Jurnal Ilmiah KURSOR
Menuju Solusi Teknologi Informasi

Vol. 6, No. 1, Januari2011 **ISSN 0216 - 0544**

DETEKSI KEBERADAAN KALIMAT SAMA SEBAGAI INDIKASI PENJIPLAKAN DENGAN ALGORITMA *HASHING*BERBASIS*N-GRAM*

[a]Diana Purwitasari, [b]Putu Yuwono Kusmawan, [c]Umi Laili Yuhana
Lab Semantik *Web*, TeknikInformatika, ITS
Gedung Teknik Informatika, Kampus ITS
Jl. Raya ITS, Kampus ITS, Sukolilo, Surabaya, 60111
E-Mail: [a]diana@if.its.ac.id

Abstrak

Maraknya kasus penjiplakan yang dilakukan oleh golongan intelektual menjadi suatu tragedi dalam dunia pendidikan Indonesia. Akibat banyaknya informasi tersedia secara *online*hal yang memprihatinkan tersebut dapat dengan mudah dilakukan dengan kebiasaan *copy–paste* tanpa menyebutkan referensi. Pada makalah ini dibahas cara untuk deteksi keberadaan kalimat sama sebagai hasil *copy–paste*. Meskipun untuk mengetahui suatu penjiplakan perlu ditelaah lebih lanjut seperti adanya penyebutan referensi yang baku. Untuk mendeteksi kesamaan kalimat antar file teks atau problem *common subsequence*digunakan algoritma Winnowing. Algoritma tersebut akan mencari *document fingerprinting* dengan mengubah rangkaian *N-gram* dari teks menjadi sekumpulan nilai-nilai *hash*. Apabila dijumpai kalimat hasil *copy–paste* maka kedua file teks memiliki *document fingerprinting* yang sama. Uji coba dilakukan untuk melihat kemampuan deteksi kalimat sama dengan perubahan parameter nilai *n* dari *n-gram*, bilangan prima *b*sebagai basis *hashing*, ukuran *window w* dan nilai ambang batas penentuan penjiplakan.

Kata kunci:deteksi kalimat sama, algoritma winnowing, *document fingerprinting*, *n-gram*, *hashing*.

Abstract

Abundant cases of plagiarism committed by some intellectual people in the Indonesia's education fields have became such tragedy.Due to the amount of information which is available online are things that make copy-paste without proper citation cause plagiarism. This paper discusses about how to detect similar sentences which is probable caused by copy-paste. However plagiarism detection still needs further examination such as the existing of citation or not. Winnowing algorithm is used for detecting similar sentences between text files which is treated as a common subsequence problem. The algorithm finds document fingerprinting by changing sequence of N-grams from text into a set of hash values.If copy-paste sentences are found then both of text files must have the same document fingerprinting. Experiment has been done to observe the capability of detecting similar sentences by analyzing on value variations of n-gram, prime base b for hashing, window w, and threshold for determining plagiarism indication.

Keywords:common subsequence problem, winnowing algorithm, document fingerprinting, n-gram, hashing.

PENDAHULUAN

Maraknya kasus penjiplakan oleh golongan intelektual menjadi suatu tragedi dalam dunia pendidikan Indonesia seperti kasus profesor termuda bidang hubungan internasional yang diberhentikan secara tidak hormat di tahun 2010 [1]. Akibat banyaknya informasi tersedia secara *online*maka kebiasaan *copy–paste* tanpa menyebutkan referensi menjadi mudah dilakukan. Sehingga karya ilmiah yang dibuat menjadi hasil plagiat dari karya ilmiah lain. Namun dikarenakan sebagian besar karya ilmiah belum dilindungi Undang-Undang Hak atas Kekayaan Intelektual (HaKI) maka plagiarisme digolongkan sebagai kejahatan akademik yang termasuk sebagai pelanggaran etika dan sulit untuk dipidanakan. Sebagai langkah awal untuk mencegah kasus serupa diperlukan cara mendeteksi kemungkinan terjadinya penjiplakan seperti di lingkungan perguruan tinggi yaitu utamanya pada hasil tugas akhir calon sarjana S1 maupun tesis calon sarjana S2 dan disertasi calon sarjana S3 yang rawan penjiplakan.

Suatu karya ilmiah dikatakan sebagai hasil penjiplakan apabila kutipan yang dilakukan tidak disertai penyebutan referensi secara benar. Sehingga dijumpai kalimat yang sama saja tidak berarti karya ilmiah tersebut dinyatakan sebagai hasil plagiat. Algoritma Winnowing [2] yang dibahas pada makalah ini adalah suatu cara untuk mendeteksi adanya kalimat – kalimat yang sama atau sering disebut juga sebagai problem *common subsequence* [3].

Aplikasi Dustball - *The Plagiarism Checker* yang dibuat oleh tim dari University of Maryland Dustball [4] menggunakan fasilitas mesin pencari dengan mencari kalimat yang diduga sebagai hasil penjiplakan dalam web, sedangkan Copyscape mendeteksi isi dari halaman web menurut alamat URL yang diisikan [5]. Kedua aplikasi tersebut mencurigai adanya penjiplakan berdasarkan urutan posisi kata dalam kalimat seperti penelitian di Universitas Gajah Mada (UGM) Yogyakarta dengan nama TESSY (*Test of Texts Similarity*) [6]. Urutan posisi kata juga dapat digunakan dalam pengaplikasian lain seperti pengecekan urutan kata dalam *spell checker* [7].

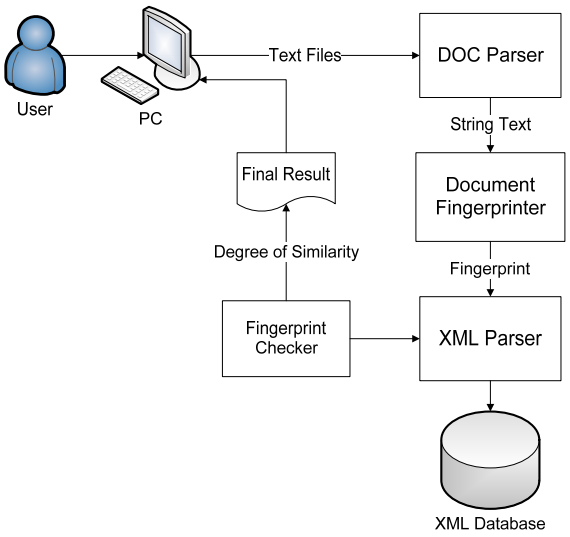
Pada makalah ini kalimat yang sama dikenali melalui *fingerprint* dari dokumen [2].Identifikasi penjiplakan dengan teknik *fingerprint*(sidik jari) akan merubah urutan kata dengan setiap panjang tertentu (*window*) menjadi suatu nilai yang dianggap sebagai sidik jari. Teknik *fingerprint* dapat mengenali frase yang dicurigai banyak dijiplak pada suatu dokumen teks meskipun telah sedikit mengalami perubahan dengan cara parafrase. Hal tersebut yang masih belum bisa dikenali dengan pendekatan urutan posisi kata dalam kalimat.

Makalah ini membahas langkah – langkah untuk deteksi keberadaan kalimat sama yang menjadi indikasi penjiplakan dengan algoritma Winnowing sebagai algoritma *hashing* berbasis *n-gram* dijelaskan pada uraian selanjutnya. Kemudian makalah akan menunjukkan implementasi sistem untuk uji coba keberhasilan deteksi keberadaan kalimat sama. Sebagai penutup akan diuraikan uji coba beserta analisa dan simpulan.

ALGORITMA WINNOWING UNTUK MEMBUAT *FINGERPRINT* FILE TEKS

Banyak cara atau metode yang dapat digunakan untuk mendeteksi penjiplakan dalam file teks dengan mengenali kalimat – kalimat yang mirip. Namun ada kebutuhan mendasar yang harus dipenuhi oleh algoritma deteksi tersebut yaitu [2]: (i)*whitespace insensitivity*yaitu pencarian kalimat mirip seharusnya tidak terpengaruh oleh spasi, jenis huruf (kapital atau normal), tanda baca dan sebagainya; (ii) *noise surpression* yaitu menghindari penemuan kecocokan dengan panjang kata yang terlalu kecil atau kurang relevan seperti ‘*the*’dan bukan merupakan kata yang umum digunakan; (iii) *position independence* yaitu penemuan kesamaan harus tidak bergantung pada posisi kata-kata sehingga kata dengan urutan posisi berbeda masih dapat dikenali jika terjadi kesamaan.

Cara kerja algoritma Winnowing yang digunakan pada makalah ini untuk mendeteksi adanya keberadaan kalimat sama sebagai suatu indikasi terjadinya penjiplakan menganut ketiga prinsip tersebut. Pada detil bahasan berikut ditunjukkan keterkaitan antara ketiga prinsip diatas dengan langkah – langkah yang dilakukan dalam algoritma Winnowing.



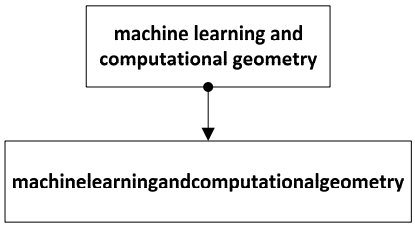
Gambar 1. Langkah-langkah Deteksi Kalimat Mirip.

Implementasi dari Algoritma Winnowing membutuhkan masukan file teks dan menghasilkan keluaran berupa sekumpulan nilai *hash* disebut *fingerprint*.

Proses untuk menghasilkan *fingerprint* sebuah dokumen yang ditunjukkan pada Gambar 1 adalah sebagai berikut:

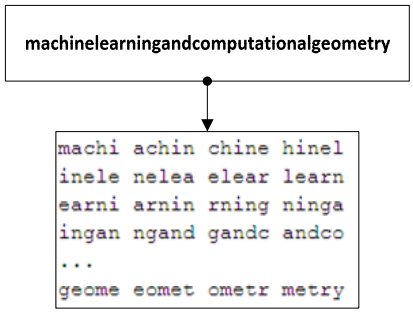
1. Membuang karakter-karakter tidak relevan seperti tanda baca.

Contoh teks:
machine learning and computational geometry



Langkah tersebut terkait dengan *whitespace insensitivity* dan pembuangan kata – kata tidak penting seperti artikel atau kata sambung maka isu *noise surpression* telah teratasi. Pembuangan kata – kata tidak penting atau sering disebut *stopword* merupakan bagian dari prapemrosesan teks.

2. Membentuk rangkaian *n-gram* dari teks, semisal *n*=5.



Untuk teks tersebut dihasilkan 35 gram.

3. Melakukan fungsi *hash* untuk setiap *n-gram*.

```
machi achin chine hinel
inele nelea elear learn
earni arnin rning ninga
ingan ngand gandc andco
...
geome eomet ometr metry
```

```
12756 11891 12203 12660
12809 13009 12411 12800
12261 12350 13582 13141
12803 12994 12351 12135
...
12497 12578 13305 13063
```

Persamaan (1) adalah perhitungan fungsi *hash* dari algoritma Winnowing [2]:

$$H_{(c_1 \dots c_k)} = c_1 * b^{(k-1)} + c_2 * b^{(k-2)} + \dots + c_{(k-1)} * b^k + c_k$$

$$H_{(c_2 \dots c_{k+1})} = \left(H_{(c_1 \dots c_k)} - c_1 * b^{(k-1)}\right) * b + c_{(k+1)}(1)$$

dengan nilai ASCII karakter *c*, nilai basis bilangan prima *b*, dan banyak karakter *k*.

Sebagai contoh *n-gram* dari "*machi*" dengan nilai *b* = *3* dan *k* = *5* memiliki nilai *hash*:

$$H_{(machi)} = ascii(m) * 3^{(4)} + ascii(a) * 3^{(3)} + ascii(c) * 3^{(2)} + ascii(h) * 3^{(1)} + ascii(i) * 3^{(0)}$$

$$H_{(machi)} = 109 * 81 + 97 * 27 + 99 * 9 + 104 * 3 + 105 * 1 = 12756$$

4. Memilih *fingerprint* dari hasil *hashing* dengan pembagian hasil *hash* berdasarkan satu nilai *window w*, dan kemudian dipilih nilai *hash* terkecil.

Semisal *w* = 4 sehingga *window* yang dibentuk dari 4 nilai-nilai *hash* adalah sejumlah 32 *window* sebagai berikut:

[12756 **11891** 12203 12660]
[**11891** 12203 12660 12809]
[**12203** 12660 12809 13009]
[12660 12809 13009 **12411**]
[12809 13009 **12411** 12800]
[13009 12411 12800 **12261**]
...
[**12114** 12880 12497 12578]
[12880 **12497** 12578 13305]
[**12497** 12578 13305 13063]

Kemudian *fingerprint* yang dihasilkan adalah sejumlah 15 nilai *hash* dari 15 *window* yaitu: 11891 12203 12411 12261 12350 ... 12450 13351 12377 12891 12114 12497. Sebagai catatan, untuk pasangan *window* 1 - 2 atau 4 – 5 atau *window* 35 – 36 yang memiliki nilai *hash* terkecil sama maka *window* kanan yang dipilih yaitu *window* 2, 5, dan 36.

Setiap file teks akan memiliki *fingerprint* sejumlah *jwindow* yang telah ditentukan atau dokumen $D = \{w_1 \dots w_j\}$. Kemudian setiap pasang dokumen akan dicari nilai *hash* yang sama dari set *window* tersebut dengan Persamaan (2):

$$Kesamaan = \frac{jumlahhashsama}{totaljumlahhash} x\ 100\% \quad (2)$$

Pencarian kesamaan tersebut menunjukkan bahwa algoritma Winnowing tidak bergantung pada posisi kata-kata dalam mencari adanya kesamaan atau disebut *position independence*.

Misalkan file teks $D_1\ dan\ D_2$ dengan *fingerprint*D_1 =
11891 12203 12411 12261 12350 12803 12351 12135 12211 12450 13351 12377 12891 12114 12497
dan *fingerprint*D_2 =
12232 12268 12411 12500 12195 12508 12756 11891 12203 12411 12261
,maka sesuai dengan Persamaan (2) akan diperoleh nilai kesamaan sebesar 19.05%.

$$Kesamaan = \frac{|11891\ 12203\ 12411\ 12261|}{21}\ x\ 100\%$$
$$= \frac{4}{21} x\ 100\% = 19.0476\%$$

SISTEM DETEKSI KALIMAT SAMA DENGAN WINNOWING

Untuk uji coba deteksi kalimat sama dengan Algoritma Winnowing telah diimplementasikan sebuah sistem dengan fungsi – fungsi utama yang ditunjukkan pada Gambar 1.

Fungsi ***pertama***[*DOC Parser*]akan mengekstrak isi dari file yang ingin dideteksi kemungkinan adanya penjiplakan.

Fungsi***kedua*** [*Document Fingerprinter*] memproses teks hasil ekstraksi [*DOC Parser*] menjadi *fingerprint*dokumen teks berupa nilai-nilai *hash*.Untuk membaca isi file berekstensi .doc, digunakan *library* yang diunduh gratis melalui http://poi.apache.org/.

Fungsi ***ketiga***[*XML Parser*] melakukan proses baca dan tulis terhadap file *xml* yang menjadi *repository* dokumen teks. Pada awalnya sistem diimplementasikan untuk uji coba mengetahui kemungkinan adanya penjiplakan dari laporan tugas perkuliahan dengan melihat adanya kalimat uang sama. Sehingga setiap file laporan akan disimpan kedalam repository sebagai file XML.*Fingerprint Checker* akan membandingkan fingerprint yang dimiliki oleh dokumen masukan serta dokumen yang akan dijadikan pembanding dari repository. Hasil dari sistem berupa *file* yang disebut *Final Result* derajat kesamaan antar dokumen teks berdasarkan perbandingan *fingerprint*.

Fungsi ***keempat***[*Fingerprint Checker*] akan membandingkan *fingerprint* yang dimiliki oleh dokumen masukan serta dokumen yang akan dijadikan pembanding dari *repository*. Hasil dari sistem berupa file yang disebut *Final Result*derajat kesamaan antar dokumen teks berdasarkan perbandingan *fingerprint*.

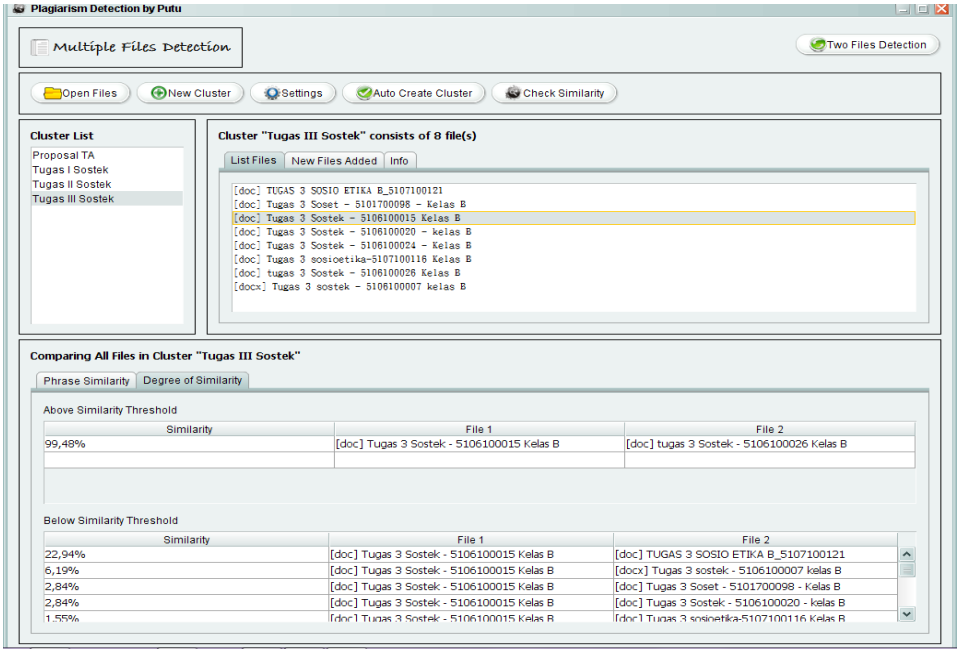
Sistem untuk uji coba dibangun dalam lingkungan pemrograman dengan spesifikasi sebagai berikut: (i) Microsoft Windows Seven Ultimate sebagai sistem operasi; (ii) NetBeans 6.7 sebagai tool utama pemrograman dengan JDK versi 1.6; (iii) prosesor Intel Pentium Core 2 Duo 1.83 Ghz dan memori 2 Gb.

Antarmuka sistem hasil implementasi untuk uji coba deteksi kalimat sama ditunjukkan pada Gambar 2. Pengguna menekan tombol "*Open File*" untuk memasukkan file yang akan dideteksi kalimatnya. Kemudian untuk deteksi kalimat sama dalam dua file teks maka pengguna menekan tombol "*Check Similarity*" guna memulai proses pemeriksaan. Untuk melihat hasil akhir, pengguna dapat menekan tombol "*Show Highlight*" yang mengaktifkan fitur penanda pada area berisi teks dari file jika terdapat kesamaan kalimat pada kedua file teks yang diperiksa (lihat Gambar 3). Pengguna juga

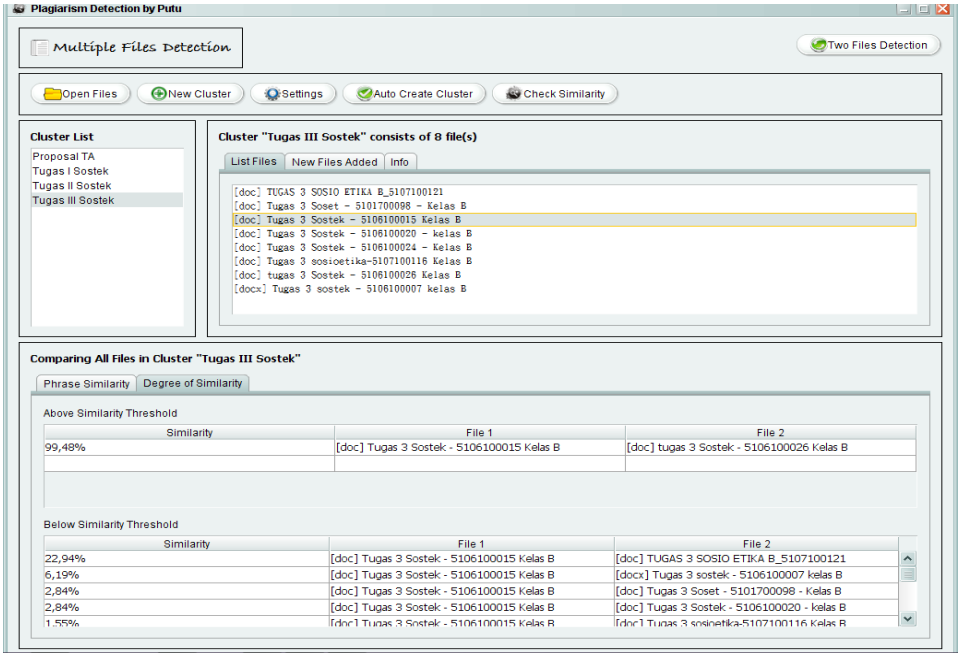
dapat membuat *cluster* atau kelompok file-file yang ingin diperiksa untuk membantu proses deteksi kalimat sama jikalau jumlah file teks yang akan diperiksa terlalu banyak.

HASIL DAN PEMBAHASAN

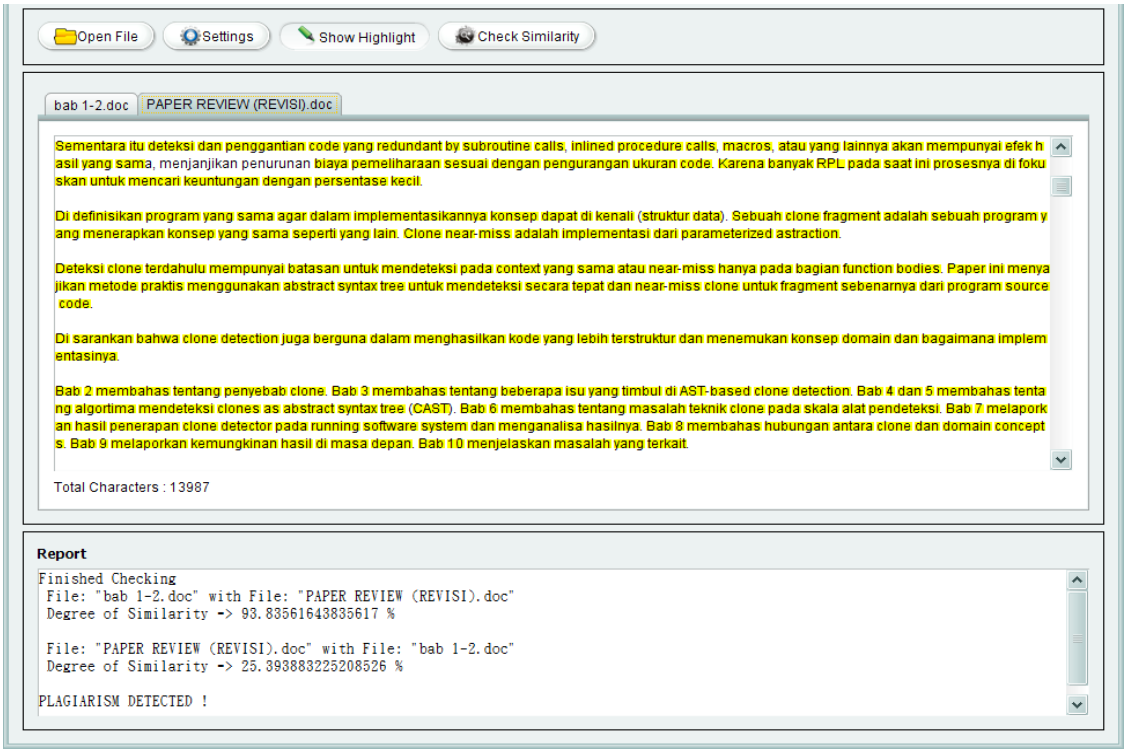
Uji coba fungsionalitas dari implementasi dilakukan untuk verfikasi dan validasi semua fungsi yang ada. Fungsi – fungsi tersebut adalah deteksi kalimat sama sebagai indikasi terjadinya penjiplakan dalam dua file teks, dan kemudian dalam banyak file dengan membuat kelompok file untuk mempermudah proses. Selain itu juga dilakukan uji coba dengan berbagai konfigurasi guna mendapatkan konfigurasi terbaik untuk deteksi kalimat sama.



Gambar 2. Antarmuka untuk Sistem Uji Coba Deteksi Kalimat Sama.



Gambar 3. Antarmukauntuk Sistem Uji Coba Deteksi Kalimat Sama**.**



Gambar 4. Antarmuka untuk Pemberian *Highlight* pada Teks yang Sama.

Tabel 1. Uji Coba Konfigurasi Parameter Algoritma *Winnowing* untuk Deteksi Kalimat Sama.

	WINDOW W	*N-GRAM*	*BASIS B*	*THRESHOLD*	*KEMIRIPAN*	*JML FILE*	*SIM. < THRES.*
SKENARIO 1	10	30	3	50%	98,9%	2	22,2%
	30	30	3	50%	99,0%	2	22,9%
	50	30	3	50%	98,2%	2	21,7%
SKENARIO 2	30	10	3	50%	99,2%	2	46,4%
	30	30	3	50%	99,0%	2	22,9%
	30	50	3	50%	64,2%	2	0,0%
	30	50	3	20%	53,3%	5	0,0%
SKENARIO 3	30	30	3	50%	99,0%	2	22,9%
	30	30	5	50%	62,4%	2	0,0%
	30	30	17	50%	66,6%	2	0,0%

Tabel 2. Hasil Deteksi Kalimat Sama dengan Algoritma *Winnowing* Berdasarkan Konfigurasi *W = 30, N = 30, B = 3*, dan *Threshold = 50%*.

NO	KEMIRIPAN	FILE TEKS 1	FILE TEKS 2
1	99,48%	[DOC] TUGAS 3 SOSTEK - 5106100015 KELAS B	[DOC] TUGAS 3 SOSTEK - 5106100026 KELAS B
2	98,47%	[DOC] TUGAS 3 SOSTEK - 5106100026 KELAS B	[DOC] TUGAS 3 SOSTEK - 5106100015 KELAS B
3	22,94%	[DOC] TUGAS 3 SOSTEK - 5106100015 KELAS B	[DOC] TUGAS 3 SOSIO ETIKA B_5107100121
4	22,92%	[DOCX] TUGAS 3 SOSTEK - 5106100007 KELAS B	[DOC] TUGAS 3 SOSET - 5101700098 - KELAS B
5	22,70%	[DOC] TUGAS 3 SOSTEK - 5106100026 KELAS B	[DOC] TUGAS 3 SOSIO ETIKA B_5107100121
...	...	...	...
28	0,14%	[DOC] TUGAS 3 SOSTEK - 5106100024 - KELAS B	[DOC] TUGAS 3 SOSIO ETIKA B_5107100121

Konfigurasi parameter yang diamati dengan hasil ditunjukkan pada Tabel 1 adalah:
(i) nilai *n* dari *n*-gram,
Parameter nilai *n* pada algoritma Winnowing digunakan saat mengubah teks sepanjang *n-gram* menjadi sekumpulan nilai-nilai *hash*.

(ii) bilangan prima *b* yang menjadi basis dalam proses *hash*,
Parameter nilai *b* digunakan saat mengubah *n-gram* menjadi nilai *hash* dengan fungsi *hashing* yang membutuhkan bilangan prima *b*.

(iii) ukuran *window w*
Parameter nilai *window w* digunakan untuk mengambil perwakilan nilai *hash* sebagai bagian *fingerprint* yang tepat.

(iv) nilai ambang batas penentuan penjiplakan.
Parameter nilai ambang batas digunakan sebagai batas munculnya kalimat sama yang melebihi nilai toleransi tertentu.

Pada Tabel 1 terdapat tiga skenario uji coba yang bertujuan untuk (a) mencari nilai *windoww* yang tepat; (b) mencari nilai *n* dari *n-gram* yang sesuai; (c) mencari batas nilai *threshold* yang cukup; dan (d) mencari nilai basis bilangan prima *b* untuk fungsi *hashing*.

Uji coba untuk tiga skenario dengan empat tujuan dilakukan menggunakan sejumlah file teks, 8 dokumen, berupa laporan tugas kuliah mahasiswa. File teks yang digunakan tidak terlalu banyak untuk mempermudah verifikasi kebenaran hasil. Pada delapan dokumen tersebut terdapat dua file teks yang sebagian besar isinya memang memiliki kalimat hasil *copy-paste*, utamanya pada bagian latar belakang tentang dasar teori dari laporan tugas.

Skenario 1 dilakukan dengan nilai n = 30, basis b = 3, dan *threshold* = 50% untuk mencari nilai *windoww* yang tepat. Terlihat dengan variasi nilai *w*, hasil menunjukkan bahwa semakin lebar nilai*w* maka kemiripan kalimat dalam file teks mudah terdeteksi. Meskipun nilai *window* yang terlalu lebar tidak berpengaruh banyak pada tingkat terdeteksinya kalimat sama.

Berdasarkan hasil uji coba pertama didapatkan nilai w = 30 yang akan digunakan pada uji coba skenario 2 bertujuan untuk mencari nilai *n* dari *n-gram* dan nilai *threshold* yang cukup. Kemudian menggunakan nilai basis b = 3 dan *threshold* = 50% didapatkan ketelitian dalam deteksi kalimat sama pada file teks berkurang sampai ±60%. Bahkan dengan nilai n = 50 yang terlalu besar artinya *commonsubsequence* sampai 50 karakter maka hampir tidak mungkin ditemui. Hal itu terlihat dari tingkat deteksi kalimat sama selain dua file teks yang memang mengandung kalimat hasil *copy-paste* adalah bernilai 0%. Selanjutnya dengan nilai *n-gram* sama yaitu n = 50 dicoba deteksi kalimat sama dengan pengurangan *threshold* sampai 20%. Jumlah file yang bisa dikenali bertambah dari dua menjadi lima dengan tingkat deteksi kalimat sama selain lima file tersebut tetap 0% karena memang jarang kalimatatau *commonsubsequence* sampai 50 karakter. Sehingga berdasarkan hasil uji coba skenario 2, disarankan nilai n = 30 untuk mendapatkan *commonsubsequence* dengan panjang kata yang cukup umum sebagai hasil *copy-paste*.

Skenario 3 dilakukan untuk analisa variasi nilai b untuk mencari basis bilangan prima dalam fungsi *hashing* yang paling sesuai. Pada skenario 3 nilai w = 30, n = 30, dan *threshold* = 50% ditetapkan berdasarkan hasil analisa dari kedua skenario uji coba pendahulu yang ditunjukkan pada Tabel 1. Variasi nilai b = {3, 5, 17} menunjukkan bahwa semakan besar basis bilangan prima maka ketelitian dalam deteksi kalimat sama pada file teks berkurang sampai ±60%. Hal tersebut juga terjadi apabila nilai *n-gram* semakin panjang. Bahkan dengan basis bilangan prima terkecil kedua yaitu $b = 5$ maka kalimat sama dari file teks sudah tidak dapat dideteksi. Oleh karena itu nilai basis bilangan prima yang paling sesuai untuk nilai w = 30, n = 30, dan *threshold* = 50% adalah nilai $b = 3$.

Tabel 2 menunjukkan hasil deteksi kalimat sama pada dua file teks dari 8 dokumen uji coba. Dihasilkan 28 kombinasi untuk pasangan dua file teks dari 8 dokumen tersebut. Deteksi kalimat sama diproses dengan konfigurasi paramater algoritma Winnowing hasil tiga skenario uji coba yang telah dilakukan. Terlihat bahwa dengan konfigurasi parameter yang sesuai maka setiap kombinasi pasangan akan tetap dapat dikenali adanya kalimat sama bahkan dengan tingkat kemiripan sampai 0.14%. Kemungkinan bagian 0.14% kalimat yang sama adalah nama jurusan atau deskripsi laporan tugas kuliah.

SIMPULAN

Makalah ini telah membahas algoritma Winnowing sebagai algoritma untuk deteksi kalimat sama sebagai indikasi terjadinya penjiplakan. Uji coba telah dilakukan untuk melihat kemampuan mendeteksi kalimat sama sebagai indikasi penjiplakan dengan perubahan nilai – nilai tertentu. Parameter – parameter yang telah diamati adalah nilai *n* dari *n*-gram, bilangan prima *b* yang menjadi basis dalam proses *hash*, ukuran *window w* dan nilai ambang batas penentuan penjiplakan. Parameter nilai *n* pada algoritma Winnowing digunakan saat mengubah teks sepanjang *n*-gram menjadi sekumpulan nilai-nilai *hash* dengan fungsi *hashing* yang membutuhkan bilangan prima *b* tertentu. Kemudian set parameter nilai *window w* yang tepat dibutuhkan untuk mengambil perwakilan nilai hash sebagai bagian *fingerprint*. Adanya indikasi penjiplakan terjadi jika kalimat sama yang muncul melebihi suatu batas toleransi tertentu. Sehingga nilai ambang batas juga diamati dalam uji coba.

Berdasarkan uji coba yang telah dilakukan terlihat bahwa konfigurasi nilai yang kurang tepat menyebabkan deteksi adanya kalimat mirip tidak terjadi. Pada makalah ini deteksi kalimat dari dua file teks dilakukan *one-to-one* sehingga akan cukup memakan waktu apabila jumlah file tugas yang akan dibandingkan mencapai 175 untuk mahasiswa satu angkatan. Untuk penelitian selanjutnya akan dicoba dilakukan pengelompokkan file berdasarkan nilai *hashing* guna mengenali jenis laporan seperti tugas laporan mata kuliah sosio etika atau laporan mata kuliah jaringan komputer. Kemudian deteksi kalimat cukup diproses dengan perwakilan nilai *hashing* kelompok.

DAFTAR PUSTAKA

[1] Editor Berita Institut Teknologi Bandung, Pernyataan Sikap ITB Terhadap Plagiarisme Mochammad Zuliansyah, 2010, URL: http://www.itb.ac.id/news /2813.xhtml, diakses tanggal 10 Januari 2011.

[2] Schleimer S, Wilkerson D, andAiken A, Winnowing: Local Algorithms for Document Fingerprinting, Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 76–85, 2003.

[3] Oetsch J, Pührer J, Schwengerer M, andTompits H, The System Kato: Detecting Cases of Plagiarism for Answer-Set Programs, Theory and Practice of Logic Programming, 10(4-6), pp 759-775, 2010.

[4] Klug B, The Plagiarism Checker, 2002, URL: http://www.dustball.com/cs/ plagiarism.checker/, diakses tanggal 10 Januari 2011.

[5] Greenspan G, Copyscape, 2006, URL: http://copyscape.com, diaksestanggal 10 Januari 2011.

[6] Editor The Jakarta Post, Tracing Plagiarism Makes Cheating Hard, 2008, url: http://www.thejakartapost.com/news/ 2008/12/26/tracing-plagiarism-makes-cheating-hard. html, diaksestanggal 10 Januari 2011.

[7] Ahmad T, Jatmiko N, dan Safii M, Perancangan danPembuatanAplikasi SMS Spell Checker Berbahasa Indonesia denganMenggunakanAlgoritma Naive Bayes pada Mobile Device, Kursor, 4 (2): 58-66,2008.