Terakreditasi SINTA Peringkat 2
Surat Keputusan Dirjen Penguatan Riset dan Pengembangan Ristek Dikti No. 10/E/KPT/2019
masa berlaku mulai Vol. 1 No. 1 tahun 2017 s.d Vol. 5 No. 3 tahun 2021

Terbit online pada laman web jurnal: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 4 No. 5 (2020) 988 - 997 ISSN Media Elektronik: 2580-0760

Analisis Fitur Stilometri dan Strategi Segmentasi pada Sistem Deteksi Plagiasi Intrinsik Teks

Sylvia Putri Gunawan[1], Lucia Dwi Krisnawati[2], Antonius Rachmat Chrismanto[3]
[1,2,3]Prodi Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana
[1]slyvia.putri@ti.ukdw.ac.id, [2]krisna@staff.ukdw.ac.id, [3]anton@ti.ukdw.ac.id*

Abstract

Two different paradigms in the field of plagiarism detection resulting in External Plagiarism Detection (EPD) and Intrinsic Plagiarism Detection (IPD) systems. The most common applied system is EPD, which requires its algorithm to make a heuristic comparison between a suspicious document with documents in a corpus. In contrast, given a suspicious document only, an algorithm of IPD should be able to find the plagiarism section by looking for text segments having different writing styles. Previous researches for Indonesian texts fell only in the field of the EPD development system. Therefore, this research focuses on and contributes to experimenting and analyzing the stylometric features and segmentation strategies to build an IPD system for Indonesian texts. The experimentation results show that the paragraph segment performs better by scoring 0.92 for Macro Averaged-Accuracy and 0.54 for Macro Averaged-F1. The stylometric features achieving the highest scores of F-1 and Accuracy are the frequency of punctuation, the average paragraph length, and the type-token ratio.

Keywords: intrinsic plagiarism detection, stylometry features, text segmentation, outlier

Abstrak

Dalam pembangunan sistem deteksi plagiasi, terdapat dua paradigma berbeda. Paradigma yang sering digunakan adalah sistem membandingkan dokumen yang diduga plagiat dengan dokumen di repositori untuk menemukan bagian plagiasinya. Paradigma ini disebut sistem Deteksi Plagiasi Eksternal (DPE). Paradigma lainnya berusaha meniru kecerdasan manusia dalam mengenali bagian teks yang berbeda gaya penulisannya, sehingga hanya diberi sebuah teks masukan, sistem Deteksi Plagiasi Intrinsik (DPI) harus mampu menemukan potongan teks yang memiliki gaya penulisan yang berbeda dari bagian teks lainnya dari dokumen yang sama. Berhubung penelitian sebelumnya dengan teks berbahasa Indonesia sebagian besar berkutat di pembangunan sistem DPE, maka penelitian ini berkontribusi dalam eksperimentasi fitur-fitur stilometri serta segmentasi teks untuk sistem DPI berbahasa Indonesia. Hasil uji coba menunjukkan bahwa segmen paragraf memberikan hasil yang lebih baik daripada segmen kalimat dengan nilai Akurasi sebesar 0.92 dan nilai F-1 sebesar 0.54. Sedangkan kombinasi fitur stilometri yang cocok digunakan adalah rata-rata jumlah tanda baca, panjang paragraf, dan rasio type-token.

Kata kunci: deteksi plagiasi, deteksi plagiasi intrinsik, fitur stilometri, segmentasi teks, outlier.

1. Pendahuluan

Pada umumnya, stilometri didefinisikan sebagai cabang ilmu linguistik yang mempelajari tentang kronologi dan perkembangan karya-karya penulis yang didasarkan pada pemikiran dan gaya penulisannya. Seiring dengan proses otomatisasi di segala bidang, maka makna stilometri mengalami pergeseran dan difenisikan oleh Halvani [1] sebagai "cabang ilmu yang menentukan kepemilikan pengarang terhadap karya-karya tulis melalui analisis statistik dan pembelajaran mesin". Analisis stilometri banyak diterapkan dalam aplikasi komputasional yang lebih kompleks seperti pada Identifikasi Kepengarangan (*authorship identification*) oleh Rexha dkk [2], atribusi dan diarisasi penulis oleh Stamatatos dkk [3], atau Deteksi Plagiasi Intrinsik (DPI) oleh Rexha [1] dan Kuznetsov dkk [4].

Berdasarkan pendekatannya, sistem deteksi plagiasi dibedakan menjadi dua yakni sistem Deteksi Plagiasi Intrinsik (DPI) dan Deteksi Plagiasi Eksternal (DPE) [5]. Algoritma DPE bekerja dengan cara membandingkan dokumen uji yang diinputkan dengan tiap dokumen yang ada di korpus untuk menemukan bagian dokumen uji yang memiliki kesamaan dengan bagian dokumen di korpus [6]. Dengan pendekatan ini,

DPE mutlak mensyaratkan ketersediaan korpus yang mengandung beberapa dokumen yang terindikasikan sebagai sumber plagiasi.

Berbeda dari DPE, Deteksi Plagiasi Intrinsik (DPI) tidak memerlukan korpus dan perbandingan dengan teks yang diasumsikan sebagai dokumen sumber. Hal ini dipengaruhi oleh fakta bahwa kandidat dokumen sumber tidak selalu tersedia di korpus. Mekanisme kerja algoritma DPI didasarkan pada peniruan keahlian manusia dalam mengenali bagian teks yang mengalami perubahan gaya penulisan sebagai tanda awal teks salinan (CoPas -- c*opy and paste*) tanpa harus membandingkan dengan teks lainnya [7]. Untuk itulah, maka DPI terkait erat dengan bidang verifikasi penulis (*author verification*) [1] [8].

Sebagai pihak yang mencetuskan istilah DPI, Eissen dan Stein [9] memperkenalkan penggunaan fitur Stilometri rata-rata kelas frekuensi kata (*averaged word frequency class*) yang didapatkan dengan mengelompokkan kata dengan frekuensi kemunculan tertinggi di kelas 0, tertinggi kedua di kelas 1 dst. Berdasarkan korpus mereka, maka *stopword* menempati kelas 0-19 [9]. Dengan panjang jendela sebesar 40-200 kata, Kelas kata ini kemudian dicari reratanya yang kemudian digunakan untuk mendeteksi adanya anomali perubahan nilai rata-ratanya. Di [9], rerata Frekuensi Kelas Kata ini mampu menunjukkan ukuran dan kekayaan kosakata yang digunakan oleh penulis.

Berbeda dari Eissen dan Stein, Stamatatos [10] menggunakan 3-gram karakter sebagai fitur stilometri yang dibentuk dari tiap jendela sepanjang 1000 karakter. Pergeseran jendela dilakukan tiap 200 karakter. Deteksi perubahan gaya penulisan dilakukan dengan mengukur jarak tiap jendela dengan menggunakan rumus *normalized dissimilarity* (nd). Setelah itu dilakukan perhitungan perubahan gaya dengan rumus sc (*style change*) yang dia perkenalkan. Kontribusi terbesar dari penelitian Stamataos ini adalah memperkenalkan kedua rumus tersebut.

Selain Stamatatos [10], n-gram karakter digunakan juga oleh Rahman [11] dan Kuznetsov dkk [4]. Rahman [11] membentuk n-gram karakter dari segmen jendela berukuran 1000, 2000 dan 5000 karakter. Hanya saja Rahman menggenerasikan profil n-gram dari tiap segmen jendela (X) yang dipisahkan dari profil n-gram keseluruhan dokumen (Y). Kemudian, perhitungan entropi relatif dan koefisien korelasi dilakukan dengan dasar profil frekuensi n-gram X dan Y. Selain n-gram karakter, Kuznetsov dkk [4] menggunakan juga n-gram kata sebagai fitur stilometri.

Kecenderungan menggunakan fitur numerik sebagai hasil kuantifikasi fitur stilometri lebih dominan dalam DPI. Sebagai contoh, daripada menggunakan bigram atau token, maka frekuensi token dan bigramlah yang dijadikan fitur stilometri. Selain itu, sebagian besar sistem DPI menggunakan beberapa fitur sekaligus daripada hanya mengandalkan fitur tunggal. Fitur stilometri lainnya yang kerap digunakan adalah frekuensi panjang kata [2], [12], Frekuensi panjang kalimat [12], [13], Frekuensi tag kelas kata (*part of speech (pos) tag frequency*) [12], Rasio *type-token* [2], [13], dan frekuensi kespesifikan kata [12].

Ditinjau dari sisi granularitas segmen teks, maka ada yang menggunakan segmen atau jendela teks di tingkat karakter seperti 1000-5000 karakter [10] - [11], di tingkat token [13], kalimat [4], [12], dan potongan teks (*snippet*) sepanjang 400 karakter pertama dari tiap sub-bahasan atau sub-sub bahasan [11]. Metode yang digunakan untuk pengukuran pun berbeda-beda dari penggunaan clustering seperti di artikelnya Elamine dkk [13], klasifikasi oleh Kuznetsov dkk [4], Rahman [12], Support Vector Machine (SVM) [12], dan perhitungan perubahan gaya penulisan di Stamatatos [10], dan Kraus [13].

Ulasan pemilihan fitur stilometri dan strategi segmentasi yang dipaparkan sebelumnya berasal dari artikel yang melaporkan penelitian di ranah Deteksi Plagiasi Intrinsik (DPI) dengan objek teks berbahasa asing (Inggris, dll). Dalam studi literatur, penulis telah mensurvei 27 artikel yang diterbitkan dari tahun 2011-2018 terkait deteksi plagiasi dengan objek teks berbahasa Indonesia. Semua artikel yang didapatkan tersebut melaporkan proses pembangunan sistem DPE dan bukan DPI. Sebagai contohnya, Sunardi dkk [14] menggunakan trigram karakter sebagai unit analisis dan algoritma Winnowing untuk seleksi fiturnya, kemudian *Jaccard coefficient* diterapkan untuk mengukur kesamaan jumlah kemiripan fitur antara dokumen uji dengan tiap dokumen di korpus. Fitur leksikal, token atau unigram diterapkan oleh Bianto dkk [15], dan Ratna dkk [16]. Perbedaannya, Ratna dkk [16] melakukan analisis semantik dari fitur yang dipilih dengan mengaplikasikan *Latent Semantic Analysis* (LSA) serta *Learning Vector Quantification* untuk menemukan plagiasi di teks lintas bahasa (Inggris-Indonesia). Sedangkan, Bianto dkk [15] menerapkan pendekatan clustering dengan ukuran model Bayesian untuk menemukan persamaan di tiap segmen dokumen uji dengan dokumen yang ada di korpusnya.

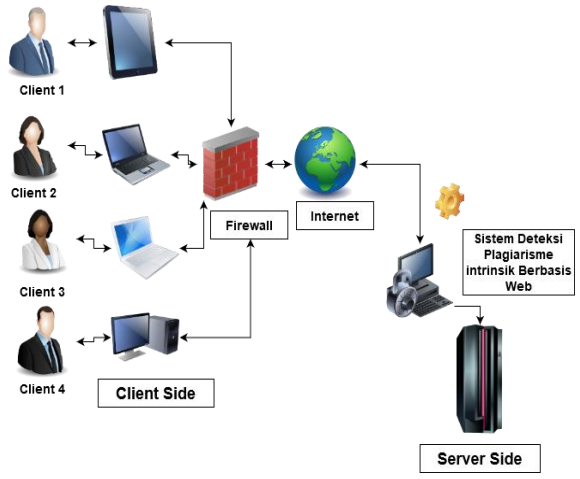
Berdasarkan survei literatur sistem Deteksi Plagiasi teks berbahasa Indonesia ini, maka penelitian ini lebih difokuskan untuk melakukan analisis beberapa fitur stilometri yang dilaporkan di referensi [1] - [13] bagi sistem DPI teks berbahasa Indonesia. Selain itu, Halvani [1] melaporkan bahwa ukuran segmentasi teks yang praktis masih menjadi pertanyaan terbuka dalam DPI. Untuk itulah, penelitian ini juga melakukan eksperimentasi terhadap dua strategi segmentasi. Tujuan penelitian ini adalah untuk menemukan kombinasi fitur dan segmentasi yang representatif untuk deteksi perubahan gaya penulisan pada teks berbahasa Indonesia. Sehingga, penelitian ini akan dapat memberikan kontribusi yaitu pembangunan sistem,

analisis fitur stilometri dan strategi segmentasi dalam bidang Deteksi Plagiasi Intrinsik (DPI) teks berbahasa Indonesia yang tidak terlalu banyak dilakukan.

Artikel ini disusun dengan urutan: 1) pendahuluan yang berisi latar belakang masalah, tinjauan pustaka dan literatur, alasan pentingnya penelitian, tujuan penelitian, dan kontribusi yang dilakukan, 2) metode penelitian yang membahas tentang metode yang dilakukan, 3) hasil dan pembahasan dari penelitian yang telah dilakukan, dan 4) kesimpulan yang berisi simpulan dan saran penelitian pada masa yang akan datang.

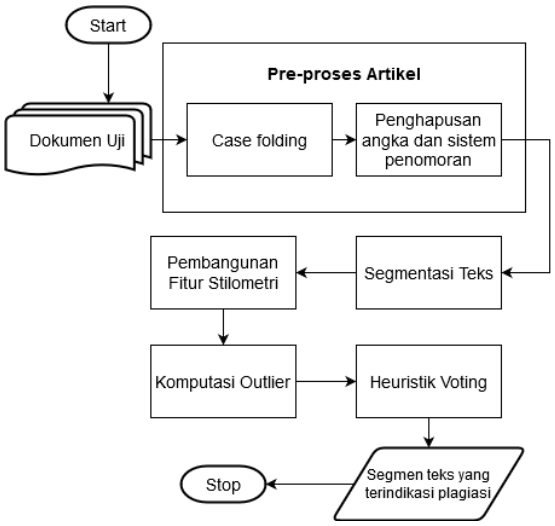
1. Metode Penelitian

Seperti yang diuraikan di bab Pendahuluan, sistem DPI tidak memerlukan repositori dokumen atau korpus untuk melakukan perbandingan. Untuk itu, Sistem DPI yang dibangun berbasis aplikasi situs Web dan pengguna bisa memberikan sebuah input dokumen ke sistem saat mereka terhubung dengan Internet. Sistem arsitektur aplikasi DPI yang penulis rancang dapat dilihat di Gambar 1.



Gambar 1. Arsitektur sistem DPI berbasis web

Saat diberi sebuah dokumen input, sistem DPI memproses dokumen tersebut ke dalam 5 langkah utama. Lima (5) proses tersebut adalah pra-pemrosesan dokumen, segmentasi dokumen, pembentukan dan ekstraksi fitur stilometri yang dilanjutkan dengan proses komputasi nilai *outlier* dari tiap segmen. Proses terakhir adalah perhitungan *voting* untuk menentukan apakah sebuah segmen benar-benar memiliki gaya penulisan yang berbeda dari segmen lainnya. Proses *voting* ini dikategorikan dalam pasca-pemrosesan. Sedangkan pra pemrosesan dokumen terdiri dari 2 langkah. Tahapan pemrosesan yang menggambarkan metode dalam pembangunan sistem DPI ini bisa dilihat di Gambar 2. Setiap tahapannya akan dijelaskan di sub bab berikutnya.



Gambar 2. Tahapan proses pembangunan sistem DPI

2.1. Pra Pemrosesan Dokumen

Sesuai paradigmanya, sistem DPI tidak melakukan perbandingan kemiripan dokumen yang diduga mengandung plagiasi (dokumen uji) dengan tiap dokumen yang ada di repositori (dokumen sumber). Untuk itu, pra pemrosesan dokumen diterapkan hanya pada dokumen uji atau yang diinputkan ke sistem. Pra pemrosesan dokumen dilakukan dengan mengubah semua huruf besar menjadi huruf kecil (*case folding*). Berhubung bahasa pemrograman yang digunakan adalah *Python*, maka normalisasi penulisan huruf kecil ini dilakukan dengan menggunakan fungsi *lower()* pada pustaka *string* di Python.

Kemudian, sistem menghapus token yang terdiri dari 1 huruf diikuti tanda baca dan angka yang diasumsikan tidak relevan dalam proses penelitian ini, yaitu poin-poin yang digunakan untuk penomoran list (*ordered list, bullet and numbering*) seperti yang ditunjukkan di Gambar 3. Untuk mendeteksi penomoran baik yang ditulis dengan angka Arab, Romawi, ataupun huruf Latin, modul RE (*Regular Expression*) milik Python digunakan dengan pola "\n +\w[.]", "\n\w[.]", "\n +\d+[.]", dan "\n\d+[.]".

```
sistem teknologi yang diacu di sini adalah teknik dan seni
bangunan atau arsitektur tradisional yang berada di
indonesia.

1. pengertian arsitektur vernacular

arsitektur merupakan produk kebudayaan yang mencerminkan
peradaban masyarakat setempat. sebuah kebudayaan dapat
bertahan karena nilai-nilainya tetap dipegang dan
diturunkan antar generasi, dan tercermin pada tampilan
```

Gambar 3. Contoh teks dengan penomoran sub bab yang dihilangkan di tahap pra pemrosesan.

2.2. Segmentasi Teks

Penelitian ini menerapkan dua tingkat segmentasi teks, yakni segmen kalimat dan paragraf. Segmentasi teks ke dalam paragraf dilakukan dengan cara 1). memecah-

mecah teks berdasarkan baris baru ("\n"). Sehingga dihasilkan daftar paragraf sementara dengan tipe data *list of string*, 2). menghapus paragraf kosong (" ") yang ada pada daftar paragraf, dan 3). menggabungkan paragraf pendek yang panjangnya kurang dari atau sama dengan 15 kata dengan paragraf selanjutnya. Paragraf pendek ini diasumsikan sebagai judul, subjudul, dan penjelasan dalam penomoran list. Sedangkan segmentasi teks ke dalam kalimat dilakukan dengan memanfaatkan fungsi *sent_tokenize(str)* dari pustaka NLTK (*Natural Language Toolkit*).

2.3. Pembangunan Fitur Stilometri

Berdasarkan segmen yang didapatkan, maka fitur yang diekstraksi dibedakan sesuai dengan segmennya. Fitur-fitur yang digunakan untuk segmentasi tingkat paragraf adalah frekuensi 4-gram kata, rata-rata jumlah tanda baca, panjang paragraf, dan rasio *type-token.* Sedangkan fitur yang diekstraksi dari segmen kalimat adalah frekuensi n-gram karakter dengan n={3,4,5}, panjang kalimat, dan jumlah tanda baca.

Fitur Frekuensi Quadgram Kata diekstraksi dengan memanfaatkan fungsi *word_tokenize(str*) dari pustaka NLTK dan membuang kata yang hanya berisi tanda baca. Kemudian, dengan fungsi ngrams(sequence, n) pada NLTK, sistem membuat dua jenis list bigram kata, yaitu list untuk tiap paragraf dan list untuk keseluruhan dokumen. Setelah itu, sistem menghitung frekuensi semua quadgram kata dengan fungsi FreqDist(list) pustaka NLTK. Lalu, sistem menghitung nd_1 tiap paragraf dibandingkan dengan keseluruhan dokumen, sehingga didapatkan perubahan gaya penulisan (*style change* atau *sc*) tiap paragraf *sc(i,D)* sesuai dengan persamaan (1) dan (2).

Untuk mengukur perubahan gaya penulisan, Stamatatos [10] memperkenalkan ukuran yang disebutnya sebagai *normalized dissimilarity measure* (nd_1) seperti pada rumus (1).

$$nd_1(A,B) = \frac{\sum_{g \in P(A)} \left(\frac{2(f_A(g) - f_B(g))}{f_A(g) + f_B(g)} \right)^2}{4|P(A)|} \quad (1)$$

dengan $f_A(g)$ dan $f_B(g)$ sebagai frekuensi kemunculan n-gram token g di teks A dan B, serta $|P(A)|$ adalah ukuran profil teks A. Profil merupakan vector yang terbentuk dari frekuensi semua n-gram yang muncul minimal sekali dalam suatu teks. Kemudian, fungsi perubahan gaya penulisan (*style change* atau *sc*) dalam dokumen *D* seperti pada rumus (2).

$$sc(i,D) = nd_1(w_i, D), i = 1 \dots |w| \quad (2)$$

dengan *w* adalah jendela. Dalam [10], dokumen dinyatakan bebas plagiasi apabila standar deviasi terhadap *sc* kurang dari 0,02. Sedangkan segmen$_i$ dokumen diduga merupakan hasil plagiasi apabila nilai SC memenuhi rumus (3).

$$sc(i', D) > M' + a * S' \quad (3)$$

sc(i',D) merujuk pada fungsi perubahan gaya penulisan setelah dikurangi segmen dengan nilai yang lebih tinggi dari M+S. Sedangkan *a* adalah parameter nilai sensitivitas terhadap metode deteksi plagiasi. Pada [10], nilai parameter ini ditentukan secara empiris dimana a=2.

Fitur rata-rata jumlah tanda baca dihasilkan dengan membagi jumlah tanda baca masing-masing paragraf dengan jumlah kalimat di paragraf tersebut. Sistem menganggap suatu karakter merupakan tanda baca apabila karakter tersebut menjadi elemen di himpunan *string.punctuation* milik pustaka *string*.

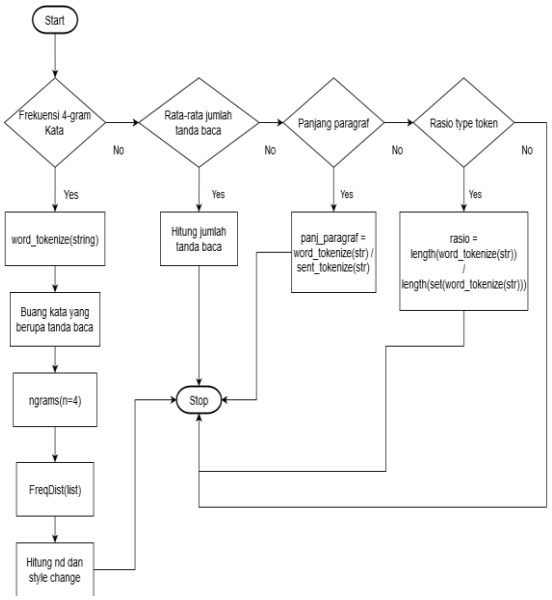
Fitur Panjang Paragraf dihasilkan dengan membagi jumlah token dengan jumlah kalimat. Jumlah token diperoleh dengan menghitung ukuran list hasil *word_tokenize()*. Perlu diketahui, tanda baca tidak diperhitungkan di sini. Sedangkan jumlah kalimat diperoleh dengan menghitung ukuran list (array) hasil segmentasi kalimat yang menggunakan fungsi *sent_tokenize(str)* pada NLTK.

Fitur Rasio Type-Token diperoleh dengan cara membagi jumlah *type* dengan jumlah *token*. Jumlah *type* diperoleh dengan mengubah list *word tokenize* ke struktur data *set*, sehingga didapatkan token unik. Kemudian, panjang *set* diukur sebagai jumlah *type*. Apabila hasil pembagian semakin mendekati angka 1, maka artinya kata-kata yang digunakan penulis semakin bervariasi.

Proses pembentukan fitur yang dijelaskan di atas adalah fitur stilometri untuk segmen paragraf yang diilustrasikan di Gambar 4. Sedangkan fitur stilometri yang digunakan untuk segmen kalimat adalah frekuensi n-gram karakter dengan n={3,4,5}, panjang kalimat, dan jumlah tanda baca.

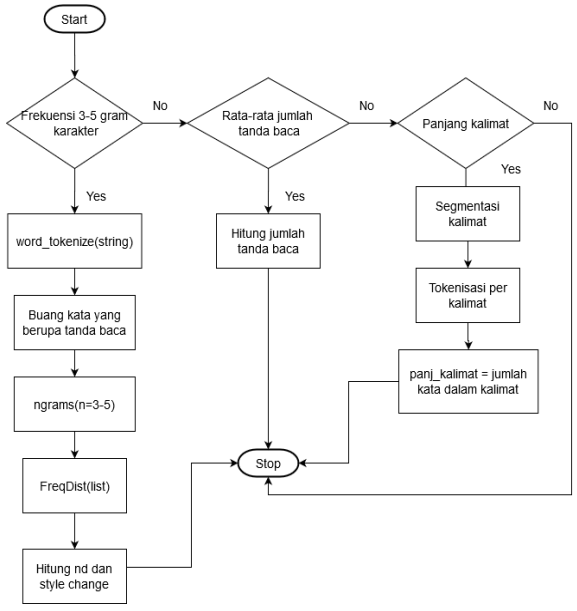
Fitur Frekuensi n-gram karakter. Setelah teks tersegmen kedalam kalimat, maka string kalimat dipecah menjadi karakter. Kemudian sistem mencari n-gram dengan n={3,4,5} menggunakan fungsi *ngrams(sequence, n)* pada NLTK dan menghasilkan dua list. List pertama diperuntukkan setiap kalimat, sedangkan list kedua untuk semua kalimat pada keseluruhan dokumen. Tiap n-gram karakter dihitung frekuensinya dengan fungsi *FreqDist(list)* pustaka NLTK. Kemudian, sistem menghitung nd_1 untuk masing-masing kalimat dibandingkan dengan keseluruhan dokumen, sehingga didapatkan *sc* tiap kalimat atau *sc(i,D)* sesuai dengan rumus (1) dan (2).

Fitur panjang kalimat didapatkan dengan menghitung jumlah token (kata) pada kalimat. Jumlah token dihitung dengan mengukur panjang list hasil *word tokenize* tanpa mengikutsertakan tanda baca.



Gambar 4. Diagram alir proses pembentukan fitur stilometri untuk segmen paragraf

Fitur Jumlah tanda baca. Berbeda dari fitur stilometri di tingkat paragraf yang menggunakan nilai rata-rata, maka di tingkat kalimat, digunakan jumlah total tanda baca saja. Komputasi jumlah tanda baca memiliki kemiripan dengan fitur rata-rata tanda baca, hanya di sini tidak dihitung reratanya. Untuk lebih jelasnya, Gambar 5 menunjukkan proses pembentukan fitur stilometri untuk segmen kalimat.



Gambar 5. Diagram alir pembentukan fitur stilometri untuk segmen kalimat.

2.4. Komputasi Outlier

Terdapat dua metode yang digunakan untuk menghitung *outlier* sesuai dengan fitur stilometri yang digunakan yakni fungsi Perubahan Gaya penulisan (sc) dan *Boxplot Outlier*. **Fungsi Gaya Perubahan Penulisan** (sc) digunakan untuk fitur frekuensi quadgram kata dan n-gram karakter. Sesuai dengan rumus (3), maka langkah perhitungannya adalah sebagai berikut: 1). menghitung rata-rata untuk *sc(i,D)* menggunakan fungsi *mean(list)* yang tersedia di pustaka *NumPy*, 2). menghitung standar deviasi untuk *sc(i,D)* dengan fungsi *std(list)* pada *NumPy*, 3). apabila standar deviasi kurang dari 0,02, maka dokumen tersebut bebas plagiasi. Namun, jika standar deviasi lebih atau sama dengan 0,02, maka lanjutkan ke langkah 4, 4). menghapus semua *sc(i,D)* yang lebih besar dari nila rata-rata ditambah standar deviasi, sehingga didapatkan *sc(i',D)*, 5). menghitung rata-rata dan standar deviasi untuk *sc(i',D)*, dan terakhir 6). mendefinisikan angka sensitivitas *a*=2,0, dan 7). Sesuai rumus (3), suatu bagian dokumen dianggap *outlier* bila nilai perubahan gaya penulisannya lebih besar dari rata-rata ditambah standar deviasi dikalikan dengan angka sensitivitas.

Fungsi Boxplot Outlier digunakan untuk fitur selain n-gram kata dan karakter. Langkah yang dilakukan yaitu: 1). menghitung Q1 dan Q3 dengan fungsi *percentile(list, percentile)* dari NumPy. List yang digunakan adalah list berisi fitur stilometri suatu segmentasi. Persentil (*percentile*) 25 digunakan untuk menghitung nilai Q1 dan persentil 75 digunakan untuk menghitung nilai Q3, 2). menghitung nilai IQR (Q3 dikurangi Q1), 3). menghitung *inner fences*. *Lower inner fences* didapat dari Q1 dikurangi dengan 1,5 IQR, sedangkan *upper inner fences* didapat dari Q3 ditambah 1,5 IQR, dan terakhir 4). melakukan pengecekan terhadap list berisi nilai vektor fitur stilometri. Nilai yang kurang dari *lower inner fences* atau lebih dari *upper inner fences* akan ditandai sebagai *outlier*.

Kedua fungsi ini akan mengembalikan *list* yang berisi nilai integer 1 dan 0, dengan angka 1 untuk data yang merupakan *outlier*, sedangkan angka 0 untuk data yang bukan *outlier*. List memiliki panjang yang sama dengan jumlah segmen, dimana setiap indeks pada list menunjukkan masing-masing segmen dokumen. Contohnya, suatu dokumen memiliki 4 paragraf dimana paragraf ke-1 dan ke-2 terdeteksi sebagai *outlier*, maka list yang terbentuk adalah [1, 1, 0, 0].

2.5. Voting Heuristik

Fitur stilometri dengan proses yang dijelaskan di sub bab sebelumnya tidaklah serta merta digunakan begitu saja. Dari berbagai fitur stilometri tersebut dikombinasikan dengan minimal jumlah kombinasi fitur adalah 3. Sehingga kombinasi yang terbentuk pada segmen paragraf adalah: frekuensi bigram kata, rata-rata jumlah tanda baca, panjang paragraf, frekuensi bigram kata, rata-rata jumlah tanda baca, rasio type-token, rata-rata jumlah tanda baca, panjang paragraf, rasio type-token, frekuensi bigram kata, rasio type-token, panjang paragraf, dan terakhir frekuensi bigram kata, rata-rata jumlah tanda baca, panjang paragraf, rasio type-token. Sedangkan untuk segmen kalimat, kombinasi yang

terbentuk hanya satu jenis, yaitu frekuensi bigram, trigram dan quadgram karakter, panjang kalimat, dan jumlah tanda baca.

Voting heuristik diterapkan pada tiap fitur stilometri yang nilainya diperhitungkan sebagai outlier dengan dua fungsi yang dijelaskan di sub bab D. Voting dilakukan dengan cara memberi nilai 1 pada tiap fitur yang terkategorikan sebagai *outlier*, kemudian nilai voting tersebut dijumlahkan untuk tiap fitur stilometrinya. Jika sebuah segmen paragraf atau kalimat memiliki nilai voting lebih tinggi dari nilai ambang yang ditetapkan, maka segmen paragraf atau kalimat tersebut diduga sebagai hasil plagiasi karena memiliki nilai perubahan gaya penulisan yang mencolok. Contoh perhitungan voting ditunjukkan pada Tabel 1, yang menampilkan teks dengan segmen paragraf. Dalam contoh tersebut teks hanya terdiri dari 4 paragraf, dan paragraf 2 memiliki voting tertinggi karena memiliki 2 fitur stilometri yang terdeteksi sebagai *outlier*. Untuk mencatat hasil akhir voting tiap segmen teks, sistem kemudian menggunakan tipe data *list* dengan elemen bilangan biner. Variabel list ini diberi nilai 1 jika segmen teks merupakan outlier dan 0 jika bukan. Sebagai contoh, hanya kolom hasil voting di Tabel I ini yang ditampung dalam variabel luaran ini dengan nilai [0, 1, 0, 0]. Dengan demikian untuk seluruh dokumen, dihasilkan 2d list integer.

Tabel 1
Contoh Penghitungan Voting Heuristik

Paragraf	Frekuensi bigram kata	Rata-rata jumlah tanda baca	Panjang paragraf	Nilai Voting
1	0	0	1	1
2	1	0	1	2
3	0	0	0	0
4	1	0	0	0

2.6. Skenario Pengujian

Stein, Lipka, dan Prettenhofer [17] mendefinisikan tugas utama algoritma Deteksi Plagiasi Intrinsik (DPI) sebagai berikut: "dengan diberi sebuah dokumen input saja, sistem harus mampu menemukan potongan teks yang dicurigai bukan ditulis oleh pengarang atau diduga plagiat" [17]. Berdasarkan kriteria ini, maka data yang dikumpulkan adalah dokumen uji yang ditulis oleh seorang penulis tunggal. Selain itu teks yang digunakan sebagai dokumen uji adalah teks yang ditulis oleh penulisnya tanpa melalui proses pengeditan oleh editor. Hal ini bertujuan supaya gaya penulisan dapat tampak pada teks tersebut. Data teks yang digunakan diperoleh dari situs https://www.kompasiana.com / dan karya tulis mahasiswa. Selengkapnya, data statistik tentang data uji ditampilkan di Tabel 2.

Untuk keperluan pengujian, Setiap teks disisipi beberapa paragraf atau kalimat dari teks lain yang ditulis oleh penulis yang berbeda sebagai paragraf atau kalimat yang terindikasi plagiat. Paragraf atau kalimat ini yang nantinya harus dapat dideteksi oleh sistem. Bagian plagiat tiap teks dicatat pada dokumen label dalam xml dengan tingkat granularitas kalimat. Pada contoh label di Gambar 6, label memiliki empat elemen <plagiarized> di dalam elemen <plagiarisms>. Artinya, teks "Banjir di Tahun Baru" ini mengandung empat kalimat plagiasi. Kalimat-kalimat ini berada di dua paragraf yang berbeda, yaitu paragraf id 19 dan 20. Paragraf dengan nomor id 19 memiliki tiga kalimat plagiasi, yaitu kalimat dengan nomor id 71 hingga 73. Sedangkan paragraf id 20 mengandung satu kalimat plagiasi, yaitu kalimat id 74.

Tabel 2
Data Statistik Tentang Dokumen Uji

Hal	Jumlah	Satuan
Genre teks		
- Artikel Populer	30	buah
- Artikel ilmiah	1	buah
Panjang teks		
- maksimal	2.235	kata
- minimal	320	Kata
- rata-rata	888	kata

```
<?xml version="1.0" encoding="UTF-8"?>
<testdoc>
  <docinfo>
    <docname>Banjir di Tahun Baru</docname>
    <docsource>
https://www.kompasiana.com/katedrarajawen/5e1559fbd541df230e3e4
1a2/tahun-baru-banjir-dan-melepaskan?page=1</docsource>
    <totalpar>22</totalpar>
    <totalsent>78</totalsent>
  </docinfo>
  <plagiarisms>
    <plagiarized parId="19" sentId="71" source=
"https://www.kompasiana.com/posmasiahaan/5e1f5995d541df18e517c2
32/pengennya-liburan-tahun-baruan-di-jakarta-malah-stress-keban
jiran"/>
    <plagiarized parId="19" sentId="72" source=
"https://www.kompasiana.com/posmasiahaan/5e1f5995d541df18e517c2
32/pengennya-liburan-tahun-baruan-di-jakarta-malah-stress-keban
jiran"/>
    <plagiarized parId="19" sentId="73" source=
"https://www.kompasiana.com/posmasiahaan/5e1f5995d541df18e517c2
32/pengennya-liburan-tahun-baruan-di-jakarta-malah-stress-keban
jiran"/>
    <plagiarized parId="20" sentId="74" source=
"https://www.kompasiana.com/posmasiahaan/5e1f5995d541df18e517c2
32/pengennya-liburan-tahun-baruan-di-jakarta-malah-stress-keban
jiran"/>
  </plagiarisms>
</testdoc>
```

Gambar 6. Contoh label untuk teks "Banjir di Tahun Baru"

Pengolahan data label, seperti yang ditunjukkan di Gambar 6, dilakukan dengan membuat list label dari data label untuk mempermudah komputasi berikutnya. Ada dua list yang dihasilkan, yaitu list label tingkat paragraf dan tingkat kalimat. Pada contoh Gambar 6, dokumen memiliki 22 paragraf. Paragraf indeks 19 dan 20 dilabeli sebagai plagiat karena diambil dari penulis lain. Maka, sistem akan membentuk list label tingkat paragraf [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0]. Begitu pula untuk label tingkat kalimat.

List label tersebut nantinya akan dibandingkan dengan list luaran sistem. Kemudian nilai Akurasi, Presisi, Sensitivitas, dan skor F-1 akan dihitung berdasarkan perbandingan dari kedua list tersebut. Keempat ukuran ini dihitung dengan menggunakan fungsi-fungsi yang tersedia di pustaka *SK-learn*. Sebagai contoh, nilai F-1 dihitung dengan cara memanfaatkan fungsi

f1_score(y_true, y_pred, average) dari pustaka *Scikit-learn*. Dengan parameter "y_true" adalah list label dokumen, "y_pred" adalah list prediksi sistem. Perhitungan dilakukan untuk keenam jenis kombinasi. Sehingga hasil *F1 score* untuk satu dokumen berupa *list of float* dengan ukuran enam elemen. Setiap indeks list digunakan sebagai representasi kombinasi. Contohnya bila list *F1 score* adalah [0,28; 0,56; 0,26; 1,00; 1,00; 0,53], maka: 1). indeks 0 (kombinasi 1): frekuensi bigram kata, rata-rata jumlah tanda baca, panjang paragraf. Pada contoh diatas, nilai F-1 kombinasi 1 adalah 0,28. 2). indeks 1 (kombinasi 2): frekuensi bigram kata, rata-rata jumlah tanda baca, rasio *type-token* memiliki skor F-1 sebesar 0,56. 3). indeks 2 (kombinasi 3): Rata-rata jumlah tanda baca, panjang paragraf, rasio *type-token* memiliki nilai F-1 sebesar 0,26. 4). indeks 3 (kombinasi 4): Frekuensi bigram kata, rasio *type-token*, panjang paragraf. Pada contoh diatas, skor F-1 bernilai 1,00. 5). indeks 4 (kombinasi 5): Frekuensi bigram kata, rata-rata jumlah tanda baca, panjang paragraf, rasio *type-token* menunjukkan nilai F-1 sebesar 1,00. Terkahir, 6). indeks 5 (kombinasi 6): Frekuensi n-gram karakter, panjang kalimat, jumlah tanda baca. Pada contoh diatas, skor F-1 kombinasi ini bernilai 0,53.

Penentuan kombinasi fitur terbaik di setiap dokumen ditentukan dengan proses yang dijabarkan dalam langkah-langkah sebagai berikut: 1). ambil satu list skor F-1 dari sebuah dokumen. 2). cari indeks yang memiliki angka skor F-1 terbesar menggunakan *max(list)* pada pustaka *NumPy*, 3). buat list dengan ukuran 6 elemen (sejumlah kombinasi yang digunakan). Isi list dengan angka 1 pada indeks yang sama dengan langkah di nomor 2, sedangkan indeks lain diisi dengan angka 0. Masing-masing indeks merepresentasikan kombinasi terbaik (angka 1) dan bukan terbaik (angka 0), 4). ulangi langkah 1 apabila masih ada list F-1 yang lain. Terakhir, 5). hasil semua list pada langkah 3 ditampung dalam satu list, sehingga terbentuk 2d list integer. Contohnya, list skor F-1 untuk sebuah dokumen adalah [0,28; 0,56; 0,26; 1,00; 1,00; 0,53], maka hasil pencarian kombinasi terbaik adalah [0, 0, 0, 1, 1, 0].

Komputasi perhitungan voting heuristik kombinasi terbaik untuk teks dokumen berbahasa Indonesia dilakukan dengan cara sebagai berikut: buat list dengan panjang 6 elemen dan isi dengan angka 0. List nantinya digunakan untuk menyimpan perolehan voting. Tiap indeks merepresentasikan satu jenis kombinasi. Ambil satu list dari 2d list yang dihasilkan pada penentuan kombinasi terbaik tiap dokumen sebelumnya. Jumlahkan setiap data pada list yang dibuat di langkah 1 dengan data pada list langkah 2. Simpan hasil penjumlahan pada list yang dibuat di langkah 1. Jika masih ada list dalam 2d list penentuan kombinasi terbaik tiap dokumen, kembali lagi ke langkah 2. Cari indeks list yang memperoleh voting tertinggi dengan fungsi *max(list)* dari pustaka *NumPy* untuk list yang telah dibuat di langkah 1. Masukkan indeks voting tertinggi ke sebuah list baru.

Contohnya, 2d list integer yang didapatkan dari 3 dokumen pada penentuan kombinasi terbaik tiap dokumen yaitu [[0, 0, 1, 0, 0, 0], [0, 0, 1, 0, 0, 0], [0, 0, 1, 0, 0, 0]]. Sehingga, perolehan votingnya adalah [0, 0, 3, 0, 0, 0]. Karena voting tertinggi ada pada indeks ke-2, maka list yang dihasilkan oleh sistem adalah [2]. Indeks ke-2 merepresentasikan kombinasi 3, yaitu kombinasi rata-rata jumlah tanda baca, panjang paragraf, rasio type-token. Kombinasi itulah yang dianggap sebagai sebagai kombinasi yang paling baik dalam sistem DPI teks berbahasa Indonesia ini.

3. Hasil dan Pembahasan

Hasil dan pembahasan berdasarkan metode penelitian dan pengujian yang telah dilakukan sebelumnya dijabarkan dalam dua sub bab berikut ini.

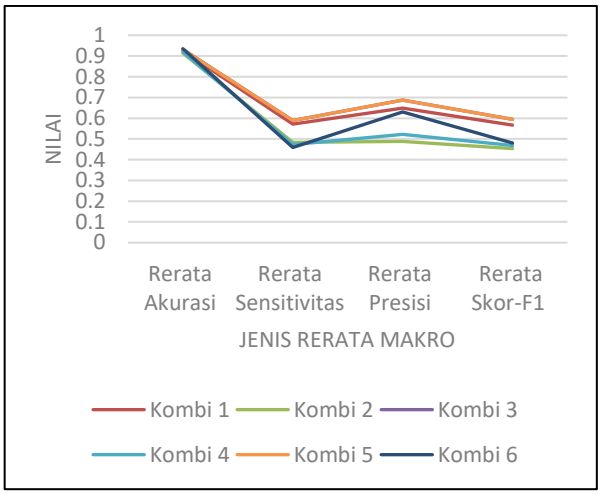
3.1. Hasil Pengujian

Pada setiap dokumen, pengujian dilakukan dengan menghitung nilai evaluasi untuk enam jenis kombinasi yang ada. Kombinasi yang memiliki nilai skor F-1 tertinggi akan dianggap sebagai kombinasi yang paling bagus untuk dokumen tersebut. Selain skor F-1, Sensitivitas, *presisi*, dan akurasi juga ditampilkan. Nilai F-1 sebenarnya merupakan nilai-rata-rata harmoni dari nilai Presisi dan Sensitivitas. Hasil evaluasi sistem ditampilkan di Gambar 7 yang menunjukkan bahwa nilai F-1 tertinggi mencapai 0.59 yang diperoleh oleh kombinasi fitur 3 dan 5. Nilai akurasi dari semua kombinasi fitur mencapai lebih dari 0.9 dan cenderung jauh lebih tinggi dari nilai Presisi. Ini disebabkan oleh dimasukkannya jumlah *True Negative* dalam perhitungan metrik Akurasi, sehingga bila jumlah *True Negative* besar, maka nilai Akurasi otomatis akan tinggi.

Selain mengevaluasi sistem dengan 4 metrik, maka kami mencoba untuk mengamati kombinasi fitur stilometri mana yang menghasilkan nilai evaluasi terbaik. Dengan proses komputasi yang dijelaskan di sub bab II.F, maka hasil nilai voting kombinasi fitur stilometri tertinggi diperoleh kombinasi 3 (rata-rata jumlah tanda baca, panjang paragraf, rasio *type-token*) dan kombinasi 5 (frekuensi bigram kata, rata-rata jumlah tanda baca, panjang paragraf, rasio *type-token*). Kedua kombinasi ini menggunakan segmentasi paragraf. Hasil voting per kombinasi fitur stilometri yang digunakan bisa dilihat di Tabel 3.

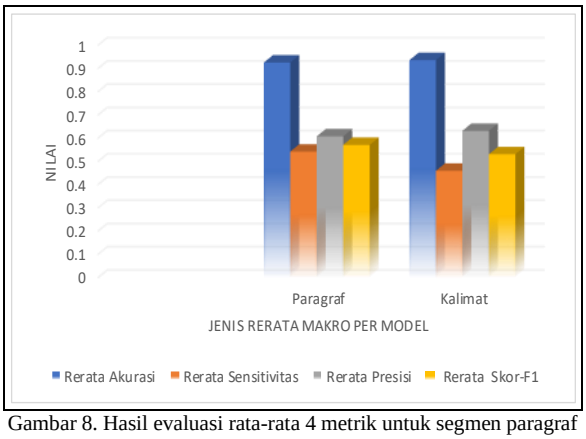
Tabel 3
Hasil Voting Kombinasi Terbaik Untuk Keseluruhan Dokumen

Kombinasi	Total	Kombinasi	Total
Kombinasi 1	14	Kombinasi 4	12
Kombinasi 2	10	Kombinasi 5	15
Kombinasi 3	15	Kombinasi 6	11



Gambar 7. Hasil pengujian sistem DPI dengan 4 metrik ukuran pengujian

Keempat metric evaluasi diterapkan juga untuk mengamati nilai rata-rata makro akurasi, Sensitivitas, presisi, dan skor F-1 untuk tingkat granularitas segmen paragraf dan kalimat. Hasil penghitungannya bisa dilihat di Gambar 8 yang menunjukkan bahwa segmen paragraf memiliki nilai F-1 (0.57) dan sensitivitas (0.54) yang lebih tinggi dibandingkan segmen kalimat. Namun nilai Presisi di segmen kalimat (0.63) melampaui nilai presisi tingkat paragraf. Nilai akurasi segmen kalimat mencapai 0.94 dan 0.92 untuk segmen paragraf.



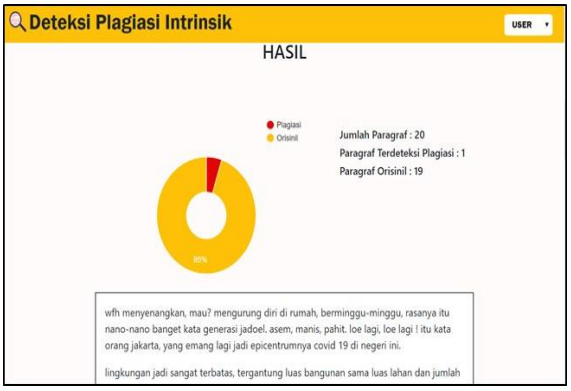
Gambar 8. Hasil evaluasi rata-rata 4 metrik untuk segmen paragraf dan kalimat

Khusus segmen kalimat, dilakukan evaluasi fitur n-gram karakter untuk mengetahui n-gram dengan panjang n yang mana yang memberikan hasil maksimal. Empat metric evaluasi diterapkan juga dalam hal ini, dan hasilnya menunjukkan bahwa n=4 atau quadgram karakter memberikan hasil yang lebih tinggi daripada trigram dan pentagram. Ini ditunjukkan dengan nilai rerata akurasi quadgram yang mencapai 0.94 dan rerata presisi yang mencapai 0.63. Hasil evaluasi sepenuhnya bisa dilihat di Tabel 4.

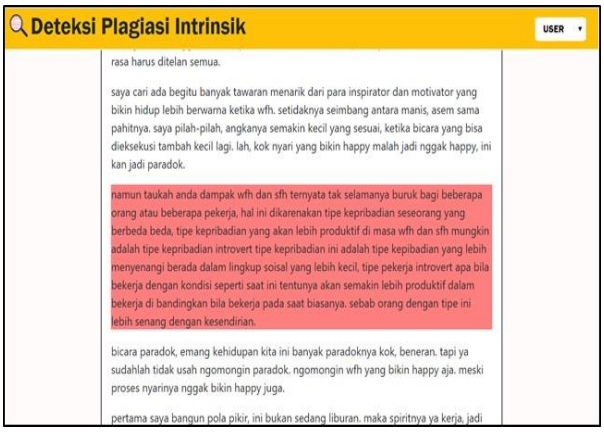
Luaran sistem berupa segmen ditampilkan ke pengguna dalam bentuk diagram yang menunjukkan persentase segmen teks yang diduga plagiat atau mengalami perubahan gaya penulisan dan segmen tersebut diberi penekanan warna yang berbeda. Tampilan antarmuka sistem DPI ini dirancang menggunakan *Django* yang merupakan *framework Python* untuk membuat aplikasi web yang dinamis. Gambar 9 dan 10 menunjukkan tampilan antarmuka sistem DPI berbasis Web ini.

Tabel 4
Hasil Pengujian N-Gram Yang Berbeda

n	Rerata Akurasi	Rerata Sensitivitas	Rerata Presisi	Rerata Skor-F1
n=3	0.93	0.46	0.61	0.52
n=4	0.94	0.46	0.63	0.53
n=5	0.93	0.46	0.59	0.52



Gambar 9. Visualisasi luaran dengan diagram Donat



Gambar 10. Visualisasi luaran dengan pewarnaan di segmen paragraf.

3.2. Pembahasan

Berdasarkan hasil pengujian yang dilakukan, strategi segmentasi tingkat paragraf menunjukkan hasil yang lebih baik dari segmentasi kalimat yang ditunjukkan dengan nilai F-1 yang lebih tinggi serta selisih nilai Presisi dan Sensitivitas yang tidak terpaut jauh. Ini dikarenakan segmen paragraf memiliki ukuran jendela (*window*) yang lebih lebar dibandingkan kalimat. Besarnya ukuran jendela ini membuat program dapat lebih mengenali gaya penulisan seseorang karena data statistik yang diperoleh sistem DPI lebih banyak.

Namun segmentasi tingkat paragraf juga memiliki kekurangan. Karena granularitasnya besar, maka sistem tidak dapat mendeteksi bagian kecil paragraf yang diduga plagiat. Apabila suatu paragraf mengandung bagian plagiasi yang tidak mendominasi paragraf tersebut, sebagai contohnya 1 kalimat di antara 8 kalimat, maka paragraf tersebut tidak terdeteksi sebagai *outlier*. Sehingga paragraf tersebut dianggap paragraf orisinil oleh sistem.

Berdasarkan hasil eksperimen, kombinasi terbaik adalah kombinasi fitur 3 (KF3) dan kombinasi fitur 5 (KF5). Perbedaan pada kedua kombinasi fitur ini adalah adanya frekuensi bigram kata pada KF5. Selain itu, semua fitur pada kedua kombinasi ini sama. Namun, Tabel 3 menunjukkan bahwa KF5 memiliki nilai voting yang sama dengan KF3. Nilai evaluasi untuk KF3 dan KF5 selalu sama untuk semua dokumen uji. Dari sini, dapat disimpulkan bahwa fitur frekuensi bigram kata tidak terlalu berpengaruh dalam mengenali gaya penulisan. Dengan kata lain, kombinasi fitur stilometri yang paling praktis digunakan dalam sistem DPI ini adalah rata-rata jumlah tanda baca, panjang paragraf, dan rasio type-token (KF3).

Frekuensi bigram kata yang tidak terlalu berpengaruh ini kemungkinan dikarenakan gaya penulisan seseorang tidak dapat dilihat dari angka frekuensi. Contohnya, penuF-1yA memiliki kebiasaan menulis “aku adalah” tiga kali dalam tiap paragraf, sedangkan penulis B menulis “aku merupakan” tiga kali dalam paragraf. Saat sebuah paragraf dari penulis B disisipkan di teks penulis A, frekuensi bigram kata yang terhitung akan sama, yaitu 3. Sistem gagal mendeteksi paragraf penulis B sebagai *outlier*. Dengan demikian penelitian ini menyarankan untuk menggunakan fitur n-gram berbasis frekuensi tertingginya sebagai ganti frekuensi n-gramnya.

Sistem DPI bekerja dengan cara meniru manusia untuk mengenali bagian teks yang berbeda dilihat dari gaya penulisannya. Namun, tidak semua jenis teks cocok untuk dideteksi oleh sistem DPI yang dibangun. Jenis teks yang tidak cocok dan sulit dideteksi melalui sistem DPI yaitu teks yang terlalu pendek, teks yang ditulis oleh beberapa penulis, dan teks yang mengandung terlalu banyak kutipan dari berbagai sumber namun tulisan orisinil penulis memiliki porsi yang sangat kecil.

Dokumen teks yang terlalu pendek sulit dideteksi oleh sistem DPI karena tidak memiliki segmen yang cukup untuk dibandingkan dengan segmen lainnya dalam teks tersebut. Selain itu, sistem DPI ini tergantung pada data statistic fitur stilometri yang ada, dan jika data statistiknya tidak mencukupi untuk dibandingkan, maka sistem tidak mampu menemukan segmen teks yang berbeda (*outlier*).

Sistem DPI didasari oleh asumsi bahwa penulis asli menulis sebagian besar teks dalam dokumennya. Sehingga jika teks yang diuji mengandung kutipan atau plagiasi yang lebih dominan dibanding bagian orisinil, hasil deteksi sistem akan terbalik. Ini dibuktikan dengan hasil uji dokumen bergenre karya ilmiah. Pada dasarnya hasil deteksi sistem telah benar bahwa tulisan asli penulis karya ilmiah tersebut merupakan *outlier* dari sebagian besar potongan teks lainnya yang merupakan kutipan dari teks lain. Namun sistem mendeteksi bagian asli yang ditulis penulis sebagai outlier atau diduga plagiat. Dalam kasus seperti ini, intervensi manusia sangat diperlukan dalam menentukan porsi plagiasi, sehingga kasus terbalik seperti hasil uji coba tersebut tidak mengecoh.

Solusi yang dapat dijadikan pertimbangan untuk penelitian selanjutnya untuk meminimalisir kejadian deteksi terbalik ini adalah penambahan modul ekstraksi sitasi dan sumber rujukan pada pra-pemrosesan. Setelah dideteksi, kalimat atau paragraf yang hasil sitasi ini tidak ikut dalam komputasi. Tujuannya adalah supaya teks yang orisinil ditulis oleh penulis dapat lebih mendominasi. Sehingga jika ada bagian teks hasil plagiasi (tidak mencantumkan sumber rujukan) akan lebih memungkinkan terdeteksi sebagai *outlier*.

4. Kesimpulan

Berbeda dari penelitian sistem DPI sebelumnya yang menggunakan teks berbahasa Inggris, objek dari penelitian ini adalah teks berbahasa Indonesia. Penelitian terdahulu di bidang deteksi plagiasi dengan teks berbahasa Indonesia berkutat di Deteksi Plagiasi Eksternal (DPE), sehingga kontribusi yang diberikan oleh penelitian ini adalah pengujian beberapa fitur stilometri serta penentuan segmen teks yang terbaik untuk sistem DPI berbahasa Indonesia. Hasil eksperimen terhadap 6 kombinasi fitur dengan 2 granularitas segmen teks menunjukkan bahwa granularitas paragraf memiliki kinerja lebih baik daripada granularitas kalimat. Skor F-1 granularitas paragraf mencapai 0.57 dan nilai akurasinya sebesar 0.92 yang terpaut 0.02 dari nilai akurasi granularitas kalimat. Sedangkan kombinasi fitur stilometri yang cocok digunakan dalam PDI teks berbahasa Indonesia adalah rata-rata jumlah tanda baca, panjang paragraf, dan rasio *type-token*. Kombinasi fitur ini dapat menghasilkan nilai 0,59 untuk rata-rata skor F-1 dan 0.93 untuk rata-rata nilai akurasinya.

Jika diamati lebih seksama, hasil nilai akurasi, sensitivitas, presisi dan F-1 antara segmen paragraf dan kalimat tidak terpaut jauh. Segmen kalimat memiliki nilai presisi (0.63) dan akurasi (0.94) yang lebih tinggi daripada segmen paragraf. Bagi pengembang perangkat lunak yang mementingkan nilai presisi dari pada keseimbangan antara nilai sensitivitas dan presisi (skor F-1), maka segmen kalimat akan lebih cocok. Selain itu, segmen kalimat memberikan hasil deteksi dengan granularitas yang lebih halus sekalipun memiliki kecenderungan bias yang lebih tinggi juga.

Hasil eksperimen menunjukkan bahwa tidak semua jenis teks cocok untuk dideteksi oleh sistem PDI. Untuk teks yang memiliki kutipan lebih dominan dibanding bagian yang orisinil dapat membuat hasil deteksi terbalik, yakni tulisan si penulisan diduga sebagai hasil plagiasi karena terdeteksi sebagai outlier dengan gaya penulisan yang berbeda dari sebagian besar teks hasil salinan langsung (CoPas). Maka dari itu, hasil deteksi sistem DPI sebaiknya digunakan sebagai rekomendasi bagi penggunanya dalam menemukan bagian teks yang gaya penulisannya berbeda. Sistem DPI ini masih memerlukan intervensi manusia dalam memverifikasi hasil deteksinya. Untuk memperbaiki kinerja sistem DPI, sebaiknya digunakan modul ekstraksi informasi sumber rujukan dan menggabungkan fitur stilometri dengan data statistiknya secara bersamaan sebagai profil tiap segmen teks. Sistem DPI yang mengandalkan kuantifikasi (data statistik) fitur stilometri saja akan memberikan hasil yang kurang memuaskan.

Daftar Rujukan

[1] Halvani, O., 2015. Register & Genre Seminar: Towards Intrinsic Plagiarism Detection, *Citeseer*, Darmstadt.

[2] A. Rexha, M. Kröll, H. Ziak and R. Kern, 2018. Authorship identification of documents with high content similarity, *Scientometrics*, vol. 115, p. 223–237

[3] Stamatatos, E., Tschuggnall, M., Verhoeven, B., Daelemans, W., Specht, G., Stein, B., and Potthast, M., 2016. Clustering by Authorship Within and Across Documents, in *PAN CLEF 2016 Evaluation Labs and Workshop – Working Notes Papers*, Évora, Portugal.

[4] Kuznetsov, M., Motrenko, A., Kuznetsova, R., and Strijov, V., Methods for Intrinsic Plagiarism Detection and Author Diarization. *CLEF 2016 Evaluation Labs and Workshop – Working Notes Papers*,, Évora, Portugal.

[5] Foltýnek, T., Meuschke, T., and Gipp, B., 2019. Academic Plagiarism Detection: A Systematic Literature Review, *ACM Computing Survey*, vol. 52, no. 6, pp. 1-42.

[6] Haryanto, N., Krisnawati , L.D., and Chrismanto, A.R., 2020. Temu Kembali Dokumen Sumber Rujukan Dalam Sistem Daur Ulang Teks. *Jurnal Teknologi dan Sistem Komputer*, vol. 8, no. 2, pp. 140-150.

[7] Chowdhury, H., and Bhattacharya, D., 2016. Plagiarism: Taxonomy, tools and detection techniques. *Proceedings of the 19th National Convention on Knowledge, Library and Information Networking (NACLIN'16)*.

[8] Krisnawati, L.D., 2016. *Plagiarism Detection for Indonesian Tex*t. Ph.D. Mumchem:Ludwig-Maximilians-Universität.

[9] Eissen, S., and Stein, B., 2006. Intrinsic Plagiarism Detecion. *ECIR 2006, LNCS 3936.*

[10] Stamatatos, E., 2009. Intrinsic Plagiarism Detection Using Character n-gram Profiles. *SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN 2009)*, vol. 2, pp. 38-46.

[11] Rahman, R., 2015. Information Theoretical and Statistical Features for Intrinsic Plagiarism Detection. *Proceedings of the SIGDIAL 2015 Conference*. Prague, czech Republic.

[12] Krause, M., 2015. Stylometry-based Fraud and Plagiarism Detection for Learning at Scale. *5th KSS Workshop*. Karlsruhe, Germany.

[13] Elamine, M., Mechti, S., and Belguith, L., 2017. Intrinsic Detection of Plagiarism based on Writing Style Grouping. *LPKM2017, Computer Science, Pshychology*..

[14] Sunardi, Yudhana, A., and Mukaromah, I., 2017. Perancangan Aplikasi Deteksi Plagiarisme Karya Ilmiah Menggunakan Algoritma Winnowing. *Seminar Nasional Serba Informatika*. Samarinda, Indonesia.

[15] Bianto, M., Rahayu, I., Huda, M., and Kusrini, 2018. Perancangan Sistem Deteksi Plagiarisme Terhadap Topik Penelitian Menggunakan Metode K-Means Clustering dan Model Bayesian. *Seminar Nasional Teknologi Informasi dan Multimedia*. Yogyakarta, Indonesia.

[16] Ratna, A., Purnamasari, P., Adhi, B.A., Ekadiyanto, F.A., Salman, M., Mardiyah and Winata, D,J., 2017. Cross-Language Plagiarism Detection System Using Latent Semantic Analysis and Learning Vector Quantization. *Algorithms*, vol. 10, no. 69, pp. 1-14.

[17] Stein, B., Lipka, N., and Prettenhofer, P., 2011. Intrinsic plagiarism analysis. *Language Resources and Evaluation*, vol. 45, no. 1, pp. 63-82.