Aplikasi Berbasis Web Pendeteksi Plagiarisme Menggunakan Algoritma Himpunan Kata

Ismail[1], Eka Widhi Yunarso[2]
[1]Teknik Komputer, [2]Manajemen Informatika, Fakultas Ilmu Terapan, Universitas Telkom
Jl. Telekomunikasi, Ters. Buah Batu, Bandung, 40257, Indonesia
[1]ismailrusli@telkomuniversity.ac.id, [2]ekawidhi@telkomuniversity.ac.id

Abstrak – Plagiarisme adalah penggunaan tulisan orang lain dalam tulisan sendiri tanpa memberikan kredit terhadap penulis asli. Tindakan ini membuat pembaca menganggap tulisan tersebut asli dan bukan kutipan. Plagiarisme melanggar kode etik dalam karya ilmiah. Untuk itu, adanya aplikasi yang membantu mendeteksi potensi plagiarisme sebuah tulisan sangatlah penting. Paper ini menguraikan hasil implementasi aplikasi berbasis web pendeteksi plagiarisme menggunakan algoritma berbasis himpunan kata dari Küppers. Oleh karena tidak adanya *corpus* Bahasa Indonesia untuk evaluasi algoritma pendeteksi plagiarisme, pengujian tidak dievaluasi menggunakan penilaian standar, seperti *recall*, *precision*, dan *granularity*. Hasil pengujian ditunjukkan dengan nilai yang menjadi ukuran kemiripan dokumen yang diuji dengan dokumen lain yang ada di basisdata. Secara rata-rata, didapatkan nilai potensi sebesar 0,8 untuk paragraf-paragraf yang sama persis, 0,6 untuk paragraf yang merupakan hasil pengubahan kata atau tanda baca, dan 0,4 untuk paragraf-paragraf yang merupakan hasil penulisan ulang.

Kata kunci – plagiarisme, aplikasi web, algoritma himpunan kata

Abstract— Plagiarism is the act of using other's text in someone's writing without giving credit to the original author. This act misleads reader to think that the text is original instead of reused. This infringes the ethical code especially in scientific articles. Hence, the existence of application to detect plagiarism potential in a writing is a great help to community. This paper describes our research in implementing web-based application for plagiarism detection using set-based approach from Küppers. Because there are no evaluation corpus for plagiarism detection in Bahasa Indonesia, we could not evaluate our implementation in standard evaluations, such as using recall, precision, and granularity. Results of experiments are shown in term of values represent the similarity degree of suspected documents and stored documents. In average, our application measures potential value of 0.8 for plagiarism by copy-pasted paragraphs, 0.6 for plagiarism by adding/replacing words/punctuations, and 0.4 for plagiarism by rephrasing paragraphs.

Keywords: plagiarism, web-based application, set-based algorithm

I. PENDAHULUAN

Plagiarisme tidak diperkenankan dalam dunia akademik. Selain itu, plagiarisme bertentangan dengan sifat jujur yang dibutuhkan di dunia ilmiah maupun akademik. Tanpa sifat jujur, ilmu pengetahuan tidak berkembang seperti sekarang. Akan tetapi, masih ada orang yang bergerak di bidang ilmu pengetahuan, baik sebagai peneliti maupun akademisi, yang melakukan tindakan plagiarisme. Hal ini dipicu salah satunya karena kurangnya kemampuan menulis artikel ilmiah.

Menentukan plagiarisme dalam sebuah tulisan tidaklah mudah. Dokumen pembanding sangatlah banyak. Apalagi di era Internet dewasa ini. Artikel banyak dituliskan di dunia maya. Untuk itu, dibutuhkan aplikasi yang dapat membantu mendeteksi potensi plagiarisme sebuah tulisan.

Paper ini menguraikan hasil implementasi aplikasi berbasis web pendeteksi plagiarisme. Implementasi dibatasi hanya untuk pengujian lokal. Artinya, dokumen pembanding yang akan dijadikan acuan adalah dokumen yang ada di basisdata aplikasi dan bukan dokumen yang ada di Internet.

Kontribusi yang diberikan penelitian ini adalah modifikasi algoritma, yaitu algoritma berbasis himpunan kata dari Küppers [1]. Küppers menggunakan *chunk* sebesar 250 karakter sementara dalam penelitian ini digunakan *chunk* yang diusahakan setara dengan paragraf.

Untuk menguji kinerja algoritma yang digunakan, dibutuhkan *corpus* Bahasa Indonesia. C*orpus* seperti itu belum pernah ada. Sebagai gantinya, ditunjukkan penilaian algoritma terhadap *chunk* dokumen yang dianggap mirip dengan *chunk* yang berada di basisdata. Untuk pengujian, dilakukan simulasi plagiarisme secara manual, yaitu dengan menyalin utuh paragraf, mengubah kata atau tanda baca dalam paragraf, dan melakukan penulisan ulang (*rephrase*).

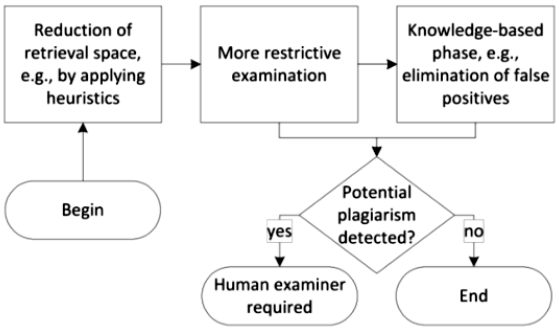
Pembahasan dalam paper ini akan mengikuti alur sebagai berikut. Di bagian II diuraikan metode penelitian termasuk di dalamnya uraian mengenai penelitian-penelitian lain yang sudah dilakukan dalam

bidang deteksi plagiarisme. Selanjutnya di Bagian III, diuraikan hasi pengujian yang akan dibahas di bagian IV. Kesimpulan dari penelitian ini akan diuraikan di bagian V.

II. METODOLOGI

A. Algoritma yang digunakan

Secara umum, pendeteksi plagiarisme otomatis memiliki tiga tahap pengujian dokumen. Gambar 1 memperlihatkan tahapan-tahapan tersebut.



Gambar 1. Langkah-langkah dalam pendeteksian plagiarisme secara otomatis [2]

Dari Gambar 1 terlihat bahwa hingga saat ini, campur tangan manusia masih diperlukan untuk mendeteksi suatu dokumen. Dengan demikian, dapat dikatakan bahwa sistem pendeteksi plagiarisme oleh komputer hanya bersifat semi-otomatis.

Tahap pertama dalam deteksi plagiarisme adalah pemilihan dokumen yang diduga sumber plagiarisme. Tahap ini diperlukan karena jumlah dokumen di basisdata sangat banyak sehingga tidak praktis untuk dilakukan pemeriksaan terhadap semua dokumen.

Menurut Potthast [3], langkah yang dilakukan dalam tahap pertama ini adalah *chunking*, *keyphrase extraction*, *query formulation*, *search control*, dan *download filtering*. Langkah tersebut dilakukan terhadap proses pengambilan dokumen dari Internet menggunakan mesin pencari. Oleh karena dalam penelitian ini tidak dilakukan pencarian di Internet, proses yang dilakukan hanya berlangsung hingga tahap dua.

Beberapa strategi untuk *chunking* adalah *no-chunking*, 50-*line chunk*, TextTiling, 4-*sentence chunking*, *paragraph chunking*, 100-*words chunking*, 5-*sentence chunks*, dan kombinasi dari strategi-strategi tersebut [3].

Dalam penelitian ini, digunakan *paragraph chunking*. Diasumsikan plagiarisme biasanya dilakukan dalam satuan paragraf. Selain itu, implementasi *paragraph chunking* relatif lebih mudah.

Sebelum masuk ke proses selanjutnya, paragraf dibersihkan dari karakter-karakter *non-alphabet* (kecuali spasi) karena karakter-karakter tersebut tidak esensial terhadap proses deteksi plagiarisme.

Tahap selanjutnya setelah *chunking* adalah *keyphrase extraction* atau pemilihan kata/frasa kunci. Salah satu proses yang biasanya dilakukan dalam tahap ini adalah penghilangan *stopwords*. Selanjutnya, kata/frasa kunci biasanya diambil dari dokumen setelah menghitung frekuensi dari setiap kata/frasa dan mengurutkannya berdasarka nilai *tf.idf*.

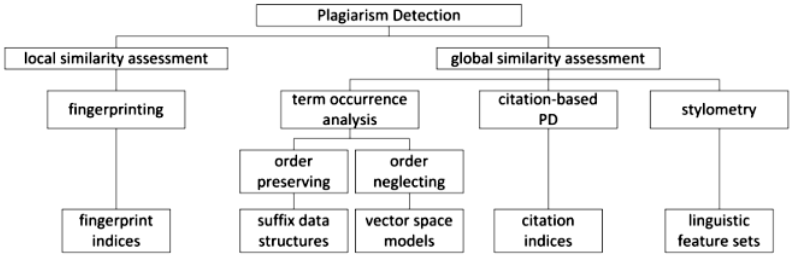
Dalam penelitian ini, digunakan metode sederhana, yaitu mengumpulkan kata dalam sebuah dokumen (setelah menghilangkan *stopwords*) serta mengurut-kannya berdasarkan frekuensi. Setengah dari kumpulan kata ini disimpan di basisdata sebagai "sidik jari" dari dokumen tersebut.

Selengkapnya, proses pemilihan dokumen dalam aplikasi adalah sebagai berikut.

1. Ekstraksi paragraf terhadap dokumen PDF yang dimasukkan ke dalam aplikasi.
2. Pembersihan karakter *non-alphabet* (kecuali spasi pemisah antarkata).
3. Penghitungan dan pengurutan berdasarkan frekuensi kata.
4. Penyimpanan di basisdata setengah dari himpunan kata dengan frekuensi terbanyak dari tahap 3.

Penentuan dokumen yang terpilih dalam tahapan satu ini adalah dengan menghitung fraksi antara jumlah kata yang sama antara dokumen yang diuji dengan dokumen yang ada di basisdata dibagi dengan jumlah kata unik dari dokumen yang diuji. Dengan ambang tertentu, dapat diambil dokumen yang diduga merupakan sumber plagiarisme dari dokumen yang diuji.

Di tahap 2, atau tahap analisis detail, metode-metode yang digunakan dapat dilihat di Gambar 2. Metode ini secara garis besar dapat dibagi menjadi deteksi berdasarkan pengujian kesaamaan lokal dan pengujian kesamaan global.



Gambar 2. Metode dalam pengujian potensi plagiarisme suatu dokumen [2]

Metode yang digunakan dalam penelitian ini adalah metode yang termasuk ke dalam *fingerprinting*. Metode ini diajukan oleh Küppers [1]. Menurut Zu Eissen [4], *fingerprinting* adalah metode yang saat ini paling banyak digunakan.

Küppers membagi dokumen ke dalam *chunk* 250 karakter (atau hingga batas kata). Dengan demikian, satu dokumen yang diuji memiliki *n chunk*. Selanjutnya, *n chunk* ini dibandingkan dengan *m chunk* dari setiap dokumen yang ada di basisdata.

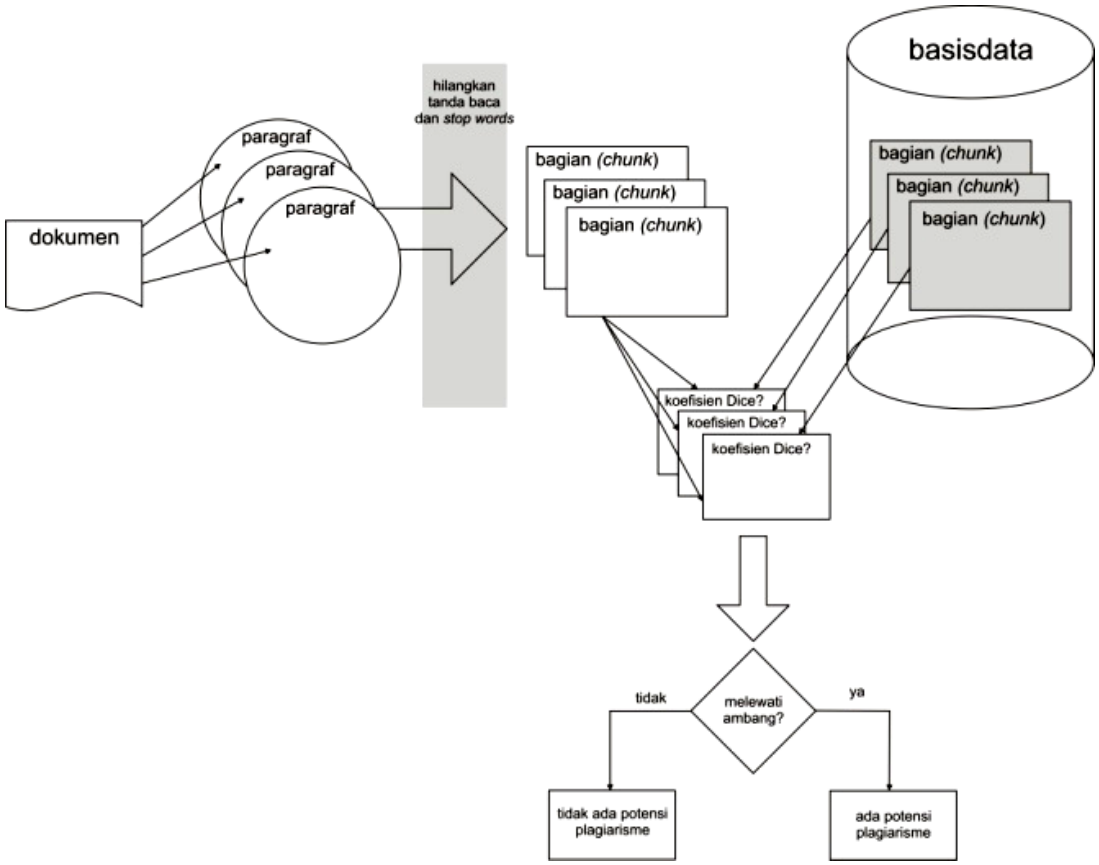
Satu *chunk* dalam dokumen dianggap sebagai himpunan kata. Himpunan kata ini dibandingkan dengan himpunan kata dari *chunk* lain dari seluruh dokumen di basisdata. Tingkat kesamaan diukur menggunakan koefisien Dice, yaitu sebagai berikut.

$$\frac{2 \times |A \bigcap B|}{|A| + |B|} \quad \text{..... (1)}$$

dengan A dan B adalah *chunk* dari dokumen yang diuji dan *chunk* dari dokumen yang ada di basisdata.

Meskipun sederhana, metode yang diajukan Küppers berada di peringkat ke-7 dalam kompetisi deteksi plagiarisme internasional PAN 2012 untuk tahap analisis detail [5].

Selengkapnya, metode yang digunakan dalam tahap analisis detail dapat dilihat di Gambar 3.



Gambar 3. Metode yang digunakan dalam penelitian ini

B. Skenario Pengujian

Dalam penelitian ini, digunakan 1554 dokumen yang diambil dari Internet. Dokumen ini berbahasa Indonesia dan sebagian besar dalam bentuk jurnal. Dalam penelitian ini, tidak digunakan *corpus* untuk melakukan evaluasi terhadap akurasi algoritma deteksi yang digunakan. Hal ini karena memang *corpus* tersebut belum pernah ada. Dengan demikian, dalam pengujian, tidak dapat diukur nilai-nilai seperti *precision*, *recall*, dan *granularity*. Sebagai gantinya, digunakan skenario pengujian sebagai berikut.

Hal pertama yang diuji adalah waktu yang dibutuhkan oleh aplikasi untuk melakukan proses deteksi plagiarisme. Dari 1554 dokumen yang ada di basisdata, dilakukan *cross-check* di antara dokumen itu sendiri. Dengan kata lain, dilakukan cek terhadap dokumen yang sudah berada di basisdata. Aplikasi tidak melakukan pengujian terhadap dokumen yang pernah diupload sehingga pengecekan ini seharusnya akan memberikan hasil nol kecuali ada indikasi plagiarisme di antara teks yang terdapat dalam 1554 dokumen tersebut.

Dalam pengujian ini diukur kecepatan eksekusi fungsi cek plagiarisme dari aplikasi dibandingkan terhadap ukuran *file* yang dicek dan juga jumlah paragraf dalam *file*.

Pengujian yang kedua adalah dengan membuat secara manual beberapa paragraf hasil plagiarisme dari

1554 dokumen yang ada di basisdata. Proses simulasi plagiarisme ini dilakukan dengan 3 cara, yaitu

1. Menyalin secara persis kata per kata dan juga tanda baca (*copy and paste*)
2. Menghapus, mengubah, atau menambah kata atau tanda baca dengan padanannya.
3. Melakukan penulisan ulang (*rephrase*) sebuah paragraf dengan mempertahankan arti dari paragraf tersebut.

Hasil pengujian ditunjukkan dengan nilai yang diberikan aplikasi terhadap tingkat kemiripan paragraf hasil simulasi ini dengan paragraf lain yang ada di basisdata.

C. Perangkat Kerasa dan Lunak

Aplikasi web ini diimplementasikan dalam bahasa pemrograman Python dan framework Django. Spesifikasi perangkat keras dan lunak yang digunakan dapat dilihat di Tabel 1.

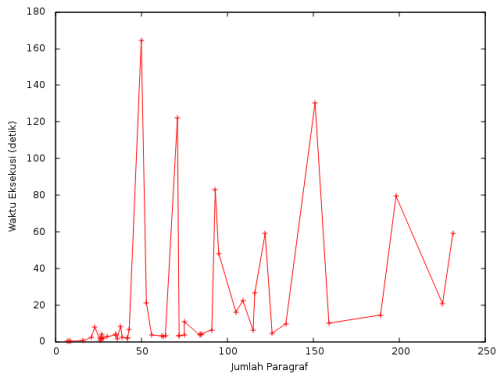
Tabel 1. Spesifikasi perangkat keras/lunak

No	Perangkat Keras/Lunak	Spesifikasi
1.	Prosesor	Intel Core i5-3330 3GHz
2.	Memori	2 × 8192MB DDR3 1600MHz
3.	Hard disk	4TB SATA 6GB/s
4.	Sistem Operasi	ArchLinux dengan kernel 3.16.1.1-ARCH 64 bit
5.	Bahasa pemrograman	Python 2.7
6.	Framework	Django Web Framework 1.6.6
7.	PDF library	Pdfminer 20140328-1
8.	Basisdata	PostgreSQL 9.3.5-1

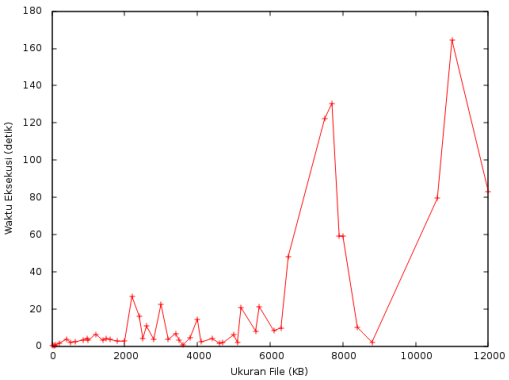
III. HASIL

A. Pengujian Kecepatan Eksekusi

Untuk pengujian, diambil sampel sebanyak 50 dokumen dari seluruh dokumen yang ada di basisdata. Selanjutnya, dilakukan proses cek potensi plagiarisme dari 50 dokumen tersebut melalui aplikasi web. Gambar 4 dan Gambar 5 menunjukkan grafik dari hasil pengujian kecepatan eksekusi deteksi yang dilakukan.



Gambar 4. Hasil pengujian kecepatan deteksi terhadap jumlah paragraf dalam dokumen yang diuji



Gambar 5. Hasil pengujian kecepatan deteksi terhadap ukuran file dari dalam dokumen yang diuji

Perlu dicatat di sini semua dokumen yang diambil sebagai sampel ternyata bersih dari potensi plagiarisme berdasarkan algoritma yang digunakan.

B. Pengujian Deteksi Plagiarisme terhadap Paragraf yang Sama Persis

Selanjutnya, dilakukan pengujian deteksi plagiarisme dengan cara membuat suatu dokumen yang berisi satu paragraf yang diambil persis dari paragraf yang ada di basisdata. Paragraf tersebut adalah sebagai berikut:

"*Kebutuhan pengendalian daya telah ada sejak lama. Sebelum ditemukan thyristor, pengendalian daya listrik* menggunakan *generator induksi, tetapi alat ini mempunyai beberapa kelemahan antara lain efisiensi yang rendah, harga yang mahal, ukurannya besar dan perawatan yang tidak mudah. Saat ini pengendalian daya menggunakan penyearah thyristor fasa terkendali yang merupakan penyearah sederhana dan lebih murah. Efisiensi dari penyearah ini umumnya berada diatas 95%. Penyearah ini dikenal sebagai konverter AC DC yang mengkonversi dari tegangan AC ke DC dan digunakan secara intensif pada aplikasi-aplikasi industri.*"

Aplikasi menemukan paragraf di basisdata dengan nilai koefisien Dice 0.969696969697 (paragraf ini

adalah paragraf asli yang tersimpan di basisdata). Paragraf tersebut adalah sebagai berikut.

"*1.1. Latar Belakang dan Permasalahan Kebutuhan pengendalian daya telah ada sejak lama. Sebelum ditemukan listrik menggunakan generator induksi, tetapi alat ini* mempunyai *beberapa kelemahan antara lain efisiensi yang rendah, harga yang mahal, ukurannya besar dan perawatan yang tidak mudah. Saat ini pengendalian daya menggunakan penyearah thyristor fasa terkendali yang merupakan penyearah sederhana dan lebih murah. Efisiensi dari penyearah ini umumnya berada diatas 95%. Penyearah ini dikenal sebagai konverter AC DC yang mengkonversi dari tegangan AC ke DC dan digunakan secara intensif pada aplikasi-aplikasi industri.*"

Aplikasi juga menemukan paragraf di basisdata dengan nilai koefisien Dice 0.289855072464. Paragraf tersebut adalah sebagai berikut.

"*PENDAHULUAN Saat ini, Motor induksi 3 phase sering digunakan berbagai aplikasi di dunia industri. Motor induksi 3 phase memiliki keunggulan diantaranya handal, tidak ada kontak antara stator dan rotor kecuali bearing, tenaga yang besar, daya listrik rendah dan* hampir *tidak ada perawatan. Akan tetapi motor induksi 3 phase memiliki kelemahan pada pengontrolan kecepatan. Kecepatan putar motor induksi bergantung pada frekuensi input, sedangkan sumber listrik memiliki frekuensi konstan. Untuk mengubah sulit daripada mengatur tegangan input. Dengan ditemukannya teknologi inverter maka hal tersebut menjadi lebih mudah dan mungkin dilakukan.*"

Untuk pengujian ini, aplikasi menampilkan 450 paragraf lain yang "mirip", dengan koefisien Dice antara 0.969696969697 hingga 0.10071942446.

C. Pengujian Deteksi Plagiarisme terhadap Paragraf yang Diubah Kata atau Tanda Baca

Paragraf yang diuji adalah hasil pengubahan paragraf pada bagian B sebelumnya dengan cara ditambah/diubah/dikurangi kata atau tanda bacanya. Paragraf yang diuji menjadi seperti berikut.

"*Kebutuhan pengendalian tenaga telah ada dari dulu. Sebelum ada thyristor, pengendalian tenaga listrik memanfaatkan generator induksi. Akan tetapi, alat ini mempunyai kekurangan seperti efisiensi yang rendah, mahal, ukurannya besar, dan perawatan yang sulit. Saat ini, pengendalian tenaga memanfaatkan penyearah thyristor fasa terkendali yang lebih sederhana dan murah. Efisiensi penyearah ini di atas 95%. Penyearah ini dikenal sebagai pengubah AC-DC yang mengubah tegangan Alternating Current ke Direct Current dan dipakai intensif di banyak industri.*"

Aplikasi menemukan paragraf di basisdata dengan koefisien Dice sebesar 0.686567164179 yang merupakan paragraf asli.

Selanjutnya, aplikasi juga menampilkan paragraf dengan koefisien Dice 0.276923076923, yaitu paragraf berikut.

"*Motor induksi tiga fasa saat ini sering digunakan pada industri dengan berbagai aplikasi. Hal ini disebabkan karena motor tiga fasa memiliki beberapa keunggulan diantaranya tidak ada kontak antara stator dan rotor kecuali bearing, tenaga yang besar, daya listrik rendah* dan *hampir tidak ada perawatan, tetapi memiliki beberapa kelemahan diantaranya pengontrolan kecepatan hanya bergantung pada frekwensi input sedangkan sumber yang ada memiliki frekwensi yang konstan. Untuk mengubah frekwensi input lebih sulit dibanding dengan mengubah ditemukannya teknologi inverter maka hal tersebut menjadi mungkin untuk dilakukan.*"

Untuk pengujian ini, aplikasi menampilkan 358 paragraf lain yang "mirip", dengan koefisien Dice antara 0.686567164179 hingga 0.101265822785.

D. Pengujian Deteksi Plagiarisme terhadap Paragraf Hasil Penulisan Ulang (*Rephrase*)

Terakhir, dilakukan penulisan ulang terhadap paragraf yang sama yang digunakan di pengujian sebelumnya. Berikut adalah paragraf yang diuji hasil penulisan ulang tersebut.

"*Sudah dari dulu kebutuhan akan pengendalian daya ada. Dulu biasanya digunakan generator induksi. Akan tetapi, alat ini mahal dengan tingkat efisiensi yang rendah. Selain itu, alat ini memiliki ukuran yang besar dan perawatan yang tidak gampang. Sejak ditemukannya thyristor, pengendalian daya listrik dapat dilakukan secara efisien dan murah. Thyristor sekarang merupakan pengendali daya standar di industri.*"

Aplikasi menemukan paragraf di basisdata dengan koefisien Dice sebesar 0.5 yang merupakan paragraf yang asli.

Selanjutnya, aplikasi juga menampilkan paragraf dengan koefisien Dice 0.296296296296, yaitu paragraf berikut.

"*Motor induksi tiga fasa saat ini sering digunakan pada industri dengan berbagai aplikasi. Hal ini disebabkan karena motor tiga fasa memiliki beberapa keunggulan diantaranya tidak ada kontak antara stator dan rotor kecuali bearing, tenaga yang besar, daya listrik rendah dan hampir tidak ada perawatan, tetapi memiliki beberapa kelemahan diantaranya pengontrolan kecepatan hanya bergantung pada frekwensi input sedangkan sumber yang ada memiliki frekwensi yang konstan. Untuk mengubah frekwensi input lebih sulit dibanding dengan mengubah ditemukannya teknologi inverter maka hal tersebut menjadi mungkin untuk dilakukan*."

Untuk pengujian ini, aplikasi menampilkan 482 paragraf lain yang "mirip", dengan koefisien Dice antara 0.5 hingga 0.101694915254.

E. Rangkuman Pengujian Deteksi Plagiarisme

Dalam penelitian ini, dilakukan juga pengujian yang sama seperti pengujian sebelumnya terhadap 4 paragraf lain. Nilai yang didapat, selanjutnya dirata-ratakan. Tabel 2 menunjukkan hasil yang didapat.

Tabel 2. Hasil pengujian deteksi plagiarisme

No	Paragraf yang Sama (detik)	Paragraf yang Diubah (detik)	Paragraf yang Ditulis Ulang (detik)
1	0,969696969697	0,686567164179	0,5
2	0,980392156863	0,64	0,35
3	0,676056338028	0,376811594203	0,30303030303
4	0,619047619048	0,547619047619	0,394736842105
5	0,96	0,723404255319	0,619047619048
Rata-rata	0,8410386167272	0,594880412264	0,433362952836

Dari Tabel 2 dapat dilihat bahwa dapat digunakan ambang sebesar 0,3 agar aplikasi dapat "menangkap" potensi plagiarisme dari paragraf yang merupakan hasil penulisan ulang dari paragraf aslinya.

IV. PEMBAHASAN

Di bagian pembahasan ini diuraikan beberapa catatan mengenai penelitian yang dilakukan.

Pertama, hasil pengujian deteksi plagiarisme yang dilakukan masih memiliki bias. Hal ini disebabkan proses simulasi plagiarisme dilakukan secara manual oleh peneliti sendiri. Upaya untuk menghilangkan bias sebenarnya dapat dilakukan terhadap teks Bahasa Indonesia seperti yang dilakukan Potthast [6] untuk teks Bahasa Inggris. Akan tetapi, belum ada peneliti di Indonesia yang melakukannya.

Kedua, dari Gambar 4 dan Gambar 5 terlihat bahwa lamanya eksekusi proses deteksi oleh aplikasi tidak mengikuti dugaan yang biasanya muncul, yaitu semakin besar atau semakin banyak paragraf, semakin lama eksekusi. Salah satu yang mengakibatkan anomali ini adalah proses pengubahan dokumen PDF ke dalam paragraf-paragraf.

PDF merupakan format dokumen untuk kepentingan pencetakan (baik melalui media display maupun media cetak). Dengan demikian, struktur dokumen berdasarkan isi, tidak tercermin dalam PDF. Ekstraksi paragraf membutuhkan analisis terhadap isi dokumen. Oleh karena algoritma yang digunakan sederhana, proses *chunking* menjadi tidak sempurna. Hal ini mempengaruhi penilaian algoritma deteksi dan lamanya eksekusi.

V. PENUTUP

A. Kesimpulan

1. Sebuah aplikasi berbasis web untuk pendeteksi plagiarisme telah diimplementasikan. Aplikasi ini menggunakan algoritma berbasis himpunan kata dari Küppers (2012). Pengujian aplikasi menunjukkan nilai potensi plagiarisme untuk 3 jenis paragraf. Ketiga jenis paragraf tersebut adalah paragraf yang sama dengan aslinya, paragraf yang merupakan hasil pengubahan kata atau tanda baca, dan paragraf yang merupakan hasil penulisan ulang.
2. Secara rata-rata, aplikasi mengukur nilai potensi sebesar 0,8 untuk paragraf-paragraf yang sama persis, 0,6 untuk paragraf yang merupakan hasil pengubahan kata atau tanda baca, dan 0,4 untuk paragraf-paragraf yang merupakan hasil penulisan ulang.
3. Dalam penelitian ini juga ditunjukkan kecepatan eksekusi dari aplikasi. Hasil yang didapatkan adalah bahwa banyak faktor lain yang mempengaruhi lamanya eksekusi deteksi selain ukuran *file* yang diuji dan juga jumlah paragraf dalam *file* tersebut. Dari 50 sampel yang diuji, proses deteksi terlama adalah terhadap dokumen dengan dengan ukuran 11MB yang terdiri dari 50 paragraf (hasil proses konversi aplikasi). Untuk mendeteksi dokumen ini diperlukan waktu 164 detik. Sementara yang tercepat adalah untuk mendeteksi dokumen sebesar 13,8 KB dengan 7 paragraf. Proses ini dilakukan dengan jumlah dokumen di basisdata sebanyak 1554 dokumen.

B. Saran

Ada beberapa perbaikan yang dapat dilakukan terhadap aplikasi yang telah dibuat. Pertama adalah menyediakan *corpus* standar untuk evaluasi algoritma pendeteksi plagiarisme. Dengan adanya *corpus* ini, evaluasi terhadap kinerja algoritma dapat dinilai secara standar menggunakan nilai-nilai seperti *recall*, *precision*, dan *granularity*. Kedua, proses ekstraksi teks dari dokumen yang diupload ke aplikasi baik untuk disimpan di basisdata sebagai referensi maupun sebagai dokumen yang diuji harus dibuat lebih "cerdas" sehingga algoritma deteksi bekerja lebih akurat. Terakhir, kecepatan eksekusi dapat ditingkatkan dengan memanfaatkan metode *heuristic* yang lain, misalnya menggunakan nilai *tf.idf*. Selain itu, algoritma lain untuk proses analisis detail dapat juga diimplementasikan untuk dibandingkan dengan algoritma Küppers yang kami gunakan.

UCAPAN TERIMA KASIH

Penelitian ini didanai oleh Direktorat Pendidikan Tinggi (Dikti) Republik Indonesia dalam skema Penelitian Dosen Pemula untuk tahun anggaran 2014.

DAFTAR PUSTAKA

[1] Küppers, Robin, dan Stefan Conrad. 2012. "A Set-Based Approach to Plagiarism Detection." *CLEF* (*Online Working Notes/Labs/Workshop*).

[2] Meuschke, Norman, dan Bela Gipp. 2013. "State-of-the-art in detecting academic plagiarism." *International Journal for Educational Integrity* 9.1

[3] Potthast, Martin, et al. 2013. "Overview of the 5th International Competition on Plagiarism Detection." *CLEF* (*Online Working Notes*/*Labs*/*Workshop*).

[4] Zu Eissen, Sven Meyer, dan Benno Stein. 2006. "Intrinsic plagiarism detection." *Advances in Information Retrieval*. 565-569.

[5] Potthast, Martin, et al. 2012. "Overview of the 4th International Competition on Plagiarism Detection." *CLEF* (*Online Working Notes*/*Labs*/*Workshop*).

[6] Potthast, Martin, et al. 2010. "An evaluation framework for plagiarism detection." *Proceedings of the 23rd international conference on computational linguistics*: Posters. Association for Computational Linguistics.