OPEN ACCESS
Indo-JC Indonesia Journal of Computing
ISSN 2460-9056
socj.telkomuniversity.ac.id/indojc

Ind. Journal on Computing
Vol. 4, Issue. 3, Dec 2019. pp. 87-98
doi:10.21108/indojc.2019.4.3.365

Deteksi Kemiripan Dokumen Bahasa Indonesia Menggunakan Algoritma Smith-Waterman dan Algoritma Nazief & Andriani

Bunga Sari [1], Yuliant Sibaroni [2]

 # Prodi Informatika, Fakultas Informatika, Universitas Telkom
Jl. Telekomunikasi No.1 Terusan Buah Batu, Bandung, Indonesia

[1] bungasari@student.telkomuniversity.ac.id
[2] yuliant@telkomuniversity.ac.id

Abstract

Technological developments are increasingly sophisticated with the internet. The internet that can be easily accessed to find information and documents can lead to acts of plagiarism. Every document that carries out plagiarism will be difficult to recognize without a system that can recognize the similarity of documents. A system that can detect plagiarism by looking for similarities to the required documents. In this study Smith-Waterman algorithm was used to identify the most significant similarity (local alignment) of two sequences of nucleotide sequences of processes (sequences) or protein sequences so that the similarities between the two documents can be seen. The Nazief & Andriani algorithm is a stemming algorithm on text-preprocessing as a supporting algorithm in the process of determining the similarity of text documents. The results obtained in this study are the comparison of two sequences with the help of preprocessing has a greater level of similarity calculation in detecting document similarities. From the average test on the original document and the test document, the level of similarity produced is more than 50%, which means that the results can be stated as a plagiarism action.

Keywords: *Document, Plagiatrism, Algoritma Smith-Waterman, Algorithm Nazief & Andriani*

Abstrak

Perkembangan teknologi semakin canggih dengan adanya internet. Internet yang dapat dengan mudah diakses untuk mencari informasi dan dokumen dapat memicu adanya tindak plagiarisme. Setiap dokumen yang melakukan tindakan plagiarisme akan susah dikenali tanpa adanya sistem yang dapat mengenali kesamaan dokumen. Sistem yang dapat mendeteksi plagiarisme dengan mencari kemiripan pada dokumen dibutuhkan. Dalam penelitian ini digunakan algoritma *Smith-Waterman* untuk mengidentifikasi kesamaan yang paling signifikan (*local alignment*) dari dua buah rangkaian *sequence* string proses (rangkaian/susunan) *nucleotide* atau *protein sequences* sehingga kemiripan antara dua dokumen tersebut dapat terlihat. Algoritma *Nazief & Andriani* merupakan algoritma *stemming* pada *text-preprocessing* sebagai algoritma pendukung dalam proses penentuan nilai kemiripan dokumen teks. Hasil akhir yang didapatkan pada penelitian ini adalah perbandingan dua *sequence* dengan bantuan *preprocessing* memiliki tingkat perhitungan *similarity* yang lebih besar dalam mendeteksi kemiripan dokumen. Dari rata-rata pengujian pada dokumen asli dan dokumen uji menunjukkan tingkat kemiripan yang dihasilkan lebih dari 50%, yang berarti hasil tersebut dapat dinyatakan terjadi tindak plagiarisme.

Kata Kunci: Dokumen, Plagiarisme, Algoritma *Smith-Waterman*, Algoritma *Nazief & Andriani*

Received on Nov 2019. Accepted on Dec 2019

I. PENDAHULUAN

MENURUT Adimihardja [1], plagiat merupakan salah satu permasalahan yang ada didunia Pendidikan saat ini. Plagiarisme adalah tindakan penyalahgunaan, pencurian dan penggunaan gagasan atau tulisan ciptaan orang lain yang diakui sebagai miliknya sendiri [2] [3]. Tindakan plagiat merupakan hal yang kriminal dimana banyaknya dokumen-dokumen dapat ditemui di internet sehingga dapat dengan mudah di akses dan di-*copy* [2]. Plagiarisme dapat merugikan pemilik sah dari sebuah laporan atau penelitiannya. Setiap dokumen yang melakukan plagiat akan susah dikenali tanpa adanya sistem yang dapat mengenali kesamaan dokumen tersebut [4]. Kemiripan pada dokumen dapat dilihat dari kemiripan kata atau terjadi perubahan pada setiap kalimat dengan menambahkan imbuhan atau mencari persamaan kata (Sinonim). Terkadang disertai dengan membolak balik susunan kata dan mengubah bentuk pasif ke aktif [4]. Sehingga diperlukan sistem untuk analisis deteksi kemiripan atau plagiarisme.

Berdasarkan permasalahan diatas, dibutuhkan analisis deteksi kemiripan untuk membantu terjadinya tindakan plagiarisme yang dapat merugikan oranglain dan mengetahui kemiripan suatu dokumen. Deteksi kemiripan merupakan suatu cara atau usaha untuk mencari kemiripan pada dokumen, dari hasil deteksi tersebut dapat dilihat berapa persen kemiripan dokumen yang dibandingkan. Beberapa algoritma atau metode yang digunakan untuk mendeteksi tindak plagiatrisma seperti: *Algoritma Winnowing* [5]*, Cosine Similarity* [6]*. Algoritma* SCAM [2]*.* Penelitian ini menggunakan algoritma *Smith-Waterman* dikarenakan algoritma klasik yang telah dikenal luas dalam bidang bioinformatika sebagai metode yang dapat mengidentifikasi *local similarities* (penyejajaran sekuens) yaitu proses penyusunan dua *local sequences* (rangkaian/susunan atau rentetan) *nucleotide* atau *protein sequences* sehingga kemiripan antara dua *sequence* tersebut akan terlihat [6]. Terdapat tahap *preprocessing* pada penelitian ini menggunakan algoritma *Nazief & Adriani*, algoritma ini digunakan dalam penelitian yang dilakukan sebagai algoritma pendukung dalam proses penentuan nilai kemiripan pada dokumen teks. Algoritma *Nazief & Andriani* membantu dalam proses pembuatan data testing, semakin baik data yang digunakan dapat mempengaruhi nilai akurasi suatu sistem.

Berdasarkan fungsi proses penyejajaran sekuens tersebut, maka algoritma ini dapat dikonversikan ke dalam pemrograman komputer untuk digunakan membantu proses pendeteksian dokumen teks yang dianggap cenderung plagiat dengan cara melihat kesamaan isi (*local similarities*) dari beberapa dokumen teks [2]. Pembuatan sistem pendeteksi plagiat ini menggunaka algoritma *Smith-Waterman* dengan *text-preprocessing* terdiri dari, *tokenization, stopword removal, synonym replacement* dan *stemming* (Algoritma *Nazief & Adriani*). Algoritma *stemming* berdasarkan pada aturan morfologi bahasa Indonesia yang luas, yang dikumpulkan menjadi satu grup dan di-enkapsulasi pada imbuhan/*affixes* yang diperbolehkan (*allowed affixes*) dan imbuhan/*affixes* yang tidak diperbolehkan (*disallowed affixes*). Algoritma ini menggunakan kamus kata dasar dan mendukung *recoding*, yakni penyusunan kembali kata-kata yang mengalami proses *stemming* berlebih [7].

II. KAJIAN LITERASI

Algoritma *Smith-Waterman* klasik dari *molecular biology domain* yang biasa untuk membandingkan dokumen teks, kesamaan lokal yang signifikan dan dapat digunakan sebagai alat yang ampuh untuk mendeteksi plagiarisme dan kolusi [8]. Implementasi algoritma *Smith-Waterman* dapat digunakan sebagai alat bantu dalam menemukan kesamaan dokumen teks Bahasa Indonesia. Pengujian dilakukan dengan tahapan *preprocessing, preprocessing* pada dokumen akan meningkatkan kinerja algoritma *Smith-Waterman* dalam menemukan kesamaan sebesar 12.14 % untuk dokumen yang menipulasi perubahan sinonim. Dokumen yang memanipulasi perubahan imbuhan sebesar 13,82%, dan dokumen yang menipulasi kombinasi perubahan sebesar 20.91 % [9].

Algoritma *Smith-Waterman* yang diimplementasikan dapat menghasilkan local *alignment* yang akurat dari perbandingan dokumen teks. *Preprocessing* membantu performa algoritma *Smith-Waterman* saat pengisian matriks kesamaan dan pembentukan *local alignment, similarity* dari sistem menjadi lebih akurat namun waktu prosesnya menjadi lama apabila dibantu dengan tahapan *preprocessing* [10]. Dalam penelitian yang lain

mengenai plagirisme algoritma ini berhasil diimplementasikan dalam sistem pendeteksi kesamaan dokumen *2)* dan mampu menemukan kesamaan antara dokumen satu dengan dokumen yang lainnya serta menampilkan presentasi kesamaan dari kedua dokumen [11].

Novanta A [12] telah melakukan studi mengenai pendeteksian plagiarisme pada dokumen teks dengan menggunakan algoritma *Smith-Waterman.* Algoritma *Smit-Waterman* tentang pendeteksian plagiarisme teks dokumen menghasilkan keakuratan yang lebih baik pada saat membandingkan dokumen yang mengandung pengubahan struktur kalimat di dalam paragraf daripada mengubah struktur kata di dalam kalimat. Dengan bantuan proses tambahan *pre-processing,* sistem pendeteksian pendeteksian plagiarisme dapat menghasilkan bobot/nilai kecenderungan terjadinya tindakan plagiat yang lebih akurat.

Su, Z et al. [13] dalam jurnalnya telah melakukan studi mengenai deteksi plagiarisme menggunakan *Levenshtein Distance* dan algoritma *Smith-Waterman.* Algoritma *Smith-Waterman* yang disederhanakan dapat digunakan untuk kesamaan yang signifikan. Dari hasil percobaan dapat dibuktikan bahwa itu alat yang ampuh untuk mendeteksi plagiarisme. Hal ini dapat mengeksplorasi heuristik yang sesuai dalam bidang perbandingan teks, dan mengurangi kelemahan *Smith–Waterman* yang lambat untuk aplikasi skala besar

Jurnal yang diunggah oleh Rahman, E. [14] menjelaskan peningkatan kinerja algoritma *Smith-Waterman* untuk pendeteksian plagiarisme pada dokumen teks. Perbandingan *error rate* dan waktu proses antara algoritma *Smith-Waterman* asli dengan *Smith-Waterman* revisi memiliki *error rate* lebih kecil daripada *Smith-Waterman* asli, meskipun waktu prosesnya masih lebih tinggi. Hasilnya prosentase penurunan *error rate* lebih besar daripada prosentase kenaikan waktu prosesnya, performansi *Smith-Waterman* revisi lebih tinggi daripada *Smith-Waterman* asli.

Berdasarkan penelitian yang telah dilakukan oleh Agusta Ledy [15]. Dengan judul perbandingan algoritma *stemming porter* dengan algoritma *Nazief dan Adriani* untuk *stemming* dokumen teks Bahasa Indonesia yang membahas mengenai algoritma *stemming* dengan hasil penelitian yaitu proses *stemming* dokumen teks Bahasa Indonesia menggunakan algoritma *Porter* memiliki prosentase keakuratan (presisi) lebih kecil dibandingkan dengan *stemming* menggunakan algoritma *Nazief dan Adriani.*

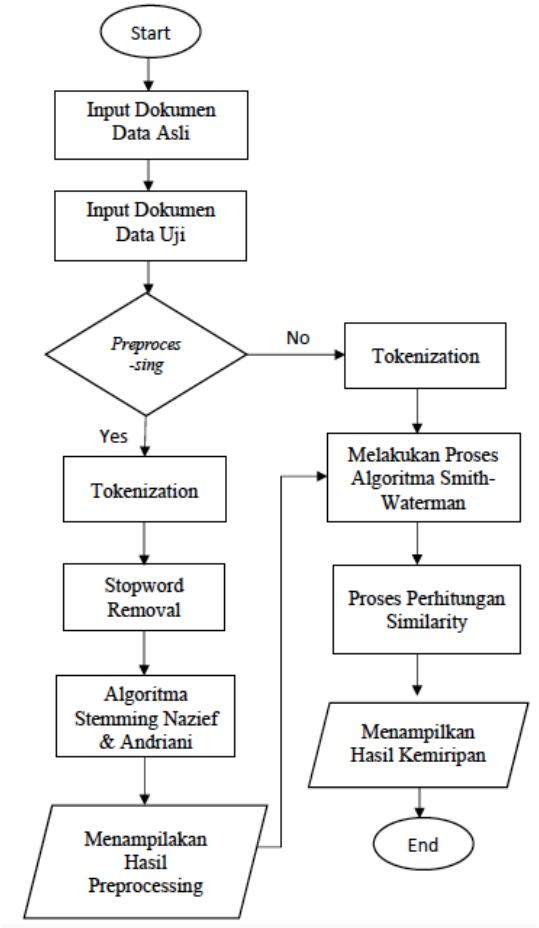
III. Metode penelitian

Pada penelitian ini dibangun sebuah sistem untuk mendeteksi kemiripan dokumen teks berbahasa Indonesia dengan menggunakan algoritma *Smith-Waterman.* Dokumen teks bahasa Indonesia yang menjadi data masukan bertipe dokumen teks dengan ekstensi .txt. Antar dokumen teks 1 dengan yang lainnya dibandingkan dengan dokumen uji, hasil akhir sistem menampilkan kemiripan dari perbandingan masing-masing dokumen.

A. Rancangan Sistem

Terdapat diagram *flowchart* atau alur sistem pada sestem deteksi kemiripan dokumen untuk mengetahui urutan proses yang berjalan dengan baik. Diagram *flowchart* dapat dilihat pada gambar I.

Pada gambar I merupakan diagram *flowchart* alur proses keseluruhan pada sistem deteksi kemiripan pada dokumen teks. Dokumen asli dan dokumen uji diinputkan pada sistem untuk menampilkan hasil perbandingan. Terdapat proses *preprocessing* untuk menghilangkan noise pada dokumen, dengan tahapan yang dilakukan yaitu, *Tokenization, Stopword Removal* dan Algoritma *Stemming Nazief & Andriani.* Disaat pengujian membutuhkan (Yes) *preprocessing* maka akan dilakukan tahapan *Tokenization, Stopword Removal* dan Algoritma *Stemming Nazief & Andriani.* Namun apabila tidak dibutuhkan (No) *preprocessing,* tahapan yang dilakukan langsung *tokenization.* Tahap selanjutnya proses mencari kemiripan dokumen dengan algoritma *Smith-Waterman* dan proses perhitungan *similarity.* Tahap akhir adalah menampilkan hasil dari kemiripan.



Gambar I. Diagram Proses Deteksi Plagiarism

B. Algoritma *Preprocessing*

Tahap *preprocessing* berfungsi untuk menghilangkan *noise* pada dokumen [10]. Sebagian besar metode deteksi berbasis konten mengasumsikan fase *preprocessing* dimana kata-kata berhenti dihapus dan kata-kata direduksi menjadi bentuk *root* mereka, langkah-langkah dilakukan untuk mengubah teks menjadi reprensentasi terstruktur dan terformat, yang lebih efektif untuk proses deteksi plagiarisme [16]. *Preprocessing* terdiri dari beberapa proses diantaranya.

1. Proses *Tokenization* adalah proses pemecahan teks terstruktur dalam suatu dokumen berdasarkan kalimat-kalimat penyusun. Penghapusan tanda baca dan mengubah semua huruf didalam dokumen menjadi kecil [10]. Adapun beberapa tanda baca yang dipisahkan adalah koma (,), titik (.), titik dua (:), garis miring (/), tanda tanya (?), dan tanda garis/pisah (-) [11].
2. Proses *Stopword Removal* merupakan pendekatan *preprocessing* mendasar yang menghilangkan kata-kata umum [17], kata yang tidak berkontribusi banyak terhadap dokumen yaitu kata yang sering muncul dalam setiap dokumen akan dihapus [18] . Tahap terakhir dari *preprocessing* adalah *stemming* yang digunakan untuk mencari kata dasar dari setiap kata yang ada dalam dokumen [10], yaitu dengan cara menghilangkan semua imbuhan (*affixes*) baik yang terdiri dari awalan (*prefixes*), sisipan (*infixes*), akhiran (suffixes) dan kombinasi dari awalan dan akhiran (*confixes*) pada kata turunannya [9] [18].

3. Proses algoritma *Stemming* yang digunakan adalah *Nazief & Andriani. Stemming* adalah salah satu cara yang digunakan untuk meningkatkan performa IR (*information retrieval*) dengan cara mentrasformasi kata-kata pada sebuah dokumen teks kebentuk kata dasar [15], Algoritma ini melakukan proses *stemming* dengan menghilangkan akhiran terlebih dahulu, kemudian diikuti dengan penghilangan awalan [12]. Urutan penghilangan imbuhan sbb [15]:

    - *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.
    - 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.

    - 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.

    - Melakukan Recoding.
    - Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai root word. Proses selesai.

C. Perbandingan Algoritma *Smith-Waterman* dan Algoritma *Nazief & Andriani*

Tabel I Perbandingan Algoritma *Smith-Waterman* dan *Algoritma Nazief & Andriani*

Algoritma *Nazief & Andriani*	Algoritma *Smith-Waterman*
• Algoritma ini merupakan algoritma *Stemming* untuk *text-preprocessing* berbahasa Indonesia [15]. • Algoritma ini digunakan dalam penelitian yang dilakukan sebagai algoritma pendukung dalam proses penentuan nilai kemiripan pada dokumen teks [6]. • Algoritma ini membantu dalam proses pembuatan data testing, semakin baik data yang digunakan dapat mempengaruhi nilai akurasi sistem [6].	• Algoritma *Smith-Waterman* merupakan algoritma yang dapat mengidentifikasi *local similarity* (penyejajaran sekuens) proses penyusunan dua *local sequence* (rangkaian/susunan atau rentetan) *nucleotide* atau *protein sequences* sehingga kemiripan antara dua dokumen tersebut dapat terlihat [12]. • Algoritma ini dapat dikonversikan dalam program kamputer untuk digunakan memantu proses deteksi dokumen yang dianggap plagiat, dengan cara melihat kesamaan ini (*local similarities*) [12].

Pada tabel I menunjukkan perbandingan kedua algoritma yaitu algoritma *Smith-Waterman* dan Algoritma *Nazief & Andriani.*

D. Algoritma *Smith-Waterman*

Algoritma *Smith-Waterman* digunakan untuk mengidentifikasi kesamaan yang paling signifikan (*local alignment*) dari dua buah rangkaian sekuens string [9]. Prinsip yang digunakan dalam *dynamic programming* di algoritma ini adalah prinsip optimasi, yaitu setiap langkah yang diambil merupakan langkah yang paling maksimum, dalam hal ini membarikan skor yang terbesar dari segala kemungkinan langkah pembentuknya. Algoritma ini memiliki kompleksitas waktu O(*nm*) dimana *n* dan *m* merupakan panjang masing-masing sekuens gen yang dibandingkan [19]. Dalam mengidentifikasi kesamaan string digunakan konsep *reward* dan *punishment.* Jika terdapat kata yang sama (*match*) akan diberikan *reward* sedangkan pada kata yang tidak cocok (*mismatch*) dan gap akan diberikan *punishment.* Gap sendiri merupakan skor pasangan karakter yang berasal dari baris dan kolom yang berlainan [8]. Sesuai dengan aturan algoritma *Smith-Waterman* untuk pasangan karakter yang cocok (*match*) diberi skor positif, untuk pasangan karakter yang tidak cocok (*mismatch*) serta gap diberi skor negatif.

Terdapat prosedur algoritma *Smith-Waterman* yang digunakan dalam proses mengidentifikasi *local similarities* (penyejajaran sekuens) yaitu proses penyusunan dua *local sequence*. Berikut ini adalah cara kerja dari algoritma *Smith-Waterman* dalam menemukan kesamaan dari dua *sequence* [11]. *Sequence* ini dapat dilihat pada gambar II Inisialisasi matrks dengan kolom pertama sebagai *sequence* A dan beris pertama sebagai *sequence* B. Adapun tahapan algoritam sebagai berikut:

Pertama melakukan inisialiasi matriks pada dua *sequence* A dan B [20]. DNA *sequence* A = {ATGAC}, DNA *sequence* B = {ACGTA} dengan algoritma yang diusulkan akan mulai diisi dari posisi (1,1), entri pertama dibaris pertama. Selanjutnya mengisi matriks dengan skor yang sesuai, dua urutan diatur dalam bentuk matriks dengan menggunakan kolom A + 1 dan baris B + 1. Pada gambar II nilai dibaris pertama dan kolom pertama diatur ke 0.

	-	A	C	G	T	A
-	0	0	0	0	0	0
A	0					
T	0					
G	0					
A	0					
C	0					

Gambar II Inisialisasi Matriks

Pengisian matriks, jika DNA ‘i’ sama dengan DNA ‘j’ maka akan diberikan nilai (*match*) = +5. Sebaliknya apabila DNA dari kedua *sequence* berbeda maka akan digunakan nilai (*mismatch*) = -2 serta gap = -3.

Rumus pengisian matriks [20]:

$$M_{(i,j)} = \text{Max}\begin{cases} M_{i,j,\ j-1} + S_{i,j}, \\ M_{i,j-1} + W, \\ M_{i-1,j} + W, \\ 0 \end{cases} \quad (1)$$

Persamaan diatas memiliki keterangan yaitu $M_{i,j,\ j-1}$ adalah skor kesamaan pada *field* (String A ke-(i-1), string B ke-(j-1)), $S(_{i,j})$ adalah skor untuk *match* atau *mismatch* string yang dibandingkan, $M_{i,j-1}$ adalah skor kesamaan pada *field* (string A ke-(i), string B ke-(j-1)). $S(_{i,j})$ adalah skor untuk *gap* string yang dibandingkan dan $M_{i-1,j}$ adalah skor kesamaan pada *field* (string A ke-(i-1), untuk *gap* string yang dibandingkan)

	-	A	C	G	T	A
-	0	0	0	0	0	0
A	0	5				
T	0					
G	0					
A	0					
C	0					

Gambar III Pengisian Matriks

Contoh ilustrasi dari rumus diatas.

$$M_{(1,1)} = \text{Max}\left[M_{(0,0)} + S_{(1,1)}, M_{(1,0)} + W,\ M_{(0,1)} + W, 0\right]$$

$$\text{Max}\ [0 + (5), 0 + (-3), 0 + (-3), 0]$$

$$\text{Max}\ [5, -3, -3, 0]$$

Hasil max = 5 (2)

Gambar III merupakan hasil dari ilustrasi persamaan (2) untuk mencari nilai keseluruhan dari tiap *cell* pada tabel matriks. Setelah semua nilai ditemukan maka proses selanjunya adalah melakukan *traceback*. *Traceback* merupakan proses untuk melacak kembali urutan untuk perataan yang sesuai, langkah terakhir untuk penyelarasan yang tepat dalam pemerataan ini adalah penyelurusan jejak. setelah melakukan perhitungan pengisian matriks kemudian dilakukan *traceback* dari nilai tertinggi dalam matriks. skor maksimum dalam matriks pada gambar IV adalah 10. Jadi, *traceback* dimulai dari nilai *cell* yang memiliki nilai tertinggi.

	-	A	C	G	T	A
-	0	0	0	0	0	0
A	0	5	2	0	0	5
T	0	2	3	0	5	2
G	0	0	0	8	5	3
A	0	5	2	5	6	10
C	0	2	10	7	4	7

Gambar IV Hasil *Traceback*

Hasil *traceback* gambar IV menunjukkan matriks kesamaan dari perbandingan *sequence* diatas, dari 2 *sequence* tersebut akan dilakukan perhitungan kemiripan untuk melihat seberapa persen kemiripan pada dokumen yang diuji. *Local similarity* dari kedua DNA tersebut adalah:

DNA *sequence* A = ATG-A

DNA *sequence* B = ACGTA

E. Perhitungan *Similarity*

Perhitungan *Similarity* dari proses perbandingan dokumen akan didapatkan informasi tentang kalimat yang dianggap memiliki kesamaan. Token atau kata yang sama pada tiap-tiap kalimat akan diakumulasi seberapa besar kesamaan antara dokumen yang dibandingkan [9].

Rumus untuk menentukan kemiripan:

$$\frac{\left(\frac{\text{Jumlah kata sama } (match)}{\text{Jumlah kata bersih } (clean)} \times 100\%\right)}{2} \quad (3)$$

Dengan persamaan no. 3 menghasilkan *similarity.* Dari hasil akhir *traceback* tedapat 8 karaker yang sama dari total karakter masing-masing dokumen adalah 5 karakter. Adapun hasil perhitungannya adalah (((8/5) x 100)/2) maka dihasilkan nilai DNA hasil *traceback* sebesar 80 persen.

IV. Hasil dan diskusi

Pada pengujian sistem ini dirancang untuk mendeteksi plagiat pada dokumen teks. Data dokumen diinputkan satu dokumen asli yang dibandingkan dengan dokumen uji yang dipilih dan menampilkan *similarity* dari hasil perbandingan yang dilakukan.

A. Skenario Hasil Pengujian

Pengujian yang dilakukan dalam penelitian ini yaitu membandingkan nilai akurasi sistem dengan menilai persamaan dari dua dokumen atau lebih dengan menggunakan metode *Smith-Waterman*. Dalam proses perbandingan akan dilihat ouput hasil sistem dengan penerapan data yang dihasilkan dari tahapan *preprocessing* dengan data yang tidak melalui tahapan *preprocessing.* Perbandingan itu juga akan dilihat perbedaan waktu yang dibutuhkan sistem dalam mengeluarkan nilai akurasi dengan data yang dilakukan tahapan *preprocessing* dengan data yang tidak melalui tahapan *preprocessing.* Dokumen asli yang digunakan merupakan karya ilmiah

yang telah melakukan pemotongan separuh dari dokumen tersebut terdapat abstrak dan pendahuluan yang diambil sebagai data asli. Dokumen uji yang digunakan merupakan dokumen asli yang telah melalui beberapa perubahan, seperti penambahan kata dan juga perubahan kata pasif ke aktif dan perubahan truktur kalimat.

Tabel II merupakan pengujian setiap data asli dan data uji pada algoritma *Smith-Waterman* dan *Preprocessing*. Tabel III adalah pengujian waktu yang dibutuhkan untuk memproses algoritma *Smith-Waterman* dan *preprocessing,* dengan setiap percobaan pada dokumen asli dan dokumen uji yang terdapat pada tabel II. Tabel IV merupakah pengujian setiap dokumen asli dan dokumen uji pada algoritma *Smith-Waterman* dan tanpa membutuhkan pengujian *preprocessing*.

Tabel V adalah pengujian waktu yang dibutuhkan untuk memproses algoritma *Smith-Waterman,* dengan setiap percobaan pada dokumen asli dan dakumen uji yang terdapat pada tabel IV.

Tabel II Pengujian Dengan Algoritma *Smith-Waterman* dan *Preprocessing*

Data Hasil Pengujian Dengan Algoritma *Smith-Waterman* dan *Preprocessing*						
No.	Dokumen	Asli 1	Asli 2	Asli 3	Asli 4	Asli 5
1.	Uji 1	71.34	83.39	49.10	73.91	89.34
2.	Uji 2	82.40	38.28	51.97	26.07	7.04
3.	Uji 3	55.46	36.87	83.45	31.45	66.77
4.	Uji 4	22.27	40.67	8.49	39.07	93.52
5.	Uji 5	51.80	2.38	61.64	88.71	72.88
Rata-rata hasil uji		56.65	40.32	50.93	51.48	65.91

Tabel II dan Tabel IV dihasilkan dengan menggunakan persamaan (3). Dengan menggunakan data yaitu berupa karya ilmiah yang telah melakukan pemotongan dengan hanya mengambil abstraksi dan pendahuluan. Adapun ilustrasi perhitunganya adalah sebagai berikut yaitu dengan membandingkan dua dokumen dengan melihat jumlah kata yang sama dan jumlah kata bersih.

$$\frac{\left(\frac{361 + 411}{541} \times 100\%\right)}{2} = 71.34$$

Tabel III Rata-Rata Nilai Pengujian Waktu dengan Algoritma *Smith-Waterman* dan *Preprocessing*

No.	Rata-Rata Nilai Pengujian Waktu dengan Algoritma *Smith-Waterman* dan *Preprocessing*	
1.	Percobaan 1	52,287 detik
2.	Percobaan 2	47,905 detik
3.	Percobaan 3	50,058 detik
4.	Percobaan 4	50,008 detik

Tabel IV Pengujian Algoritma *Smith-Waterman*

Data Hasil Pengujian Algoritma *Smith-Waterman*						
No.	Dokumen	Asli 1	Asli 2	Asli 3	Asli 4	Asli 5
1.	Uji 1	69.98	81.96	44.89	72.05	88.04
2.	Uji 2	77.74	37.81	50.60	25.98	5.59
3.	Uji 3	53.05	36.24	79.84	28.69	66.13
4.	Uji 4	21.19	37.88	4.62	38.15	92.89
5.	Uji 5	46.90	2.38	61.30	84.27	70.64
Rata-rata hasil uji		53,77	39.25	48.25	49.83	64.66

Tabel V Rata-Rata Nilai Pengujian Waktu dengan Algoritma *Smith-Waterman*

No.	Rata-Rata Nilai Pengujian Waktu dengan Algoritma *Smith-Waterman*	
1.	Percobaan 1	16,386 detik
2.	Percobaan 2	15,392 detik
3.	Percobaan 3	14,986 detik
4.	Percobaan 4	15,428 detik

B. Hasil Pengujian

Setelah dilakukan pengujian berdasarkan skenario pengujian yang telah dibuat didapatkan hasil pengujian yaitu nilai akurasi dengan menggunakan algoritma *Smith-Waterman* dan *preprocessing* menghasilkan nilai akurasi yang lebih tinggi serta waktu yang dibutuhkan oleh sistem dalam proses mengeluarkan nilai akurasi lebih besar dikarenakan sistem harus melewati tahapan-tahapan *preprocessing* seperti *tokenization, stopword removal,* algoritma *stemming* (*Nazief & Andriani*). Perbandingan ini dapat dilihat pada tabel II, tabel III, tabel IV dan tabel V.

C. Analisis Hasil Pengujian

Berdasarkan setiap skenario pengujian yang menggunakan algoritma *Smith-Waterman* dengan bantuan proses *preprocessing* pada Tabel II dan Tabel III, dapat diketahui bahwa pengujian memiliki keakuratan yang lebih baik pada saat membandingkan dokumen. Keakuratan yang lebih baik tersebut diperoleh dengan waktu proses yang lebih lama dibandingkan pengujian tanpa bantuan proses *preprocessing* yang bisa dilihat pada Tabel III dan Tabel V. Begitu juga pada Tabel IV dan Tabel V yang menggunakan algoritma *Smith-Waterman*. Perbedaan hanya pada tingkat akurasi yang kurang baik dalam membandingkan dokumen. Penggunaan algortima *Smith-Waterman* dengan bantuan proses *preprocessing* menghasilkan tingkat akurasi yang lebih baik.

Hasil akurasi dari algoritma *Smith-Waterman* dipengaruhi oleh adanya data berupa kata dasar dalam *database* sistem. Dimana pada *database* sistem tersebut jumlah katanya dipengaruhi oleh dokumen yang digunakan oleh user. Selain itu faktor lain yang bekerja pada sistem ialah faktor yang berpengaruh pada minimalisasi waktu yaitu dikarenakan adanya proses *stopword.* Faktor-faktor yang telah disebutkan tersebut merupakan faktor penentu akurasi sistem yaitu merupakan hasil dari *preprocessing* yang digunakan dalam sistem yang menerapkan dari algoritma *Smith-Waterman.*

V. KESIMPULAN

Dari hasil pengujian yang dilakukan pada skenario yang dibuat, didapatkan bahwa perbandingan dua *sequence* yang diuji dengan menggunakan *preprocessing* mendapatkan nilai perhitungan *similarity* dengan tingkat kemiripan lebih besar dibandingkan dengan tanpa *preprocessing.* Hal tersebut disebabkan karena penghilangan *noise* pada dokumen dapat meningkatkan nilai *similarity* tingkat kemiripan pada dokumen. Selain itu, pengujian menggunakan *preprocessing* membutuhkan waktu yang lebih lama dibandingkan tanpa *preprocessing,* karena terdapat proses *tokenization, stopword removal* dan algoritma *stemming Nazief & Andriani* yang dilakukan pada suatu dokumen. Algoritma *Nazief & Andriani* membantu dalam proses *preprocessing* (*stemming*) yang mendukung algoritma *Smith-waterman* dalam proses deteksi kemiripan dokumen teks. Dari rata-rata pengujian pada dokumen asli dan dokumen uji menunjukkan tingkat kemiripan yang dihasilkan lebih dari 50%, yang berarti hasil tersebut dapat dinyatakan terjadi tindak plagiarisme.

PUSTAKA

[1] Adimihardja, "Plagiarisme. Makalah Disampaikan dalam Lokakarya Etika di Perguruan Tinggi yang Dilaksanakan di Medan pada Tanggal 19—20 April 2005.Fakultas Ekonomi Universitas Sumatera Utara. Medan.," vol. 24, 2005.

[2] H. Oktaviani, "Sistem Deteksi Kemiripan Antra Dokumen Teks Menggunakan Algoritma SCAM".

[3] F. B. Djafar, A. Lahinta and H. Lillyan , "Penerapan Algoritma *Smith-Waterman* Dalam Sistem Pendeteksi Kesamaan Dokumen".

[4] Helmy Darmawan1, "Analisis dan Implementasi Algoritma *Smith-Waterman* pada Proses Identifikasi Kesamaan Dokumen," pp. 1-7, 2009.

[5] Wibowo, R. Karisma, H. and K. , "Penerapan Algoritma Winnowing Untuk Mendeteksi Kemiripan Teks Pada Tugas Akhir Mahasiswa," *Techno.COM,* vol. 15, pp. 303-311, November 2016.

[6] A. Firdaus, Ernawati and A. Vatresia, "Aplikasi Pendeteksi Kemiripan Pada Dokumen Teks Menggunakan Algoritma *Nazief & Andriani* Cosine Similarity," vol. 10 Nomor 1, April 2014.

[7] D. Wahyudi, T. Susyanto and D. Nugroho, "Implementasi dan Analisis Algoritma Stemming *Nazief & Andriani* dan Algoritma Porter Pada Dokumen Berbahasa Indonesia," *Jurnal Ilmiah SINUS,* no. ISSN (Print) : 1693-1173 ISSN (Online) : 2548-4028, July 2017.

[8] R. W. Irving, "Plagiarism and Collusion Detection Using the *Smith-Waterman* Algorithm," pp. 1-22.

[9] H. Darmawan, "Analisis dan Implementasi Algoritma *Smith-Waterman* pada Proses Identifikasi Kesamaan Dokumen," pp. 1-7, 2009.

[10] M. Y. Rimaldo and . S. , "Sistem Pendeteksi Plagiat Dokumen Teks Menggunakan Algoritma *Smith-Waterman* Serta Algoritma Arifin dan Setiono," 2012.

[11] F. B. Djafar, A. Lahinta and L. Hadjaratie, "Penerapan Algoritma *Smith-Waterman* Dalam Sistem Pendeteksi Kesamaan Dokumen".

[12] A. Novanta, "Pendeteksian Plagiarisme Pada Dokumen Teks Dengan Menggunakan Algoritma *Smith-Waterman*," Vols. ISSN 1414-9999, no. 2009, 2009.

[13] Z. Su, B. R. Ahn, K. y. Eom, M. k. Kang, J. P. Kim and M. K. Kim, "Plagiarism Detection Using the Levenshtein Distance and *Smith-Waterman* Algorithm," *Intetnational Conference on Innovative Computing Information and Control,* Vols. 978-0-7695-3161-8/08, 2008.

[14] E. R. S, "Peningkatan Kinerja Algorit- ma *Smith-Waterman* untuk Pendeteksian Plagiarisme pada Dokumen Teks," 2010.

[15] L. Agusta, "Perbandingan Algoritma Stemming Porter Dengan Algoritma *Nazief & Andriani* Untuk Stemming Dokumen Teks Bahasa Indonesia," *Konferensi Nasional Sistem dan Informatika,* no. KNS&I09-036, 2009.

[16] M. E. B. Menai, "Detection of Plagiarism in Arabic Documents," *I.J. Information Technology and Computer Science,* no. september , pp. 80-89, 2012.

[17] C. F. Zdenek Ceska, "The Influence of Text Pre-processing on Plagiarism Detection," *International Conference RANLP,* p. 55–59, 2009.

[18] F. Z. Tala, "A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia," *Institute for Logic, Language and Computation Universiteit van Amsterdam The Netherlands.*

[19] Agarwal, M. Vaidya and P. K, "Local Alignment and Substitution Matrices," *CPS260/BGT204.1 Algorithms in Computational Biology,* 2003.

[20] D. B. C and N. V, "Parallel *Smith-Waterman* Algorithm for Gene Sequencing," *International Journal on Recent and Innovation Trends in Computing and Communication,* no. ISSN: 2321-8169, pp. 3237 - 3240, 2015.

[21] Kusumawati, F. T. and R. , "Pembuatan Program Aplikasi untuk Pendeteksian Kemiripan Dokumen Teks dengan Algoritma *Smith-Waterman*".

[22] T. Smith and M. Waterman, "Identification of common molecular subsequences.," *Journal of Molecular Biology,* p. 147:195 – 197, 1981.