ISSN 2355-3286

Pengaruh Algoritma *Stemming* Nazief-Adriani Terhadap Kinerja Algoritma Winnowing Untuk Mendeteksi Plagiarisme Bahasa Indonesia

Hargyo Tri Nugroho I.
Program Studi Sistem Komputer, Universitas Multimedia Nusantara, Tangerang, Indonesia
hargyo@umn.ac.id

Diterima 5 Mei 2017
Disetujui 16 Juni 2017

Abstract*—Winnowing algorithm is one among many algorithms for detecting document similarity and plagiarism. Some studies show that Winnowing algorithm performs quite well. One form of plagiarism is paraphrase plagiarism. Paraphrase plagiarism can be done by changing sentence structure, changing vocabulary, and adding or changing affixes. Based on some of our previous experiments, detecting document resemblances can be enhanced by changing the words containing affixes to their basic words. In computer science, this technique is known as stemming - a technique to extract the basic word from an affixed word. Usually this technique is required in the filtering process to save storage media. For Indonesian, the Nazief-Adriani stemming algorithm is by far the most appropriate. This study examines how the effect of Nazief-Adriani stemming algorithm on Winnowing algorithm's performance against Indonesian texts. The results showed that the stemming process using Bloom-Filter on the Winnowing algorithm tends to decrease the similarity level achieved, but it accelerates processing time by approximately 30%.

***Keywords*—Algoritma Nazief-Adriani, Algoritma Winnowing, Bloom-Filter, Plagiat, Plagiat Checker**

I. PENDAHULUAN

Kemajuan teknologi informasi yang pesat memungkinkan penggunanya untuk berbagi informasi dengan cepat dalam berbagai format digital. Berbagai artikel berbahasa Indonesia dapat kita baca melalui laman *web* secara daring maupun mengunduhnya dalam format Ms. Word atau PDF secara luring. Dokumen-dokumen tersebut dapat dengan mudah kita temukan melalui mesin pencari maupun mengunjungi repository yang banyak tersedia di internet [1]. Kemudahan mengakses dokumen-dokumen yang tersedia dalam format digital ini tentu saja banyak memberi dampak positif namun juga mempermudah dilakukannya plagiasi.

Rawannya plagiat pada dokumen digital mendorong para peneliti untuk mengembangkan piranti lunak *plagiarism checker* untuk mendeteksi plagiasi dengan cara mengukur tingkat kemiripan dokumen tersebut dengan dokumen-dokumen lainnya. Hal ini penting karena kredibilitas akademisi maupun penerbit ditentukan oleh originalitas artikelyang diterbitkannya [2]. Salah satu algoritma yang banyak digunakan adalah algoritma Winnowing. Algoritma ini sederhana namun cukup dapat diandalkan untuk mendeteksi plagiat [3] .

Bahasa Indonesia memiliki karakteristik yang berbeda dari bahasa yang lain. Penambahan imbuhan (afiksasi) pada sebuah leksem (morfem dasar) dapat dilakukan dengan berbagai cara; sebagai awalan (prefix), akhiran (sufix), maupun sisipan (infix) [4].

Afiksasi pada leksem tidak hanya merubah bentuknya, namun juga maknanya secara gramatikal, sedangkan maknanya yang semula yaitu makna leksikal sedikit banyak tidak berubah [5].

Plagiat parafrase dapat dilakukan dengan mengubah struktur kalimat, merubah kosakata, serta menambah atau mengubah imbuhan [6]. Berdasarkan beberapa percobaan kami sebelumnya, kemiripan dokumen dapat ditingkatkan dengan mengubah kata-kata yang mengandung imbuhan menjadi kata dasarnya. Pada ilmu komputer, teknik ini dikenal sebagai *stemming* yaitu suatu teknik untuk mengekstraksi kata dasar dari suatu kata [7]. Umumnya teknik ini diperlukan pada proses *filtering* untuk menghemat media penyimpanan. Untuk bahasa Indonesia, algoritma *stemming* Nazief-Adriani sejauh ini dirasa paling sesuai [8]. Paper ini membahas bagaimana pengaruh algoritma *stemming* Nazief-Adriani terhadap kinerja algoritma Winnowing terhadap teks-teks berbahasa Indonesia.

.

II. RISET SEBELUMNYA

Kinerja Algoritma Winnowing dibandingkan algoritma *fingerprint* oleh [9]. Hasil pengujian menunjukkan bahwa walaupun secara kinerja algoritma Winnowing (91.8%) tidak sebaik algoritma *fingerprint* (92.8%), namun algoritma Winnowing memiliki tingkat relevansi topik yang lebih baik daripada algoritma *fingerprint*.

Algoritma Winnowing diimplementasikan oleh Ryan et al. [10] menggunakan metode K-Gram dipadukan dengan Synonim Analysis untuk meningkatkan akurasi *plagiarism checker*. Hasilnya sangat memuaskan khususnya terhadap artikel jiplakan yang banyak mengalami perubahan kosakata. Selain itu algoritma Winnowing juga menunjukkan kinerja yang baik pada [11] yang mengimplementasikannya pada jurnal *online*.

Teknik *stemming* digunakan oleh Sagala [12] dengan memadukan algoritma Enhanced Confix Stripping bersama dengan algoritma Winnowing namun hasilnya perpaduan ini justru menghasilkan kinerja yang kurang baik. Namun, hasil yang berbeda ditemukan oleh Alfikri et al. [13] yang menyatakan bahwa *stemming* akan memperbaiki kinerja sistem namun tidak signifikan. Perbedaan hasil ini mengarah pada suatu hipotesa bahwa pengaruh *stemming* terhadap kinerja algoritma winnowing akan berbeda-beda tergantung beberapa kondisi yang akan diteliti pada penelitian ini.

III. TINJAUAN PUSTAKA

A. *Algoritma Winnowing*

Algoritma Winnowing [3] adalah algoritma untuk menghasilkan suatu deret bilangan unik (*fingerprint*) yang mewakili suatu dokumen. Dengan *fingerprint* tersebut kita bisa mengetahui tingkat kemiripan satu dokumen dengan dokumen yang lain.

Secara garis besar, algoritma Winnowing bekerja sebagai berikut:

1. Penghapusan karakter-karakter yang tidak relevan (*whitespace insensitivity*).
2. Pembentukan rangkaian gram dengan ukuran k.
3. Penghitungan nilai hash.
4. Membagi ke dalam window tertentu.
5. Pemilihan beberapa nilai hash menjadi *fingerprint* dokumen

B. *Algoritma Stemming Nazief-Adriani*

Stemming adalah cara yang digunakan untuk meningkatkan performa *Information Retrieval* dengan cara mentransformasikan kata-kata dalam sebuah dokumen teks ke kata dasarnya [8].

Proses *stemming* pada teks Bahasa Indonesia digunakan untuk menghilangkan sufiks, konfiks, dan prefiks. Hal ini berbeda dengan teks Bahasa Inggris, di mana *stemming* digunakan untuk mengilangkan sufiks [8].

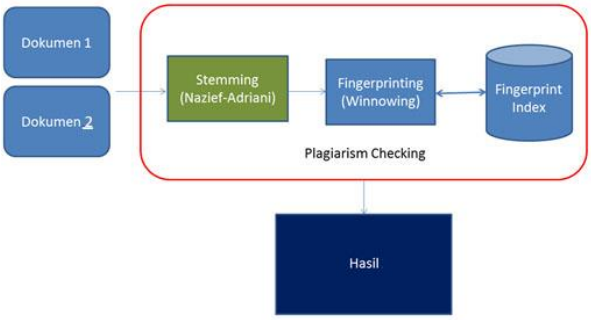
Algoritma yang dibuat oleh Bobby Nazief dan Mirna Adriani memiliki tahapan sebagai berikut:

1. Cari kata yang akan distem dalam kamus. Jika ditemukan maka diasumsikan bahwa kata tesebut adalah root word. Maka algoritma berhenti.
2. Inflection Suffixes(“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”) dibuang. Jika berupa particles (“-lah”, “-kah”, “-tah” atau “-pun”) maka langkah ini diulangi lagi untuk menghapus Possesive Pronouns(“-ku”, “-mu”, atau “-nya”), jika ada.
3. Hapus Derivation Suffixes(“-i”, “-an” atau “-kan”). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah 3a
    a. Jika “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.
    b. Akhiran yang dihapus (“-i”, “-an” atau “-kan”) dikembalikan, lanjut ke langkah 4.
4. Hapus Derivation Prefix. Jika pada langkah 3 ada sufiks yang dihapus maka pergi ke langkah 4a, jika tidak pergi ke langkah 4b.
    a. Periksa tabel kombinasi awalan-akhiran yang tidak diijinkan. Jika ditemukan maka algoritma berhenti, jika tidak pergi ke langkah 4b.
    b. For i = 1 to 3, tentukan tipe awalan kemudian hapus awalan. Jika root word belum juga ditemukan lakukan langkah 5, jika sudah maka algoritma berhenti. Catatan: jika awalan kedua sama dengan awalan pertama algoritma berhenti.
5. Melakukan Recoding.
6. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai root word. Proses selesai.

IV. ARSITEKTUR SISTEM

Seperti terlihat pada gambar 1, pada penelitian ini algoritma Winnowing diimplementasikan dengan diawali dengan preprocessing. Penelitian ini menggunakan dua jenis preprocessing yaitu *case folding*, dan *stemming*. Pengujian akan menggunakan dua jenis implementasi preprocessing yaitu case folding saja, dan case folding ditambah dengan *stemming* untuk mengetahui pengaruh *stemming* terhadap hasil akhir algoritma Winnowing.

Proses ini *stemming* dimulai ketika *Nazief-Adriani Stemmer* menerima kalimat dari *case folding*. Kalimat ini merupakan rangkaian kata tanpa huruf kapital. Pada sebagian besar implementasi yang ditemui, kalimat ini kemudian diolah dengan menggunakan aturan pembuangan imbuhan dan pencocokan kata dengan data yang ada di dalam kamus atau basis data. Pencocokan kata pada database ini memiliki kelemahan yaitu lamanya waktu yang dibutuhkan untuk melakukan *string matching.*



Gambar 1. Arsitektur Sistem

Pada penelitian ini diajukan solusi berupa proses tambahan, yaitu dengan menggunakan *Bloom Filter* [14]. Pemanfaatan *Bloom Filter* ini dimulai dengan membuat suatu ruang penyimpanan di dalam memori yang memiliki ukuran sesuai dengan jumlah data dari tabel di database. Kemudian ruang penyimpanan ini diisi dengan nilai hashing dari setiap kata dasar yang ada di database. Setelah itu proses *Bloom Filter* dapat digunakan oleh proses *Nazief-Adriani Stemmer* untuk melakukan pencocokan kata dasar. Dengan demikian proses pencarian kata dasar tidak lagi menggunakan database, namun menggunakan struktur data Bloom Filter yang siap pakai pada RAM.

Setelah preprocessing, file teks dibentuk menjadi rangkaian substring senilai k atau k-gram. Nilai dari rangkaian substring selanjutnya diproses menjadi rangkaian hash. Hashing adalah proses untuk mengubah string menjadi bilangan integer yang disebut nilai hash. Proses pengubahan menjadi nilai hash menggunakan fungsi rolling hash. Rolling hash merupakan salah satu metode hashing yang menghitung nilai hash dengan tanpa mengulangi semua string.

Setelah rangkaian hash terbentuk, proses selanjutnya adalah pembentukan window dari rangkaian hash. Window merupakan substring dari nilai hash sepanjang wgram. Proses ini menghasilkan *fingerprint* yang digunakan untuk menentukan tingkat kemiripan dokumen.

V. HASIL PENGUJIAN

A. *Pengaruh Penggunaan Bloom-Filter*

Percobaan dilakukan dengan menggunakan 4 buah artikel yang dipilih secara acak berisikan kalimat yang merupakan rangkaian kata dasar maupun kata berimbuhan. Seperti terlihat pada tabel 1, rata-rata kecepatan pengolahan (*processing speed*) *stemming* menggunakan database adalah 9,15 kata/detik. Sedangkan apabila menggunakan Bloom-Filter kecepatan pengolahan meningkat sangat tajam menjadi rata-rata 8039,73 kata/detik. Hal ini membuat perbedaan waktu yang semakin signifikan, berbanding lurus dengan jumlah kata dalam artikel.

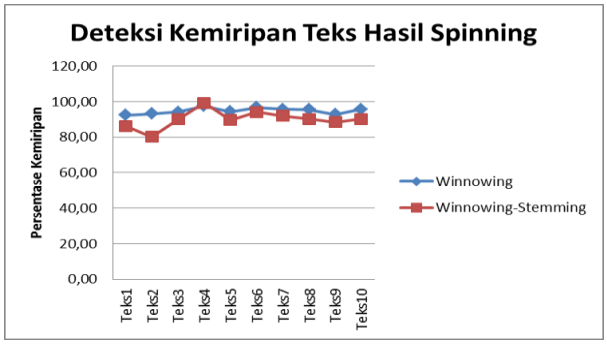
Bloom-Filter lebih cepat diakses karena berada di RAM. Pencocokan datanya pun tidak dilakukan dengan *string matching* melainkan melalui proses *hashing*. Tentu saja ini sangat jauh berbeda dengan database yang berada di hard disk yang memiliki *access time* yang jauh lebih lambat dibanding RAM.

Tabel 1. Perbandingan Kecepatan Stemming Menggunakan Database vs Bloom Filter

File	Jumlah Kata	Database		Bloom Filter	
		Waktu	Kecepatan (kata/detik)	Waktu	Kecepatan (kata/detik)
Teks3	651,00	65,86	9,88	0,09	7233,33
Teks5	712,00	76,69	9,28	0,10	7340,21
Teks6	460,00	54,76	8,40	0,05	10000,00
Teks9	622,00	68,85	9,03	0,08	7585,37
Rata-rata			**9,15**		**8039,73**

B. *Pengaruh Stemming Nazief-Adriani Pada Winnowing*

Gambar 2 menunjukkan hasil pengujian algoritma Winnowing – *Stemming* untuk mendeteksi kemiripan artikel yang direkayasa menggunakan teknik *spin*, yaitu mengganti beberapa kata secara acak dengan sinonimnya. Rekayasa ini tanpa merubah struktur kalimat. Dapat dilihat bahwa algoritma Winnowing mampu mendeteksi kemiripan artikel yang dibuat dengan teknik *spinning* sebesar rata-rata 94,84% sedangkan algoritma Winnowing – *Stemming* mendeteksi kemiripan artikel sebesar rata-rata 90,03%.

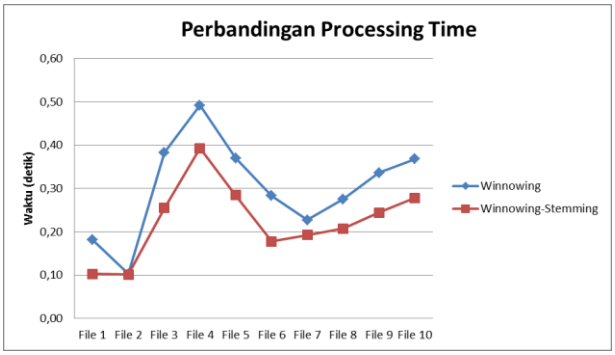


Gambar 2. Deteksi Kemiripan Terhadap Teks Hasil *Spinning*

Hasil yang berbeda ditunjukkan gambar 3. Pada percobaan ini beberapa artikel diparafrase menggunakan teknik *spin* dan perubahan struktur kalimat. Perubahan struktur kalimat yang dimaksud mencakup mengubah kalimat dari pasif menjadi aktif (dan sebaliknya), melakukan penambahan imbuhan, serta menambahkan kata penghubung.. Algoritma Winnowing-*Stemming* rata-rata mendeteksi kemiripan dokumen yang telah dirubah struktur kalimatnya sebesar 45,60%. Sedangkan algoritma Winnowing murni mampu mendeteksi kemiripan di atas 60%.

Gambar 3. Deteksi Kemiripan Terhadap Teks Hasil Parafrase

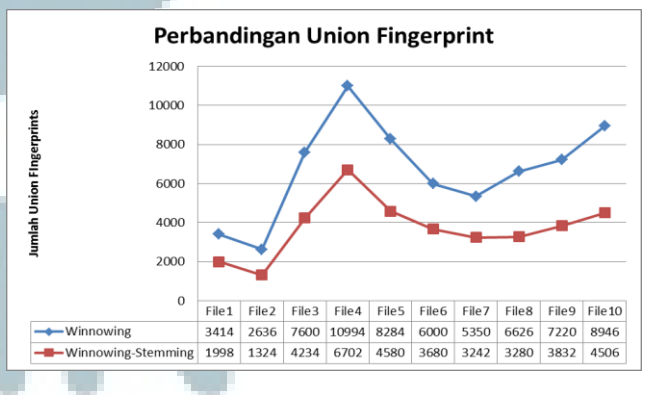
Untuk plagiasi berbasis parafrase yang merubah struktur kalimat, baik algoritma Winnowing murni maupun algoritma Winnowing-*Stemming* menunjukkan kinerja yang lebih rendah dibandingkan pada saat deteksi kemiripan dokumen berbasis spinning. Hal ini menunjukkan bahwa, dengan merubah struktur kalimat, *n-gram* maupun *fingerprints* yang terbentuk lebih berbeda dibandingkan hanya sekedar mengganti kata-kata dengan sinonimnya saja.Jumlah *fingerprints* yang semakin sedikit, memperkecil peluang munculnya *fingerprints* yang sama. Sehingga hasil kalkulasi koefesien Jaccard akan lebih kecil.



Gambar 4. Perbandingan Processing Time Algoritma Winnowing vs Winnowing - *Stemming*

Seperti terlihat di gambar 4, *stemming* membuat algoritma Winnowing berjalan lebih cepat. Hal ini disebabkan karena berkurangnya jumlah karakter yang harus diproses lebih lanjut hingga menjadi *fingerprints*. Terlihat pada gambar 5, *stemming* memotong jumlah *union fingerprint* yang linear dengan penurunan processing time apabila dibandingkan dengan algoritma Winnowing murni.

Berdasarkan pengujian di atas maupun analisis algoritma, algoritma Winnowing memiliki potensi perbaikan terutama dalam proses pembentukan *n-gram* yang memakan waktu relatif tinggi.



Gambar 5. Perbandingan Jumlah Union Fingerprints

Seluruh pengujian di atas menggunakan *k-gram* 5 dan *w-gram* 4. Baik *w-gram*, maupun *k-gram* memiliki pengaruh terhadap hasil *similarity* maupun waktu proses (*processing time*). Semakin besar *w-gram*, maupun *k-gram*, semakin lama waktu proses yang diperlukan. Namun, pada penelitan ini *k-gram* lebih menentukan tingkat *similarity* maupun waktu prosesnya. Semakin kecil nilai *k-gram* maka semakin kecil jumlah karakter yang akan dicocokan dan semakin sering rangkaian karakter tersebut akan ditemukan dalam teks.

VI. KESIMPULAN

Dari hasil pengujian dan analisis maka dapat disimpulkan beberapa hal sebagai berikut: Algoritma Winnowing sangat efektif untuk mendeteksi plagiarisme dokumen baik dengan teknik *spinning* maupun parafrase yang merubah struktur kalimat. Proses *stemming* pada algoritma Winnowing cenderung menurunkan tingkat *similarity* yang dicapai, namun mempercepat processing time kurang lebih sebesar 30%. Penggunaan Bloom-Filter dalam proses *stemming* terbukti efektif untuk mempercepat processing time sekitar 1000 kali lebih cepat.

DAFTAR PUSTAKA

[1] Ali, A.E.T., H.D. Abdulla and V. Snasel, "Survey of plagiarism detection methods", Proceedings of the 5th Asia Modelling Symposium, May 24-26, 2011, Manila, Philippines, pp: 39-42.

[2] Khan, M.A., A. Aleem, A. Wahab and M.N. Khan, "Copy detection in Urdu language documents using n-grams model", Proceedings of the International Conference on Computer Networks and Information Technology, July 11-13, 2011, Abbottabad, Pakistan, pp: 263-266.

[3] Schleimer, S., Wilkerson, D. S., & Aiken, A., "Winnowing: local algorithms for document fingerprinting", Proceedings of the 2003 ACM SIGMOD international conference on Management of data (pp. 76-85). ACM 2003.

[4] Kridalaksana, H., "Pembentukan kata dalam bahasa Indonesia", Gramedia pustaka utama, 1989.

[5] Purnanto, Dwi, "Kajian morfologi derivasional dan infleksional dalam bahasa Indonesia", Kajian Linguistik dan Sastra, Vol. 18, No. 35, 2006: 136-152

[6] Clough, P., "Old and New Challenges in Automatic Plagiarism Detection", Sheffield, UK : Department of Information Studies, University of Sheffield, 2003.

[7] Utomo, M. S., & Winarko, E., "Design And Implementation of Document Similarity Search System For WEB-Based Medical Journal Management", IJCCS-Indonesian Journal of Computing and Cybernetics Systems, 5(1), 2013.

[8] Agusta, Ledy, "Perbandingan Algoritma *Stemming* Porter dengan Algoritma Nazief & Adriani untuk *Stemming* Dokumen Teks Bahasa Indonesia", Prosiding Konferensi Nasional Sistem dan Informatika, November 14th, 2009, Bali, Indonesia, pp: 196-201

[9] Wibowo, A. T., Sudarmadi, K. W., & Barmawi, A. M., "Comparison between fingerprint and winnowing algorithm to detect plagiarism fraud on Bahasa Indonesia documents", Information and Communication Technology (ICoICT), 2013 International Conference of (pp. 128-133). IEEE, March 2013.

[10] Ryan, Mudafiq, E.B. Cahyono, & G.I. Marthasari, "Aplikasi Pendeteksi Duplikasi Dokumen Teks Bahasa Indonesia Menggunakan Algoritma Winnowing Dengan Metode K-Gram Dan Synonym Recognition", Universitas Muhammadiyah, Malang. Tersedia daring: http://www.mudafiqriyan.net/wp-content/uploads/2012/03/Mudafiq-AplikasiPendeteksiDuplikasiDokumen.pdf

[11] Kharisman, O., Susanto, B., & Suwarno, S., "IMPLEMENTASI ALGORITMA WINNOWING UNTUK MENDETEKSI KEMIRIPAN PADA DOKUKEN TEKS", Jurnal Informatika, 9(1), 2013.

[12] Sagala, A. C. S., "Pendeteksian Kesamaan Pada Dokumen Teks Menggunakan Kombinasi Algoritma Enhanced Confix Stripping Dan Algoritma Winnowing", 2009.

[13] Alfikri, Z. F., & Purwarianti, A., "The Construction Of Indonesian-English Cross Language Plagiarism Detection System Using Fingerprinting Technique", Jurnal Ilmu Komputer dan Informasi, 2012, 5(1), 16-23.

[14] Mullin, James K. "A second look at Bloom filters." Communications of the ACM 26.8 (1983): 570-571.