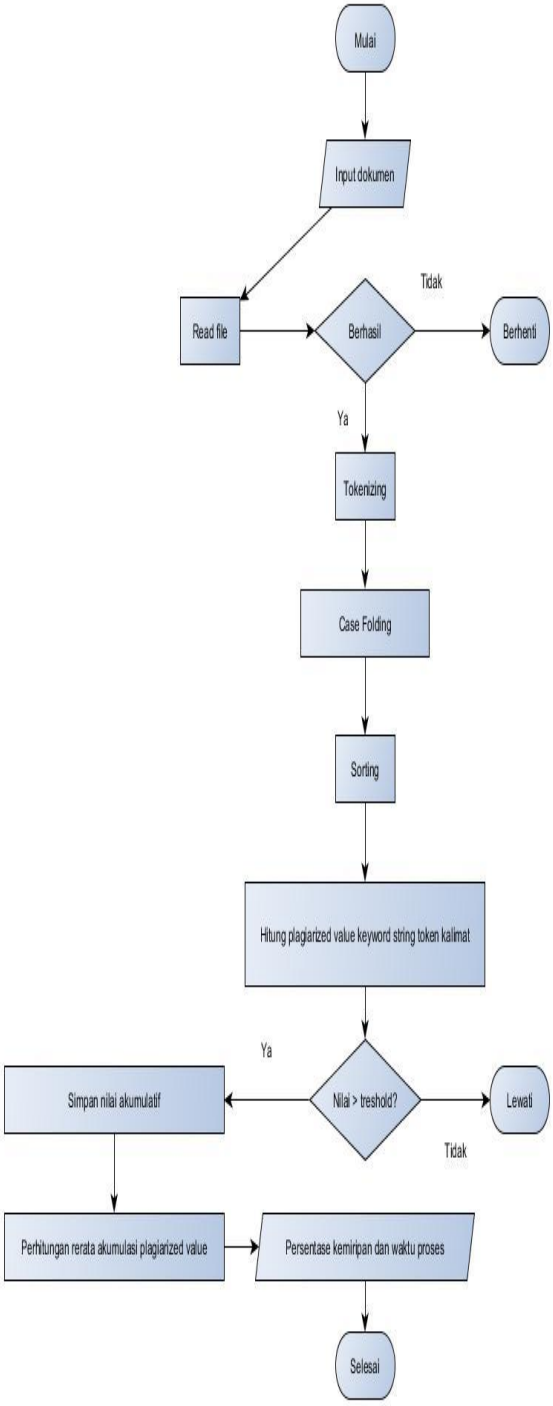
Threshold sendiri merupakan standar ambang batas bagi *plagiarized value* hasil perbandingan dokumen yang akan diakumulasi dan dicari reratanya. *Threshold* dapat diatur sesuai keinginan *user*. Penggunaan nilai *threshold* dapat berpengaruh terhadap hasil akhir.

2.5 Metode Penelitian

Metode penelitian yang digunakan pada penelitian ini adalah salah satu metode pengembangan perangkat lunak, yaitu metode pengembangan sistem Waterfall. Menurut Roger S. Pressman (1992: 39), model Waterfall adalah metode klasik yang bersifat sistematis, berurutan dalam membangun software. Inti dari metode ini adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau linear. Setelah disesuaikan dengan kebutuhan aplikasi, maka tahapan-tahapan dari metode Waterfall yang diterapkan pada penelitian ini yaitu:

a. Perancangan Sistem (System Engineering)
b. Analisis Kebutuhan Perangkat Lunak (Software Requirement Analysis)
c. Perancangan (Design)
d. Pengkodean (Coding)
e. Pengujian (Testing)
f. Pemeliharaan (Maintenance)

Gambar berikut menampilkan flowchart keseluruhan aplikasi:



Gambar 2. Flowchart keseluruhan aplikasi

3. Pengujian

3.1 Data Uji 1

Pengujian ini dilakukan dengan membandingkan dua dokumen yang sama tetapi telah diubah letak struktur kalimatnya. Misalnya, dengan mengubah paragraf induktif menjadi deduktif atau

e-ISSN: 2597- 4475
http://doi.org/10.21009/pinter.1.1.4

sebaliknya. Berikut contoh pengubahan struktur kalimat didalam paragraf:

a. Paragraf A

> Banyak penjual kerudung di jalan keluar pabrik. Di sisi-sisi penjual itu juga nampak penjual kurma dan buah lainnya. Penjual makanan juga memenuhi trotoar jalan, yang sebagian besar adalah penjual dadakan. Mulai dari menu takjil hingga makanan berbuka puasa tersedia disini. Belum lagi penjual pakaian yang memanjang hingga ujung jalan. Kerumunan pedagang tersebut makin membuat lalu lintas padat. Terlebih ketika para pekerja pulang dari pabrik, tetapi hal itu dapat dimaklumi. Memang, setiap menjelang Hari Raya Idul Fitri banyak orang menjadi pedagang karena keuntungannya yang besar.

b. Paragraf B

> Banyak penjual kerudung di jalan keluar pabrik. Di sisi-sisi penjual itu juga nampak penjual kurma dan buah lainnya. Penjual makanan juga memenuhi trotoar jalan, yang sebagian besar adalah penjual dadakan. Mulai dari menu takjil hingga makanan berbuka puasa tersedia disini. Belum lagi penjual pakaian yang memanjang hingga ujung jalan. Kerumunan pedagang tersebut makin membuat lalu lintas padat. Terlebih ketika para pekerja pulang dari pabrik, tetapi hal itu dapat dimaklumi. Memang, setiap menjelang Hari Raya Idul Fitri banyak orang menjadi pedagang karena keuntungannya yang besar.

3.2 Data Uji 2

Pengujian ini dilakukan dengan membandingkan dua dokumen yang struktur kata didalam kalimatnya telah diubah, yaitu dengan mengubah kalimat pasif menjadi aktif atau sebaliknya, serta dengan penukaran posisi kata dan perubahan kata berimbuhan. Kondisi dokumen pembanding pada data uji coba ini mirip dengan tindakan plagiarisme mosaik, dimana plagiarisme tidak dilakukan kata demi kata, melainkan diselang-seling atau disisip-sisipkan sehingga memberikan kesan bahwa dokumen yang dihasilkan adalah dokumen asli.

Berikut contoh pengubahan struktur kata didalam kalimat:

a. Paragraf A

> Beberapa kelebihan dimiliki kelengkeng Pingpong jika dibandingkan dengan kelengkeng lain sehingga sangat cocok untuk dibudidayakan. Ukuran buah sebesar bola pingpong dimiliki oleh kelengkeng pingpong. Kelengkeng Pingpong dapat berbuah dengan cepat. Pada usia 1,5 tahun sudah bisa mulai berbuah. Bila buah-buahan lain rata-rata berbuah setahun sekali, kelengkeng pingpong berbuah lebih dari sekali dalam setahun. Karena buah bisa tumbuh dari ranting-ranting secara bergantian, boleh dikatakan Kelengkeng Pingpong berbuah dengan tidak mengenal musim. Kelengkeng Pingpong bisa ditanam pada media pot, bagi masyarakat perkotaan yang memiliki lahan terbatas. Kelengkeng Pingpong menjadi potensi agrobisnis yang cukup menggiurkan karena kelebihan-kelebihan tersebut.

b. Paragraf B

e-ISSN: 2597- 4475
http://doi.org/10.21009/pinter.1.1.4

Beberapa kelebihan dimiliki kelengkeng Pingpong jika dibandingkan dengan kelengkeng lain sehingga sangat cocok untuk dibudidayakan. Kelengkeng Pingpong memiliki ukuran buah yang besar sebesar bola pingpong. Selain itu bisa tumbuh di dataran rendah. Kelengkeng Pingpong dapat berbuah dengan cepat. Pada usia 1,5 tahun sudah bisa mulai berbuah. Kelengkeng Pingpong dapat berbuah lebih dari satu kali dalam satu tahun. Sedangkan buah-buahan lain rata-rata berbuah satu tahun sekali. Karena buah bisa tumbuh dari ranting-ranting secara bergantian, boleh dikatakan Kelengkeng Pingpong berbuah dengan tidak mengenal musim. Kelengkeng Pingpong bisa ditanam pada lahan sempit bahkan media pot, bagi masyarakat perkotaan yang memiliki lahan terbatas. Kelengkeng Pingpong menjadi potensi agrobisnis yang cukup menggiurkan karena kelebihan-kelebihan tersebut.

3.3 Hasil Data Uji 1

Uji coba dilakukan dengan menguji data menggunakan sistem aplikasi yang telah dijabarkan. Nilai *threshold* yang digunakan adalah >= 60.

Berdasarkan skenario uji coba data uji 1, maka proses penghitungan paragraf A dengan paragraf B adalah sebagai berikut:

A1: banyakdijalankeluarkerudungpabrikpenjual
B3: banyakditemukanjalankeluarkerudungpabrikpenjual
Levenshtein *distance* = 9
Plagiarized value = 81,63%

A2: buahdandiitujugakurmalainnyanampakpenjualsisisisi
B4: buahdandiitujugakurmanampakpenjualsisisisi
Levenshtein *distance* = 7
Plagiarized value = 85,71%

A3:
adalahbesardadakanjalanjugamakananmemenuhipenjualsebagiantrotoaryang
B6:
besardadakanjalanmakananmemenuhipenjualsebagiantersebuttrotoaryang
Levenshtein *distance* = 18
Plagiarized value = 73,52%

A4:
berbukadaridisinihinggamakananmenumulaipuasatakjiltersedia
B5:
berbukadaridisinihinggamakananmenumulaipuasatakjiltersedia
Levenshtein *distance* = 0
Plagiarized value = 100%

A5:
belumhinggajalanlagimemanjangpakaianpenjualujungyang
B7:
belumhinggajalanlagimemanjangpakaianpenjualujungyang
Levenshtein *distance* = 0
Plagiarized value = 100%

A6:
kerumunanlalulintasmakinmembuatpadatpedagangtersebut
B9:
kerumunanlalulintasmakinmembuatpadatpedagangtersebut
Levenshtein *distance* = 0
Plagiarized value = 100%

A7:
dapatdaridimaklumihalituketikapabrikparapekerjapulangterlebihtetapi
B9:
dapatdaridimaklumihalituketikapabrikparapekerjapulangterlebihtetapi
Levenshtein *distance* = 0
Plagiarized value = 100%

A8:
banyakbesarfitrihariidulkarenakeuntungannyamemangmenjadimenjelangorangpedagangrayasetiapyang
B1:
banyakbesarfitrihariidulkarenakeuntungannyamenjadimenjelangorangpedagangrayayang
Levenshtein *distance* = 12
Plagiarized value = 86,95%

Akumulasi PV yang lolos *threshold* = 727, 81.
Max(Max(kd1, kd2), Lolos) = 9

Dengan menggunakan rumus perhitungan nilai persentase akhir kemiripan dokumen, diperoleh nilai persentase dugaan kemiripan isi teks dokumen yang dibandingkan sebesar 80,87%.

3.4 Hasil Data Uji 2

e-ISSN: 2597- 4475
http://doi.org/10.21009/pinter.1.1.4

Uji coba dilakukan dengan menguji data menggunakan sistem aplikasi yang telah dijabarkan. Nilai *threshold* yang digunakan adalah >= 60.

Berdasarkan skenario uji coba data uji 2, maka proses penghitungan paragraf A dengan paragraf B adalah sebagai berikut:

A1:
beberapacocokdengandibandingkandibudidayakandimilikijikakelebihankelengkenglainpingpongsangatsehinggauntuk
B1:
beberapacocokdengandibandingkandibudidayakandimilikijikakelebihankelengkenglainpingpongsangatsehinggauntuk
Levenshtein *distance* = 0
Plagiarized value = 100%

A3: berbuahcepatdapatdengankelengkengpingpong
B4: berbuahcepatdapatdengankelengkengpingpong
Levenshtein *distance* = 0
Plagiarized value = 100%

A4: 15berbuahbisamulaipadasudahtahunusia
B5: 15berbuahbisamulaipadasudahtahunusia
Levenshtein *distance* = 0
Plagiarized value = 100%

A6:
berbuahbergantianbisabolehbuahdaridengandikatakankarenakelengkengmengenalmusimpingpongrantingrantingsecaratidaktumbuh
B8:
berbuahbergantianbisabolehbuahdaridengandikatakankarenakelengkengmengenalmusimpingpongrantingrantingsecaratidaktumbuh
Levenshtein *distance* = 0
Plagiarized value = 100%

A7:
bagibisaditanamkelengkenglahanmasyarakatmediamemilikipadaperkotaanpingpongpotterbatasyang
B9:
bagibahkanbisaditanamkelengkenglahanmasyarakatmediamemilikipadaperkotaanpingpongpotsempitterbatasyang
Levenshtein *distance* = 12
Plagiarized value = 88,12%

A8:
agrobisniscukupkarenakelebihankelebihankelengkengmenggiurkanmenjadipingpongpotensitersebutyang
B10:
agrobisniscukupkarenakelebihankelebihankelengkengmenggiurkanmenjadipingpongpotensitersebutyang
Levenshtein *distance* = 0
Plagiarized value = 100%

Akumulasi PV yang lolos *threshold* = 588, 11.
Max(Max(kd1, kd2), Lolos) = 10

Dengan menggunakan rumus perhitungan nilai persentase akhir kemiripan dokumen, diperoleh nilai persentase dugaan kemiripan isi teks dokumen yang dibandingkan sebesar 58,81%.

4. Kesimpulan

Hasil percobaan yang dilakukan terhadap Data Uji 1, menunjukkn bahwa aplikasi masih dapat mendeteksi kemiripan diantara kedua dokumen. Hal ini terbukti dengan tingginya persentase yang dihasilkan, yaitu 80,87%.

Sedangkan hasil percobaan yang dilakukan terhadap Data Uji 2 dimana dokumen pembandingnya memiliki struktur kata yang berbeda di dalam kalimatnya dibandingkan dengan dokumen aslinya, menunjukkan bahwa aplikasi tidak bisa mendeteksi dengan baik kemiripan dokumen dengan pola kalimat yang diubah-ubah letak susunan katanya. Hal ini terbukti dengan persentase kemiripan dokumen yang rendah, yaitu 58,81%.

Berdasarkan hasil uji coba dan analisis, dapat ditarik kesimpulan bahwa setelah melalui tahapan *preprocessing*, algoritma Levenshtein *distance* dapat diimplementasi pada aplikasi untuk menghitung persentase kemiripan isi teks antar dokumen tugas siswa/mahasiswa sehingga dapat digunakan oleh guru/dosen dalam membantu menduga terjadinya tindak plagiarisme.

Hasil analisis menunjukkan bahwa, dengan metode *preprocessing* yang digunakan pada penelitian ini, aplikasi yang mengimplementasi algoritma Levenshtein *distance* ini lebih cocok digunakan untuk mendeteksi plagiarisme kata demi kata (*word for word plagiarizing*).

Kelemahan dari implementasi algoritma Levenshtein *distance* pada aplikasi di penelitian ini adalah tidak dapat mendeteksi plagiarisme ide secara akurat. Hal ini disebabkan oleh cara kerja algoritma yang menghitung kemiripan *token* string yang dibandingkan berdasarkan karakter yang menyusun masing-masing *token* string tersebut, bukan berdasarkan makna dari masing-masing *token* string itu sendiri.

Daftar Pustaka:

[1] Andhika, Fatardhi R. 2010. Penerapan String Suggestion dengan Algoritma Levenshtein Distance dan Alternatif Algoritma Lain dalam Aplikasi, Sekolah Tinggi Teknik Informatika, Institut Teknologi Bandung (ITB). Bandung.
[2] Gull Liaqat, Ahmad dan Aijaz Ahmad. 2011. Plagiarism Detection in Java Code. School of Computer Science, Physics and Mathematic Linnaeus University.
[3] Sastroasmoro, Sudigdo. 2010. Beberapa Catatan tentang Plagiarisme. Departemen Ilmu Kesehatan Anak, Fakultas Kedokteran Universitas Indonesia. Jakarta
[4] Janowski, Tomasz dan Hrushikesha Mohanty. 2010. Distributed Computing and Internet Technology. Springer-Verlag. Berlin.
[5] Winoto, Henri. 2012. Deteksi Kemiripan Isi Dokumen Teks Menggunakan Algoritma Levenshtein Distance [skripsi]. Malang:

e-ISSN: 2597- 4475
http://doi.org/10.21009/pinter.1.1.4

Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim.

[6] Pressman, Roger. 2010. Software Engineering: A Practitioner's Approach, Seventh Edition. McGraw-Hill. New York.