Jurnal Teknik Informatika Vol. 14 No. 2, Oktober 2021
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v14i2.22841

JURNAL TEKNIK INFORMATIKA
Homepage : http://journal.uinjkt.ac.id/index.php/ti

PENDEKATAN SEMANTIK DALAM DETEKSI BERBAGAI TIPE PLAGIARISME PADA DOKUMEN TEKS

Surya Agustian[1]

[1]Teknik Informatika, Fakultas Sains dan Teknologi
[1]Universitas Islam Negeri Sultan Syarif Kasim Riau
[1]Jl. H.R. Soeberantas km 11,5 Simpang Baru Panam, Pekanbaru, Riau
E-mail: [1]surya.agustian@uin-suska.ac.id

Artikel:
Diterima: 18 Oktober, 2021
Direvisi: 06 Januari, 2022
Diterbitkan: 10 Januari, 2022

***Alamat Korespondensi:**
surya.agustian@uin-suska.ac.id

ABSTRACT

Plagiarism detection is a complex task. In-text, it should be able to find fragments of a text that is suspected of being illegally plagiarized from other sources. Aligning the plagiarized passages of suspicious documents from the source document is an issue that was discussed a lot, of which we can measure the percentage of the plagiarized text. This research proposes a semantic approach of text (fragments in documents) alignment between source and suspicious documents, using Jackard similarity method. Experimental results on the PAN competition for plagiarism detection competition, yielding average of 66.9% detection scores, increased more than twice if compared to the baseline method provided by the organizer, which is 28,4%. This approach is potential as a starting point to find offset match and length of plagiarized text in a plagiarism detection system.

Keywords: *Plagiarism Detection, Text Alignment, Semantic, Similarity*

Jurnal Teknik Informatika Vol. 14 No. 2, Oktober 2021
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v14i2.22841

ABSTRAK

Deteksi plagiarisme merupakan suatu task yang kompleks. Dalam teks, ia harus dapat menemukan fragmen dari suatu teks yang dicurigai diplagiasi dari sumber lain secara tidak sah. Menyelaraskan bagian teks yang diplagiasi dari suatu dokumen yang dicurigai dengan dokumen sumbernya merupakan suatu problem yang banyak didiskusikan, karena berdasarkan pencocokan inilah kita dapat mengukur berapa persen kasus plagiarisme pada teks. Penelitian ini mengusulkan pendekatan semantik untuk penyelarasan posisi *passage* (fragmen pada dokumen teks) di antara dokumen sumber dan dokumen yang dicurigai, berdasarkan metode Jackard similarity. Hasil eksperimen pada dataset PAN *shared task* menunjukkan skor deteksi plagiarisme rata-rata keseluruhan adalah 66,9%, meningkat lebih dari 2 kali lipat dibandingkan dengan metode *baseline* yang disediakan oleh organizer, yaitu sebesar 28,4%. Pendekatan semantik ini sangat potensial sebagai langkah awal untuk menemukan posisi ofset dan panjang teks yang diplagiasi dalam sistem deteksi plagiarisme.

Kata Kunci: *Plagiarisme, Penyelarasan Teks, Semantik, Similaritas*

I. PENDAHULUAN

Pertumbuhan teknologi informasi yang sangat cepat telah berakibat pada produksi dokumen teks, gambar, dan video dengan jumlah yang sangat massif di internet. Khusus untuk dokumen teks, pertumbuhan jumlah dokumen yang besar tersebut bisa sangat bervariasi, baik dari bentuk (artikel, rubrik, laporan, publikasi ilmiah, dan sebagainya) maupun topiknya, mulai dari hukum, ekonomi, teknik, hobi, politik, dan apa saja.

Dokumen teks yang di-*share* di internet dapat bernilai ekonomis, baik secara langsung maupun tidak langsung. Salah satu nilai ekonomis dari dokumen teks, bisa berbentuk patent maupun hak cipta dari suatu laporan ilmiah yang dikembangkan dari penelitian yang lama. Bisa juga berupa teori-teori, formula, dalil, yang disimpulkan setelah melakukan sejumlah survey, pengumpulan data, bahan dan peralatan, eksperimen, yang semuanya menghabiskan biaya yang mahal untuk memperolehnya.

Selain itu, dokumentasi proyek, model suatu barang produksi, dan karya sastra seperti novel dan lagu, juga tidak terlepas dari kekayaan yang dilindungi hak cipta. Apabila ada pihak lain yang ingin mengutip sebagian dari laporan tersebut, harus menggunakan tata cara perujukan (sitasi) yang layak, sampai izin tertulis dari pemilik dokumen agar kegiatan pengutipan tersebut tidak melanggar kaidah penulisan ilmiah. Pengutipan tanpa izin maupun tanpa menyebutkan sumbernya, dapat dianggap sebagai tindakan plagiarisme.

Menurut kamus Merriam-Webster[1], plagiarisme merupakan kegiatan mencuri dan mengakui ide, perkataan, tulisan atau lainnya milik orang lain sebagai miliknya sendiri, atau menggunakan hasil produk orang lain tanpa menyebutkan sumbernya. Di dalam literasi tertulis, plagiarism adalah perbuatan mencuri literasi, yaitu menampilkan suatu ide sebagai sesuatu yang baru dan orisinil, padahal diturunkan dari sumber-sumber yang telah ada.

Sedangkan menurut Dictionary.com[2], plagiarism adalah suatu tindakan menggunakan atau sengaja meniru dari bahasa atau pemikiran seseorang tanpa autorisasi dan mengakui karya orang lain sebagai miliknya, tanpa memberi kredit yang layak kepada pemilik aslinya.

Taksonomi dari plagiarism dibagi ke dalam dua tipe prinsip deteksi keterhubungan, yaitu salinan utuh (*verbatim* atau *exact copy*) dan salinan yang dimodikasi (*modified copy*), misalnya dengan mengubah kalimat, maupun mengubah kata-kata atau paraphrase ([12] dalam [11]).

[1] https://www.merriam-webster.com/

[2] https://www.dictionary.com/

Jurnal Teknik Informatika Vol. 14 No. 2, Oktober 2021
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v14i2.22841

Modified copy lebih disukai daripada *exact copy*, karena lebih sulit untuk dideteksi oleh sistem anti plagiarisme. Pelaku plagiat akan menambah, mengurangi beberapa kata atau menggantinya dengan sinonim kata atau frase (dikenal dengan teknik *paraphrase*). Lebih lanjut, frase-frase yang jarang dipakai, dicari dari kamus *thesaurus* untuk digunakan sebagai pengganti kata-kata dalam teks. Akhirnya, teknik ini akan menyulitkan sistem pendeteksi plagiarisme untuk menemukan bagian teks yang diplagiasi, serta di mana posisinya di dalam dokumen sumber.

Beberapa metode dan algoritma telah diperkenalkan untuk mendeteksi plagiarisme. Sistem SCAM diusulkan untuk mendeteksi subset atau bagian dari dokumen-dokumen yang diplagiat [1] berdasarkan frekuensi kata. Ia dikembangkan untuk meningkatkan beberapa kelemahan dari sistem COPS [2] yang kurang bagus performanya ketika mendeteksi kalimat.

Metode Winnowing yang pertama kali diperkenalkan dalam [3], bekerja berdasarkan pembentukan *window-window* dan mengekstraksi *fingerprint* (sidik jari) dari setiap *window* tersebut, berupa kata-kata yang dipilih untuk mewakili *window* tersebut. Metode ini diklaim efisien, bahwa kecocokan pada panjang tertentu dijamin dapat terdeteksi.

Sejak saat itu, *winnowing* cukup populer digunakan untuk klasifikasi dan klasterisasi teks, seperti mengusulkan sistem klasterisasi berbasis *winnowing* [4], dan [5] yang menggunakan Hartigan Index untuk mengelompokkan klaster dokumen sebelum menghitung kemiripan atau *similarity* dari *fingerprint winnowing*-nya.

Metode *winnowing* dikembangkan juga di dalam [6] [7], yaitu dekteksi plagiarism setelah dokumen dibagi ke dalam kelompok-kelompok (klaster). Yang lainnya dengan melalui 2 fase, di mana pada fase pertamanya mereka juga menerapkan *winnowing* [8]. *Biword winnowing* diusulkan pada [9] dan [10] untuk mendeteksi kasus plagiarisme pada dokumen panjang seperti laporan skripsi atau tugas akhir mahasiswa.

Kebanyakan algoritma *fingerprint* tidak mempedulikan semantik dalam bahasa alami, karena teks yang panjang akan diwakili oleh sejumlah kecil angka sebagai sidik jari dari teks tersebut. Ia tidak dapat menunjukkan posisi pasti dari fragmen yang disalin utuh kata per kata (*verbatim copy*), apalagi paraphrase kalimat. Ia hanya mencocokkan *fingerprint* dari dokumen A dan B dan menghitung berapa jumlah kecocokan (*match*) sebagai pengukuran kesamaan (*similarity*).

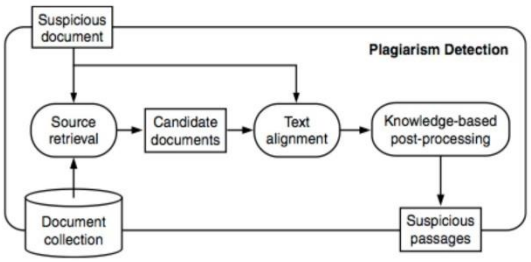
Winnowing [3]-[10] memiliki kelemahan bahwa sering kali hasil *fingerprint* adalah suatu *gram* yang tidak dapat dibaca, tidak memiliki arti, karena ia diperoleh dari memotong kalimat ke dalam karakter *n-gram* tanpa spasi. Dengan demikian, untuk kasus plagiarisme yang lebih canggih dengan teknik *paraphrase*, ringkasan dan lainnya, metode ini akan banyak mengalami kegagalan. Oleh karena itu, winnowing tidak dapat menentukan di mana posisi teks yang diplagiasi dari dokumen yang panjang.

Penelitian ini bertujuan untuk menemukan posisi dari suatu fragmen (*passage*) di dalam suatu dokumen yang dicurigai mengandung plagiarisme, dan posisi passage yang diplagiasi dari dokumen sumbernya. Pada sistem deteksi plagiarisme, tugas *text alignment* ini merupakan tahap kedua setelah *source retrieval*, yaitu menemukan kandidat dokumen sumber dari *passage* (fragmen teks) yang dicurigai mengandung plagiarisme.

Sebagai kontribusi penelitian adalah menjelaskan proses tahapan penyelarasan teks (*text alignment*) sebagai proses lanjutan pada sistem deteksi plagiarisme dokumen. Penelitian sejenis masih belum banyak berkembang di Indonesia, oleh karena itu penelitian ini dapat menjadi inspirasi bagi peneliti selanjutnya untuk pengembangan metode dan peningkatan performa *score* deteksinya.

II. METODOLOGI

Pemecahan problem deteksi plagiarisme bukanlah hanya pada memeriksa kemiripan antar dokumen, karena teks yang diplagiasi biasanya hanya sebagian fragmen. Namun banyaknya fragmen yang berasal dari berbagai sumber bisa meningkatkan *score* plagiarisme secara kumulatif.



Gambar 1. Model sistem deteksi plagiarisme [18]

Jurnal Teknik Informatika Vol. 14 No. 2, Oktober 2021
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v14i2.22841

Pendekatan metode *search engine* (mesin pencari) yang membandingkan satu kueri dengan seluruh isi dokumen kurang tepat untuk dipakai pada deteksi fragmen. Mesin pencari bisa digunakan untuk tahap *source retrieval*: menemukan dokumen relevan dari koleksi dokumen (korpus). Namun untuk *passage alignment*, perlu dilakukan analisis mendalam pada bagian-bagian teks. Inilah yang menjadi tantangan dalam *shared task Text Alignment* pada PAN 2013 dan PAN2014.

Tujuan *task* ini adalah menemukan kasus plagiarisme dari pasangan dokumen yang dicurigai mengandung plagiat (*suspicious document*) dan dokumen sumber (*source document*) sebagai hasil pencarian oleh mesin pencari pada tahap *source retrieval*, sebagaimana Gambar 1.

Apabila kasus plagiarisme ditemukan, maka kita perlu menentukan mana fragmen teks yang diambil dari dokumen sumber. Dalam hal ini, satuan performa yang diukur adalah posisi karakter awal (*offset*) dan akhir dari dokumen yang dicurigai (*suspicious doc*) dan pasangannya di dalam dokumen sumber (*source doc*). Performa system yang disubmit pada *shared task* diukur dengan *macro-average precision and recall*, *granularity* dan *plagiarism detection score* [21].

2.1 Analisa *Dataset*

Penelitian ini menggunakan himpunan data (*dataset*) dari *plagiarism detection shared task* yang diadakan oleh PAN (*Plagiarism analysis, Authorship Identification and Near-Duplicate Detection*) tahun 2013 dan 2014 [18]. Semua dokumen dalam *dataset* menggunakan bahasa Inggris. *Dataset* ini disusun secara otomatis menggunakan skrip program tertentu (di-*generate* oleh mesin) dengan meniru kebiasaan manusia saat melakukan plagiasi [21]. Jumlah kata yang diambil dari *source doc* minimal adalah 50 kata, lalu fragmen teks ini diselipkan di dalam *suspicious doc* pada posisi yang acak, bisa berada di tengah kalimat.

Ditinjau dari cara membuat bingung/sulit (*obfuscation*) atau agar menjadi kabur/tidak jelas dan sulit dideteksi oleh mesin, data set ini dibagi atas 4 kategori, yaitu [21]:

1. *No Obfuscation* (*Verbatim Copy*)

   Kalimat disalin secara utuh kata-per-kata tanpa perubahan apa pun dari *source doc*. Kemudian fragmen teks ini disisipkan pada sembarang posisi di dalam dokumen yang dicurigai. Dokumen sumber dan dokumen hasil plagiat memiliki topik yang sama, keduanya diambil dari dokumen di internet (artikel dan sebagainya). Deteksi dari jenis plagiarisme ini cukup mudah karena susunan kata-per-kata tidak ada perubahan dan tanpa ada kata baru yang dihapus atau ditambahkan.

2. *Random Obfuscation*

   Pendekatan yang diambil adalah secara *naïve* atau sederhana mengacak bagian fragmen teks, sedemikian sehingga teks tersebut tidak dapat dibaca/dipahami oleh manusia maksud kalimatnya. Tujuannya untuk menguji apakah algoritma penyelarasan teks mampu mengidentifikasi kalimat yang diplagiasi, dari sudut pandang model *bag-of-word*s. Kalimat yang diplagiasi berasal dari fragmen teks yang telah melalui pengacakan urutan, penambahan, penghapusan, dan penggantian kata atau frase pendek. Penggantian kata dilakukan dengan menggunakan sinonim atau kata yang mirip melalui *Wordnet*. Sedangkan frase diacak dengan tetap memperhatikan urutan *part-of-speech* aslinya.

3. *Translation Obfuscation*

   Teks yang diplagiasi berasal dari suatu kalimat yang sekurang-kurangnya mengalami 2 kali proses translasi ke bahasa lain. Misalnya teks asli ditranslasikan (diterjemahkan) ke bahasa kedua, kemudian ditranslasikan lagi ke bahasa ketiga, ditranslasikan kembali ke bahasa kedua, dan akhirnya ditranslasikan ke bahasa asal (Inggris). Google Translate, Microsoft Translator, dan MyMemory digunakan sebagai mesin penerjemahnya (*translator*).

4. *Summary Obfuscation*

   Rasionale yang digunakan adalah bahwa suatu kasus plagiarisme dapat berbentuk *summary* atau ringkasan dari suatu dokumen. *Summary* dibangkitkan dengan menggunakan metode-metode dari hasil riset di bidang *automatic text summarization*.

Jurnal Teknik Informatika Vol. 14 No. 2, Oktober 2021
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v14i2.22841

Selain itu, ada sejumlah pasangan dokumen dalam dataset yang tidak mengandung kasus plagiarisme (*no plagiarism* atau kelas netral). Pasangan dokumen dapat berasal dari topik yang sama maupun berbeda. Sistem yang diajukan oleh peserta harus dapat dengan tepat memeriksa bahwa untuk kasus ini memang tidak terjadi proses plagiarisme, walaupun berasal dari topik yang sama dan kebetulan ada beberapa kata yang digunakan bersama.

2.2 Pemrosesan Awal Teks

Pemrosesan awal teks (*text preprocessing*) dilakukan sebagai tahap awal dari penyelesaian *task text alignment* ini, yaitu;

1. *Text Cleaning*

   Dokumen dibersihkan dari tanda baca, karakter khusus dan kata-kata yang hanya berupa angka saja atau yang jumlah karakternya kurang dari 3.

2. *Stop Word Removal*

   Kata-kata yang sering muncul di dalam korpus, yang tidak memiliki pengaruh signifikan untuk diterapkan pada metode, dapat dihilangkan.

3. *Stemming*

   Pemotongan imbuhan dan pengembalian token ke bentuk akar kata (*root*) dilakukan bila proses tokenisasi mengambil kata-kata sebagai token. Dalam penelitian ini digunakan algoritma *Porter stemmer*[3] [22].

4. *Tokenization*

   Proses tokenisasi dilakukan untuk memecah dokumen pada korpus menjadi potongan-potongan *gram* atau token, dapat berupa *word-n-gram* atau *char-n-gram*. Pemilihan bentuk token juga dapat menentukan seberapa baik performa metode yang dipakai dalam kasus yang akan diselesaikan. Dalam penelitian ini, digunakan bentuk token kata atau *word*-1-*gram* sebagai fitur pengukuran *similarity* antar *fragment*.

5. *Frequency Analysis*

   Analisis dan solusi pada berbagai metode dalam IR maupun NLP banyak menggunakan frekuensi sebagai fitur dasarnya, seperti Naïve Bayes, model ruang vektor (TF-IDF), SVM, neutral network, dan banyak lainnya. Pada metode konvensional, frekuensi sering digunakan sebagai variabel pembobotan kata (*term weighting*). Penelitian ini menggunakan fitur frekuensi token untuk pengukuran *similarity* dengan menggunakan *Jackard* di antara pasangan *passage* pada *source* dan *suspicious doc*.

2.3 Token *n-gram*

Secara semantik, pendeteksian plagiarisme banyak yang mengandalkan teknik *n-gram matching*, dengan *word* (kata) sebagai basis (*word-n-gram*). Pendekatan menggunakan *char-n-gram* (basis karakter/huruf) biasanya akan banyak menghilangkan makna semantik, disebabkan oleh setiap token atau *gram* menjadi terpotong untuk kata-kata yang lebih panjang dari *n*, dan juga *gram* yang terbentuk dari gabungan beberapa karakter dari kata sebelum dan/atau sesudahnya.

Tabel 1. Variasi token berdasarkan char-n-gram dan word-n-gram

Char-4-gram	***Char-5-gram***	***Word-1-gram***	***Word-2-gram***
Topi	Topik	Topik	Topik ini
Opik	Opiki	Ini	Ini sangat
Piki	Pikin	Sangat	Sangat
Kini	Ikini	Menarik	Menarik
Inis	Kinis		
Nisa	Inisa		
Isan	Nisan		
Sang	Isang		
Anga	Sanga		
Ngat	Angat		
Gatm	Ngatm		
Atme	Gatme		
Tmen	Atmen		
Mena	Tmena		
Enar	Menar		
Nari	Enari		
Arik	Narik		

Pencocokan *n-gram* di antara dua dokumen dalam *task text alignment*, adalah dengan membandingkan token yang dimiliki oleh masing-masing dokumen. Cara yang populer

[3] https://tartarus.org/martin/PorterStemmer/

Jurnal Teknik Informatika Vol. 14 No. 2, Oktober 2021
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v14i2.22841

untuk menghitung kesamaan dokumen, antara lain dengan menghitung jumlah *n-gram* yang *match* (cocok) atau *overlap* (tumpang tindih).

Penelitian ini menggunakan pendekatan *word-n-gram*, karena tidak menghilangkan makna semantik kata. Dari segi penanganan memori dan operasi *matching* juga akan lebih efisien, karena jumlah variasi *token* yang terbentuk (*vocabulary*) dari dokumen yang dibandingkan, jauh lebih sedikit daripada *char-n-gram*. Lebih spesifik lagi, digunakan *n* yang digunakan adalah 1 (*token* berbentuk kata).

2.4 Pengukuran Kemiripan (*Similarity*)

Beberapa cara mengukur *similarity* antar dokumen, dapat didasarkan pada jumlah *n-gram match* yang telah ditemukan seperti di atas, seperti *Jackard coefficient*, *n-gram overlap count*, dan lainnya. Metode pengukuran similarity lainnya dapat berdasarkan *bag-of-word* (BoW), yaitu token yang dikumpulkan dari keseluruhan dokumen di dalam korpus untuk membentuk suatu list *vocabulary*. Selanjutnya, *similarity* untuk BoW dapat dihitung dengan menggunakan perhitungan jarak (misalnya *Euclidean Distance*, *Mahnattan Distance*, *Hamming Distance*), atau perhitungan sudut vektor BoW antar pasangan dokumen (*cosine similarity*). Dalam penelitian ini, digunakan Jackard similarity (*J*) sebagaimana persamaan (1), dengan tanda mutlak menyatakan jumlah token yang unik.

$$J = \frac{|susp| \cap |source|}{|susp \cup source|} \quad (1)$$

2.5 *Seeding* (Pembibitan)

Pencocokan atau penyelarasan bagian teks (*passage alignment*) dilakukan dengan mencari pasangan *passage* (teks *fragment*) yang bisa saja tidak dimulai dari awal kalimat. Hal ini menambah kesulitan pada sistem untuk menetapkan *offset* dan *length* (panjang) *passage* baik pada dokumen sumber maupun pada dokumen yang dicurigai agar dapat mendeteksi dengan benar.

Plagiarism detection score diukur dengan mempertimbangkan nilai-nilai ini (*offset* dan *length*). Gambar 2 berikut memperlihatkan betapa sulitnya menentukan batas-batas teks yang dianggap plagiat, dari sistem yang melakukan pencocokan secara *phrase matching* (*word n-gram matching*).

Untuk menjawab tantangan tersebut, diusulkan metode pendeteksian *fragment* teks, yang dibagi atas kalimat-kalimat (*sentence alignment*). Pada tahap awal, dokumen *source* dan *suspicious* dipecah ke dalam kalimat-kalimat dengan panjang maksimum *k* kata. Apabila jumlah kata dalam kalimat kurang dari *fragment threshold* T_f (jumlah minimum kata dalam setiap *fragment*, yang ditentukan secara empiris), maka kalimat tersebut digabung dengan kalimat setelahnya. Apabila masih di bawah *threshold*, digabung kembali ke kalimat setelahnya. Posisi *offset* dan panjang masing-masing kalimat disimpan.

Gambar 2. *Word-n-gram matching*

Langkah selanjutnya adalah membandingkan *fragment* dari *suspicious doc*, dengan seluruh *fragment* di *source doc* satu per satu (*one-to-one comparison*). Apabila ditemukan nilai kemiripan yang mencapai *similarity threshold* T_s, maka kalimat berserta posisi *offset* dan panjangnya disimpan dalam suatu *Seeds set S*. Bila masih ditemukan lagi kalimat yang sama atau mirip, maka disimpan lagi dalam *set S*.

Pengukuran kemiripan (*similarity*) dapat dilakukan dengan beberapa cara, seperti TF-IDF, *n-gram match count*, *n-gram overlap*, *Jackard similarity*, *L1-norm*, atau *L2-norm* (*Euclidean distance*). Dalam penelitian ini, penulis menggunakan *Jackard* untuk menghitung *similarity* antar *fragment*, sebagaimana persamaan (1). Perhitungan *similarity* dengan metode lainnya disimpan untuk pekerjaan penelitian selanjutnya.

Dalam bentuk *pseudo code*, algoritma dasar (*baseline*) untuk menemukan kandidat kasus plagiarisme (*seeds*) di atas dapat dilihat sebagai mana Gambar 3 berikut. *Baseline method* adalah metode sederhana yang menjadi tolok ukur pengembangan metode yang diusulkan. Sebagai *baseline*, metode ini dapat menemukan dengan baik kasus plagiarisme yang dikategori-

Jurnal Teknik Informatika Vol. 14 No. 2, Oktober 2021
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v14i2.22841

kan sebagai *verbatim copy* (*no-obfuscation*). Sedangkan pada kategori *obfuscation* lainnya, performanya kurang begitu baik.

```
1  Susp[id, stc, pos, len] ←
   fagment(doc_susp)
2  Src[id, stc, pos, len]  ← fragment
   (doc_src)
3  Detection = ϕ
4  For stc_susp in Susp[stc]:
       Candidate = ϕ
       for stc_src in Src[stc]:
           J = Jackard(stc_susp, stc_src)
           If J > Ts:
           Cadidate ← add(Susp[id], Src[id], J)
       sort(Candidate, J, descending)
       Detection ← add(Candidate[0])
```

Gambar 3. Algortima proses *seeding*

Objektif yang ingin dicapai dalam penelitian ini, adalah sistem yang dikembangkan harus dapat melampaui hasil *baseline score* yang ditetapkan oleh *organizer*. Dengan kata lain, metode yang dikembangkan harus dapat memecahkan problem deteksi plagiarisme yang lebih rumit, yaitu *translation*, *random* dan *summary obfuscation*.

2.6 Deteksi Plagiarisme dengan Pendekatan Semantik

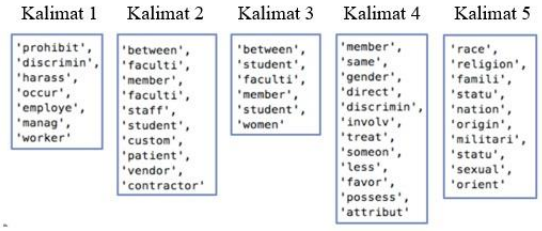
Langkah-langkah dalam pendekatan yang diusulkan ini adalah dengan memeriksa semantik dari kalimat-kalimat di dalam kedua dokumen, yaitu urutan *token kata* yang terpanjang yang dapat membentuk suatu *passage* atau *fragment* dari teks (bisa berbentuk satu atau lebih kalimat, bahkan mendeteksi bila mulainya kasus plagiarisme berada di tengah kalimat, seperti pada Gambar 2.

Berikut adalah langkah-langkah algoritma dari deteksi plagiarisme dengan menggunakan pendekatan semantik yang diusulkan dalam penelitian ini.

Step 1: Tahap *Preprocessing*

- Baca file *suspicios* dan *source* dalam *stream* karakter
- Deteksi pembatas kalimat (tanda titik) catat *offset* (posisi karakter awal) dan *length* (panjang kalimat)-nya
- Lakukan *text cleaning*
- Untuk setiap kalimat, gabung kalimat dengan kalimat sebelumnya kalau jumlah kata kurang dari *threshold* T_f
- *Split* kalimat/fragment menjadi *token*
- Generate set fitur (*unigram, bigram, name entity*)
- Remove *stopword* berdasarkan *common* English *stoplist*
- Lakukan *stemming* dengan *Porter stemmer*
- Generate *inverted index* {*token*: *sentence ID*: fitur(urutan kata dalam kalimat)}

Misalnya untuk fitur *unigram*, *token* pada beberapa kalimat dalam suatu dokumen dapat dilihat pada gambar 4.



Gambar 4. Fitur *token unigram* pada 5 kalimat berbeda dalam suatu dokumen

Step 2: Tahap *Seeding*

Temukan *property alignment*: *shared unigram*, *shared bigram*, *important words*, *name entities*, yang mungkin saja adalah topik dari *fragment* yang diplagiasi (deteksi kasus *random obfuscation*).

- Hitung kemiripan antara pasangan dokumen (*sus*, *src*) dengan menghitung *Jackard similarty* seperti persamaan (1).
- Catat ID pasangan *fragment* yang memiliki nilI *Jackard* melebihi *threshold* T_s pada kedua dokumen {*sus_stc_ID* : *src_stc_ID*} dari s*eeds Set S.*

Gambar 5 adalah ilustrasi set *S* dari *seeds* (bibit/kandidat kalimat) dari pasangan ID kalimat pada dokumen *suspicios* dan *source*. Pada bahasa pemrograman *Perl*, output *print* ditampilkan dalam variabel *$VARx* (dengan *x* adalah nomor urut pasangan kandidat). Dalam contoh tersebut, terdapat 17 pasangan kalimat yang dicurigai memiliki kesamaan.

```
$VAR1 = '8:33';
$VAR2 = '33:5';
$VAR3 = '33:10';
$VAR4 = '34:11';
$VAR5 = '35:5';
$VAR6 = '35:10';
$VAR7 = '35:12';
$VAR8 = '36:13';
$VAR9 = '37:14';
$VAR10 = '38:15';
$VAR11 = '39:3';
$VAR12 = '39:14';
$VAR13 = '39:16';
$VAR14 = '45:3';
$VAR15 = '45:13';
$VAR16 = '89:33';
$VAR17 = '90:32';
```

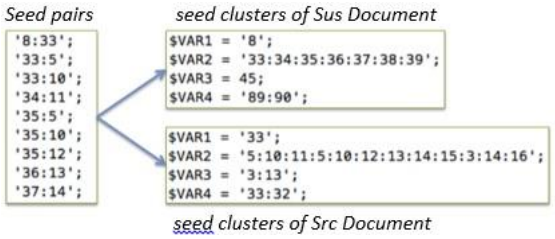
Gambar 5. *Seeds set S* yang *match*

Jurnal Teknik Informatika Vol. 14 No. 2, Oktober 2021
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v14i2.22841

Step 3: Tahap *Clustering*

- Generate *Set* $C_{Sus} = \{C_{sus,1}, \dots, C_{sus,x}\}$ dan *Set* $C_{Src} = \{C_{src,1}, \dots, C_{src,y}\}$ dengan formula (2).
  $C_{Sus} \leftarrow s_{i=1}$ dan $C_{Src} \leftarrow s_{j=1}$ (2)
  dengan *i* dan *j* adalah indeks *seed Sus* dan *seed Src*.
- Untuk pasangan *seed* selanjutnya, lakukan pengelompokan *seeds* yang berdekatan atau berurutan.
- Diberikan *rule skip n-seeds*, dengan hipotesa ada 2 *seed* (kalimat atau *fragment*) yang terpisah oleh sejumlah *n seed* lain yang tidak terdeteksi cocok atau *match* (mengantisipasi kasus ringkasan atau *summary obfuscation*).
- Jika ada *seeds* berdekatan dengan jarak *n (n* ditentukan secara empiris), maka *seeds* tersebut diambil menjadi anggota *cluster* C_{sus} dan pasangannya diambil menjadi anggota C_{src}, sebagaimana persamaan (3).
  $C_{sus} \leftarrow s_i, s_{i+1}, s_{i+\dots}, s_{i+n}, s_{i+n+1}$ (3)
  $C_{src} \leftarrow s_j, s_{j+1}, s_{j+\dots}, s_{j+n}, s_{j+n+1}$
- Sejumlah *n fragment* yang tidak *match* tersebut di-*skip* dari keharusan memiliki pasangan *seeds*.

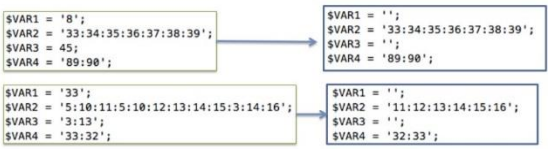
Gambar 6 memperlihatkan tahap *clustering*, dengan beberapa *seeds* yang berurutan dikelompokkan, demikian juga pasangannya pada dokumen lainnya (terlihat pada *$VAR2* dan *$VAR4* antara *source* dan *suspicious*. *Seeds* pada kotak sebelah kiri sesuai dengan Gambar 6, digambarkan sebagian untuk menghemat *space*). Terbentuk 4 *cluster plagiarism passage candidate* (*$VAR1-$VAR4*).



Gambar 6. Tahap *seeds clustering*

Step 4: Tahap *Extension : passage smoothing*

- Urutkan *seeds ID* pada masing-masing *cluster* seperti pada gambar 6.
- Hapus *seeds* yang *double* atau lebih pada *cluster* (ambil 1 *seed* saja) bila ditemukan
- Selipkan (*insert*) *seeds ID* pada seeds yang diperoleh dari operasi *skip n-seeds*.
- Tetapkan panjang *passage* minimal (*threshold plagiarism passage* T_p), menyatakan suatu *fragment* layak disebut sebagai plagiarisme. Misalnya *Tp*=3 kalimat
- Apabila jumlah seeds dalam suatu *cluster* dari salah satu dokumen kurang dari *threshold plagiarism passage Tp*, maka *seeds* dihapus.
- Apabila dalam satu *cluster src* terdapat 2 kandidat fragmen (*seeds* yang berurutan) namun terpisah jauh (*skip n-seeds* dengan jumlah *n* yang melebihi yang ditetapkan), maka ambil seeds yang terpanjang saja.



Gambar 7. Tahap *smoothing*

Gambar 7 menunjukkan operasi tahap *exstension* (*passage smoothing*) dari program *Perl*, yang menghapus *seeds* dengan panjang kurang dari *threshold* T_p, dengan *seeds* yang sudah terurut. Pada *$VAR2* di dalam *cluster src*, *seeds ID* {3,4,5} dari hasil *smoothing* terhadap *skip n-seeds* {3 dan 5}, dibuang/dihapus. Karena seeds {3,4,5} tidak bisa digabung dengan {10,11,12,…,16} untuk membentuk *fragment (n* terlalu besar bila diterapkan *skip n-seeds)*, maka diambil *fragment* yang terpanjang saja.

Sedangkan anggota *cluster* yang hanya punya 1 *seed* (*$VAR1* dan *$VAR3*) dihapus. Akhirnya terdapat 2 kandidat kasus plagiarisme, yaitu *susp_ID* {33,34…39} dan {89, 90}, secara berurutan berpasangan dengan *src_ID* {11,12,13,14,15,16} dan {32,33}.

2.7 Menentukan *offset suspicious-source*

Langkah selanjutnya adalah menghitung *offset* dan panjang *passage* dari *seeds* dan kemudian ditulis ke dalam file *XML* sesuai dengan format yang ditentukan oleh *organizer*. Data *offset* dan panjang *passage* didapat dari mengkonversi *seeds ID* pada anggota cluster masing-masing (C_{sus} dan C_{src}), yang dilakukan dengan me-*look up* panjang *fragment* dan *offset* pada *inverted index* dari tahap *Preprocessing*. Total panjang fragment adalah penjumlahan dari panjang masing-masing seeds, sedangkan offset adalah diambil dari seed pertama dari anggota pasangan C_{sus} dan C_{src}.

Jurnal Teknik Informatika Vol. 14 No. 2, Oktober 2021
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v14i2.22841

Sebagai contoh seperti Gambar 8, kandidat *seed $VAR2* dimulai dari *seed ID* 33, dan berakhir pada *seed ID* 39 untuk *susp_doc*. Kemudian dari *inverted index* ditemukan *offset seed ID* 33 adalah 1146 (posisi karakter di dalam file *susp_doc*), dan panjangnya diperoleh sebanyak 669 karakter. Hal yang sama dilakukan juga terhadap dokumen sumber *src_doc*.

```
<document reference="suspicious-document00005.txt">
<feature name="detected-plagiarism"
source_length="669" source_offset="1146"
source_reference="source-document01090.txt"
this_length="791" this_offset="3640" />
<feature name="detected-plagiarism"
source_length="219" source_offset="3412"
source_reference="source-document01090.txt"
this_length="220" this_offset="9350" />
</document>
```

Gambar 8. Output deteksi plagiarisme dalam *file XML*

Hasil file XML ini kemudian diinputkan ke dalam mekanisme perhitungan *score* yang disediakan oleh *organizer*.

III. HASIL DAN PEMBAHASAN

Eksperimen ini dilaksanakan dengan menguji coba sistem yang menggunakan *baseline* dan *proposed method* seperti diterangkan di bagian II, menggunakan komputer MacBook dengan memori 4 GB dan processor Intel Pentium Core i-5 1.8 GHz, dan sistem operasi IOS High Sierra.

Dataset yang digunakan adalah PAN-2013 *Corpus* 1 *dataset* (Tabel 2), yang terdiri atas 4 model *obfuscation* seperti yang telah diterangkan pada bab 2.B. Dalam *dataset* ini, terdapat pula set data *Corpus* 2 yang dinyatakan sebagai set yang kurang bagus karena banyak memiliki *noise*. Oleh karena itu, eksperimen pada penelitian ini tidak menguji set yang berisi *noise* tersebut [18].

3.1 *Set up* Eksperimen

Skema pengujian dalam penelitian ini adalah mencoba terlebih dahulu berbagai kombinasi *threshold* dari sistem deteksi plagiarisme yang diusulkan. Tahap pencarian parameter optimal dilakukan, dengan melihat hasil *score* deteksi.

Penelusuran pertama adalah menetapkan secara empiris nilai *threshold* untuk eksperimen awal, yaitu T_s (*similarity threshold* atau *Jackard threshold*) dan T_f (*fragment threshold*, atau *max_word*, jumlah kata maks dalam kalimat). Kemudian secara bertahap, nilai T_s diturunkan, untuk menambah sensitifitas deteksi (meloloskan *seeds* yang tidak terlalu mirip).

Tabel 2. Data Korpus Penelitian

Corpus Name 1 : pan13-text-alignment-test-corpus1-2013-03-08	
Suspicious	*398 Documents*
Source	*490 Documents*
Case:	
No plagiarism	*90 doc of susp doc (no alignment)*
No obfuscation	*105 pairs*
Random obfuscation	*121 pairs*
Translation obfuscation	*108 pairs*
Summary obfuscation	*94 pairs*
Corpus Name 2 : pan13-text-alignment-test-corpus2-2013-01-21	
Suspicious	*1826 Documents*
Source	*3169 Documents*
Case:	
No plagiarism	*1000 doc of susp doc (no alignment)*
No obfuscation	*1000 pairs*
Random obfuscation	*1000 pairs*
Translation obfuscation	*1000 pairs*
Summary obfuscation	*1185 pairs*

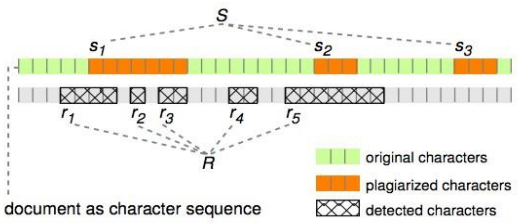
Penelusuran dilakukan sampai pada nilai *plagdet score* yang sudah konvergen (tidak berubah lagi), kemudian pencarian dilakukan terhadap parameter T_f, sebagaimana dapat dilihat pada Tabel 3.

Tabel 3. Set up eksperimen

Exp	T_s	T_f	Penamaan Method
1	0.35	15	T=0.35, max_word=15
2	0.3	15	T=0.3, max_word=15
3	0.25	15	T=0.25, max_word=15
4	0.25	13	T=0.25, max_word=13
5	0.25	10	T=0.25, max_word=10

3.2 *Scoring*

Pengukuran unjuk kerja sistem (*scoring*) dilakukan dengan menggunakan *tools* yang disediakan oleh *organizer*. Satuan yang diukur adalah *precision*, *recall*, *granurality*, dan *plagiarism detection score* sebagai-mana dalam [21] [23]. Masing-masing satuan pengukuran dihitung berdasarkan persamaan (4) sampai (7).



Gambar 9. Sekuensi karakter yang terdeteksi pada dokumen (*sus* dan *src*) [21]

Jurnal Teknik Informatika Vol. 14 No. 2, Oktober 2021
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v14i2.22841

Official score yang akan digunakan untuk unjuk kerja sistem secara keseluruhan adalah *plagiarism detection score* (persamaan (7)). Sensitivitas sistem diukur sampai pada level karakter, seperti terlihat pada Gambar 9.

Bagian yang diplagiasi diperlakukan sebagai unit *retrieval* dasar. Dalam hal ini, setiap sumber (*source*) $s_i \in S$ didefinisikan sebagai kueri di mana algoritma deteksi plagiarisme menghasilkan suatu set *retrieved* $R_i \subseteq R$.

Recall dari algoritma deteksi plagarisme (*Plagiarism Detection Algorithm*) *recPDA*, didefinisikan sebagai rata-rata dari fraksi bagian plagiarisme, dengan rata-rata di keseluruhan bagian di dalam *S* adalah seperti persamaan (4) berikut.

$$rec_{PDA}(S,R) = \frac{1}{|S|}\sum_{s\in S}\frac{|s \sqcap \bigcup_{r\in R} r|}{|s|}\ , \qquad (4)$$

dengan tanda ⊓ menghitung posisi *overlap* dari karakter. Selain dari bagain plagiasi *S*, presisi di mana kueri di dalam *S* telah terjawab kemudian diukur sebagai recall dari *R* di bawah *S*. Dengan menghitung rata-rata dari $r \in R$ kita akan mendapatkan suatu *rule* komputasi yang terdefinisi sebagai *retrieval precision* untuk *S*, sebagaimana persamaan (5) berikut ini.

$$prec_{PDA}(S,R) = \frac{1}{|R|}\sum_{r\in R}\frac{|r \sqcap \bigcup_{s\in S} s|}{|r|}\ . \qquad (5)$$

Nilai *recPDA* dan *precPDA* tidak sensitif terhadap berapa kalinya suatu $s \in S$ terdeteksi di dalam hasil *R*. Oleh karena itu diperkenalkan suatu ukuran disebut *granurality* dari *R*.

Granurality dari hasil *R* untuk suatu set bagian yang diplagiasi *S*, didefinisikan sebagai ukuran rata-rata dari lingkup yang ada: suatu hasil deteksi $r \in R$ bagian dari cakupan C_s dari suatu $s \in S$ jika dan hanya jika *s* dan *r overlap* (saling tumpang tindih). Bila $S_R \subseteq S$ menyatakan dari set kasus sedimikian sehingga untuk setiap $s \in S : |C_s| > 0$. Maka, *granurality* dari *R* dengan syarat *S* didefinisikan sebagai persamaan (6).

$$gran_{PDA}(S,R) = \frac{1}{|S_R|}\sum_{s\in S_R}|C_s|\ , \qquad (6)$$

Untuk menghitung *ranking* secara keseluruhan, digunakan satuan *overall score* dari seluruh kelas *obfuscation* yang meliputi ketiga pengukuran di atas, yaitu sebagaimana persamaan 7 berikut ini, dengan *F* adalah *harmonic mean* dari *precision* dan *recall*.

$$overall_{PDA}(S,R) = \frac{F}{\log_2(1+gran_{PDA})}, \qquad (7)$$

3.3 Data Hasil Eksperimen

Dari *set up* eksperimen yang telah ditetapkan, dilakukan pengujian terhadap metode sistem yang diusulkan di atas. Pengukuran performa dilakukan dengan menggunakan *metric tools* yang disediakan oleh *organizer*, dan dituangkan di dalam Tabel 4 sampai 7.

Variasi percobaan ditetapkan sebagai nama metode (untuk penggambaran pada grafik hasil eksperimen pada Gambar 9 dan 10 mengikuti Tabel 3 seperti diterangkan pada sub bagian 3.1.

Tabel 4. Plagiarism Detection Score

No	Threshold		Obfuscation				Ave
	T_s	T_f	No	Rnd	Trn	Sum	
1	0.35	15	0,91	0,43	0,69	0,02	0.523
2	0.3	15	0,88	0,68	0,76	0,21	0.646
3	0.25	15	0,88	0,68	0,76	0,21	0.646
4	0.25	13	0,88	0,66	0,76	0,25	0.649
5	**0.25**	**10**	**0,87**	**0,72**	**0,78**	**0,25**	**0.669**
6	*Baseline*		0,93	0,07	0,11	0,04	0,284

Tabel 5. Precision

No	Threshold		Obfuscation				Ave
	T_s	T_f	No	Rnd	Trn	Sum	
1	0.35	15	0,90	0,27	0,57	0,01	0,443
2	0.3	15	0,88	0,53	0,68	0,12	0,564
3	0.25	15	0,88	0,53	0,68	0,12	0,564
4	0.25	13	0,89	0,52	0,68	0,15	0,570
5	**0.25**	**10**	**0,89**	**0,59**	**0,73**	**0,14**	**0,600**
6	*Baseline*		0,89	0,98	0,98	0,91	0,943

Tabel 6. Recall

No	Threshold		Obfuscation				Ave
	T_s	T_f	No	Rnd	Trn	Sum	
1	0.35	15	0,93	0,96	0,95	0,99	0.959
2	0.3	15	0,88	0,95	0,87	0,99	0.921
3	0.25	15	0,88	0,95	0,87	0,99	0.921
4	0.25	13	0,88	0,94	0,88	0,99	0.921
5	**0.25**	**10**	**0,86**	**0,92**	**0,83**	**0,99**	**0.898**

Jurnal Teknik Informatika Vol. 14 No. 2, Oktober 2021
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v14i2.22841

6	*Baseline*	0,99	0,04	0,09	0,04	0,286

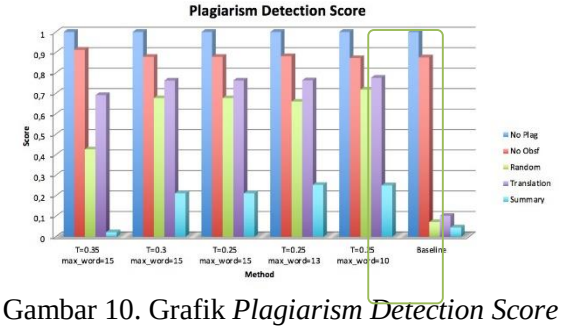
Tabel 7. Granularity

No	*Threshold*		*Obfuscation*				Ave
	T_s	T_f	No	Rnd	Trn	Sum	
1	0.35	15	1,00	1,00	1,04	1,00	1,010
2	0.3	15	1,00	1,01	1,00	1,00	1,003
3	0.25	15	1,00	1,01	1,00	1,00	1,003
4	0.25	13	1,00	1,01	1,01	1,00	1,005
5	**0.25**	**10**	**1,00**	**1,00**	**1,01**	**1,00**	**1,003**
6	*Baseline*		0,88	1,01	1,04	1,65	1,126

Untuk *no plagiarism case*, semua metode deteksi bernilai 1, artinya semua sistem (yang diusulkan dan *baseline*) benar tidak mendeteksi adanya kasus plagiarisme dari pasangan dokumen *suspicious* dan *source*. Data ini tidak dituliskan di dalam Tabel 4 sampai 7, karena ruang yang tidak mencukupi.

3.4 Analisa dan Diskusi

Dari data hasil eksperimen yang diperoleh seperti dilaporkan pada bagian 3.3, dapat kita lihat bahwa sistem yang dikembangkan sudah cukup baik untuk mendeteksi kasus plagiarisme modern yang mengandung berbagai kategori *obfuscation*, seperti diilustrasikan oleh grafik pada Gambar 10. *Baseline system* yang disediakan oleh *organizer* [18] sangat baik mendeteksi plagiarisme dengan tipe *verbatim copy* (*no-obfuscation*), seperti ditunjukkan pada *cluster bar* paling kanan. Namun untuk mendeteksi kasus plagiarisme dengan ketiga jenis *obfuscation* lainnya, tidak berhasil atau sangat rendah sekali, seperti terlihat pada Gambar 10 berikut.



Gambar 10. Grafik *Plagiarism Detection Score*

Sedangkan dengan pencarian parameter optimal, pada percobaan awal menggunakan kombinasi T_s=0.35 dan maksimum kata di dalam kalimat (*max_word* T_f) adalah 15 kata, diperoleh *plagdet score* yang sangat tinggi untuk *verbatim copy*. Namun untuk *summary obfuscation* sangat rendah sekali, bahkan di bawah performa dari *baseline system*. Setelah dilakukan pencarian awal dengan mengurangi *threshold* T_s, diperoleh peningkatan deteksi untuk kasus *obfuscation* lainnya, namun deteksi terhadap *verbatim-copy* menurun. Ada kompromi di dalam *plagdet score*.

Dengan menurunkan T_s menjadi 0.25, tidak terjadi perubahan *score*. Artinya *threshold* sebesar 0.25 atau 0.3 tidak signifikan terhadap hasil deteksi. Kemudian parameter T_f diturunkan menjadi 13 kata. Artinya bila kalimat kurang dari 13 kata, maka kalimat digabungkan dengan kalimat setelahnya. Dengan parameter ini, terjadi peningkatan *score* deteksi untuk kasus *summary obfuscation*, sedangkan yang lainnya relatif tetap.

Dari hasil *tuning* parameter T_s dan T_f, pendekatan semantik dalam deteksi plagiarisme yang dipilih/diusulkan dalam penelitian ini, adalah pada nilai *Ts*=0.25 dan T_f =10 kata, karena memberikan hasil paling optimal. Di dalam Gambar 10 adalah pada posisi yang ditandai dengan kotak warna hijau.

Ditinjau dari segi *presicion*, seperti terlihat pada Tabel 5, hasil yang diperoleh pada eksperimen ini sudah cukup baik. *Precision* tertinggi didapatkan pada parameter pencarian awal. Namun karena fokus sistem deteksi plagiarisme adalah pada *recall* yang tinggi (kuantitas kasus plagiarisme yang dapat ditemukan), maka kompensasi peningkatan *recall* akan berpengaruh pada nilai *precision score* yang menurun, sebagaimana terlihat pada Tabel 6 tentang *recall* untuk eksperimen ini.

Dari grafik *recall* terlihat bahwa pemilihan parameter akhir, yaitu T_s=0.25 dan T_f=10 memberikan *recall* yang meningkat secara signifikan pada kasus plagiarisme sulit, dengan *random*, *translation* dan *summary obfuscation*. Karena nilai *similarity threshold* diperkecil, maka untuk kasus *exact copy* yang berada di tengah *fragment*, bila porsi kata yang sama lebih sedikit dari panjang seluruh kata pada *fragment* yang ada, maka nilai *similarity*-nya menjadi kecil. Walaupun *recall* pada kasus *verbatim copy* (*no obfuscation*) menurun, secara keseluruhan Recall tetap yang tertinggi.

Dari segi *granularity*, sistem yang terbaik adalah bila nilai *granularity*-nya mendekati 1, dalam arti bahwa antara *fagment* kata-kata hasil deteksi dan *fragment* kata-kata pada sumber dapat saling meliputi/tumpang tindih (saling

Jurnal Teknik Informatika Vol. 14 No. 2, Oktober 2021
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v14i2.22841

overlap). Apabila ada *fragment* yang tidak *overlap* (misalnya ada kata yang berlebih di salah satu), menyebabkan nilai *granularity*-nya menjadi lebih besar daripada 1. Sistem yang diajukan memperlihatkan *granularity* yang sangat baik, seperti terlihat pada Tabel 7, yaitu semuanya mendekati atau sama dengan nilai 1. Sedangkan *baseline system*, *granularity* terbaik adalah pada kasus *no obfuscation* (*verbatim copy*) dan *random obfuscation*.

3.5 Kesimpulan Hasil Eksperimen

Bila kita hitung secara keseluruhan, dengan menjumlahkan *plagdet score* dari semua kelas *obfuscation*, maka sistem yang diusulkan dengan parameter optimal dapat meningkatkan performa *baseline system* sebesar lebih kurang 2.4 kali (236%), sebagaimana terlihat pada Tabel 8 dan Gambar 11 berikut.

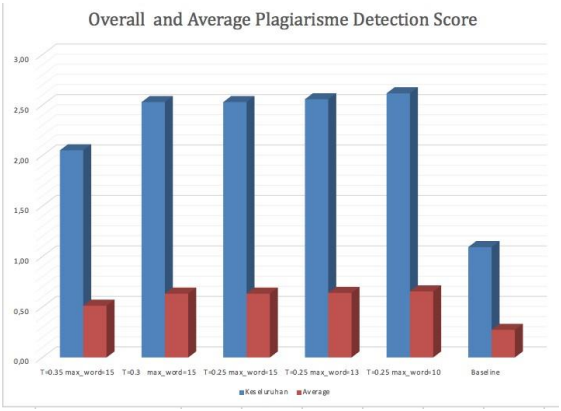
Dari segi *recall*, metode yang diusulkan dapat menemukan lebih banyak kasus plagiarise dibandingkan dengan *baseline*, yaitu 89,8% dibandingkan dengan 28,6%. Peningkatannya sekitar 313% atau 3x lebih baik.

Dari segi *precision*, penentuan *offset* sangat signifikan sehingga walaupun dari segi kasus, plagiarisme dapat ditemukan, namun penempatan *offset* posisi teks yang diplagiasi kurang tepat (karena diambil dari awal kalimat yang mengandung plagiat), maka *precision*-nya menjadi menurun. Pada *baseline* nilai *precision*-nya tinggi disebabkan karena kasus yang ditemukan sangat sedikit, yang terlihat dari angka *recall* yang sangat rendah.

Secara umum, penelitian ini sudah berhasil melakukan peningkatan skor deteksi dari *baseline* pada [18] menjadi sebesar 236%, dan peningkatan *recall* yang tinggi yaitu 313%.

Tabel 8. Plagiarisme Detection Score secara keseluruhan dan rata-rata

Metric	*Obfuscation*				Average
	No	Rnd	Trn	Sum	
Metode yang diusulkan (T_s=0.25 dan T_f=10)					
Det score	0,87	0,72	0,78	0,25	0.669
Precision	0,89	0,59	0,73	0,14	0,600
Recall	0,86	0,92	0,83	0,99	0.898
Baseline					
Det Score	0,93	0,07	0,11	0,04	0,284
Precision	0,89	0,98	0,98	0,91	0,943
Recall	0,99	0,04	0,09	0,04	0,286



Gambar 11. *Plagiarism Detection Score* secara keseluruhan

IV. PENUTUP

Dari penelitian yang telah dilakukan, serta eksperimen dan hasil yang diperoleh, dapat disimpulkan sebagai berikut:

1. Sistem deteksi plagiarisme modern harus dapat mendeteksi kasus-kasus dengan kategori kebingungan (*obfuscation*) yang sulit.
2. Kategori tersulit adalah mendeteksi kasus *summary plagiarism*, yaitu plagiasi berbentuk ringkasan atau pengembangan teks sumber.
3. Metode yang diusulkan dalam penelitian ini, yaitu pendekatan semantik pada *n-gram matching* telah dapat meningkatkan performa *baseline* secara signifikan. Dari segi *plagiarism detection score* dapat meningkatkan sebesar 236% (dari 28,4% menjadi 66,9%), dan recall dapat ditingkatkan menjadi 313%.

4.1 Saran untuk penelitian selanjutnya

Untuk penelitian selanjutnya, eksperimen dapat dilakukan terhadap *Corpus*-2 yang banyak mengandung *noise*. Selain itu, teknik lainnya dalam fragmentasi masih dapat dikembangkan, untuk dapat mendeteksi secara lebih baik sampai level karakter, khususnya dalam kasus-kasus yang dengan *obfuscation* yang rumit.

Sejalan dengan pengembangan teknik fragmentasinya, fitur *similarity* antar *fragment* juga dapat divariasikan, misalnya menggunakan model vektor *Bag-of-Word*, TF-IDF, dan lainnya. Selain itu, formulasi untuk *similarity* dapat dicoba menggunakan berbagai fungsi *distance* seperti *Hamming distance*, *Manhattan distance*, dan *Euclidean distance*, maupun

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v14i2.22841

pengukuran sudut antara vektor dengan *cosine similarity*.

DAFTAR PUSTAKA

[1] N. Shivakumar and H. Garcia-Molina, "SCAM: A Copy Detection Mechanism for Digital Documents," in 2nd International Conference in Theory and Practice of Digital Libraries (DL 1995), Austin, Texas, 1995.

[2] S. Brin, J. Davis and H. Garcia-Molina, "Copy Detection Mechanism for Digital Documents," in ACM SIGMOD 1995, San Jose, CA, 1995.

[3] S. Schleimer, D. S. Wilkerson and A. Aiken, "Winnowing: Local Algorithms for Document Fingerprinting," in Proceeding of ACM SIGMOD 2003, 2003.

[4] J. Parapar and A. Barreiro, "Winnowing-Based Text Clustering," in CIKM 2008, Napa Valley, California, 2008.

[5] D. Purwitasari, I. W. S. Priantara and P. Y. Kusmawan, "The Use of Hartigan Index for Initializing K-Means++ in Detecting Similar Texts of Clustered Documents as a Plagiarism Indicator," Asian Journal of Information Technology, vol. 10, no. 8-12, pp. 341-347, 2011.

[6] D. Zou, W.-J. Long and Z. Ling, "Winnowing-Based Similar Text Positioning Method," in International Conference on Internet Technology and Applications, 2010.

[7] D. Zou, W.-J. Long and Z. Ling, "A Cluster-Based Plagiarism Detection Method," in Lab Report for PAN at CLEF, 2010.

[8] D. Zou, W.-J. Long and Z. Ling, "A Two- Phase Plagiarism Detection Method," in Internationl Conference on Internet Technology and Applications (iTAP), 2011.

[9] S. Hidayatullah, "Source Detection pada Kasus Plagiarisme Dokumen menggunakan Biword Winnowing dan Retrieval berbasis OKAPI BM25", Bachelor Thesis, UIN Sultan Syarif Kasim, Riau, 2014.

[10] S. Agustian dan A. Sucipto, "Source Retrieval pada Deteksi Plagiarisme Berdasarkan Biword Fingerprint dengan Model Ruang Vektor", in Seminar Nasional Teknologi Informasi, Komunikasi dan Industri, SNTIKI 12, Desember 2020.

[11] A. Daud, J. A. Khan, J. A. Nasir, R. A. Abbasi, N. R. Aljohani and J. S. Alowibdi, "Latent Dirichlet Allocation and POS Tags Based Method for External Plagiarism Detection: LDA and POS Tags Based Plagiarism Detection," International Journal on Semantic Web and Information Systems, vol. 14, no. 3, pp. 53-69, 2018.

[12] N. Meuschke, V. Stange, M. Schubotz and B. Gipp, "HyPlag: A Hybrid Approach to Academic Plagiarism Detection," in SIGIR '18: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018.

[13] S. M. Alzahrani, N. Salim and A. Abraham, "Understanding Plagiarism Linguistic Patterns, Textual Features and Detection Methods," IEEE Transaction on System, Man and Cybernetics, vol. 42, no. 2, pp. 133-149, 2012.

[14] Z. Ceska, M. Toman and K. Jezek, "Multilingual Plagiarism Detection," Lecturer Notes in Computer Science, vol. 5253, pp. 83-92, 2008.

[15] M. Potthast, B. Stein, A. Eiselt, A. Barrón-Cedeño and P. Rosso, "Overview of the 1st International Competition on Plagiarism Detection," in 3rd Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN 2009) at SEPLN, 2009

[16] M. Potthast, A. Barrón-Cedeño, A. Eiselt, B. Stein and P. Rosso, "Overview of the 2nd International Competition on Plagiarism Detection," in Working Notes Papers of the CLEF 2010 Evaluation Labs,, 2010.

[17] M. Potthast, A. Eiselt, A. Barrón-Cedeño, B. Stein and P. Rosso, "Overview of the 3rd International Competition on Plagiarism Detection," in Working Notes Papers of the CLEF 2011 Evaluation Labs, 2011.

[18] M. Potthast, M. Hagen, T. Gollub, M. Tippmann, J. Kiesel, P. Rosso, E. Stamatatos and B. Stein, "Overview of the 5th International Competition on Plagiarism Detection," in 3rd Workshop

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v14i2.22841

on Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN 2013) at CLEF 2013, 2013

[19] F. Rangel, M. Montes-y-Gómez, M. Potthast and B. Stein, "Overview of the 6th Author Profiling Task at PAN 2018: Cross-domain Authorship Attribution and Style Change Detection," in CLEF 2018 Evaluation Labs and Workshop – Working Notes Papers, Avignon, France, 2018.

[20] W. Daelemans, M. Kestemont, E. Manjavacas, M. Potthast, F. Rangel, P. Rosso, G. Specht, E. Stamatatos, B. Stein, M. Tschuggnall, M. Wiegmann and E. Zangerle, " Overview of PAN 2019: Author Profiling, Celebrity Profiling, Cross-domain Authorship Attribution and Style Change Detection," in 10th International Conference of the CLEF Association (CLEF 2019), 2019.

[21] A. Barrón-Cedeño, M. Potthast, P. Rosso, B. Stein and A. Eiselt, "Corpus and Evaluation Measures for Automatic Plagiarism Detection," in LREC 2010, Seventh International Conference on Language Resources and Evaluation, 2010.

[22] C. v. Rijsbergen, S. Robertson and M. Proter, "New models in probabilistic information retrieval," British Library, London, 1980

[23] M. Potthast, B. Stein, A. Barrón-Cedeño and P. Rosso, "An Evaluation Framework for Plagiarism Detection," in 23rd International Conference on Computational Linguistics (COLING 2010), Beijing, China, 2010