PENERAPAN ALGORITMA WINNOWING UNTUK MENDETEKSI KEMIRIPAN TEKS PADA TUGAS AKHIR MAHASISWA

Reynald Karisma Wibowo[1], Khafiizh Hastuti[2]
[1,2]Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
Jalan Imam Bonjol, 50131, telepon
E-mail : 111201206601@mhs.dinus.ac.id[1], afis@dsn.dinus.ac.id[2]

Abstrak

Tugas Akhir merupakan dokumen yang merepresentasikan penelitian dan riset yang dilakukan oleh mahasiswa jenjang Strata Satu. Untuk menghasilkan Tugas Akhir yang berkualitas di butuhkan penelitian yang kompeten, salah satu faktornya adalah originalitas. Kemampuan mahasiswa untuk menciptakan penelitian yang original menjadi faktor penting. Teks plagiasi adalah bentuk plagiasi yang paling sering dijumpai. Deteksi plagiarisme dibedakan menjadi dua berdasarkan tugasnya yaitu secara intrinsik dan ekstrinsik. Algoritma winnowing merupakan metode deteksi plagiarisme secara ekstrinsik. Metode ini merupakan ekstensi dari metode Rabin-Karp Fingerprint dengan menambahkan fitur window pada proses agar hasil deteki lebih optimal. Penelitian ini menerapkan algoritma winnowing untuk mendeteksi kemiripn teks pada dokumen tugas akhir mahasiswa Universitas Dian Nuswantoro.

Kata Kunci: *Tugas akhir, deteksi plagiasi, plagiarisme, winnowing, rolling hash, jaccard coefficient.*

Abstract

Final Project (TA) itself is a document that represents the research and study author. TA is a obligation for each student to do in order to get Bachelor degree. To produce a exemplary final project it`s need research competence, one of the factor is its originality. The ability of students to create original research is an important factor, with the internet searching for resources become more easy and the more act of plagiarism that could occurs.Plagiarism detection is a process of displaying the document, analyze its content, launching plagiarize parts, and bring the sources of the same document, if available. Detection of plagiarism itself is divided into two categoryof Intrinsic and Extrinsic plagiarism. Winnowing method is a method of plagiarism detection in extrinsic category. This method is an extension of the fingerprinting method by adding features in order to winnow the process to increase it`s accuration matching results. In this study the authors will apply the Winnowing method to create it`s plagiarism detection applications for student final project. This study is expected to be useful for future research institutions as well as against plagiarism and text mining.

Keywords: *Final project, plagiarism detection, plagiarism, winnowing, rolling hash, jaccard coefficient*

1. PENDAHULUAN

Plagiarisme adalah suatu tindakan mencuri dan publikasi dari penulis lain baik itu berupa bahasa, pikiran, gagasan, atau ekspresi dan merepresentasikan mereka sebagai salah satu karya asli sendiri [1]. Bentuk dan taksonomi dari plagiarisme sangat luas namun salah satu bentuk plagiarisme yang paling tinggi probabilitas kemunculanya ialah Teks plagiarisme. Teks plagiarisme mungkin salah satu bentuk tertua dari plagiarisme, yang, sampai hari ini, masih sulit untuk diidentifikasi dalam praktek. Karena perkembangan cepat teknologi informasi, informasi apapun dapat ditemukan dengan mudah melalui internet. Informasi ini banyak disalahgunakan oleh berbagai pihak yang menyebabkan plagiarisme.

Permasalahan yang terjadi bermacam-macam mulai dari hal yang sepele hingga permasalahan yang kompleks. Kebutuhan sistem komputerisasi untuk deteksi plagiarisme dikarenakan ketidakmampuan manusia untuk memproses dokumen skala besar dan untuk mengambil semua dibukti bagian plagiasi dan sumber-sumber asli. Dari berbagai cara-cara pencegahan plagiarisme salah satunya adalah penggunaan deteksi plagiarisme. Beberapa software untuk deteksi plagiarisme seperti Turnitin, WCopyFind yang digunakan oleh University of Virginia, Crosscheck yang ditemukan oleh Elsevier, Springer yang diciptakan oleh Massachusetts Institute of Technology pers, dan penerbit ternama lainnya, Docoloc yang terintegrasi dengan sistem manajemen konferensi EDAS [2]. Saat ini, banyak alat antiplagiarisme telah dibangun baik oleh individu maupun institusi atau kelompok dan dapat ditemukan di Internet. Kebanyakan dari mereka, mengatasi beberapa macam tekstual plagiarisme dengan modifikasi superficial demi menciptakan Deteksi Plagiarisme terbaik.

Berdasarkan pemaparan singkat tentang beberapa hal di atas yang menjadi landasan untuk dilakukannya penelitian ini dibuat suatu aplikasi deteksi plagiarisme untuk memproses file skala besar yaitu file tugas akhir mahasiswa Universitas Dian Nuswantoro, dalam hal ini yang akan menjadi objek penelitian adalah algoritma dan menggunakan data set tugas akhir mahasiswa Universitas Dian Nuswantoro Semarang. Aplikasi yang diharapkan dapat menampilkan hasil dari algoritma winnowing dan menghasilkan presentasi tingkat kesamaan dengan similarity metrik pada aplikasi berbasis web.

2. METODE PENELITIAN

2.1 N-Gram

Ngrams adalah rangkaian token dengan panjang n.. Dalam konteks komputasi linguistik, token ini dapat berupa kata-kata, meskipun mereka dapat berupa karakter atau himpunan bagian dari karakter. Nilai n hanya mengacu pada jumlah token. Metode n-grams ini digunakan untuk mengambil potongan-potongan karakter huruf sejumlah n dari sebuah kata yang secara kontinuitas dibaca dari teks sumber hingga akhir dari dokumen. Berikut ini adalah contoh n-grams dengan n=5:

“Teks: A do run run run, a do run run” dari teks tersebut dilakukan penghilangkan spasi menjadi.

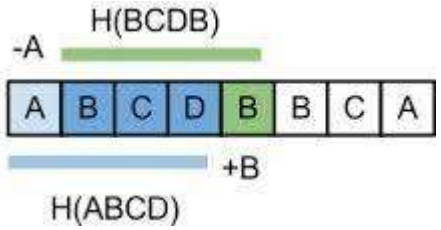
“Teks: adorunrunrunadorunrun” dari teks tersebut dihasilkan rangkaian 5-grams yang diturunkan menjadi “Teks: adoru dorun orunr runru unrun nrunr runru unrun nruna runad unado nador adoru dorun orunr runru unrun”.

2.2 Preprocessing

Preprocessing, proses yang sering digunakan dalam deteksi plagiasi dimana input file akan melalui berbagai macam proses manipulasi text. Macam-macam metode preprocessing meliputi *case folding, filtering, stemming,* dan *tokenizing*. *Case folding* adalah proses manipulasi case-sensitive, semua input teks data akan diubah menjadi huruf kecil/lower-case. Sedangkan filtering atau sering dikenal dengan istilah *stopword removal* adalah proses penghapusan kata yang tidak relevan dalam teks [3]. *Stemming* adalah pengubahan kata menjadi kata dasar, dan *tokenizing* adalah proses pengubahan kata string menjadi sekumpulan token.

2.3 Rolling Hash

Rolling hash merupakan salah satu metode hashing yang memberikan kemampuan untuk menghitung nilai hash tanpa mengulangi seluruh string. Berikut ini contoh string "abcdbbca" dan anda ingin menemukan pola "bcdb" dalam string ini.



Gambar 1. window rolling hash

2.4 Algoritma Winnowing

Winnowing adalah algoritma yang digunakan untuk melakukan proses pengecekan kesamaaan kata (dokumen fingerprinting) untuk mendeteksi plagiarisme [4]. Secara teknis *winnowing* adalah ekstensi dari implementasi algoritma *rabin-karp fingerprint* dengan penambahan metode *window*. Berikut penjelasan metode dari algoritma *winnowing.*

Preprocessing, proses awal dari metode dimana input file akan melalui berbagai macam proses manipulasi text. Macam-macam metode *preprocessing* meliputi *case folding, filtering, stemming, dan tokenizing.* Namun pada penelitian ini *preprocessing* yang digunakan *case folding, dan filtering. Case folding* adalah proses manipulasi case-sensitive. Sedangkan *filtering* atau sering dikenal dengan istilah *stopword removal* adalah proses penghapusan kata yang tidak relevan dalam teks. *Stemming* ialah proses pemisahan kata menjadi kata dasar. *Tokenizing* adalah proses pemisahan kata berdasarkan susunan kata [3]. Hasil dari proses ini akan menghasilkan dokumen teks yang relevan untuk diproses dan dicari kecocokanya. Pada preprocess file teks akan dibentuk menjadi rangkaian substring senilai *k* atau *k-gram.* Berikut contoh dari *preprocess.*

Terdapat sebuah string *S1* = "*--Reynald Karisma--".* *Preprocessing* akan melakukan *case folding* dan *filtering* sehingga *S1 = "reynaldkarisma".* Lalu string tersebut akan disubstring menjadi rangkaian string senilai *k-gram.* Pada contoh ini diberikan perumpamaan *k-gram = 4.* Dari *S1* akan menjadi rangkaian ["reyn", "eyna", "ynal", "nald", "aldk", "ldka", "dkar","kari","aris","rism","isma"].
Setelah rangkaian string dibentuk dari nilai rangkaian tersebut akan diproses menjadi rangkaian hash. *Hashing* adalah proses untuk mengubah karakter string menjadi bilangan integer yang disebut nilai hash. Proses pengubahan menjadi nilai hash menggunakan fungsi *rolling hash.* Berikut rumus *rolling hash.*

$$H(c_1 \dots c_k) = c_1 * b^{k-1} + c_2 * b^{k-2} * \dots + c_{k-1} * b + c_k$$

(1)

Pada *rolling hash* proses pengubahan nilai hash pada iterasi pertama

menggunakan rumus. $H(c_1 \dots c_n)$ merupakan nilai hash pertama, dimana *b* adalah konstan bilangan prima, *k* adalah nilai *kgram* dan c_n merupakan nilai ascii dari karakter. Pada perhitungan nilai hash kedua dan seterusnya tidak perlu dilakukan perhitungan lagi dari iterasi pertama namun dengan melakukan perhitungan rumus (2).

$$H(c_2 \dots c_{n+1}) = (H(c_1 \dots c_n) - c_1 * b^{(n-1)}) * b + c_{(n+1)} \quad (2)$$

Setelah proses rangkaian hash terbentuk dilanjutkan dengan proses *winnowing*. *Winnowing* ialah proses pembentukan *window* dari rangkaian hash. *Window* adalah proses pembentukan substring dari nilai hash sepanjang *wgram*. Dari proses *winnowing* akan menghasilkan fingerprint yang nanti akan digunakan untuk pencocokan plagiasi.

Keterangan	Teks1
Rangkaian Gram	rey eyn yna nal ald ldk dka kar ari ris ism sma
Nilai Hash	15026 13662 15948 14485 13025 14275 13374 14128 13096 15064 14079 15211
Window	{ 15026 13662 15948 14485 } { 13662 15948 14485 13025 } { 15948 14485 13025 14275 } { 14485 13025 14275 13374 } { 13025 14275 13374 14128 } { 14275 13374 14128 13096 } { 13374 14128 13096 15064 } { 14128 13096 15064 14079 } { 13096 15064 14079 15211 }
Fingerprint	[13662,1] [13025,4] [13096,8]

Gambar 2. Illustrasi Proses Winnowing

2.5 Jaccard Coefficient

Jaccard adalah nama yang sering digunakan untuk mengukur kemiripan, ketidakmiripan, dan jarak dari dataset. Mengukur *Jaccard similarity* coefficient antara dua dataset adalah hasil dari pembagian antara jumlah data yang sama dari kedua dataset dibagi dengan jumlah semua data pada dataset, seperti rumus berikut:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \quad (3)$$

2.6 Tahapan Penelitian

Data yang digunakan pada penelitian ini adalah data mahasiswa TA yang berasal dari program studi Teknik Informatika Universitas Dian Nuswantoro.

Dari data tersebut diambil atribut utama yang dibutuhkan pada penelitian ini, yaitu nama penulis, dan data teks tugas akhir bab 1 sampai bab 3. Atribut tersebut akan menjadi atribut acuan penting untuk pembandingkan tingkat plagiasi antara file TA satu dengan yang lainnya.

Pada penelitian ini penulis menggunakan metode *winnowing*. Metode ini menggunakan konsep *Rabin-Karp fingerprint* dengan penambahan konsep *window*. Dari *window* tersebut nilai hash akan disubstring dan dipilih nilai terendah.

Gambar 3. Blok diagram proses metode winnowing

Dokumen harus melalui proses yang

sama untuk mendapatkan nilai similarity antar 2 file dokumen (gambar 3). Metode ini bekerja secara sistematis, yang artinya pengerjaan dari tiap proses dilakukan secara berurutan atau linear.

a. Preprocessing

Pertama, sebelum file dokumen dicocokan, dokumen harus melalui langkah *preprocessing*. Langkah-langkah *preprocessing* meliputi *case folding, parsing, filtering, stemming, dan tokenizing*. Namun pada penelitian ini *preprocessing* yang digunakan *case folding, dan filtering. Case folding* adalah proses manipulasi case-sensitive. Pada penelitian ini, semua input teks data akan diubah menjadi huruf kecil/*lower-case*. Sedangkan *filtering* atau sering dikenal dengan istilah *stopword removal* adalah proses penghapusan kata yang tidak relevan dalam teks [3].

Tabel 1: Contoh preprocessing dengan string sample

Tipe Preprocessing	Hasil
string awal	@$%Reynald Karisma Wibowo&*
case folding	@$%reynald karisma wibowo&*
whitespace insensitivity	@#$%reynaldkarismawibowo&*
Filter	reynaldkarismawibowo

b. Pembentukan Rangkaian N-Gram / N-Gram Substring

Output dari *preprocessing* akan menghasilkan data teks murni yang akan digunakan pada proses *n-gram substring*. *N-gram* adalah substring penggabungan karakter sejumlah *k* pada teks dokumen. Tiap-tiap substring tersebut akan diproses dengan *rolling hash* dan akan menghasilkan rangkaian nilai hash.

Tabel 2: Contoh proses pembentukan rangkaian n-gram

Atribut	Nilai Array
Rangkaian string dengan kgram 3 terhadap string "reynald"	[0] => rey [1] => eyn [2] => yna [3] => nal [4] => ald

c. Penghitungan Nilai Hash / Rolling Hash

Proses transformasi dari rangkaian string menjadi rangkaian nilai hash menggunakan *rolling hash. Rolling hash* memungkinkan untuk menghitung nilai hash tanpa melakukan rehashing kembali dari iterasi pertama.

$$H(c_1 \dots c_k) = c_1 * b^{k-1} + c_2 * b^{k-2} * \dots + c_{k-1} * b + c_k \quad (4)$$

Untuk menghitung nilai hash pertama dilakukan perhitungan menggunakan rumus diatas. $H(c_1 \dots c_n)$ merupakan nilai hash pertama, dimana *b* adalah konstan bilangan prima, *k* adalah nilai *kgram* dan c_n merupakan nilai ascii dari karakter. Pada perhitungan nilai hash kedua dan seterusnya tidak perlu dilakukan perhitungan lagi dari iterasi pertama namun dengan melakukan perhitungan rumus dibawah.

$$H(c_2 \dots c_{n+1}) = \left(H(c_1 \dots c_n) - c_1 * b^{(n-1)}\right) * b + c_{(n+1)} \quad (5)$$

Berikut contoh dari perhitungan *rolling hash* terhadap substring "rey", dan "eyn" dengan *k-gram 3* dapat dilihat pada tabel 3.

Tabel 3. Tabel contoh perhitungan rolling hash

Atribut	Nilai Array

Perhitungan rolling hash iterasi pertama	[0] => rey r = 114, e = 101, y = 121, b = 7, prime = 100007 H=c_r*b^(k-1)+c_e*b^(k-2)+c_y*b^(k-3) H=114*7^2+101*7^1+121*7^0 H=5586+707+121 H=6414
Perhitungan rolling hash iterasi kedua dan seterusnya	[1] =>eyn e = 101, y = 121, n = 110, b = 7, prime = 100007 H(c_2…c_(n+1))=(H(c_1…c_n)-c_1*b^((n-1)))*b+c_((n+1)) =(6414-(7^2*114))*7^ +110 H=828 *7+110 H=5906

d. Pembentukan Window Dari Rangkaian Hash & Pemilihan Fingerprint

Proses winnowing membutuhkan parameter w-gram dimana nilai hash yang didapat akan disubstring menghasilkan rangkaian nilai hash sejumlah w. Dari rangkaian nilai hash inilah akan dipilih nilai hash terkecil, jika terdapat 2 atau lebih nilai yang sama, nilai hash terkecil yang paling kanan yang akan di pilih. Seperti pada gambar 4 adalah contoh dari rangkaian window dengan wgram = 4 terhadap nilai hash pada gambar 3.

```
77 74 42 17 98 50 17 98 8 88 67 39 77 74 42
17 98
```

Gambar 4. Contoh nilai hash

```
(77, 74, 42, 17)   (74, 42, 17, 98)
(42, 17, 98, 50)   (17, 98, 50, 17)
(98, 50, 17, 98)   (50, 17, 98,  8)
(17, 98,  8, 88)   (98,  8, 88, 67)
( 8, 88, 67, 39)   (88, 67, 39, 77)
(67, 39, 77, 74)   (39, 77, 74, 42)
(77, 74, 42, 17)   (74, 42, 17, 98)
```

Gambar 5. Contoh rangkaian window dari nilai hash

Pemilihan fingerprint diambil berdasarkan nilai terkecil dari nilai hash tiap window. Jika pada window terdapat dua nilai terkecil yang sama maka akan diambil nilai yang paling kanan dari window tersebut. Berikut adalah nilai fingerprint yang diambil dari window pada gambar 6.

```
17 17 8 39 17
```

Gambar 6. Contoh fingerprint yang dipilih

Nilai fingerprint yang telah dipilih tersebut akan digunakan untuk pencocokan terhadap fingerprint dari dokumen lain.

```
[17,3] [17,6] [8,8] [39,11] [17,15]
```

Gambar 7. Contoh fingerprint dengan informasi posisi nilai hash

e. Pencocokan Fingerprint Dengan Jaccard Coefficient

Nilai fingerprint dari proses winnowing tiap dokumen akan dicocokan menggunakan jaccard coefficient untuk mengukur prosentase kemiripan teks. Output dari proses ini adalah nilai persentasi kemiripan antara 2 dokumen.

$$similarity(A,B) = \frac{|A \cap B|}{|A \cup B|} x100\% \quad (6)$$

Penghitungan jaccard coefficient dilakukan berdasarkan rumus(4) berikut: $J(A,B)=|A \cap B||A \cup B|$, dimana J(A, B) adalah nilai kemiripan antara dataset A dan B, $A \cap B$ adalah irisan/data yang sama dari A dan B, dan $A \cup B$ adalah union/gabungan data dari A dan B. Dari hasil tersebut dikalikan 100 untuk menghasilkan nilai persentase. Berikut rumus dari jaccard coefficient

3. HASIL DAN PEMBAHASAN

Dari percobaan yang telah dilakukan data yang diperoleh meliputi data dari tabel pengujian. Data yang digunakan adalah file TA tugas akhir penulis yang akan dibandingkan dengan 10 file TA.

Hasil dari percobaan terhadap 10 file TA ditampilkan pada tabel 4.16, 4.17, 4.18, 4.19, 4.20, dan 4.21. Tiap tabel

menampilkan hasil uji 10 dataset dengan nilai n-gram yang berbeda. Berikut hasil dari pengujian terhadap 10 dataset file TA.

Tabel 4. Hasil tabel pengujian k-gram 6 & w-gram 6

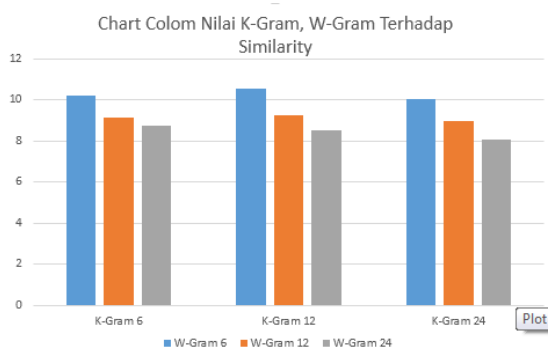
Id	*k-gram*	*w-gram*	Similarity (%)	Waktu Proses	
				Fingerprint	Pencocokan
1	6	6	13.89 %	24.67	43.27
2	6	6	12.33 %	24.67	27.88
3	6	6	6.86 %	24.67	4.6
4	6	6	14.59 %	24.67	57.3
5	6	6	11.32 %	24.67	18.8
6	6	6	13.45 %	24.67	42.95
7	6	6	13.82 %	24.67	50.56
8	6	6	12.13 %	24.67	22.78
9	6	6	12.94 %	24.67	34.18
10	6	6	12.65 %	24.67	29.83

Hasil pengujian pada tabel 4 menunjukan nilai rata-rata similarity dari pengujian 10 file. Dapat disimpulkan bahwa file TA yang diuji memiliki nilai tingkat plagiasi yang rendah dengan rata-rata 12.356 %. Dari percobaan diatas dapat diamati bahwa nilai tersebut merupakan pendeteksian umum kata-kata yang sering muncul pada Tugas Akhir. Meskipun sudah dilakukan *filterisasi* terhadap data teks hal ini masih dapat terjadi dikarenakan melakukan *preprocessing* file Tugas Akhir tidak mudah karena terdapat beberapa syntax format penulisan yang masih lolos dari *filter s*ehingga membuat objek yang dideteksi menjadi tidak relevan. Namun untuk mendeteksi plagiasi secara tekstual algoritma *winnowing* cukup efektif karena dapat mendeteksi plagiasi seperti copy paste dan relokasi kata.

Pada pengujian berikutnya dilakukan observasi lebih dalam terhadap k-gram dan w-gram. Pengujian dilakukan dengan menggunakan 2 file TA yang telah dipilih dan dilakukan proses dengan k-gram dan w-gram senilai 6, 12, dan 24. Berikut hasil pengujian tersebut pada tabel 4.2

Tabel 5. Tabel pengaruh k-gram dan w-gram terhadap similarity

No	*k-gram*	*w-gram*	Similarity (%)	Waktu Proses	
				Fingerprint File User	Fingerprint File Data
1	6	6	10.21 %	39.98	10.79
2	12	6	10.55 %	39.02	10.76
3	24	6	10.02 %	38.54	11.41
4	6	12	9.11 %	35.45	9.37
5	12	12	9.26 %	33.92	9.05
6	24	12	8.98 %	33.92	9.34
7	6	24	8.75 %	35	8.75
8	12	24	8.49 %	33.15	8.65
9	24	24	8.04 %	33.31	8.66

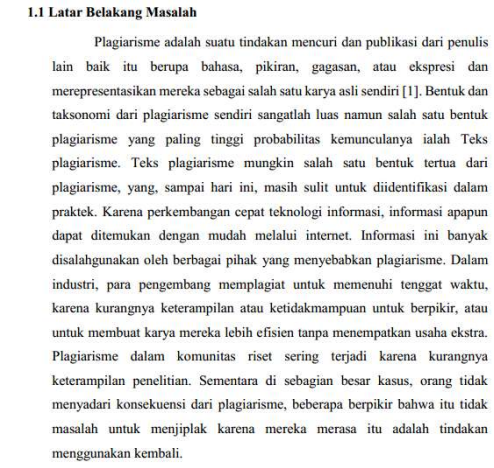


Gambar 8. Grafik pengaruh k-gram dan w-gram terhadap similarity

Tabel 5 menunjukkan bahwa nilai similarity rata-rata pada penggunaan w-gram 6 adalah 12.395%, rata-rata similarity pada penggunaan w-gram 12 adalah 9.11, dan rata-rata similarity pada penggunaan w-gram 24 adalah 8.42. Waktu proses hanya berbeda beberapa milisecond pada penggunaan *w-gram* yang sama, namun penggunaan *w-gram* yang lebih tinggi memiliki waktu proses yang lebih rendah. Hal ini dikarenakan proses pemilihan substring dan perhitungan hash lebih sedikit, tiap-tiap substring memuat konten karakter lebih banyak sehingga jumlah dari banyaknya substring akan berkurang.

Dari data tabel 5 dan grafik pada gambar 8 dapat disimpulkan bahwa semakin tinggi nilai *n-gram* maka akan semakin

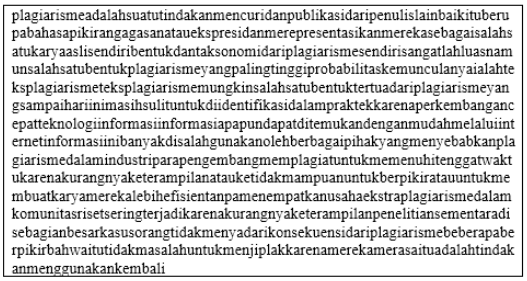
rendah nilai similarity yang dihasilkan dan semakin rendah nilai *n-gram* maka akan semakin tinggi nilai similarity. Namun bukan berarti nilai *n-gram* yang rendah akan memberikan nilai akurasi yang akurat. Semakin kecil nilai *n-gram* maka semakin kecil karakter yang akan dicocokan dan semakin sering karakter tersebut akan ditemukan dalam teks.
Analisis terhadap hasil *preprocessing* dilakukan pengamatan terhadap file sebelum di lakukan *preprocessing* dengan data teks setelah di lakukan *preprocessing.* Pada gambar 8 dapat kita lihat dan bandingkan dengan hasil *preprocessing* pada tabel 4.



1.1 Latar Belakang Masalah

Plagiarisme adalah suatu tindakan mencuri dan publikasi dari penulis lain baik itu berupa bahasa, pikiran, gagasan, atau ekspresi dan merepresentasikan mereka sebagai salah satu karya asli sendiri [1]. Bentuk dan taksonomi dari plagiarisme sendiri sangatlah luas namun salah satu bentuk plagiarisme yang paling tinggi probabilitas kemunculanya ialah Teks plagiarisme. Teks plagiarisme mungkin salah satu bentuk tertua dari plagiarisme, yang, sampai hari ini, masih sulit untuk diidentifikasi dalam praktek. Karena perkembangan cepat teknologi informasi, informasi apapun dapat ditemukan dengan mudah melalui internet. Informasi ini banyak disalahgunakan oleh berbagai pihak yang menyebabkan plagiarisme. Dalam industri, para pengembang memplagiat untuk memenuhi tenggat waktu, karena kurangnya keterampilan atau ketidakmampuan untuk berpikir, atau untuk membuat karya mereka lebih efisien tanpa menempatkan usaha ekstra. Plagiarisme dalam komunitas riset sering terjadi karena kurangnya keterampilan penelitian. Sementara di sebagian besar kasus, orang tidak menyadari konsekuensi dari plagiarisme, beberapa berpikir bahwa itu tidak masalah untuk menjiplak karena mereka merasa itu adalah tindakan menggunakan kembali.

Gambar 9. Contoh latar belakang file TA penulis yang diuji



plagiarismeadalahsuatutindakanmencuridanpublikasidaripenulislainbaikituberu pabahasapikirangagasanataueks presidanmerepresentasikanmerekasebagaisalahs atukaryaaslisendiribentukdantaksonomidariplagiarismesendirisangatlahluasnam unsalahsatubentukplagiarismeyangpalingtinggiprobabilitaskemunculanyaialahte ksplagiarismeteksplagiarismemungkinsalahsatubentuktertuadariplagiarismeyan gsampaihariinimasihsulituntukdiidentifikasidalampraktekkarenaperkembanganc epatteknologiinformasiinformasiapapundapatditemukandenganmudahmelaluiint ernetinformasiinibanyakdisalahgunakanolehberbagaipihakyangmenyebabkanpla giarismedalamindustriparapengembangmemplagiatuntukmemenuhitenggatwakt ukarenakurangnyaketerampilanatauketidakmampuanuntukberpikiratauuntukme mbuatkaryamerekalebihefisientanpamenempatkanusahaekstraplagiarismedalam komunitasrisetseringterjadikarenakurangnyaketerampilanpenelitiansementaradi sebagianbesarkasusorangtidakmenyadarikonsekuensidariplagiarismebeberapabe rpikirbahwaitutidakmasalahuntukmenjiplakkarenamerekamerasaituadalahtindak anmenggunakankembali

Gambar 10. Hasil preprocessing latar belakang file TA penulis

Dari hasil analisis *preprocessing* tersebut dapat diambil kesimpulan bahwa proses *preprocessing* pada aplikasi ini mampu memfilter latar belakang file TA penulis menghasilkan hasil data teks yang relevan untuk pencocokan. Namun pada penelitian ini sangat dibutuhkan input dari file yang akan diupload dengan format yang tepat sesuai *filter preprocessing*.

Filter kata “latar belakang masalah” yang merupakan format penulisan sub bab yang akan dihilangkan. Lalu terdapat penulisan yang salah menjadi “ltar belakng msalah” maka *preprocessing* tidak akan mendeteksinya sebagai salah satu kata yang akan dihapus dan akan dimasukan kedalam hasil teks yang akan diubah menjadi nilai hash.

4. KESIMPULAN

Berdasarkan hasil kegiatan penelitian yang telah dilakukan penulis pada penelitian ini, maka didapatkan kesimpulan:

1. Algoritma winnowing dapat di implementasikan pada sistem berbasis web. Algoritma tersebut mampu mendeteksi plagiasi file TA dalam waktu yang cukup cepat. Secara tekstual algoritma ini sangat efektif dalam menangani plagiarisme copy paste dan relokasi kata pada standart mesin aritmatik.
2. Kelemahan dari algoritma ini adalah tidak dapat memberikan jaminan dan bukti terhadap plagiarisme yang ditemukan. Pendeteksian hanya dapat dilakukan jika nilai modulo hash adalah 0. Namun tanpa penggunaan modulo nilai hash yang dihasilkan akan overflow.
3. Pemilihan nilai n-gram akan mempengaruhi nilai similarity dan waktu proses dari sistem. Nilai n-gram yang kecil akan memberikan nilai similarity yang lebih besar dan nilai n-gram yang besar akan memberikan nilai similarity yang lebih kecil. Hal ini dikarenakan semakin kecil substring maka relatif semakin kecil karakter yang akan

mempengaruhi nilai hash dan dapat memberikan nilai fingerprint yang identik sehingga pencocokan menjadi tidak relevan.

4. Penggunaan algoritma *winnowing* untuk mendeteksi plagiasi file TA membutuhkan pengecekan lebih lanjut terhadap file yang akan diproses, karena efektitas *preprocessing* algoritma *winnowing* yang membutuhkan file teks tanpa ada error penulisan yang sering muncul pada Tugas Akhir.

5. SARAN

Ada beberapa hal yang perlu diperhatikan dalam melakukan penelitian lebih lanjut terkait sistem rekomendasi ini, yaitu:

1. Algoritma *winnowing* akan lebih baik jika digunakan pada objek penelitian lain yang berskala lebih kecil, hal ini dikarenakan mendeteksi file Tugas Akhir dibutuhkan *preprocessing* konten yang spesifik untuk dapat menghasilkan hasil yang akurat.
2. Algoritma *winnowing* pada saat ini belum mampu menampilkan bukti teks yang memplagiasi. Diharapkan penelitian kedepan dapat mengatasi kekurangan algoritma *winnowing* tersebut.
3. Penelitian kedepan terhadap penentuan *n-gram* terhadap teks yang efisien untuk meningkatkan performa algoritma *winnowing*. Pemilihan *n-gram* yang tepat dapat meningkatkan akurasi dari deteksi plagiarisme.

DAFTAR PUSTAKA

[1] V. Stepchyshyn and R. S. Nelson, Library plagiarism policies, Chicago: America Library Association, (2007).

[2] S. M. Alzahrani, N. Salim and A. Abraham, "Understanding Plagiarism Linguistic Patterns, Textual Features and Detection Methods," *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C:,* vol. 42, no. 2, pp. 133-149, March 2012.

[3] A. T. Wibowo, K. W. Sudarmadi and A. M. Barnawi, "Comparison Between Fingerprint and Algorithm to Detect Plagiarism Fraud on Bahasa Indonesia Document," in *International Conference of Information and Communication Technology*, Bandung, 2013.

[4] A. Hidayat, "Algoritma Winnowing," 14 April 2016. [Online]. Available: arfianhidayat.com/algoritma-winnowing.html. [Accessed 30 July 2016].