# *Source Retrieval* pada Deteksi Plagiarisme Berdasarkan *Biword Fingerprint* dengan Model Ruang Vektor

**Surya Agustian[1], Agung Sucipto[2]**
[1,2]Teknik Informatika UIN Suska Riau
Jl. H.R. Soeberantas km 11.5, Simpang Baru Panam, Pekanbaru
e-mail: [1]surya.agustian@uin-suska.ac.id, [2]mr.agung.sucipto@gmail.com

***Abstrak***

*Kasus plagiarisme dokumen berkembang semakin banyak seiring dengan pertambahan sumber digital yang pesat yang tersimpan di jaringan internet. Kesulitan ditemui saat harus menguji apakah suatu karya mengandung plagiarisme, dan di mana menemukan sumber aslinya (source retrieval) dari jutaan artikel dan dokumen yang ada tersebut. Penelitian ini bertujuan untuk melakukan pendeteksian kasus plagiarisme dari banyak dokumen sumber. Sistem pencarian sumber dokumennya menggunakan model ruang vektor, dengan kueri berbentuk frase word-n-gram, dengan n dari 3, 4 dan 5 (triword, quadword dan pentaword). Eksperimen dengan berbagai kombinasi, dilakukan dengan atau tanpa stemming, dan variasi pada frekuensi kata. Hasil yang diperoleh memberikan rekomendasi dokumen mana yang paling mirip dengan dokumen input dari sejumlah dokumen hasil pencarian awal. Hal ini sangat membantu manusia dalam menemukan dokumen sumber yang paling banyak diplagiasi.*

***Kata kunci:*** *deteksi plagiarisme, word n-gram, stemming, source retrieval, frekuensi kata*

***Abstract***

*The case of document plagiarism is growing along with the rapid increase in digital sources stored on the internet. It would be difficult when we have to examine whether a work contains plagiarism, where is to find the original source of the millions of articles and documents those exist. This study aims to detect plagiarism from many sources of documents. The source document search system (or source retrieval) uses a vector space model, where the query is in a phrase form of word-n-gram, with n of 3, 4 and 5 (triword, quadword and pentaword). Experiments with various combinations, carried out with or without stemming, and variations in word frequency. The results obtained provide recommendations on which document is the closest to the input document, among the initial search results. This method is very helpful for human in finding the most plagiarized source documents.*

***Keywords****: plagiarism detection, source retrieval, word n-gram, stemming, word frequency*

## 1. Pendahuluan

Plagiarisme merupakan sebuah tindakan penggunaan atau mengutip sebagian isi karya tulisan orang lain tanpa mencantumkan sumber tulisan yang kemudian diakui sebagai miliknya sendiri [1]. Pada masa ini, sangat mudah menghasilkan karya tulis dengan cara menjiplak dan meniru sumber tulisan lain yang telah ada sebelumnya dan dapat dengan mudah diperoleh dari internet. Masalahnya adalah bagaimana menemukan dokumen yang dijadikan sumber plagiasi dari jutaan artikel dan dokumen di internet, untuk dievaluasi sebagai kandidat dokumen sumber. Pendeteksian plagiarisme modern harus dapat menemukan sumber dokumen dari jutaan koleksi tersebut, kemudian menemukan fragmen atau bagian teks, bisa kalimat maupun paragraf, di dalam dokumen yang dicurigai (*suspicious document*) dari dokumen sumbernya (*source document*).

Dalam suatu tugas deteksi plagiarisme, dokumen dibandingan satu lawan satu (*one-to-one*) atau *apple to apple*. Akan sulit melakukannya bila harus membandingan satu per satu dokumen yang tersimpan di internet, karena jumlahnya yang sangat banyak. Untuk itu, dalam suatu sistem pendeteksi plagarisme, perlu adanya penggabungan metode pendeteksian plagiarisme *one-to-one* untuk deteksi fragmen plagiarisme, dengan *information retrieval* yang bertujuan mencari sumber dokumennya.

Beberapa algoritma terdahulu untuk deteksi plagiarisme pada dokumen, masih bekerja untuk pencocokan teks (*text alignmen*t) di antara 2 dokumen. Rafles [1] menyebutkan beberapa algoritma antara lain algoritma *winnowing* [2], algoritma manber [3], pendekatan kata *trigram* [4], algoritma *longest common subsequence*, teknik dot, algoritma *boyer-moore* dan algoritma

lainnya. Algoritma-algoritma ini, dapat diterapkan untuk mendeteksi bentuk plagiarisme *verbatim copy* (menyalin kata perkata) dan *paraphrase* yang sederhana.

Sejumlah penelitian juga dilakukan untuk membangun aplikasi deteksi plagiarisme dokumen berbasis *winnowing,* antara lain menggunakan metode *biword winnowing* [5] dan pendekatan *n-gram* berbasis frasa untuk *fingerprint*-nya [1]. Namun metode-metode tersebut hanya diterapkan untuk mendeteksi bentuk plagiarisme pada tugas *text alignment* (pencocokan fragmen) pada *one-to-one detection* seperti *verbatim copy, copy-paste* atau *word-by-word copying* (istilah yang sama untuk salinan kata perkata), dan *para-phrase plagiarism* (mengganti beberapa kata menjadi kata majemuk atau sinonimnya).

Untuk menemukan sumber dokumen dari sejumlah koleksi, pendekatan aplikasi *information retrieval* dapat digunakan dengan pengaturan bentuk kueri. Aplikasi *information retrieval* terdiri dari berbagai metode dalam pengukuran kemiripan antara kueri pendek dan dokumen, diantaranya algortima model ruang vektor [6, 7], dan *okapi BM25* [8, 9]. Penerapan teknik *fingerprinting* dengan *biword winnowing* sebagai kueri yang lebih panjang telah diujikan dalam penelitian [10] dan cukup berhasil menemukan dokumen sumber dari koleksi yang terbatas.

Penelitian ini bertujuan untuk dapat membantu mendeteksi apakah dalam suatu dokumen mengandung plagiarisme, dengan mencari dari banyak sumber dokumen yang telah terorganisasi dalam sebuah korpus sistem. Bila ditemukan kandidat sumbernya, lalu dicari fragment teks yang diplagiasi.

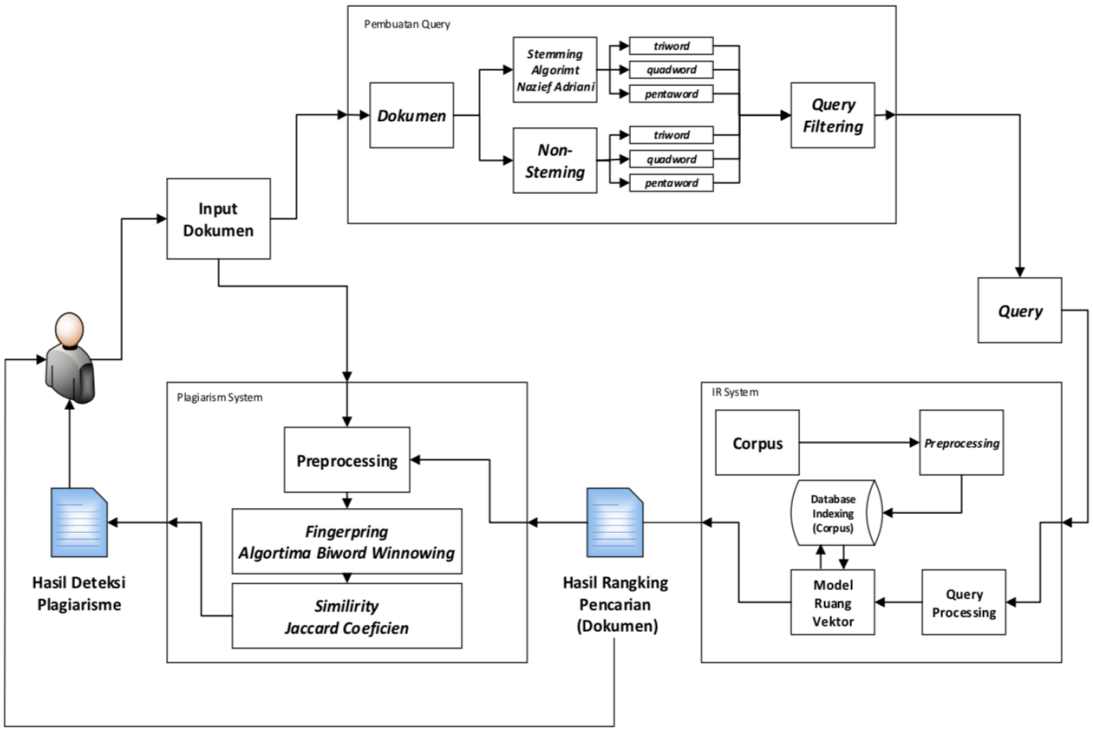
Penelitian ini menerapkan teknik *fingerprinting* dengan *winnowing* untuk menemukan dokumen sumber dari dokumen yang dicurigai mengandung plagiat, kemudian melakukan pencocokan fragmen teks (*text alignment*) yang diplagiasi paa *source document* dan *suspicions document.* Proses *text alignement*-nya menggunakan pendekatan *winnowing* seperti pada [1, 5] dengan basis yang digunakan untuk pemilihan *fingerprint* adalah pendekatan *n-gram* berbasis frasa dalam bentuk *word n-gram (triword*, *quadword, dan pentaword*).

Hasil yang diperoleh dari pengujian terhadap koleksi terbatas dengan topik Laporan Kerja Praktek dan Tugas Akhir mahasiswa S1 Teknik Informatika dapat menemukan kutipan terpanjang yang sama di antara dua dokumen teks dan mengukur kemiripan dokumen teks..

## 2. Metode Penelitian

Terdapat tiga metode untuk mendeteksi plagiarisme pada dokumen teks [11], yaitu metode berbasis *grammar* (tata bahasa), semantik, dan gabungan (*hybrid*). Suatu sistem pendeteksi plagiarisme dokumen harus memenuhi kebutuhan dasar [2] sebagai berikut:

1) *Whitespace Insensitivity*
   Harus tidak sensitif terhadap *whitespace*, yang meliputi spasi, jenis huruf (huruf kapital atau huruf kecil), tanda baca, dan sebagainya.
2) *Noise Surpression*
   Harus dapat menghidari penemuan kecocokan (*matching*) dengan fragmen kata yang terlalu pendek atau kurang relevan.
3) *Position Independence*
   Penemuan kecocokan (matching) dapat mendeteksi kata-kata di mana pun posisinya, tidak harus sama struktur kalimat dan susunan kata per kata. Hal ini untuk mengatasi plagiarisme berbentuk paraphrase atau pengubahan urutan kata atau penggunaan sinonim.




Gambar 1. Metode Deteksi Plagiarisme yang diusulkan

### 2.1. Sistem *Source Retrieval*

Untuk pengembangan sistem deteksi plagiarisme dan memenuhi kebutuhan dasar di atas, dalam penelitian ini dilakukan tiga tahap utama sebagai berikut:

**Tahap 1**: *Query Formulation* (formulasi kueri)

Proses ini dilakukan terhadap dokumen yang akan diuji apakah mengandung plagiarisme atau tidak. Secara umum, prosesnya dapat terdiri atas:

1) *Text cleaning* (pembersihan teks dokumen dari berbagai tanda baca dan karakter non-alfabet).
2) *Stop-word removal* (menghilangkan kata-kata yang tidak penting, seperti kata hubung, kata ganti dan kata-kata yang paling sering muncul dalam koleksi umum teks Bahasa Indonesia)
3) *Stemming* (proses pemotongan imbuhan, menggunakan metode dari [12].
4) Tokenisasi (melakukan ekstraksi dokumen kedalam token kata)
5) Pembentukan variasi token menjadi *triword, quadword* dan *pentaword*
6) *Frequency analysis*: menghitung frekuensi tiap variasi token.
7) Pembentukan kueri dengan menyusun kueri dari token yang terpilih di antara variasi token tersebut. Dipilih 5 token dengan frekuensi tertinggi dari variasi *triword, quadword,* dan *pentaword*. Dalam eksperimen, pengujian dengan 5 frekuensi terendah dan 5 frekuensi pertengahan juga dilakukan untuk melihat sifat-sifat *retrieval*.

**Tahap 2**: *Source Retrieval* (temu kembali dokumen sumber)

Proses ini untuk menemukan kandidat dokumen yang paling mirip dengan dokumen yang dicurigai. Oleh sebab itu, dilakukan proses terhadap dokumen-dokumen sumber (*source document*) di dalam koleksi (korpus). Secara garis besar, tahapannya meliputi:

A. Pembentukan korpus: mengumpulkan dokumen sumber yang akan diindeks.
   1) *Text cleaning* (pembersihan teks dokumen dari berbagai tanda baca dan karakter non-alfabet).
   2) *Stop-word removal* (menghilangkan kata-kata yang tidak penting, seperti kata hubung, kata ganti dan kata-kata yang paling sering muncul dalam koleksi umum teks Bahasa Indonesia)
   3) *Stemming* (proses pemotongan imbuhan, menggunakan metode dari [12].
   4) Tokenisasi (melakukan ekstraksi dokumen kedalam token kata)
   5) Pembentukan *inverted index*: *term* (token) terhadap dokumen.
B. Pengukuran *similarity* dan pe-*ranking*-an dengan model ruang vektor seperti pada [7].

1) Menemukan kandidat dokumen sumber yang mengandung kata-kata dalam kueri
2) Mengukur *similarity* kueri (output tahap 1) terhadap dokumen dari langkah B.1
3) Melakukan pe-*ranking*-an kandidat dokumen sumber berdasarkan *similarity*-nya

**Tahap 3**: Pengukuran tingkat plagiarisme

Proses ini menghitung seberapa besar kasus plagiarisme dari teks dokumen yang dicurigai terhadap setiap dokumen hasil kandidat yang ditemukan dari Tahap 2. Langkah-langkahnya adalah:

1) Pembentukan *fingerprint* dokumen sumber dan dokumen yang dicurigai, dalam bentuk *biword* (*word 2-gram*) dengan pendekatan *winnowing* [2].
2) Menghitung tingkat kemiripan (*similirity*) dokumen dengan menggunakan persamaan *jaccard coefficient*, agar output yang dihasilkan dapat berupa persentase antara jumlah kata-kata yang *match* (cocok) dengan jumlah keseluruhan kata di dalam *fingerprint*-nya.

Metode yang dilakukan untuk tahap 3 ini dijelaskan pada sub bagian 2.2 dan 2.3 untuk lebih detilnya.

### 2.2. *Fingerprint* Dokumen

Tujuan pembentukan *fingerprint* dari dokumen adalah untuk 'mempersingkat' dokumen yang akan menggambarkan ciri-ciri dokumen. Hal ini berbeda dari ringkasan, di mana ringkasan masih memperhatikan struktur kalimat, sedangkan *fingerprint* hanya berupa kata-kata yang mungkin penting dan spesifik terhadap dokumennya. Banyak dokumen dapat diketahui topiknya berdasarkan kata-kata yang terpilih tersebut.

Tahap pembentukan *fingerprint* dilakukan dengan mengambil kembali hasil tokenisasi (pembentukan token) dari kedua dokumen (*source* dan *suspicious*) dari hasil tahap 1. Kemudian dari token tersebut, dilakukan pembentukan *fingerprint* sebagai berikut:

A. Pembentukan *bag of hash*: daftar hash untuk setiap token biword.
   1) Membentuk daftar token *word 2-gram* (*biword*) dari seluruh kata (*bag of biword*).
   2) Menghitung nilai *hash* untuk setiap *biword* token dengan fungsi MD5. MD5 adalah fungsi enkripsi yang sering dipakai pada sistem keamanan komputer. Nilai hash akan unik dan memiliki panjang (jumlah karakter) yang sama [5]. Ini adalah solusi untuk masalah panjang kata yang tidak sama yang langsung dipakai untuk menghitung rolling hash pada [2].
   3) Mengkonversi nilai *hash* (yang berupa gabungan karakter huruf dan angka) ke dalam *integer* dengan metode *rolling hash* dari [2].

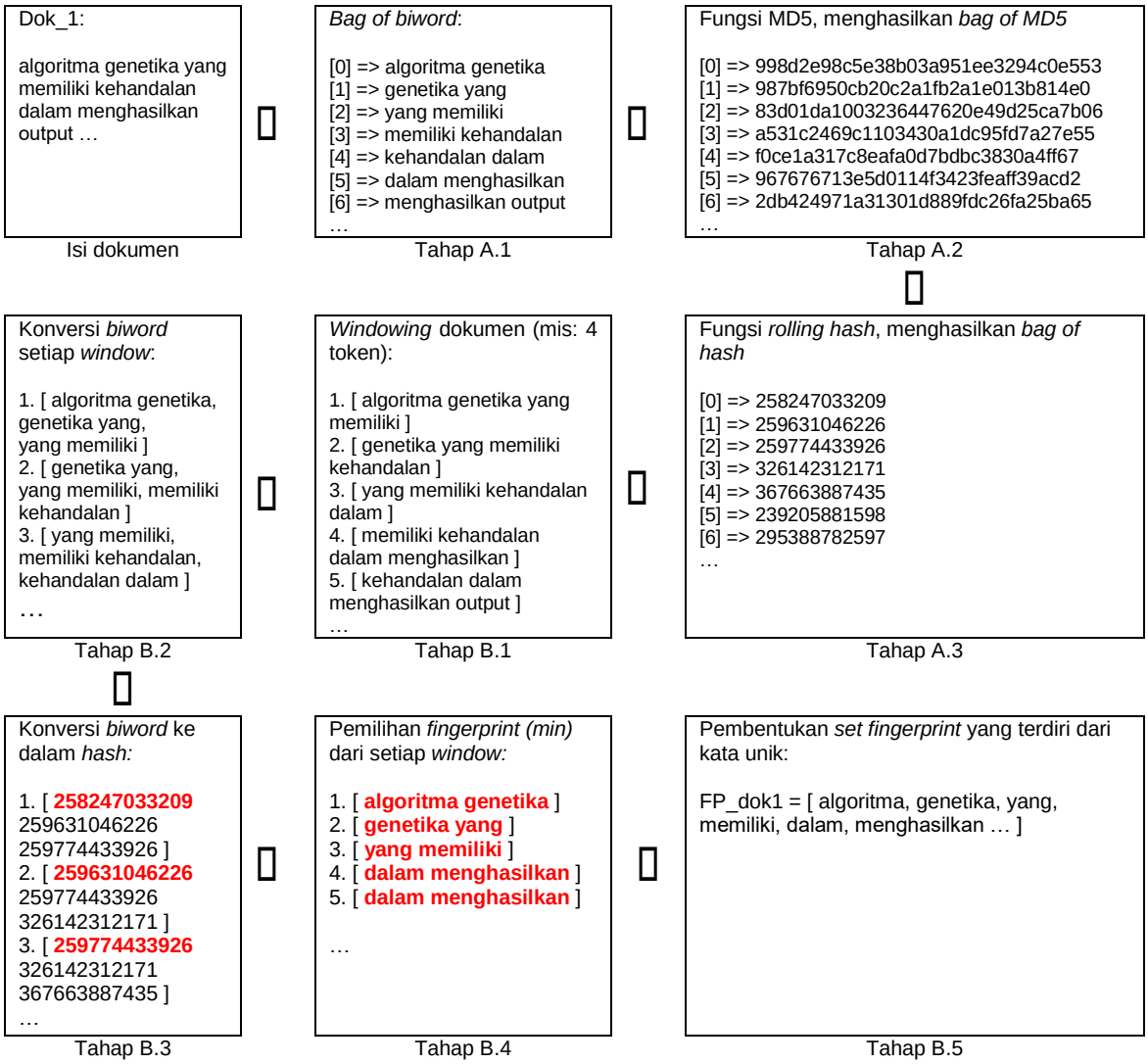
B. Pembentukan *fingerprint* dokumen: daftar *hash* (atau kata-kata yang mewakili dokumen) sebagaimana [5]. Ilustrasi dari proses *fingerprinting* (proses pembentukan *fingerprint*) dokumen dapat dilihat pada gambar 2.
   1) Membentuk *window* yang bergerak di sepanjang dokumen. Ukuran window divariasikan dalam eksperimen. Ukuran *window* misalnya 4 kata.
   2) Mengkonversi setiap token menjadi *biword* pada *window*
   3) Mengkonversi setiap window menjadi nilai *rolling hash*-nya, yang berdasarkan daftar *bag of hash*.
   4) Memilih di antara nilai *hash*, berupa nilai tengah, nilai maksimum atau minimum sebagai *fingerprint* dari setiap *window*.
   5) Kembalikan ke bentuk token *biword* (cek daftarnya dari *bag of hash*).

### 2.3. *Similarity* Dokumen

Pengukuran similarity dokumen yang dilakukan dalam sistem ini ada 2 metode, yaitu model ruang vektor pada proses *source retrieval* seperti digunakan pada [7], dan model *Jackard coefficient* seperti pada [1, 5] pada pengukuran tingkat plagiarisme. Pada model ruang vektor, vektor dokumen yang dicurigai dan dokumen sumber dibentuk dari pembobotan TF-IDF (*w*) dari kata-kata ke *i* sampai *N* di dalam *bag of words*. Similarity-nya dihitung berdasarkan persamaan *cosine similarity* (1) berikut.

$$Cosine(Susp, Src) = \frac{\sum_i^N \quad (w_{i,susp} \cdot w_{i,src})}{\sqrt{\sum_i^N \quad w_{i,susp}^2 \cdot \sum_i^N \quad w_{i,src}^2}} \tag{1}$$

*ISSN (Printed) : 2579-7271*
*ISSN (Online ) : 2579-5406*




Gambar 2. Proses Pembentukan *Fingerprint* dan Kueri Dokumen

Sedangkan *similarity* dengan menggunakan *Jackard coefficient* dihitung berdasarkan jumlah kata yang beririsan dibagi jumlah keseluruhan kata unik pada dokumen yang dibandingkan (*source* dan *suspicious*), sebagaimana persamaan (2) di bawah ini.

$$Jackard\ (Susp, Src) = \frac{|W(Susp) \cap W(Src)|}{|W(Susp) \cup W(Src)|} \quad (2)$$

## 3. Hasil Pengujian dan Pembahasan

Pengujian dilakukan terhadap korpus terbatas (topik dan jumlahnya) yang terdiri atas 20 data dokumen Bab 2 (Landasan Teori) dari laporan KP (Kerja Praktek) dan TA (Tugas Akhir) mahasiswa jurusan Teknik Informatika UIN Suska Riau. Pemilihan Bab 2 ini karena diyakini banyak sekali kasus kalimat dan paragraf yang mirip dari laporan-laporan tersebut, misalnya latar belakang mengapa mata kuliah KP diadakan untuk jurusan Teknik Informatika, struktur organisasi perusahaan tempat KP, pengembangan sistem informasi, dan sebagainya. Sedangkan data uji berupa dokumen yang dicurigai mengandung plagiarisme sebanyak 5 dokumen.

### 3.1. Setup Pengujian

Variasi skema pengujian atau eksperimen yang dilakukan adalah:

a. Kombinasi *stemming (stemming* atau *no-stemming)*
b. Kombinasi pembentukan *word n-gram* (*triword, quadword* atau *pentaword*) untuk *inverted index* pada *source retrieval*
c. Pemilihan 5 nilai frekuensi *word n-gram* (max, min atau median) sebagai kueri pada *query formulation*.
d. Mengambil 3 *ranking* teratas dari nilai *similarity* dokumen *suspicious* terhadap dokumen *source* untuk pengukuran tingkat plagiarisme.
e. Mengukur tingkat *similarity* dari dokumen *suspicios* terhadap 3 dokumen *source* hasil *retrieval*.

### 3.2. Pengukuran

Sesuai dengan tujuan penelitian, maka pengukuran yang dilakukan adalah menghitung tingkat kemiripan dokumen, yaitu antara dokumen yang dicurigai, dengan dokumen-dokumen hasil pencarian dalam korpus. Pengukuran tingkat kemiripan ini dalam bentuk koefisien jackard, yaitu seberapa banyak *n-gram* kata-kata yang beririsan dari daftar *fingerprint* setiap dokumen yang diukur kemiripannya.

Pengukuran *precision* dan *recall* tidak dilakukan karena dalam penelitian ini, kita tidak memiliki data label relevan dan tidak relevan terhadap dokumen yang panjang, di mana kueri dan topik dokumennya tidak ditentukan oleh manusia, tetapi oleh mesin (sistem) berdasarkan *fingerprint*.

Tabel 1. Resume Hasil yang Terbaik dari Berbagai Kombinasi Setup Eksperimen

| Nomor eksperimen | Susp doc | Variasi Eksperimen | | | Susp query | Hasil | | |
|---|---|---|---|---|---|---|---|---|
| | | Stemming | Word n-gram | Frek (max, min, med) | | Src doc | Cosine | Jackard (tingkat plagiarisme) |
| 1 | dok kp1.txt | tidak | 3 | max | olah data system satu entitas luar lingkung entity sifat | dok kp12.txt<br>dok kp9.txt<br>dok kp7.txt | 0,2358<br>0,1830<br>0,1103 | 65,32%<br>42,36%<br>11,79% |
| 2 | dok kp5.txt | tidak | 3 | max | tahap analisis sitem salah satu entitas desain luar lingkung | dok kp 9.txt<br>dok kp12.txt<br>dok kp7.txt | 0,1858<br>0,1807<br>0,1441 | 36,46%<br>31,35%<br>9,04% |
| 3 | dok ta1.txt | tidak | 3 | max | hasil iris urut aplikasi deteksi plagiarisme dua dokumen teks biword triword ikut gambar diagram | dok ta7.txt<br>dok kp8.txt<br>dok kp12.txt | 0,2070<br>0,1200<br>0,0453 | 1,96%<br>1,20%<br>1,51% |
| 4 | dok ta5.txt | tidak | 3 | max | sistem temu kembali informasi model ruang vektor kembali | dok ta7.txt<br>dok ta16.txt<br>dok kp6.txt | 0,0673<br>0,0245<br>0,0120 | 9,84%<br>1,23%<br>1,91% |
| 5 | dok ta4.txt | tidak | 3 | max | olah data sistem satu entitas luar lingkung entity sifat information retrieval sistem jumlah dokumen relevan | dok ta7.txt<br>dok kp8.txt<br>dok kp12.txt | 0,2358<br>0,1830<br>0,1103 | 15,89%<br>2,51%<br>2,54% |

### 3.3. Data Pengujian

Dari setup pengujian, telah dilakukan kombinasi dari variasi-variasi skema pengujian di sub bagian 3.1.(a-c), dan menghasilkan kombinasi terbaik adalah menggunakan: (a) tanpa *stemming*, (b) *triword*, (c) frekuensi maksimum, pada tahap *source retrieval*, sehingga pada saat analisis *text alignment*-nya, diperoleh rata-rata kemiripan antar dokumen *suspicious* dengan *source*-nya (dari 3 ranking teratas) sebesar 15,66%.

Ringkasan data pengujian dari hasil yang terbaik dari kombinasi tersebut, dapat dilihat pada table 1. Sedangkan contoh cuplikan fragmen yang terdeteksi plagiat dapat dilihat pada gambar 3 (jenis plagiat *verbatim copy*).

*ISSN (Printed) : 2579-7271*
*ISSN (Online ) : 2579-5406*


Gambar 3. Fragmen Teks yang Terdeteksi Plagiat

### 3.4. Analisa Hasil Pengujian

Dengan kombinasi terbaik seperti telah diuraikan dalam sub bab 3.3, seperti terlihat pada tabel 1, sistem dapat mendeteksi tingkat plagiarisme yang tinggi sampai yang rendah dari berberapa sampel dokumen pengujian (*suspicios doc*). Sistem dapat bekerja dengan baik untuk koleksi dokumen bila tidak dilakukan *stemming* pada teks, kemudian menggunakan *word 3-gram* (*triword*), dan memilih top 5 token dengan frekuensi tertinggi sebagai kueri dari dokumen yang dicurigai. Hal ini disebabkan karena hampir seluruh kasus plagiarisme yang terdapat pada dokumen yang diujikan adalah *verbatim copy*.

Penggunaan *Jackard coefficient* untuk mengukur seberapa besar tingkat plagiarisme, cukup baik karena dapat menggambarkan seberapa besar kasus plagiarisme yang dideteksi. Walaupun dengan kueri otomatis yang dibangkitkan oleh sistem, dan memperoleh dokumen sumber hasil *retrieval* yang nilai *cosine similarity*-nya cukup rendah, namun pada saat proses pengukuran *similarity* antar dokumen *suspicious* dan sumber, *similarity*-nya menjadi sangat tinggi. Hal ini karena *fingerprint* yang digunakan lebih banyak, dibandingkan kueri untuk source retrievalnya (ilustrasi pemilihan kueri pada source retrieval dijelaskan pada Gambar 2).

Dari eksiperimen ke-2 terlihat bahwa bagian teks yang diplagiasi cukup tinggi, yaitu 36,46%, yang meliputi sebagian sub bab pada bab dasar teori. Hal ini terlihat pula dari tangkapan layar bagian yang mengandung plagiarisme seperti pada Gambar 3. Dalam hal keseluruhan bab 2 dari dokumen *suspicios* dan *source*, memang tidak seluruhnya sama, karena pekerjaan KP yang dikerjakan berbeda. Terdapat sub bab lain yang isinya berbeda.

## 4. Kesimpulan

Dari hasil-hasil yang diperoleh pada eksperimen penelitian ini, dapat disimpulkan bahwa:

1) *Fingerprint* pada dokumen yang terdiri atas kata-kata yang dipilih dengan teknik *winnowing* dapat menggambarkan topik dokumen.
2) Penggunaan *fingerprint* pada dokumen panjang dapat membantu dalam deteksi plagiarisme dan mengurangi poses pencocokan kalimat per kalimat maupun kata per kata, dengan hasil yang memuaskan.
3) Penggunaan *Jackard coefficient* dapat menghitung tingkat kemiripan dokumen yang dicurigai dengan sumbernya dengan nilai yang rasional sesuai dengan proporsi dokumen secara keseluruhan.

Saran untuk penelitian selanjutnya, proses *text-alignment* dilakukan untuk mendapatkan offset karakter mulai dan sampai ke berapa yang merupakan teks plagiarisme, seperit shared task yang telah dilakukan pada PAN 2013, 2014 dan 2015 [13, 14, 15]. Pengukuran tingkat plagiarisme dapat menggunakan *micro-similarity*, yaitu kemiripan antara *fragment* plagiat dan *fragment* sumber (misalnya antar paragraf atau *passage*), bukan pengukuran terhadap keseluruhan isi dokumen (jumlah kata) atau keserluruhan *fingerprint*-nya.

**Daftar Pustaka**

[1] Rafles, Adek, *Pengembangan Aplikasi Pendeteksi Plagiarisme Dokumen Dengan Pendekatan k-gram Berbasis Frasa*, Tugas Akhir Teknik Informatika, UIN Suska Riau, 2013.
[2] Schleimer, Saul, Daniel S. Wilkerson, and Alex Aiken. *Winnowing: Local Algorithms for Document Fingerprinting.* San Diego: In Proceedings of the ACM SIGMOD International Conference On Management Of Data. 2003
[3] Manber, Udi. *Finding similar files in a large file system*. San Fransisco: In *Proceedings of the USENIX Winter.* 1994
[4] Lyon C, Malcolm J, and Dickerson B. *Detecting short passages of similar text in large document collections*. Hertfordshire: Proceedings of EMNLP (Empirical Methods in Natural Language Processing). 2001
[5] Ridho, Muhammad, *Rancang Bangun Aplikasi Pendeteksi Penjiplakan Dokumen Menggunakan Algoritma Biword Winnowing*, Tugas Akhir Teknik Informatika, 2013.
[6] Syahroni, Raja, *Sistem Temu Balik Informasi (Stbi) Laporan Kerja Praktek Dan Tugas Akhir Menggunakan Model Ruang Vektor (Studi Kasus : Teknik Informatika),* Tugas Akhir Teknik Informatika, 2012.
[7] Agustian, Surya dan Imelda Sukma Wulandari, *Sistem Qur'an Retrieval Bahasa Indonesia Berbasis Web dengan Reorganisasi Korpus*, Prosiding Konferensi Nasional Sistem Informasi (KNSI), 2014
[8] Bintana, Rizqa Raaiqa, *Penerapan Model Okapi BM25 pada Sistem Temu Kembali Informasi*, Prosiding Seminar Nasional Teknologi Informasi dan Industri (SNTIKI), 2012
[9] Agustian, Surya dan Rizqa Raaiqa Bintana, *Pengembangan Sistem Qur'an Retrieval Terjemahan Bahasa Inggris dengan Metode Okapi BM25 dan Porter Stemmer*, Prosiding Seminar Nasional Teknologi Informasi dan Industri (SNTIKI), 2014
[10] Hidayatullah, Syarif, *Source Detection* pada Kasus Plagiarisme Dokumen Menggunakan Metode *Biword Winnowing* dan *Retrieval Berbasis Okapi BM25*, Laporan Tugas Akhir, UIN Suska Riau, 2014.
[11] Wang Tao, Fan Xiao-Zhong, Liu Jie, *Plagiarism Detection in Chinese Based on Chunk and Paragraph Weight*. Kunming: in Proceedings of the Seventh International Conference on Machine Learning and Cybernetics. 2008
[12] Bobby A.A. Nazief dan Mirna Adriani, *Confix Stripping: Approach to Stemming Algorithm for Bahasa Indonesia*, Faculty of Computer Science, University of Indonesia, 1996
[13] M. Potthast, M. Hagen, T. Gollub, M. Tippmann, J. Kiesel, P. Rosso, E. Stamatatos dan B. Stein, "Overview of the 5th International Competition on Plagiarism Detection," dalam *3rd Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN 2013) at CLEF 2013*, 2013.
[14] Efstathios Stamatatos, Walter Daelemans, Ben Verhoeven, Martin Potthast, Benno Stein, Patrick Juola, Miguel A Sanchez-Perez, Alberto Barrón-Cedeño, *Overview of the author identification task at PAN 2014,* Proceeding of CLEF 2014 Evaluation Labs and Workshop Working Notes Papers, Sheffield, UK, 2014.
[15] Stamatatos E., Potthast M., Rangel F., Rosso P., Stein B., *Overview of the PAN/CLEF 2015 Evaluation Lab*. In: Mothe J. et al. (eds) *Experimental IR Meets Multilinguality, Multimodality, and Interaction*. CLEF 2015. Lecture Notes in Computer Science, vol 9283. Springer, Cham. https://doi.org/10.1007/978-3-319-24027-5_49.