Implementasi Algoritma *Merging Context Seeds* untuk *Plagiarism Detection*

Tugas Akhir

Diajukan untuk memenuhi salah satu syarat memperoleh gelar sarjana dari Program Studi Teknik Informatika
Fakultas Informatika
Universitas Telkom

Yusuf Anugrah Putra Aditama
1103120030

Program Studi Sarjana Informatika
Fakultas Informatika
Universitas Telkom
Bandung
2017

Lembar Pernyataan

Dengan ini saya menyatakan tugas akhir dengan judul "**Implementasi Algoritma *Merging Context Seeds* untuk *Plagiarism Detection***" beserta seluruh isinya adalah benar-benar karya saya sendiri dan saya tidak melakukan penjiplakan dan pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan. Atas pernyataan ini saya siap menanggung risiko adanya pelanggaran terhadap etika keilmuan dalam karya saya ini, atau ada klaim dari pihak lain terhadap keasilan karya ini.

Bandung, 8 Juni 2017
Yang membuat pernyataan,

Yusuf Anugrah Putra Aditama
NIM. 1103120030

Lembar Pengesahan

Implementasi Algoritma *Merging Context Seeds* untuk *Plagiarism Detection*

Merging Context Seeds Algorithm Implementation for Plagiarism Detection

Yusuf Anugrah Putra Aditama
NIM : 1103120030

Tugas akhir ini telah diterima dan disahkan untuk memenuhi sebagian syarat memperoleh gelar pada Program Studi Sarjana Teknik Informatika
Universitas Telkom

Bandung, 8 Juni 2017

Menyetujui,

Pembimbing 1,	Pembimbing 2,
Ir. Moch. Arif Bijaksana	Syahrul Mubarok
NIP. 03650312-4	NIP. 10830757-3

Mengesahkan,
Ketua Program Studi
Sarjana Teknik Informatika

Ir. Moch. Arif Bijaksana
NIP. 03650312-4

Abstrak

Plagiat merupakan masalah yang sering ditemukan di masyarakat, bahkan menurut survey 89% responden sering menemukan kasus plagiat pada bidangnya masing-masing. Tindak plagiat ini dapat berupa mengambil tulisan orang lain yang digunakan untuk kepentingan diri sendiri. Adapun salah satu pendekatan yang dapat dilakukan untuk mendeteksi tindak plagiat ini adalah dengan *Text Alignment*. Sehingga pada penelitian ini diusung salah satu metode yaitu *Merging Context Seeds* yang bekerja dengan cara menggabungkan ciri yang ada pada *suspicious-document* dan *source-document* dengan metode ekstraksi ciri *n-skip-k-grams*. Dengan diimplementasikannya metode *Merging Context Seeds*, penilitian ini mendapat nilai F_1 sebesar 0.532.

Kata kunci: *merging context seeds*, *seeds*, *merge*

Abstract

Plagiarism is a case that can be found in society especially in educational field, even on survey 89% responden often found plagiarism case in their own field. Plagiarsm can be form as using other people writing, or idea as their own to gain benefit from it. As for, there is an approach that can be done to detect plagiarism which is Text Alignment. Therefore in this research, researcher submit method called Merging Context Seeds that works by merge feature between suspicous-document and source-document generated by n-skip-k-grams feature extraction method. With the implementation of Merging Context Seeds method, this research get F_1-Score 0.532.

Keywords: *merging context seeds, seeds, merge*

Lembar Persembahan

Yang Utama Dari Segalanya,
Sembah sujud serta syukur kepada **Allah SWT** yang membekali penulis dengan ilmu yang berlimpah, dan karna izin-Nya pula lah penelitian ini dapat selesai. Sholawat serta salam selalu terlimpahkan keharibaan **Rasullah Muhammad SAW**.

Pembimbing Tugas Akhir
Pak Moch. Arif Bijaksana dan Pak Syahrul Mubarok selaku pembimbing dalam pengerjaan tugas akhir ini. Terimakasih pak untuk bimbingan dan pencerahannya. Tanpa bapak mungkin tugas akhir ini tidak akan selesai. Mohon maaf bila ada kekurangan selama penulis menjadi murid bimbingan bapak.

Dosen Wali dan IF-36-02
Terima kasih untuk Bu Tisa (Siti Saadah) yang sudah membantu dan memberikan arahan selama masa kuliah. Dan anak-anak IF-36-02 yang menjadi rumah selama masa kuliah ini. Terutama Rizki yang sering berbagi ilmu baik didalam maupun luar kuliah.

Kontrakan Bahagia
Babeh, Ali, Lutpi dan Woempa yang menemani hidup dikosan, dan bertahan hidup di dakol. Semoga semuanya cepet lulus yaaa.

Keluarga Besar BASDAT dan IFLAB
Terima kasih sudah diberi tempat untuk mengasah kemampuan. Terima kasih terutama untuk *Foya Foya, Kamar Kos, ASLAB 15/16 dan ASLAB 16/17*. Semoga semua cepet lulus juga yaaa.

Dan kepada seluruh pihak yang tidak sempat penulis sebutkan, semoga kesuksesan selalu menyertai kita semua.

Dan yang tidak mungkin terlupa,
Terima kasih untuk **Papah**, **Mamah**, **Teteh**, **Bebi**, dan **Gege**. Yang selalu mendukung dan memberi semangat tidak hanya selama pengerjaan tugas akhir ini tetapi juga selama menjalani kuliah dan kehidupan ini. Dan terima kasih juga udah nanyain tiap hari gimana tugas akhirnya, yang akhirnya selesai. Terima Kasih. *This is for you.*

Kata Pengantar

Dengan menyebut nama Allah SWT yang Maha Pengasih lagi Maha Penyayang, penulis panjatkan puji dan syukur atas kehadirat-Nya, atas limpahan rahmat kepada penulis, sehingga penulis mampu menyelesaikan penulisan buku tugas akhir ini yang berjudul "Implementasi Algoritma *Merging Context Seeds* untuk *Plagiarism Detection*".

Penulis juga mengucapkan terima kasih sebesar-besarnya kepada seluruh pihak yang secara langsung maupun tidak langsung turut membantu dalam penelitian ini. Semoga kelebihan maupun kekurangan yang ada pada penelitian ini dapat berguna untuk masyarakat kedepannya. Segala masukan, kritik, dan saran kepada penulis sangat dinantikan untuk penelitian selanjutnya agar lebih baik.

Bandung, 8 Juni 2017

Yusuf Anugrah Putra Aditama
NIM. 1103120030

Daftar Isi

Daftar Gambar

Daftar Tabel

Daftar Istilah

1. ***Passage*** : Bagian pada dokumen berupa kumpulan karakter.
2. ***Passage Reference/Seeds*** : Ciri pada *source-document* dan *suspicious-document* yang saling beririsan.
3. ***Pair*** : Pasangan dokumen *source* dan *suspicious*
4. ***Perimeter*** : Luas dari *cluster* yang berbentuk persegi.
5. ***Seeds*** : Fitur yang sama yang ada pada *pair*.
6. ***Char map*** : Tabel yang menunjukan kemunculan suatu karakter pada suatu dokumen.
7. ***Word map*** : Tabel yang menunjukan kemunculan awal dan akhir suatu kata pada suatu dokumen.
8. ***Obfuscation*** : Tingkat tindak plagiat.

Bab 1. Pendahuluan

1.1. Latar Belakang

Menurut hasil survey yang dilakukan oleh iThenticate[3], sebanyak 89% responden yang ditemui menjawab sering menemui kasus plagiat pada bidangnya masing-masing. Dan lebih dari 25% responden menyatakan bahwa plagiat merupakan masalah serius yang harus diselesaikan. Namun dengan makin banyaknya dokumen yang terkumpul akan semakin sulit mendeteksi tindak plagiat secara manual. *Text alignment* adalah solusi yang ada untuk menyelesaikan masalah plagiat yang ada, dengan cara membangkitkan bagian pada dua buah dokumen yang terindikasi plagiat.

Paragraf atau kalimat yang serupa yang digunakan pada dua buah dokumen dapat dideteksi dari penggunaan kata dan penataannya. Hal ini yang menjadi alasan pendekatan *text alignment* dapat dilakukan, yaitu dengan menjajarkan seluruh fitur yang ada pada dua buah dokumen dan mencari irisan antara dua buah dokumen tersebut. Dari seluruh metode *text alignment* yang ada, dipilih *Merging Context Seeds* yang merupakan salah satu metode yang diajukan pada PAN[4] yang memiliki nilai perfomansi *plagdet* 0.826. *Merging context seeds* terfokus kepada *seeds* yang merupakan fitur yang beririsan antara dua buah dokumen, dan melakukan *clustering* atau pengelompokan data pada seeds yang ada. Dari kelompok data yang didapat, dipilih yang merupakan tindak plagiat.

Permasalahan yang akan diselesaikan pada penilitian ini adalah, bagaimana mengimplementasikan algoritma *merging context seeds* dan membangun sistem yang mampu mendeteksi tindak plagiat dari berbagai tipe tindak plagiat yang ada secara akurat.

1.2. Rumusan Masalah

Pada tugas akhir ini masalah yang dibahas terfokus pada *text alignment* yang merupakan tahap kedua dari *task plagiarism detection*. Dimana sepasang dokumen *suspicous-document* dan *source-document* atau dapat disebut sebagai *pair* akan diolah dengan metode *merging context seed* untuk membuktikan adanya tindak plagiat yang ada pada *pair*. Kumpulan *pair* ini didapat dari proses *source retrieval* pada tahap awal *plagiarism detection*.

Terdapat 5 kategori tindak plagiat yang akan dibuktikan, yaitu :

1. ***No-Plagiarism***
   Pasangan dokumen tidak tedardapat tindak plagiat.

2. ***No-Obfuscation***
   Pasangan dokumen melakukan tindak plagiat berupa *copy-paste*, yaitu menggunakan kalimat sumber secara utuh tanpa melakukan perubahan apapun.

3. ***Random-Obfuscation***
   Pasangan dokumen melakukan tindak plagit berupa penghapusan, penambahan, dan/atau penggantian kata pada kalimat.

4. ***Translation-Obfuscation***
   Dokumen yang ada ditranslasi ke bahasa lain, yang kemudian di translasi ke bahasa inggris.

5. ***Summary-Obfuscation***
   Tindak plagiat berupa parafrase, yaitu merangkum intisari dari kalimat sumber.

Keluaran yang diharapkan dari sistem yang dibangun adalah letak kemunculan bagian yang terbukti plagiat pada *suspicous-document* dan *source-document* beserta nilai perfomansi dari pair yang ada.

1.3. Batasan Masalah

Adapun yang menjadi batasan masalah pada penelitian ini adalah :

1. Tahapan *plagiarism detection* yang dilakukan hanya *text alignment*.
2. *Datasets* didapat dari proses *source retrieval* yang sudah dilakukan oleh PAN[4].
3. *Datasets* yang digunakan merupakan dokumen berbahasa inggris.
4. *Datasets* yang digunakan merupakan dokumen *plain text*.

1.4. Tujuan

Membangun sistem yang dapat mendeteksi dan menunjukan tindak plagiat *no plagiarism, no obfuscation, random obfuscation, translation obfuscation* dan *summary obfuscation* dari pasangan dokumen *suspicious* dan *source* dengan mengimplementasikan metode *Merging Context Seeds*.

1.5. Metodologi

Adapun metodologi yang digunakan untuk memecahkan masalah yang ada yaitu sebagai berikut :

1. Studi Literatur
   Mengumpulkan dan mempelajari kajian yang digunakan untuk menyelesaikan masalah yang ada, yang dapat membantu dalam metode *Merging Context Seeds*.

2. Analisis Kebutuhan Sistem
   Dilakukan analisis terhadap kebutuhan sistem untuk mencapai tujuan pada tugas akhir ini.

3. Perancangan Sistem
   Merancang alur sistem yang akan dibangun untuk tugas akhir ini, dimulai dari *input*, proses, hingga *output*.

4. Implementasi
   Mengimplementasikan metode yang dipelajari kedalam sistem, mulai dari *preprocessing*, ekstraksi ciri, implementasi *Merging Context Seed* dan *clustering* menggunakan *Agglomerative single-link clustering*.

5. Pengujian Sistem
   Melakukan uji coba dengan menjalankan sistem yang telah dibuat dan melakukan analisis sementara.

6. Analisis
   Menganalisis hasil *output* yang dikeluarkan dari sistem. Menghitung nilai *precision* dan *recall* berdasarkan hasil *running* dari *datasets* yang ada.

7. Penyusunan Laporan
   Pembuatan laporan mengenai kegiatan dan sistem yang di bangun yang meliputi latar belakang, rumusan masalah, tujuan, implementasi sistem hingga hasil analisis yang dilakukan selama pengerjaan tugas akhir.

Bab 2. Studi Literatur

2.1. *Plagiarism*

Plagiarism atau plagiat merupakan tindakan mengklaim suatu ide, gagasan ataupun tulisan orang lain sebagai miliknya sendiri. Gagasan atau tulisan yang di klaim dapat berupa jurnal, buku, ucapan ataupun hasil diskusi.

Tindak plagiat pada hasil tulisan dapat berupa menghilangkan atau menambahkan satu, atau beberapa kata dari tulisan sumber seseorang dan digunakan pada karya / tulisan orang lain[5]. Kedua teks di bawah ini merupakan contoh tindak plagiat dengan menambah dan mengurangi kata pada teks sumber,

Teks Asli : *Dengan menggunakan metode* ***k-Nearest Neighbor*** *kita dapat mengetahui derajat ketetangaan suatu node dengan node lainnya.*
Teks Plagiat : *Dengan metode* ***k-Nearest Neighbor*** *kita dapat mengetahui derajat ketetangaan suatu node dengan node lain disekitarnya.*

Selain itu ada juga *paraphrase*. Yaitu, mengubah tataan suatu kalimat menjadi bentuk lain. Tindak plagiat ini sulit di deteksi karena perubahaan dapat banyak terjadi khususnya apabila ada perubahan kalimat aktif menjadi pasif, atau kebalikannya. Teks di bawah merupakan contoh tindak plagiat *paraphrase*,

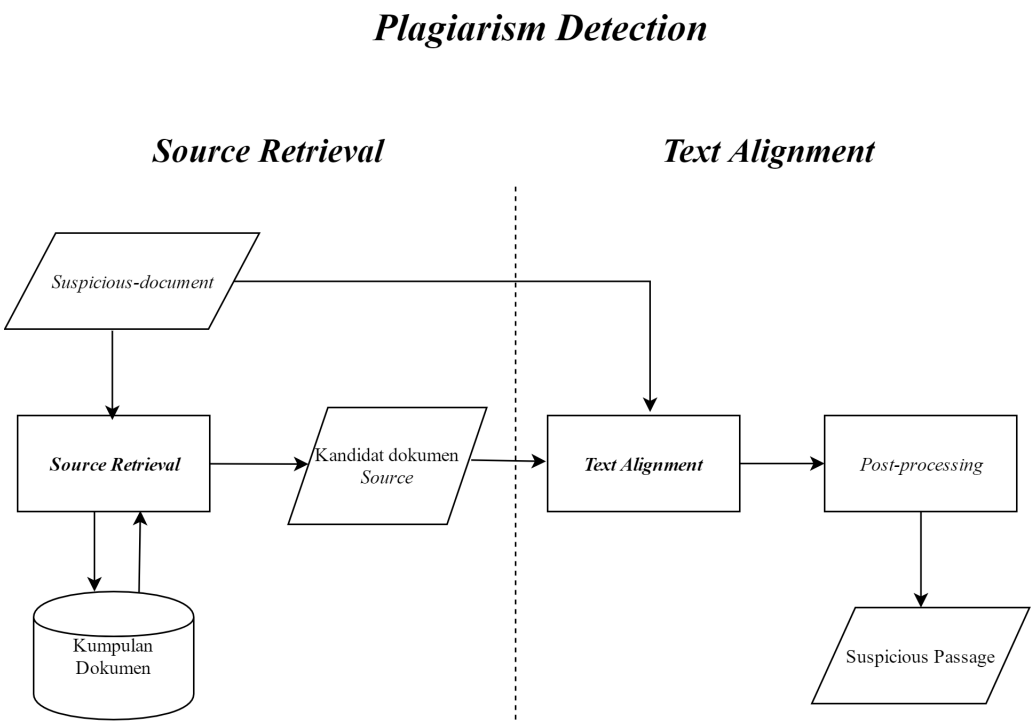
Teks Asli : *Dengan menggunakan metode* ***k-Nearest Neighbor*** *kita dapat mengetahui derajat ketetangaan suatu node dengan node lainnya.*
Teks Plagiat : *Untuk menghitung derajat ketetanggaan suatu node dapat menggunakan metode* ***k-Nearest Neighbor****.*

Tetapi walaupun suatu teks diubah dengan *paraphrase* tindak *plagiarism* masih dapat di indetifikasi karena ada kemiripan penggunaan kata yang ada pada dokumen *source* dan *suspicious*.

2.2. *Plagiarism Detection*

Plagiarism Detection adalah solusi untuk masalah *plagiarism*. Tujuan utama dari *Plagiarism Detection* ini adalah mengidentifikasi suatu dokumen yang disebut dengan *suspicious-document*, apakah dokumen tersebut mempunyai

teks atau bagian yang diambil dari satu atau beberapa dokumen lain (*source-document*). Dan membuktikan apabila terdapat teks yang diambil dari *source-document* di *suspicious-document*. Gambar 2.1 menunjukan alur *Plagiarism Detection* secara umum[4, 1].



Gambar 2.1: Alur keseluruhan*Plagiarism Detection*[1]

2.2.1 *Source Retrieval*

Source Retrieval merupakan *task* pertama untuk *Plagiarism Detection*. Pada tahap ini suatu dokumen akan diuji dengan cara melakukan pencarian perkalimat dari dokumen yang diuji. Kalimat yang diuji akan dimasukan kedalam *query* mesin pencarian yang berisikan jurnal atau dokumen sejenis. Kemudian apabila ada kemiripan dengan suatu dokumen, maka akan dihasilkan informasi bahwa dokumen yang diuji terindikasi plagiat dengan dokumen yang ditemukan[6]. Informasi atau pasangan dokumen ini disebut juga dengan *pair*.

2.2.2 *Text Alignment*

Text Alignment merupakan pendekatan yang dapat menguji *pair* yang didapat dari tahap *Source Retrieval*. Pada tahap ini *pair* dari proses *Source Retrieval* akan diuji, apakah dokumen tersebut terbukti memplagiat dokumen sumber atau tidak dengan cara mengekstrasi ciri yang ada pada dokumen yang terindikasi, dan dokumen sumber yang kemudian diolah dengan metode khusus.

Plagiarism Detection dengan pendekatan *Text Alignment* ini mempunyai 3 tahapan dasar, yaitu :

1. *Seeding*
   Terdapat pasangan *suspicious* dan *source*, setiap elemen yang sama antar kedua dokumen tersebut disebut dengan *seed*. *Seed* ini dapat berupa fitur yang diekstrak dari kedua dokumen.

2. *Extension*
   Seed yang ada akan diolah dengan dijajarkan dengan *seed* lainnya hingga mendapatkan bagian yang dijadikan sebagai bagian yang diduga plagiat.

3. *Filtering*
   Menghapus kumpulan *seed* yang dianggap tidak memenuhi kriteria bagian yang termasuk plagiat. Sedangkan kumpulan *seed* sisanya, dianggap menjadi bagian yang terbukti plagiat.

2.3. *Skip Gram*

Skip Gram atau yang sering disebut dengan k-skip-n-grams merupakan salah satu metode untuk mengekstraksi ciri dari suatu dokumen yang juga merupakan bentuk lain dari *N-Gram*. Dengan *skip gram* rangkaian karakter/kata yang ada pada dokumen akan di lompati untuk mendapatkan fitur. k menunjukan jumlah karakter/kata yang dilompati. Sedangkan n menunjukan panjang karakter untuk satu buah fitur. Fitur hasil dari metode k-skip-n-grams sendiri dapat didefinisikan pada Persamaan 2.1[7].

$$\{w_{i_1}, w_{i_2}, ...w_{i_n} | \sum_{j=1}^{n} i_j - i_{j=1} < k\} \tag{2.1}$$

Sebagai contoh, kalimat :

" A sentence is a group of words that are put together to mean something. "

Apabila kalimat diatas di ekstraksi menggunakan 2-skip-2-grams maka akan didapat fitur sesuai pada Tabel 2.1 :

Tabel 2.1: Contoh Ekstraksi Ciri dengan 2-skip-2-grams

Daftar Fitur
A_IS
SENTENCE_A
IS_GROUP
A_OF
GROUP_WORDS
OF_THAT
WORDS_ARE
THAT_PUT
ARE_TOGETHER
PUT_TO
TOGETHER_MEAN
TO_SOMETHING
MEAN_*
SOMETHING_*

2.4. *Merging Context Seed*

Merging Context Seed merupakan salah satu metode dengan pendekatan *Text Alignment*. Metode ini memiliki tahapan yang serupa pada pendekatan *Text Alignment*[8, 1] pada umumnya.

Metode ini akan mengolah dokumen pada level karakter. Diketahui sebuah dokumen terdiri dari bagian-bagian (huruf, kata, atau kalimat) yaitu P, dimana $P = \{x_i : 0 \leq a \leq i < b \leq n\}$ dimana $x_i = (c, i), c \in C$. C merupakan kumpulan simbol, dan i merupakan letak kemunculan karakter. P juga dapat dinotasikan dengan $P = [x_{a_i}, x_{b_i}]$

2.4.1 *Seed generation*

Terdapat dokumen yang terindikasi bernama dokumen X *(suspicious-document)* dan dokumen sumbernya bernama dokumen Y *(source-document)*. Setiap karakter yang ada pada tiap dokumen dipetakan kedalam *index map* untuk mengetahui letak kemunculan karakter dan fitur nantinya.

Preprocessing

Kedua dokumen akan melalui proses *preprocessing* dengan tahapan yaitu :

1. Menghapus *white space* dan *enter space*.
2. Mengubah seluruh karakter yang ada menjadi huruf kecil.
3. Menghapus seluruh karakter yang tidak termasuk kedalam *alphanumeric character*.

4. Menghapus seluruh *stopwords*.

Hal ini dilakukan untuk mengurangi jumlah fitur yang akan dihasilkan, sekaligus menghilangkan ada kemungkinan fitur yang tidak relevan sehingga membuat *noise* pada data yang akan diolah.

Ekstraksi Ciri

Dokumen yang telah melalui tahap *preprocessing* kemudian diekstraksi cirinya dengan menggunakan *k-skip-n-grams* dengan nilai $k = 1, 2, 3, 4$ dan $n = 2$, atau dapat disebut dengan $1 - 4skip - bigram$ yang dinotasikan oleh Persamaan 2.2.

$$\varphi(x_i) = \begin{cases} \{w_\beta\_w_\alpha\}\beta = \alpha - 4, ..., \alpha & x_i = w_\alpha[0] \\ 0 & \text{Lainnya} \end{cases} \tag{2.2}$$

Tabel 2.2 menunjukan *feature map* yang berisikan fitur, *token* dan letak kemunculan fitur dari kata :

" A sentence is a group of words that are put together to mean something. "

Tabel 2.2: *Feature Map*

Offset	Token	f1	f2	f3	f4
2	sentence	*_sentence	*_sentence	*_sentence	*_sentence
16	group	sentence_group	*_group	*_group	*_group
25	words	group_words	sentence_words	*_words	*_words
41	put	words_put	group_put	sentence_put	*_put
45	together	put_together	words_together	group_together	sentence_group
57	mean	together_mean	put_mean	words_mean	group_mean
61	something	mean_something	together_something	put_something	words_something

Kolom $f1$ menunjukan fitur yang dihasilkan melalui proses *1-skip-bigram*, sedangkan $f2$ menunjukan fitur yang dihasilkan melalui proses *2-skip-bigram*, dan seterusnya. *Offset* merupakan letak kemunculan kata/karakter pada kalimat sebelum melalui proses preprocessing, sehingga nantinya dapat diketahui letak pasti kata/karakter yang diplagiat. Sedangkan karakter $*$ digunakan sebagai penanda *NULL* atau karakter kosong.

Feature Relevance Filtering

Pada tahap ini fitur yang ada akan dihitung jumlah kemunculannya pada dokumennya. Apabila jumlah kemunculan fitur sesuai dengan *threshold threshold* $1 \leq |X(f)| \leq \varrho$ maka fitur akan disimpan, apabila melebihi *threshold* maka fitur akan dihapus.

Seed Generation

Seluruh fitur yang ada pada dokumen X dan dokumen Y dipetakan menjadi *index map* dari XY 2.3.

$$\iota_{XY} : F \to \wp(X \times Y), \quad f \mapsto \{(x_i, y_j) \quad | \quad f \in \varphi(x_i) \text{ and } f \in \varphi(j_i)\} \tag{2.3}$$

Persamaan di atas akan memetakan seluruh fitur antara dokumen X dan dokumen Y. Lalu dari index map XY karakter yang mempunyai fitur yang sama pada kedua dokumen disebut sebagai *passage reference*, sedangkan kumpulan dari *passage reference* ini dapat disebut sebagai *seed set*. *Passage reference* ini merupakan titik awal untuk proses dari pendekatan *text alignment* pada metode *Merging Context Seeds*.

2.4.2 *Merging*

Kriteria *Merge*

Merging merupakan proses dimana *passage reference* yang didapat pada proses *Seed Generation* akan di *cluster*. Diketahui untuk setiap *passage* pada dokumen X adalah *Passage*, dimana $P = [x_{a_i}, x_{b_i}]$ yang merupakan bagian dari dokumen X.

Untuk menghitung jarak antar *passage* pada dokumen X, berlaku 2.4.

$$dist(P_1, P_2) = \min\{|x_1 - x_2| : a_1 \leq x_1 \leq b_1, a_2 \leq x_2 \leq b_2\} \tag{2.4}$$

Untuk $P_1 = [x_{a_1}, x_{b_1}]$ dan $P_2 = [x_{a_2}, x_{b_2}]$.

Diketahui pula, *perimeter* atau luas dari satu *passage reference* adalah 2.5

$$\pi(r) = 2(b-a) + 2(d-c) \quad \text{jika,} r = [x_a, x_b] \times [y_c, y_d] \text{ dan } \pi(0) = 0 \tag{2.5}$$

Sedangkan untuk menghitung jarak antara 2 *passage reference*, $P_1 \times Q_1, P_2 \times Q_2 \subseteq X \times Y$ adalah 2.4.2

$$dist(P_1 \times Q_1, P_2 \times Q_2) = \frac{2 \times dist(P_1, P_2) + 2 \times dist(Q_1, Q_2)}{\sigma + \pi(P_1 \times Q_1) + \pi(P_2 \times Q_2)} \tag{2.6}$$

Clustering

Setelah mengetahui kriteria proses *merging* maka *passage reference* dapat di kluster menggunakan *single linkage clustering* yang merupakan bagian dari *Agglomerative Clustering*. *Clustering* dilakukan hingga tidak ada jarak antar *cluster/passage reference* yang kurang dari batas $\tau \geq 0$.

2.4.3 *Filtering*

Membuang seluruh *cluster* yang panjangnya kurang dari ν, dan sisa *cluster* yang ada akan dianggap sebagai bagian yang terbukti melakukan plagiat. *Cluster* yang tersisa membangun 2 buah *Passage* $P = [x_{c_{i_{min}}}, x_{c_{i_{max}}}]$ untuk dokumen *suspicious* dan dokumen *source*. Membuat *output* berupa lokasi karakter awal plagiat dan lokasi akhir karakter plagiat pada dokumen X dan dokumen Y.

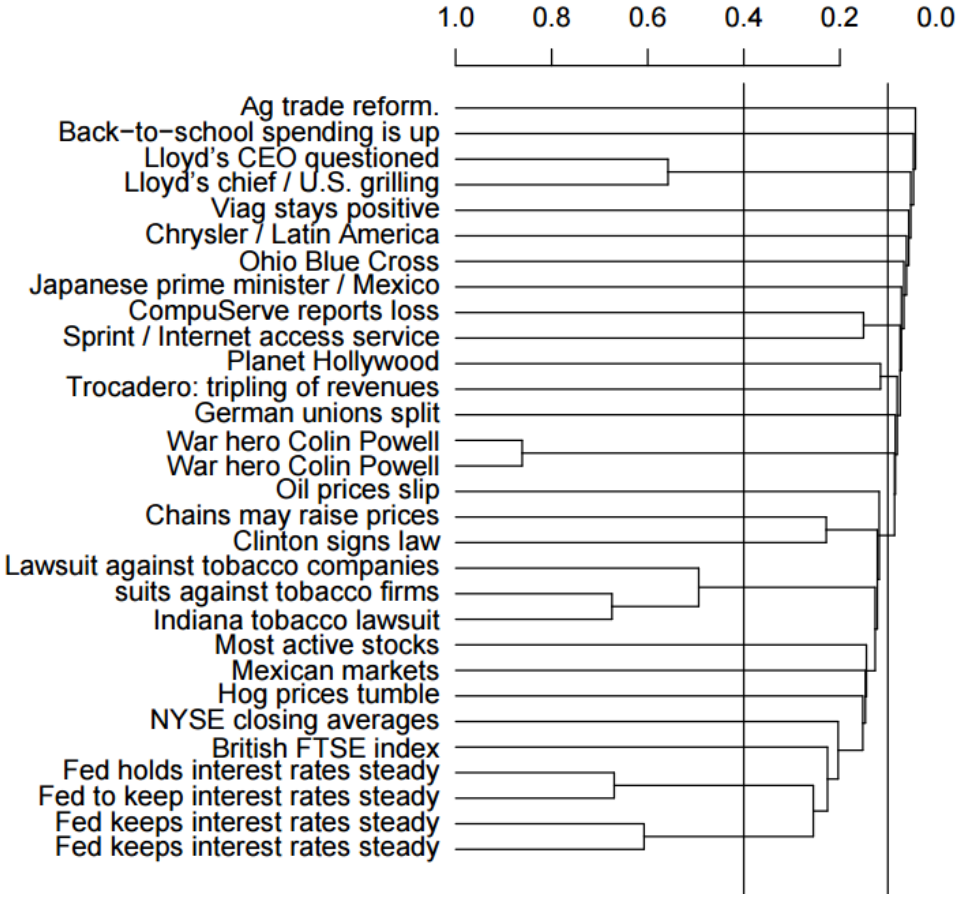
Jika terdapat *passage* yang terdeteksi plagiat maka akan dianggap r untuk set R. r dapat direpresentasikan sebagai :

Tabel 2.3: Format r

Suspicous P Start	Suspicious P End	Source P Start	Source P End

2.5. Single Linkage Clustering

Merupakan salah satu metode dari *hierarchical clustering*[2] yang bekerja dengan cara *bottom-up*. Setiap elemen yang ada dianggap sebagai *cluster* yang berdiri sendiri. *Clustering* akan dilakukan dengan menggabungkan jarak terdekat elemen 2 cluster dari seluruh yang ada, hingga menjadi 1 *cluster*. Gambar 2.2 menunjukan contoh *clustering* yang dilakukan pada 30 buah dokumen.

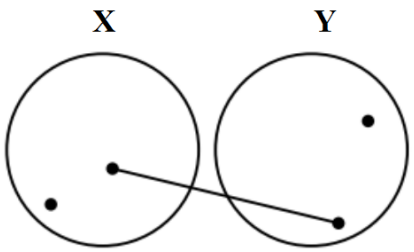


Gambar 2.2: Contoh *Dendogram* dari *Single Linkage Clustering*[2]

Untuk tugas akhir ini penggabungan *cluster* yang ada menggunakan jarak yang dinotasikan[9] oleh Persamaan 2.7.

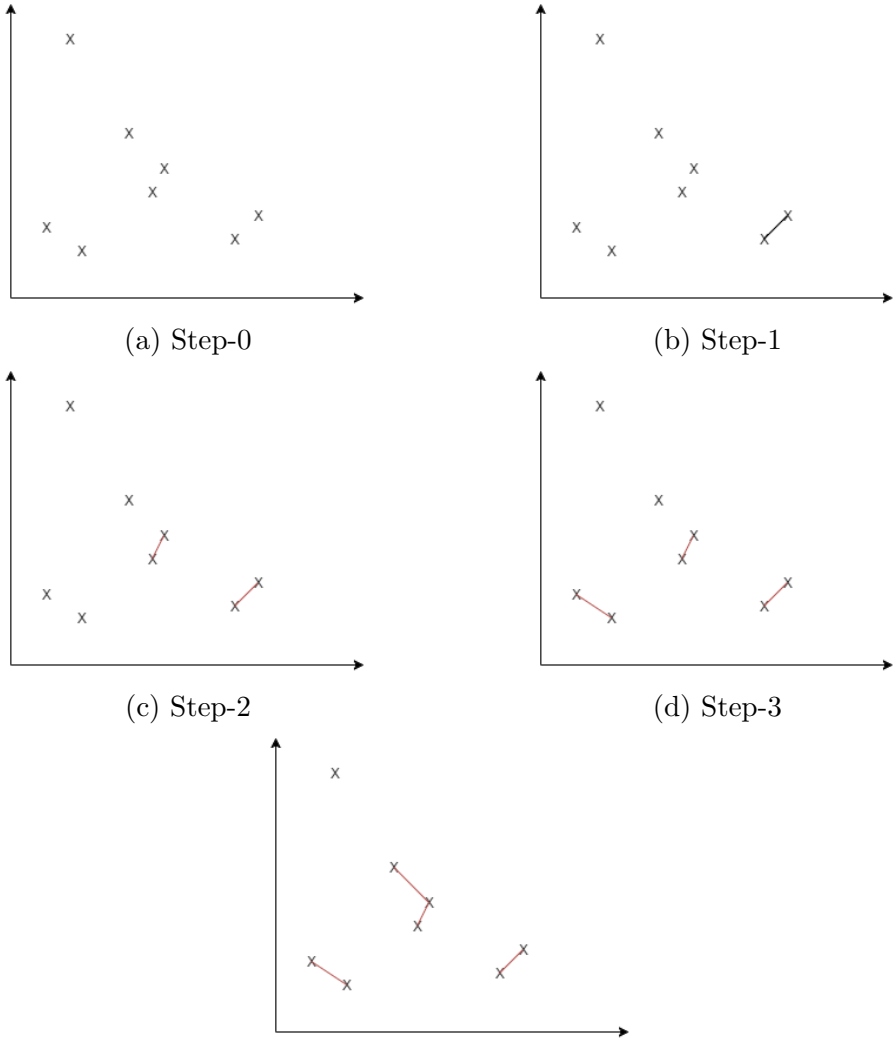
$$D(X,Y) = min\ d(x,y);\ x \in X, y \in Y \tag{2.7}$$

Atau dapat digambarkan seperti pada Gambar 2.3



Gambar 2.3: Representasi *Single Linkage Clustering*

Selain itu pada tugas akhir ini yang menjadi kondisi terminasi adalah $D(X, Y)$. Apabila jarak terpendek pada *cluster* $\geq \gamma$, maka proses *cluster* terhenti. Gambar 2.4 menunjukan contoh proses *clustering* dengan *Single Linkage Clustering*.



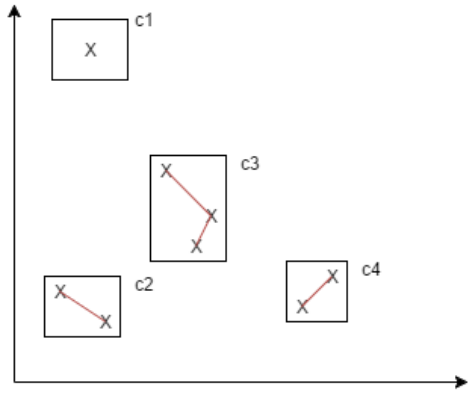
(a) Step-0 (b) Step-1

(c) Step-2 (d) Step-3

(e) Step-4

Gambar 2.4: Contoh Proses *Clustering*

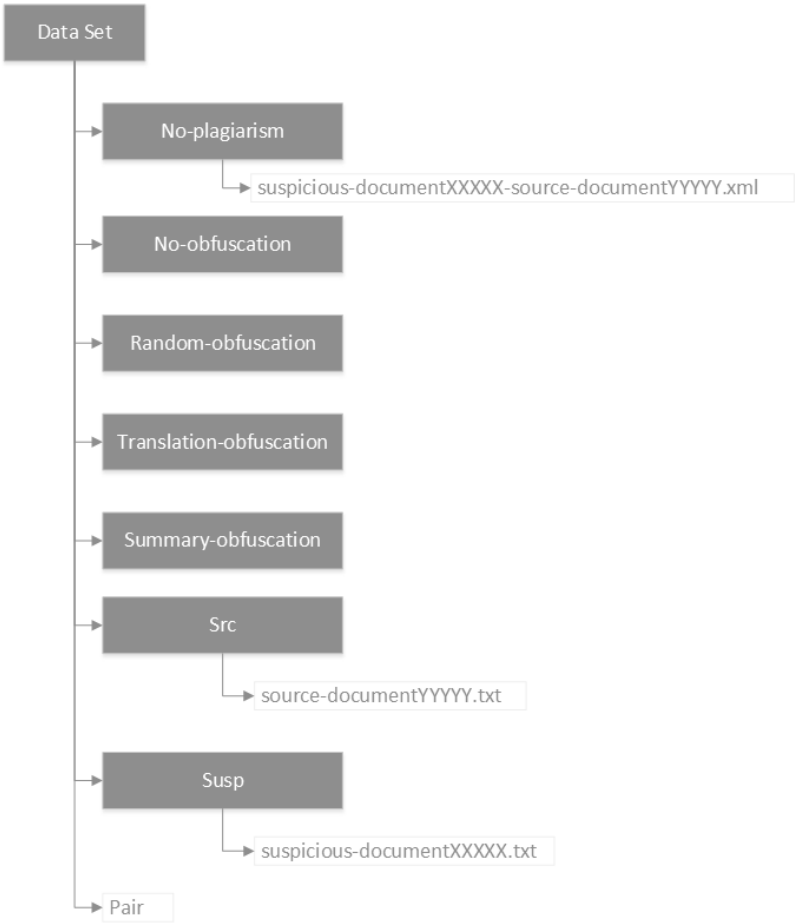
Gambar 2.5 merupakan hasil akhir dari proses *clustering* di atas.



Gambar 2.5: *Cluster* Akhir

2.6. *Datasets*

Datasets yang digunakan pada tugas akhir ini diambil di *web* resmi PAN[4, 10]. Gambar 2.6 menunjukan hierarki dari *dataset* yang digunakan.



Gambar 2.6: *Datasets*

Dataset di atas mempunyai bagian-bagian umum sebagai berikut :

1. **Susp**
   Kumpulan dokumen yang terindikasi plagiat melalui proses *Source Retrieval*.

2. **Src**
   Merupakan dokumen sumber di plagiat oleh dokumen Susp melalui proses *Source Retrieval*.

3. **Pair**
   Merupakan informasi pasangan dokumen yang terindikasi, formatnya adalah *suspicious-documentXXXXX.txt source-documentYYYYY.txt*. Dimana *suspicious-documentXXXXX.txt* merujuk pada dokumen yang ada di Susp, dan *source-documentYYYYY.txt* merujuk pada dokumen yang ada di Src.

4. ***No-Plagiarism, No-Obfuscation, Random-Obfuscation, -Obfuscation, Summary-Obfuscation***
   Bagian ini merupakan kelas hasil klasifikasi tindak plagiat dari file Pair.

   (a) ***No-Plagiarism*** : Tidak terdeketsi tindak plagiat.

   (b) ***No-Obfuscation*** : Tindak plagiat berupa *copy-paste*.

   (c) ***Random-Obfuscation*** : Tindak plagiat berupa menghilang / menambahkan kata pada kalimat yang digunakan.

   (d) ***Translation-Obfuscation*** : Tindak plagiat berupa menerjemahkan kata.

   (e) ***Summary-Obfuscation*** : Tindak plagiat berupa merangkum suatu kalimat/paragraf.

Sebagai contoh, pada file *pairs* terdapat informasi sebagai berikut :

suspicious-document00005.txt source-document01496.txt

Berarti, menurut proses proses *Source Retrieval* sebelumnya terdapat indikasi plagiat dari dokumen *suspicious-document00005.txt* bersumber dari dokumen *source-document01496.txt*. Dan menurut data yang ada, tindak plagiat pada dokumen tadi termasuk ke dalam *No-Obfuscation*, dimana tindak plagiat yang dilakukan berupa *copy-paste*. Dan setelah diteliti, tindak plagiat terbukti dikarenakan pada kedua dokumen terdapat kalimat yang sama.

Selain itu terdapat $suspicious-document00005.txtsource-document01496.xml$ yang isinya merupakan meta data letak plagiat pada kedua buah dokumen tersebut, yang berguna untuk menghitung perfomansi dari sistem yang dibangun. Isi dari meta data tersebut adalah diantaranya :

1. *Suspicious-document*.

2. *Source-document*.

3. Jenis plagiat.
4. Letak plagiat *source-document*.
5. Panjang karakter yang di plagiat pada *source-document*.
6. Letak plagiat *suspicious-document*.
7. Panjang karakter yang plagiat pada *suspicious-document*.

Berikut merupakan potongan dokumen *suspicious-document00005.txt* :

*.....WAYS TO SEND YOUR DOCUMENTATION Fax to 304-724-0909 Scan and email to DSA@apus.edu Mail to APUS ATTN : Disability Accomodations 10110 Battleview Parkway Suite 114 Manassas, VA 20109 x ED502 * required berfore student may register for courses Prepare a short essay of approximately 300 words (one page) describing why you are interested in this particular degree program. Your sample should preferably be written in Word and double-spaced. The Education degree coordinator will use your writing sample to assess your written communications skills.*
** Writing Sample Two character references are required from people who can attest to your moral and ethical character. Example of such people include supervisors, religious leaders, military commanders, school officials, or others who know you well and can provide credible information about you. He served at the Pentagon as par of Joint Staff in support of Noble Eagle and Enduring Freedom.....*

Berikut merupakan potongan dokumen *source-document01496.txt* :

*x ED502 * required berfore student may register for courses Prepare a short essay of approximately 300 words (one page) describing why you are interested in this particular degree program. Your sample should preferably be written in Word and double-spaced. The Education degree coordinator will use your writing sample to assess your written communications skills.*
** Writing Sample Two character references are required from people who can attest to your moral and ethical character. Example of such people include supervisors, religious leaders, military commanders, school officials, or others who know you well and can provide credible information about you. Forms will be provided to you by your admissions representative. Once forms are completed and signed by the references, send them to APUS following the document submission instructions below.....*

Bagian yang di *highlight* merupakan bagian yang berhasil diindikasi adanya tindak plagiat.

2.7. Perfomansi

Untuk menghitung perfomansi[8, 11] dari sistem yang dibangun, diketahui S sebagai kumpulan kasus plagiat yang didapat dari data yang ada pada *datasets*.

Dan R yang merupakan kasus plagiat yang didapatkan dari sistem. Kedua set S dan R akan dicari irisan tiap anggotanya, dengan $S \cdot R = \{s \cap r | s \in S, r \in R\}$. Apabila terdapat irisan berarti sistem berhasil mendeteksi R sesuai dengan *datasets* S.

Didefinisikan *perimeter* yang digunakan untuk menghitung luas bagian *passage reference* sebagai $\pi(r) = 2(b-a) + 2(d-c);$ dimana $r = [x_a, x_b]x[y_c, y_d]$

2.7.1 Precision

Precision merupakan nilai yang menunjukan tingkat kesesuaian atau relevansi prediksi sistem yang mengacu pada *datasets*. Untuk menghitung nilai *precision* dari pasangan dokumen yang sudah diolah menggunakan Persamaan 2.8

$$prec(S, R) \;=\; \frac{\pi(S \cdot R)}{\pi(R)} \tag{2.8}$$

2.7.2 Recall

Recall merupakan jumlah seluruh data relevan/sesuai yang berhasil diprediksi oleh sistem. Untuk menghitung nilai *recall* menggunakan Persamaan 2.9

$$rec(S, R) \;=\; \frac{\pi(S \cdot R)}{\pi(S)} \tag{2.9}$$

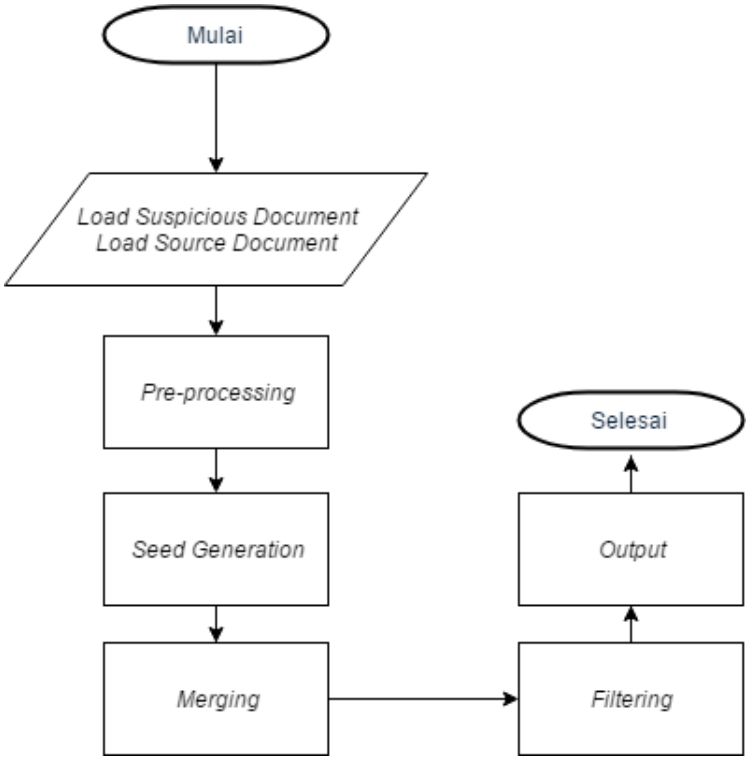
$+ F_1 Score$ merupakan nilai akurasi sistem dalam kemampuannya untuk menggolongkan *passage* ke dalam kelas plagiat. Untuk menghitung nilai F_1 digunakan Persamaan 2.10

$$F_1(S, R) = 2 \cdot \frac{prec(S, R) \cdot rec(S, R)}{prec(S, R) + rec(S, R)} \tag{2.10}$$

Bab 3. Rancangan Sistem

3.1. Gambaran Umum Sistem

Tujuan dari sistem yang dibangun adalah untuk mencari tindak plagiat yang ada pada dua buah dokumen dengan metode *Merging Context Seeds*. Alur secara umum yang ada pada sistem ditunjukan pada Gambar 3.1 dimana titik awal sistem bekerja berdasarkan *pairs* dan hasil akhir adalah bagian pada teks yang melakukan plagiat.



Gambar 3.1: Gambaran Umum Sistem

Load document
Dari satu buah *pair*, akan dimuat 2 buah dokumen yaitu *suspicious-document* (Dokumen X) dan *source-document* (Dokumen Y) yang merupakan *plain text* yang berisikan sebuah artikel.

Untuk *pre-processing, seed generation, merging, filtering,* dan *output* dijelaskan pada Sub-Bab 3.3

3.2. Analisis Kebutuhan

Analisis kebutuhan mencakup dari penentuan spesifikasi perangkat yang akan digunakan, baik perangkat lunak maupun perangkat keras.

3.2.1 Kebutuhan Fungsional

Kebutuhan fungsional sistem diantaranya adalah :

1. Sistem dapat membaca *input plain-text* yaitu *suspicious-document*, *source-document*, dan *pairs*.
2. Sistem mengolah data yang masuk ke dalam sistem dengan metode *Skipwrod-grams*, *Merging Context Seeds* dan *Single-linkage Clustering*.
3. Menampilkan *log*, dan hasil akhir berupa *highlight* pada *suspicious-document* dan *source-document* apabila ditemukan tindak plagiat.

3.2.2 Kebutuhan Non-Fungsional

Spesifikasi Perangkat Lunak

Perangkat lunak yang akan digunakan untuk tugas akhir ini adalah sebagai berikut :

1. Sitem Operasi : Windows 10 64-bit.
2. Bahasa Pemrograman : Python 2.7
   Daftar library yang diperlukan untuk menjalankan sistem yang dibangun ditunjukan pada Tabel 3.1.

Tabel 3.1: Daftar *Library* yang Digunakan

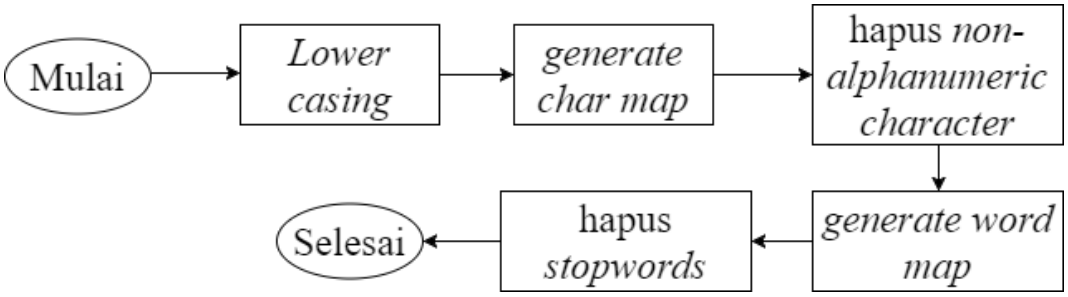
Library	Fungsi
nltk	Daftar *stopwords* dalam bahasa Inggris
xml.etree.ElementTree	Membaca informasi xml dari *datasets*
pyexcel_xls	Menyimpan hasil pengujian ke dalam file Excel

3. *Tools :* Sublime Text 3, Git Bash.

3.3. Perancangan Sistem

3.3.1 *Preprocessing*

Gambar 3.2 menunjukan rincian tahapan untuk proses *preprocessing* yang merupakan bagian dari tahapan *merging context seeds* pada Gambar 3.1.



Gambar 3.2: Alur *Preprocessing*

Diketahui dokumen X(*suspicious-document*) dan dokumen Y (*source-document*) yang dijadikan *input* untuk sistem. *Preprocessing* dilakukan terhadap kedua dokumen. Sebagai contoh, proses yang ditunjukan pada Sub-Bab ini merupakan proses untuk dokumen *suspicious-document00044.txt*

Lower casing* dan *Generate Character Map

Preprocessing dimulai dengan *Lower-casing*, atau mengubah seluruh alfabet yang ada pada dokumen menjadi huruf kecil. Kemudian memetakan seluruh karakter yang ada pada dokumen ke dalam *char map* untuk mengetahui letak kemunculan tiap karakter pada dokumen.

```
A GED is accepted by all public and most private colleges
and universities, as well as most employers.  GMAT is the
most effective test available for admission to business scho-
ols.emand for qualified professionals in education and re-
search industry is increasing.  Prepare for the GRE and get
going.  Getting admission in a law school is now easy wi-
th the LSAT Preparation course which thoroughly teaches you
the techniques and helps to build skills to take the LSAT
test.  The cost of the course is $125 and is open to the
first 40 individuals who meet the following qualifications:
.......
```

Akan membangun *char map* seperti pada Tabel 3.2.

Tabel 3.2: Contoh *Char Map*

index	karakter
0	a
1	
2	g
3	e
4	d
5	
6	i
7	s
8	
....	....
12160	
12161	g
12162	e
12163	t

Hapus karakter *non-alphanumeric*

Langkah berikutnya menghapus seluruh baris yang memiliki karakter yang tidak termasuk ke dalam *alphanumeric* karakter. Karakter yang termasuk ke dalam karakter *non-alphanumeric* ditunjukan pada Tabel 3.3.

Tabel 3.3: Daftar Karakter *Non-alphanumeric*

‘	∼	!	@	#	$
%	ˆ	&	*	(	)
-	_	+	=	{	[
}	]	—	\	;	:
”	’	<	,	>	.
?	/				

Generate Word Map

Setelah menghilangkan seluruh karakter *non-alphanumeric* maka dibangun *word map* dari karakter yang tersisa yang ada pada *char map* dengan menyimpan informasi kemunculan awal dan akhir karakter pada kata tersebut. Tiap kata akan dibangun dengan pemisah berupa karakter spasi. Contoh dari tabel *word map* ditunjukan pada Tabel 3.4. Dengan banyak jumlah data : **129**

Tabel 3.4: Contoh *Word Map*

Indeks awal	Indeks akhir	Kata
0	0	a
2	4	ged
6	7	is
9	16	accepted
18	19	by
...	...	...

Hapus *stopwords*

Tahap akhir *preprocessing* adalah menghilangkan *stopwords*. *Stopwords* merupakan kata yang umum muncul, sehingga dianggap memberikan *noise* pada data. Daftar *stopwords* dapat dilihat pada Tabel 3.5

Tabel 3.5: Daftar *Stopwords*

i	herself	was	because	from	any	t
me	it	were	as	up	both	can
my	its	be	until	down	each	will
myself	itself	been	while	in	few	just
we	they	being	of	out	more	don
our	them	have	at	on	most	should
ours	their	has	by	off	other	now
ourselves	theirs	had	for	over	some	
you	themselves	having	with	under	such	
your	what	do	about	again	no	
yours	which	does	against	further	nor	
yourself	who	did	between	then	not	
yourselves	whom	doing	into	once	only	
he	this	a	through	here	own	
him	that	an	during	there	same	
his	these	the	before	when	so	
himself	those	and	after	where	than	
she	am	but	above	why	too	
her	is	if	below	how	very	
hers	are	or	to	all	s	

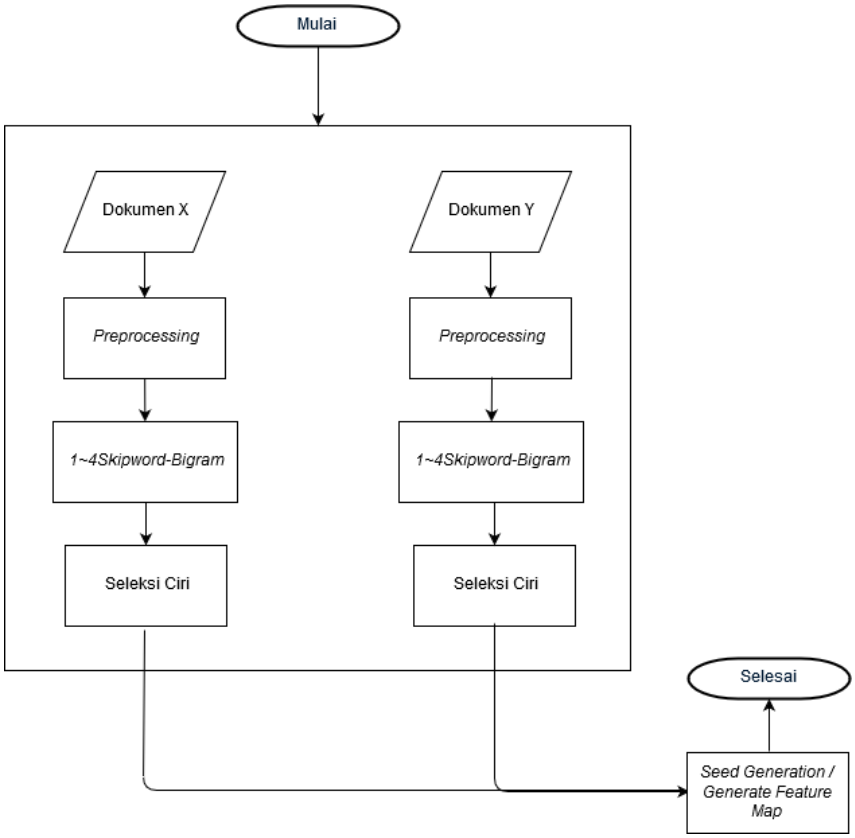
Sehingga pada akhir tahap *preprocessing*, *word map* yang dihasilkan ditunjukan pada Tabel 3.6. Dengan jumlah data : **84**

Tabel 3.6: Contoh Word Map yang Telah Dihilangkan *Stopwords*

Indeks awal	Indeks akhir	Kata
2	4	ged
9	16	accepted
25	30	public
41	47	private
49	56	colleges
...	...	...
12149	12152	what

3.3.2 *Seed Generation*

Gambar 3.3 menunjukan rincian tahapan untuk proses *seed generation* yang merupakan bagian dari tahapan *merging context seeds* pada Gambar 3.1. Seluruh proses yang ditampilkan merupakan proses yang dilakukan terhadap dokumen X *(suspicious-document00017.txt)* dan dokumen Y *(source-document00135.txt)*.



Gambar 3.3: Alur *Seeds Generation*

Ekstraksi Ciri

Pada tahap ini dokumen X(*suspicious*) dan dokumen Y(*source*)yang telah melalui tahap *preprocessing* akan melalui proses ekstraksi ciri dengan *skipword-bigram* 1-4 dengan tetap menyimpan informasi letak kemunculan karakter pada kata/*token*. Dari proses ekstraksi ini akan dibangun *feature map* yang ditunjukan oleh Tabel 3.7 seperti yang dijelaskan pada Persamaan 2.2.

Tabel 3.7: *Feature Map* Dokumen X

i awal	i akhir	Kata	f1	f2	f3	f4
2	4	ged	*_ged	*_ged	*_ged	*_ged
9	16	accepted	ged_accepted	*_accepted	*_accepted	*_accepted
25	30	public	accepted_public	ged_public	*_public	*_public
41	47	private	all_private	public_private	ged_private	*_private
49	56	colleges	private_colleges	public_colleges	accepted_colleges	ged_colleges

Seleksi Fitur/Ciri

Pada metode *Merging Context Seeds*, fitur yang mempunyai jumlah kemunculan yang tinggi dianggap tidak relevan. Sehingga apabila ada fitur yang kemunculannya melebihi **4**, fitur akan dihapus. Nilai 4 merupakan parameter yang sudah diuji[8] untuk mendapatkan fitur yang optimal.

Sebagai contoh : ***jika*** *fitur* ***patch_alabama*** *pada dokumen X muncul sebanyak* ***6****. Maka fitur tersebut akan dihapus seluruhnya dari feature map, baik fitur tersebut muncul di f1,f2,f3, atau f4.*

Seed Generation

Pada tahap ini fitur yang ada pada dokumen X dan dokumen Y. Akan dicari irisan antara kedua dokumen tersebut dari fitur yang ada. Apabila terdapat fitur yang sama, maka akan dibangun *seed set* yang merupakan titik awal pendeteksian *passage* yang diplagiat pada kedua dokumen, atau disebut dengan *passage reference*. Tabel 3.8 menunjukan hasil *seed generation* antara dokumen X *(suspicious-document00017.txt)* dan dokumen Y *(source-document00135.txt)*.

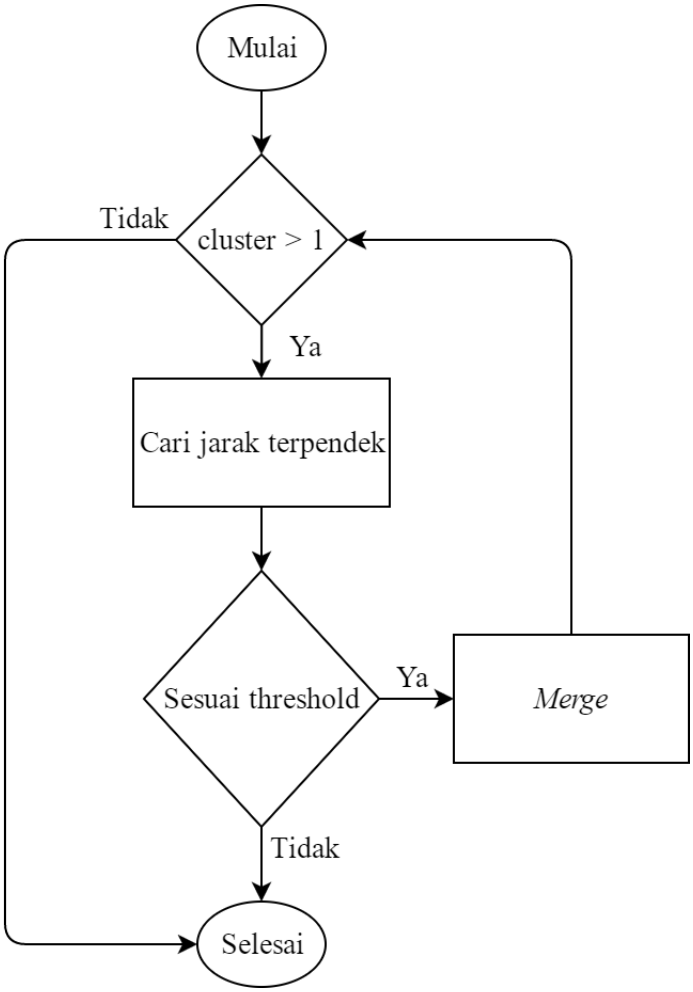
Tabel 3.8: *Passage Reference* Dokumen X dan Dokumen Y

pr_x	f	Kata	$Y_{f_{start}}$	$Y_{f_{end}}$	$X_{f_{start}}$	$X_{f_{start}}$
pr_0	gmat_test	test	1181	1199	130	133
pr_1	gmat_questions	questions	1191	1199	1043	1051
pr_2	gmat_questions	questions	1191	1199	1103	1111
pr_3	gmat_questions	questions	1191	1199	1308	1316
...	...	...	...	...	...	...
pr_{461}	takers_right	right	1300	1304	9868	9872

Kolom $Y_{f_{start}}$ menunjukan kemunculan awal kata pada *feature map* pada dokumen Y, sedangkan $Y_{f_{end}}$ menunjukan akhir kemunculan kata pada *feature map* pada dokumen Y. Sedangkan $X_{f_{start}}$ dan $X_{f_{end}}$ menunjukan kemunculan pada dokumen X. Seluruh letak kemunculan pada Tabel 3.8 didapatkan dari proses ekstraksi ciri.

3.3.3 Merging

Gambar 3.4 menunjukan rincian tahapan untuk proses *merging* yang merupakan bagian dari tahapan *merging context seeds* pada Gambar 3.1.



Gambar 3.4: Alur *Merging*

Proses *merging* merupakan proses dimana sistem akan meng-*cluster* setiap *passage reference* menggunakan algoritma *single linkage clustering*. Kriteria *merge* yang digunakan adalah jarak terpendek antara 2 *passage reference* dari kumpulan *passage reference* yang ada. Proses *merging* akan terhenti apabila memenuhi salah satu dari dua kondisi terminasi berikut :

1. Jarak terpendek $\leq \tau; \tau = 7$.
   Nilai 7 dianggap nilai yang paling optimal dari pengujian sementara dengan range nilai 5 - 15.

2. Jumlah cluster = 1.
   Yang berarti seluruh *seed* yang didapat sudah masuk kedalam satu *cluster*.

3. Jumlah cluster = 0.
   Tidak terdapat *seed* pada proses *seed generation* sebelumnya.

Tabel 3.9 menunjukan jarak antar *seed* atau *passage reference* yang didapat dari proses *seed generation*.

Tabel 3.9: Jarak Antar *Passage Reference*

Dist	pr_0	pr_1	pr_2	pr_3	pr_4	...	pr_{15}	...
pr_0	x	41.6	44.2	53.3	178.2	...	...	...
pr_1	41.6	x	1.8	8.1	90.2	...	...	...
pr_2	44.2	1.8	x	6.3	88.3	...	...	...
pr_3	53.3	8.1	6.3	x	81.6	...	...	...
pr_4	178.2	90.2	88.3	81.6	x	...	...	...
...	...	...	...	...	...	...	...	...
pr_{13}	...	...	...	...	...	...	0.0	...
...	...	...	...	...	...	...	...	...

Dari hasil *loop* pertama diketahui jarak antara *passage reference* terdekat ada di pr_{13} dan pr_{15} dengan jarak 0.0, maka pr_{13} dan pr_{15} akan di *merge*. Data kandidat yang ditunjukan pada Tabel 3.10. Nilai kemunculan pada dokumen X dan dokumen Y didapatkan dari proses ekstraksi ciri pada tahap sebelumnya.

Tabel 3.10: Kandidat *Passage Reference* yang Akan Di-*merge*

pr_x	Kata	$Y_{f_{start}}$	$Y_{f_{end}}$	$X_{f_{start}}$	$X_{f_{end}}$
pr_{13}	reasoning	1861	1869	3874	3882
pr_{15}	reasoning	1861	1869	3874	3882

Pada Tabel 3.11 terlihat bahwa nilai $X_{f_{start}}$ dan $X_{f_{end}}$ diambil dari titik terendah untuk $X_{f_{start}}$, dan titik terjauh untuk $X_{f_{start}}$ dikarenakan penggabungan dua buah *passage reference*. Dikarenakan *passage reference* yang akan digabung merupakan fitur yang ada pada kata yang sama, sehingga tidak ada perubahan untuk nilai kemunculannya

Tabel 3.11: Hasil Merge

pr_x	Kata	$Y_{f_{start}}$	$Y_{f_{end}}$	$X_{f_{start}}$	$X_{f_{end}}$
$pr_{13,15}$	reasoning, reasoning	1861	1869	3874	3882

Proses akan diulang hingga tidak ada jarak antar *passsage reference* yang melebihi τ. Tabel 3.9 menunjukan jarak terpendek antar *passage reference* pada *looping* pertama.

3.3.4 *Filtering*

Pada tahap ini *cluster* akhir akan dipilihi, apabila *cluster* mempunyai jumlah anggota atau *passage reference* $\geq \tau; \tau = 15$[8] maka $pr_{n..m}$ dianggap menjadi bagian yang diplagiat (r).

3.3.5 *Output*

Dari tahap *filtering* didapat lokasi tindak plagiat, pada tahap *output* ini mengembalikan bagian dari kedua dokumen.

Tabel 3.12: $r \in R$

r	$Y_{f_{start}}$	$Y_{f_{end}}$	$X_{f_{start}}$	$X_{f_{end}}$
r_1	8	979	3805	4427
r_2	1160	1620	9073	9461
r_3	1719	2134	8269	8580

Tabel 3.12 menunjukan $r \in R$ pada *pair suspicious-document00044.txt-source-document01326.txt* yang didapat dari hasil akhir *merging* yang didapat dari tahap sebelumnya. Kemudian dibangkitkan nilai kemunculannya mejadi teks pada dokumen sesungguhnya, sehingga hasil yang dimunculkan ditunjukan seperti pada Tabel 3.13.

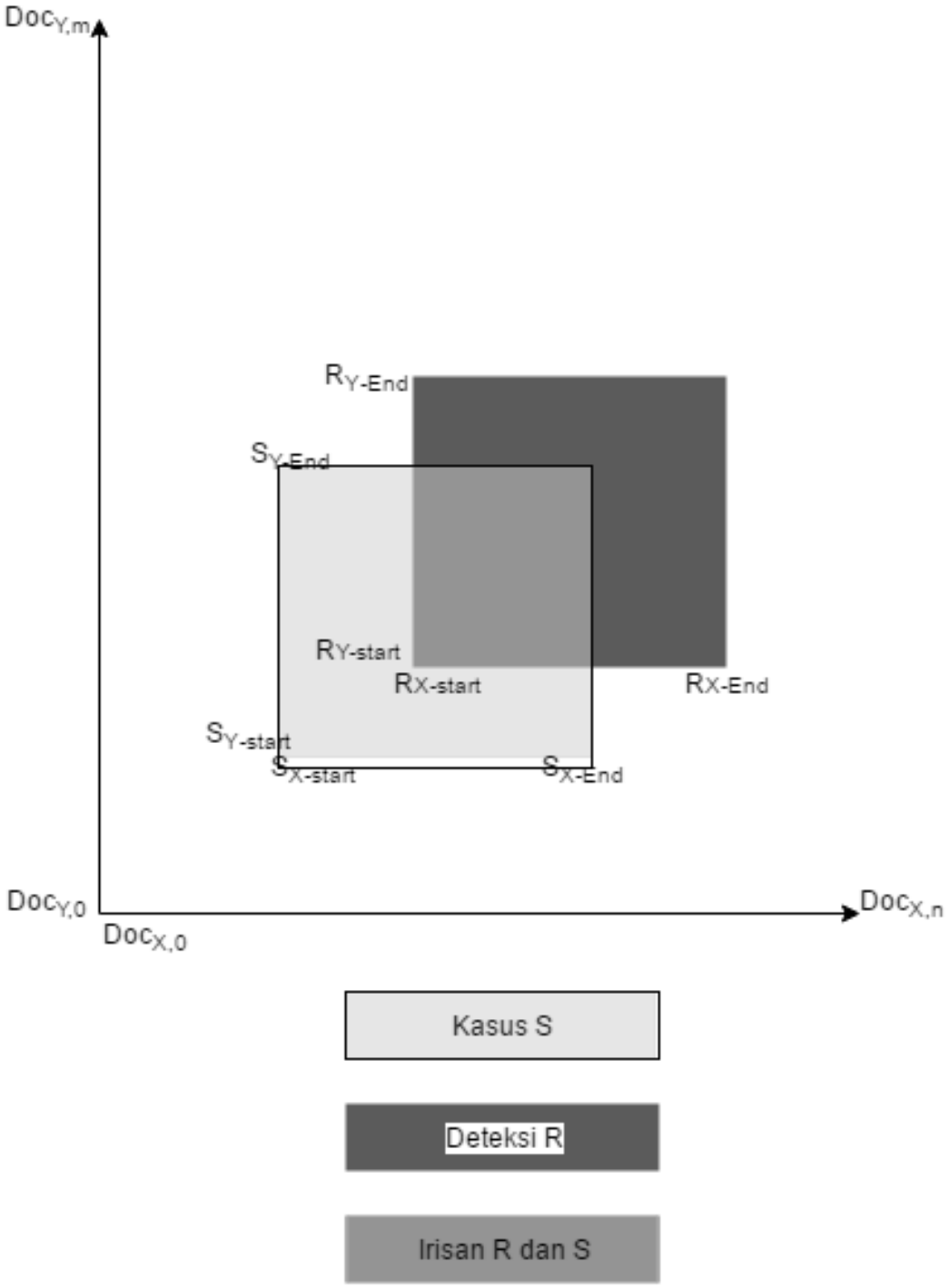
Tabel 3.13: *Output* akhir berupa teks yang diplagiat

r	Teks Pada Dokumen X	Teks Pada Dokumen Y
r_1	questions of three question types – reading comprehension, critical reasoning, and sentence correction. you are allowed 75 minutes to complete this entire section. verbal section each passage engages with a specialized topic or opinion in either the humanities, social sciences, science, or business, but no specific outside knowledge of the material is required; all questions refer to what is stated or implied in the text. the directions for these questions look like this: each passage is followed by questions about its content. after reading a passage, select the best answer to each question among the five choice	comprehension this is a partial free sample of our prep guide. to view the remainder of this page, purchase the . typically, your verbal test will include 4 reading comprehension passages, with 3 to 4 questions per passage, for a total of 12 to 14 questions of the 41 verbal questions. each passage engages with a specialized topic or opinion in either the humanities, social sciences, science, or business, but no specific outside knowledge of the material is required; all questions refer to what is stated or implied in the text. the directions for these questions look like this: each passage is followed by questions about its content. after reading a passage, select the best answer to each question among the five choices. answer all questions following a passage on the basis of what the passage states or implies.directions: a passage and a corresponding question look like this: the screen will split into two with the passage on the left and the question
r_2	computer-adaptive test) so the questions will begin at an intermediate skill level and . in general, average test takers will get about 50% right of the questions right. as result, higher scorers are effectively taking a completely different test from lower scorers and their strategies will be adjusted accordingly. higher scorers will get longer and more challenging essays and question	computer-adaptive test) so the questions will begin at an intermediate skill level and . in general, average test takers will get about 50% right of the questions right. as result, higher scorers are effectively taking a completely different test from lower scorers and their strategies will be adjusted accordingly. higher scorers will get longer and more challenging essays and questions. this chapter has sections specifically designed to help higher scorer

r_3	adapt to your performance by changing in difficulty if you are extremely good at sentence correction and weak at reading comp and critical reasoning.... guess what? your skill in sentence correction will make the gmat deliver you very hard reading comp and critical reasoning questions. the moral of the stor	adapt to your performance by changing in difficulty if you are extremely good at sentence correction and weak at reading comp and critical reasoning.... guess what? your skill in sentence correction will make the gmat deliver you very hard reading comp and critical reasoning questions. the moral of the story.... be balanced on verbal and skilled at all three question types.how the cat impacts verbal difficult

3.3.6 Evaluasi

Evaluasi dilakukan untuk mengukur perfomansi sistem yang dibangun. Evaluasi pada tahap ini dilakukan pada level *pair*. Cara untuk mengukur perfomansi adalah dengan membandingkan $r \in R$ keluaran sistem, dengan $s \in S$ dari *dataset*. Tiap elemen R akan dicari irisan untuk tiap elemen S untuk mendapatkan nilai ***True Positive***. Sedangkan elemen R dianggap ***Test Outcome Positive*** dan elemen S sebagai ***Condition Positive***. Gambar 3.5 menunjukan bagaimana perhitungan untuk nilai ***True Positive, Test Outcome Positive, Condition Positive***. Dimana pada Gambar 3.5 menunjukan area plagiat yang ada pada *dataset* dan area plagiat yang dideteksi oleh sistem.



Gambar 3.5: Perhitungan Nilai Perfomansi

Tabel 3.14 merupakan *set case* $s \in S$ atau bagian yang dinyatakan plagiat untuk *pair suspicious-document00044.txt-source-document01326.txt* yang diapat dari *dataset*.

Tabel 3.14: $s \in S$

s	$Y_{f_{start}}$	$Y_{f_{end}}$	$X_{f_{start}}$	$X_{f_{end}}$
s_1	298	741	3975	4418
s_2	1143	1548	9028	9433
s_3	1713	2032	8240	8559

Sedangkan Tabel 3.15 menunjukan jumlah luas irisan area plagiat atau *passage* dari $r \in R$ dan $s \in S$.

Tabel 3.15: $R \cap S$

$\pi(r_x \cap s_y)$	r_1	r_2	r_3
s_1	1772	0	0
s_2	0	1496	0
s_3	0	0	1204

Sehingga perhitungan perfomansi untuk *pair suspicious-document00044.txt-source-document01326.txt* ditunjukan pada Tabel 3.16.

Tabel 3.16: Contoh Perhitungan Perfomansi pada Level 1 Dokumen

	True Positive	4474
Condition Positive	True Positive + False Negative	4468
Test Outcome Positive	True Positive + False Positive	6334
Precision	True Positive / Prediction Positive	0.706
Recall	True Positive / Condition Positive	0.958
F_1	$2 \cdot \frac{prec(S,R) \cdot rec(S,R)}{prec(S,R) + rec(S,R)}$	0.813

Dari Tabel 3.16 didapat informasi bahwa *test accuracy* atau nilai F_1 untuk *pair suspicious-document00044.txt-source-document01326.txt* adalah 0.813.

Bab 4. Pengujian dan Analisis

4.1. Tujuan Pengujian

Tujuan pengujian pada sistem yang dibangun adalah untuk mendapatkan nilai perfomansi berupa *precision*, *recall*, dan F_1 untuk keseluruhan sistem terhadap *datasets* yang ada. *Datasets* yang digunakan merupakan dataset yang sudah dijelaskan pada Bab 2, dan jumlah dataset yang digunakan pertiap kelas plagiat adalah kurang lebih sebanyak 900 data. Sehingga secara keseluruhan *datasets* yang digunakan adalah sebanyak 1500 buah pasangan dokumen *suspicious* dan *source*.

4.2. Pengujian

Pengujian dilakukan pada *level* karakter dan dokumen dengan menggunakan parameter berikut dimana pemilihan nilainya sudah dijelaskan pada Bab 3 :

1. *Relevance Threshold* : $\varrho \leq 4$
   Relevance Threshold yang merupakan batas jumlah kardinalitas suatu fitur dianggap relevan.

2. *Distance* : $dist(P_1 \times Q_1, P_2 \times Q_2) \leq 7$
   Jarak maksimal yang digunakan untuk *merge* 2 buah *passage reference*, dan juga batas terminasi *clustering*.

3. *Plagiarism Case* : $\tau \geq 15$
   Jumlah minimal *passage reference* pada suatu *cluster* untuk dianggap sebagai bagian yang di plagiat.

Hasil pengujian dibagi menjadi 2 jenis, yaitu *No Plagiarism* dan *Plagiarism*. Dimana pengujian *No Plagiarism* terdiri dari *datasets No Plagiarism*, sedangkan *Plagiarism* terdiri dari *No Obfuscation, Random Obfuscation, Translation Obfuscation*, dan *Summary Obfuscation*. Pembagian ini dilakukan karena untuk kasus *No Plagiarism* ada kemungkinan pembagian dengan 0. Sehingga perhitungan dilakukan dengan cara melihat apakah pada ada bagian yang terindikasi plagiat. Bila ada, maka sistem dianggap gagal mengenali dokumen.

4.3. Pembanding

Tabel 4.1 menunjukan perbandingan hasil perhitungan nilai *precision* terhadap penilitian yang sama berdasarkan Overview of the 5th International Competition on Plagiarism Detection[12].

Tabel 4.1: Perbandingan Nilai *Precision* Terhadap Penelitian Lain

No.	Tim	Tahun	*None*	*Random*	*Translation*	*Summary*	Keseluruhan
1	Nourian	2013	0.9292	0.9627	0.9586	0.9997	0.9471
2	Jayapal	2012	0.9854	0.9598	0.8959	0.8326	0.9451
3	R.Torrejón	2013	0.9006	0.9100	0.8951	0.9075	0.8948
3	Oberreuter	2012	0.8904	0.8792	0.9033	0.9898	0.8944
4	Gillam	2012	0.8813	0.9557	0.9727	0.9959	0.8853
5	Gillam	2013	0.8809	0.9597	0.9727	0.9959	0.8849
6	Jayapal	2013	0.9199	0.9231	0.8565	0.6883	0.8790
7	Shrestha	2013	0.8093	0.9234	0.8801	0.9046	0.8746
8	Kueppers	2012	0.8326	0.8989	0.8999	0.8624	0.8692
9	Saremi	2013	0.8268	0.9181	0.8482	0.9460	0.8651
10	Kong	2012	0.8079	0.8937	0.8542	0.9640	0.8530
11	Suchomel	2012	0.8168	0.8758	0.8515	0.8748	0.8444
12	Kong	2013	0.7608	0.8622	0.8574	0.9638	0.8286
13	R.Torrejón	2012	0.8131	0.8388	0.8116	0.9267	0.8254
14	Palkovskii	2012	0.7922	0.8484	0.8322	0.9474	0.8237
15	Palkovskii	2013	0.7997	0.9314	0.8221	0.6760	0.8170
16	Suchomel	2013	0.6932	0.8297	0.6849	0.6709	0.7251
17	**Penelitian Ini**	**-**	**0.871**	**0.619**	**0.656**	**0.107**	**0.577**
18	Sánchez-Vega	2012	0.4034	0.4952	0.3730	0.4518	0.3986

Sedangkan pada Tabel 4.2 merupakan nilai recall pada kasus yang sama pada.

Tabel 4.2: Perbandingan Nilai *Recall* Terhadap Penelitian Lain

	Tim	Tahun	*None*	*Random*	*Translation*	*Summary*	Keseluruhan
1	Kong	2012	0.9484	0.7790	0.8500	0.2989	0.8245
2	Kong	2013	0.9068	0.7868	0.8463	0.3002	0.8134
3	Saremi	2013	0.9542	0.6888	0.8047	0.1021	0.7712
4	Oberreuter	2012	0.9993	0.6532	0.7959	0.0708	0.7686
5	Suchomel	2013	0.9964	0.6889	0.6662	0.5630	0.7659
6	R.Torrejón	2013	0.9526	0.6337	0.8112	0.2159	0.7619
7	Palkovskii	2012	0.9938	0.7513	0.6667	0.1609	0.7618
8	R.Torrejón	2012	0.9641	0.6028	0.7909	0.2901	0.7532
9	Shrestha	2013	0.9990	0.7146	0.6362	0.0990	0.7381
10	Suchomel	2012	0.9984	0.5195	0.5011	0.3531	0.6467
11	Sánchez-Vega	2012	0.7445	0.4350	0.5813	0.2216	0.5623
12	Palkovskii	2013	0.8505	0.3642	0.4967	0.0808	0.5356
13	Kueppers	2012	0.8385	0.3687	0.4243	0.0927	0.5107
14	**Penelitian Ini**	-	**0.472**	**0.627**	**0.363**	**0.860**	**0.456**
15	Nourian	2013	0.8763	0.2361	0.2857	0.0762	0.4338
16	Jayapal	2013	0.8604	0.1818	0.1941	0.0724	0.3819
17	Baseline	-	0.9996	0.0418	0.0880	0.0364	0.3422
18	Gillam	2012	0.8718	0.0242	0.0061	0.0010	0.2699
19	Gillam	2013	0.8378	0.0214	0.0061	0.001	0.2699
20	Jayapal	2012	0.5188	0.1114	0.0919	0.0457	0.2228

4.4. Evaluasi Hasil

Tabel 4.3 merupakan hasil pengujian sistem terhadap data ***No Obfuscation, Random Obfuscation, Translation Obfuscation*** dan ***Summary Obfuscation*** pada sistem yang dibangun.

Tabel 4.3: Perfomansi Sistem Pada *Level* Karakter

Tipe Plagiat	Jumlah Data	*Precision*	*Recall*	*F1*
No Obfuscation	952	0.871	0.472	0.612
Random Obfuscation	998	0.619	0.627	0.623
Translation Obfuscation	992	0.656	0.363	0.468
Summary Obfuscation	1185	0.107	0.860	0.190

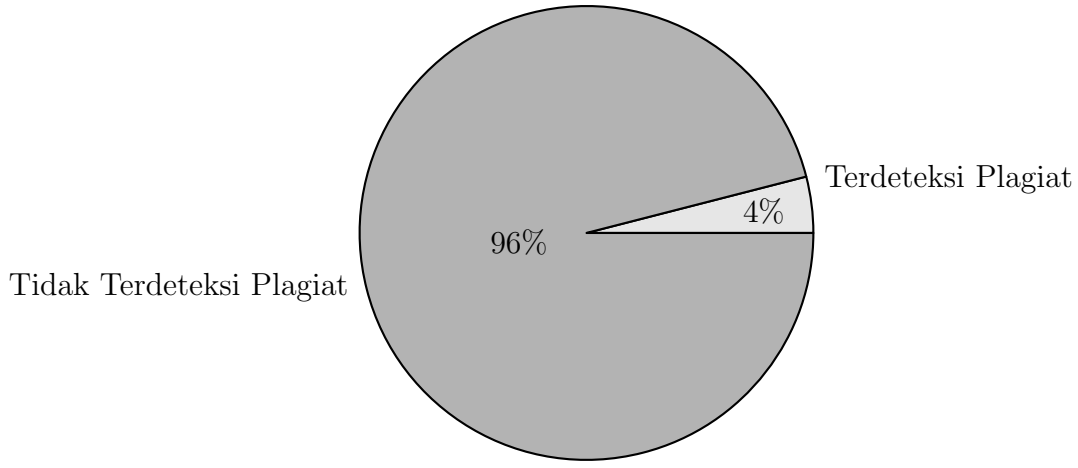
Sedangkan Tabel 4.4 merupakan hasil pengujian sistem terhadap data ***No Plagiarism***.

Tabel 4.4: Jumlah Deteksi pada *No Plagiarism*

Tipe Plagiat	Jumlah Data	Terdeteksi Plagiat	%
No Plagiat	1000	40	96%

4.4.1 No Plagiarism

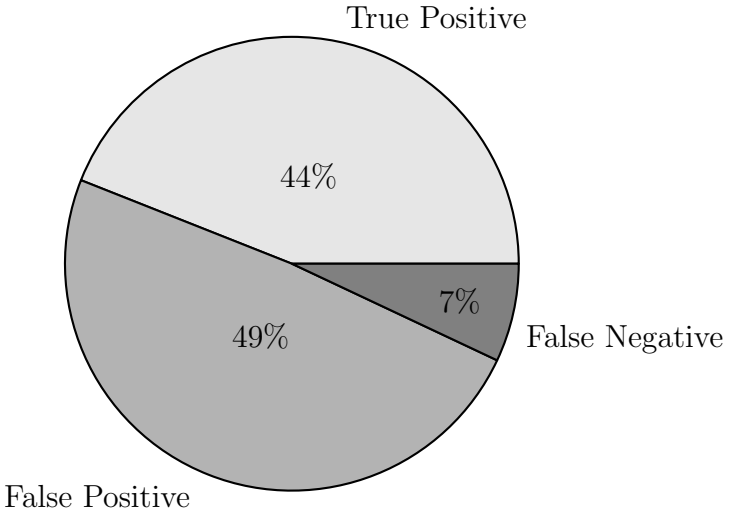
Pada Gambar 4.1 menunjukan persentase jumlah dokumen yang terdeteksi plagiat pada tipe plagiat *No Plagiarism*. Berdasarkan hasil yang didapat, sistem masih mendeteksi plagiat pada 40 dari 1000 dokumen yang ada pada *datasets* ***No Plagiarism***. Sehingga dapat dikatakan bahwa sistem masih teralu sensitif dalam mengkategorikan dokumen kedalam kategori plagiat.



Gambar 4.1: Persentase Nilai Perfomansi Tipe Plagiat ***No Plagiarism***

4.4.2 No Obfuscation

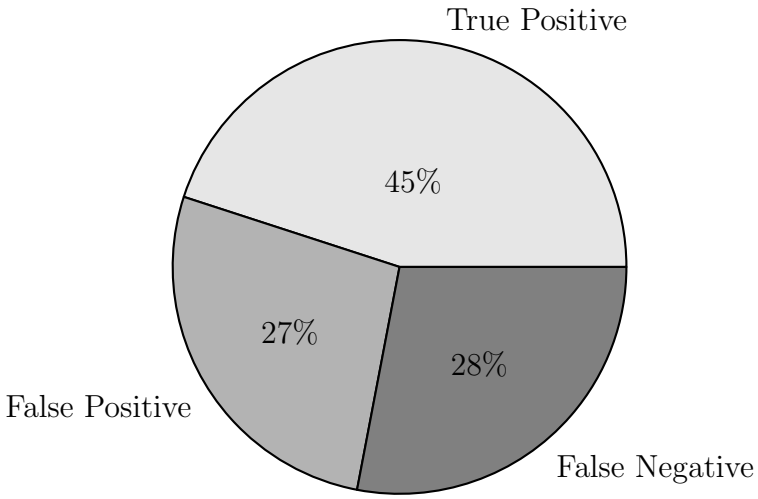
Pada Gambar 4.2 menunjukan persentase nilai perfomansi untuk tipe plagiat **No Obfuscation**. Dari hasil yang didapat, sistem mendeteksi bagian yang dianggap plagiat teralu luas/teralu banyak dari yang diharapkan, sehingga nilai *False Positive* yang didapat cukup tinggi, sebanyak 49%. Dimana bagian yang tidak plagiat dianggap plagiat oleh sistem. Nilai *False Positive* yang didapat bahkan melebihi nilai *True Positive* yaitu 44%. Sedangkan untuk bagian plagiat yang tidak terdeteksi oleh sistem *False Negative* dapat dikatakan cukup kecil, yaitu nilai 7%. Hingga hasil akhir yang didapat adalah nilai *precision* 0.871, *recall* 0.472 dan $_F1$ 0.612.



Gambar 4.2: Persentase Nilai Perfomansi Tipe Plagiat ***No Obfuscation***

4.4.3 Random Obfuscation

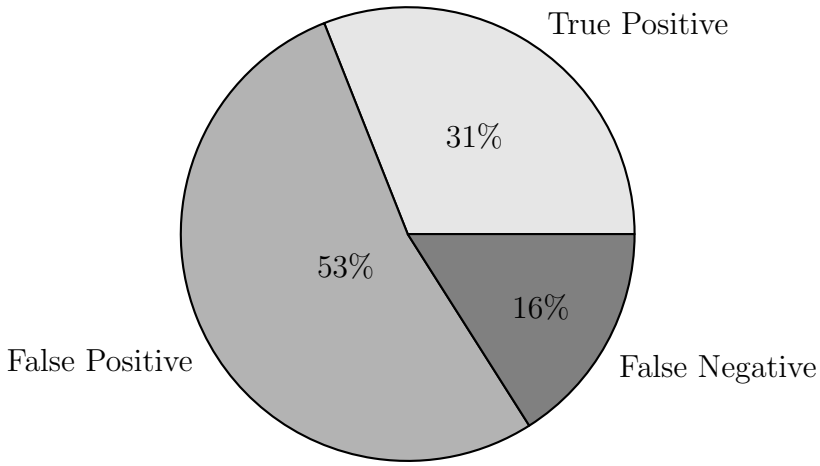
Sedangkan untuk *datasets Random Obfuscation*, seperti yang ditunjukan Gambar 4.3 penurunan perfomansi pada nilai *precision*. Penurunan nilai ini dikarenakan sistem tidak dapat mendeteksi bagian yang plagiat secara baik. Sehingga nilai *False Negative* yang didapat tinggi, yaitu 28%. Hingga hasil akhir yang didapat adalah nilai *precision* 0.619, *recall* 0.627 dan $_F1$ 0.623.



Gambar 4.3: Persentase Nilai Perfomansi Tipe Plagiat ***Random Obfuscation***

4.4.4 Translation Obfuscation

Untuk *datasets Translation Obfuscation* berdasarkan Gambar 4.4, menunjukan bahwa sistem yang dibangun masih mendeteksi bagian yang plagiat teralu sensitif, sehingga nilai *False Positive* yang didapat cukup tinggi yaitu 53% dan hanya mendapat nilai *True Positive* sebesar 31%. Hingga hasil akhir yang didapat adalah nilai *precision* 0.656, *recall* 0.363 dan $_F$1 0.468.

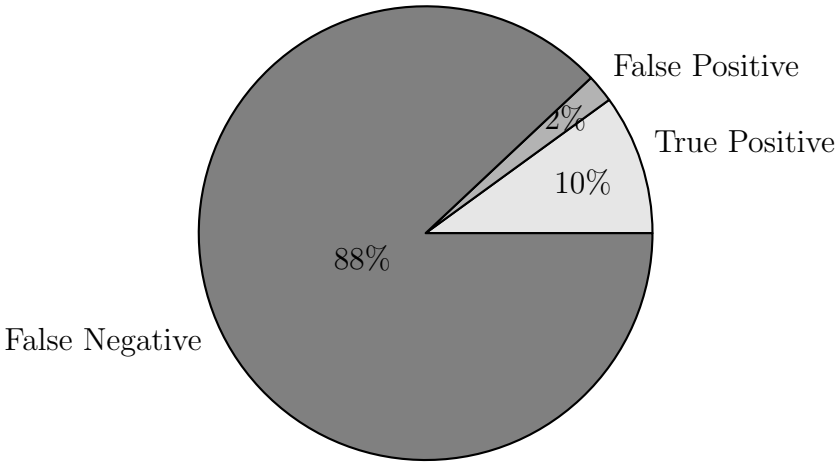


Gambar 4.4: Persentase Nilai Perfomansi Tipe Plagiat ***Translation Obfuscation***

4.4.5 Summary Obfuscation

Pada tipe plagiat *Summary Obfuscation* sistem dapat dikatakan tidak mampu mengatasi masalah yang ada. Hal ini dapat dilihat pada Gambar 4.5. Nilai *True Positive* yang didapat hanya 10%. Sedangkan nilai *False Negative* sebanyak 88%. Yang berarti sistem tidak mendeteksi hampir seluruh bagian yang

diplagiat. Hingga hasil akhir yang didapat adalah nilai *precision* 0.107, *recall* 0.860 dan $_F1$ 0.190.



Gambar 4.5: Persentase Nilai Perfomansi Tipe Plagiat ***Summary Obfuscation***

4.5. Analisis Hasil

Dari pengujian yang dilakukan untuk *datasets no obfuscation, random obfuscation, translation, dan summary obfuscation*, sistem yang dibangun masih mengalami masalah dalam proses *merging*. Hal ini dikarenakan oleh banyak bagian pada dokumen yang diolah yang tidak termasuk plagiat dideteksi sebagai plagiat. Hal ini disebabkan oleh beberapa hal. Seperti fitur yang dihasilkan teralu banyak dan saling berdekatan sehingga pada proses *merging* banyak fitur yang tidak diinginkan ikut terseret menjadi fitur yang dianggap plagiat. Jarak minimal yang teralu besar juga membuat sistem yang dibangun teralu sensitif, sehingga fitur yang di-*merge* semakin menumpuk. Hal ini dapat dilihat dari hasil pengujian dimana nilai *false positive* pada *datasets* diatas yang tinggi.

Untuk *datasets no plagiarism* sistem sudah mampu menangani tipe plagiat ini. Karena dokumen yang dideteksi plagiat hanya sedikit. Adanya dokumen yang dianggap plagiat diakibatkan karena masih teralu banyak fitur yang dihasilkan oleh sistem.

Sedangkan untuk *datasets summary obfuscation* fitur yang dihasilkan oleh sistem untuk *datasets* ini teralu sedikit, sehingga saat proses *merge* sistem tidak mampu fitur dari dokumen plagiat kedalam *cluster*. Sehingga nilai *false negative*, atau bagian plagiat yang dapat dideteksi oleh sistem kecil.

Bab 5. Kesimpulan

5.1. Kesimpulan

Sistem yang dibangun teralu sensitif dalam mengenali bagian yang ada pada dokumen untuk tipe plagiat tertentu. Hal ini terbukti dari nilai *False Positive* yang tinggi pada pengujian untuk tipe plagiat *No Obfuscation, Random Obfuscation, dan Translation Obfuscation*. Selain itu hal ini didapatkan karena pada tipe plagiat *No Plagiarism* sistem masih menemukan adanya tindak plagiat pada pasangan dokumen. Walaupun dari tipe plagiat *No Plagiarism* hanya 4%.

Nilai *False Positive* yang tinggi ini dikarenakan pada proses *merge*, *seed* yang ada pada bagian yang di plagiat dan bagian yang tidak di plagiat ikut tergabung. Sehingga banyak bagian yang tidak plagiat, dianggap sebagai plagiat. Hal ini juga dapat dikarenakan masih banyak fitur yang tidak relevan yang terbangun. Ataupun penggunaan parameter yang kurang cocok untuk seluruh dokumen yang di proses.

Pada tipe plagiat *Random Obfuscation, dan Translation Obfuscation* sistem juga tidak mampu menangani adanya perubahan pola kata pada bagian yang diplagiat sehingga teralu banyak bagian plagiat yang tidak terdeteksi oleh sistem.

Sedangkan pada tipe plagiat *Summary Obfuscation*, dimana tipe plagiat ini merangkum bagian pada dokumen *source*, sistem tidak dapat mengenali adanya tindak plagiat pada tipe plagiat ini secara baik.

5.2. Saran

Saran dari penulis untuk membangun sistem yang lebih baik dari penilitian ini diantaranya adalah :

1. Menambahkan atau menggunakan parameter yang lebih baik daripada yang penulis gunakan pada penilitian ini. Seperti pada prosses generasi fitur, agar fitur yang dihasilkan lebih relevan dibanding fitur yang dihasilkan sistem saat ini.
2. Menggunakan ekstraksi ciri yang berbeda agar mengurangi *seed* yang tidak relevan.

3. Menambahkan metode lain untuk membantu menangani tipe plagiat *Summary Obfuscation* dimana parafrase sulit ditemukan oleh sistem yang dibangun saat ini.

Daftar Pustaka

[1] Martin Potthast, Matthias Hagen, Anna Beyer, Matthias Busse, Martin Tippmann, Paolo Rosso, and Benno Stein1. *Overview of the 6th International Competition on Plagiarism Detection*. 2014.

[2] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. *An Introduction to Information Retrieval*. 2009.

[3] iThenticate: Professional Plagiarism Prevention. *2012 Survey Highlights: Scholarly Plagiarism*. 2012.

[4] PAN. Plagiarism detection, 2014. URL: `http://pan.webis.de/clef14/pan14-web/plagiarism-detection.html`.

[5] Purwani Istiana. *Panduan Anti Plagiarisme*. 2016.

[6] Van Rijsbergen. *Information Retrieval*. 2009.

[7] David Guthrie, Ben Allison, Wei Liu, Louise Guthrie, and Yorick Wilks. *A Closer Look at Skip-gram Modelling*. 2006.

[8] Philipp Gross and Pashutan Modaresi. *Plagiarism Alignment Detection by Merging Context*. 2014.

[9] Kilian Q. Weinberger and Lawrence K. Saul. *Distance Metric Learning for Large Margin Nearest Neighbor Classification*. 2009.

[10] Martin Potthast, Tim Gollub, Francisco Rangel, Paolo Rosso, Efstathios Stamatatos, and Benno Stein. *Improving the Reproducibility of PAN's Shared Tasks: Plagiarism Detection, Author Identification, and Author Profiling*. 2014.

[11] Martin Potthast, Matthias Hagen, Tim Gollub, Martin Tippmann, Johannes Kiesel, Paolo Rosso, Efstathios Stamatatos, and Benno Stein. *Information Access Evaluation. Multilinguality, Multimodality, and Visualization, Lecture Notes in Computer Science*. 2013.

[12] Martin Potthast, Matthias Hagen, Anna Beyer, Matthias Busse, Martin Tippmann, Paolo Rosso, and Benno Stein1. *Overview of the 5th International Competition on Plagiarism Detection*. 2014.

Lampiran

Tabel 5.1 hasil evaluasi level karakter pada seluruh tipe plagiat.

Tabel 5.1: Lampiran perfomansi - 1

Tipe	Evaluasi Level Karakter				Evaluasi Level Karakter		
	TP	FP	FN	TN	TP %	FP %	FN %
No Plagiarism	0	333488	0	27700777987	0	100	0
No Obfuscation	2232012	2498768	331266	83780690919	44.09308015	49.36280705	6.544112795
Random Obfuscation	1491462	887130	917400	70619313053	45.25077731	26.91541727	27.83380542
Translation Obfuscation	1722584	3018878	901592	1.0603E+11	30.52574014	53.49723749	15.97702237
Summary Obfuscation	243084	39698	2027578	32991195514	10.52147717	1.718260358	87.76026247
Jumlah	5689142	6777962	4177836	3.21122E+11			
Precision	0.577						
Recall	0.456						
F1-measure	0.509						

Tabel 5.2 perhitungan nilai *Percision, Recall,* dan *F1.*

Tabel 5.2: Lampiran perfomansi - 2

Tipe	Evaluasi Level Karakter			Rata-rata Perkelas			Max			Min			Jumlah Data	/
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1		
No Plagiarism	#DIV/0!	0	#DIV/0!	0	0	0	0	0	0	0	0	0	1000	1000
No Obfuscation	0.871	0.472	0.612	0.863	0.705	0.741	1.000	1.000	0.998	0	0	0	952	1000
Random Obfuscation	0.619	0.627	0.623	0.610	0.708	0.626	1.000	1.000	0.990	0	0	0	998	1000
Translation Obfuscation	0.656	0.363	0.468	0.676	0.677	0.634	1.000	1.000	0.998	0	0	0	992	1000
Summary Obfuscation	0.107	0.860	0.190	0.025	0.110	0.036	0.937	1.000	0.967	0	0	0	1185	1185
Rata-rata	0.563	0.580	0.473	0.543	0.550	0.510	0.984	1.000	0.988	0.000	0.000	0.000		

Tabel 5.3 jumlah deteksi plagiat pada tipe plagiat *No Plagiarism* dari 300 data.

Tabel 5.3: Lampiran Perfomansi - 3

No Plagiarism	Detected	%
1000	40	96