

# Pengaruh Algoritma *Stemming Porter* Terhadap Kinerja Algoritma *Rabin Karp* Untuk Mendeteksi Plagiarisme Teks Bahasa Indonesia

<sup>1</sup>Yulia Darnita, <sup>2</sup>Dany Adhi Wibowo, <sup>3</sup>Rozali Toyib, <sup>4</sup>Nica Aulia

<sup>1,2,3</sup> Program Studi Teknik Informatika Universitas Muhammadiyah Bengkulu

<sup>4</sup> Program Studi Sistem Informasi Universitas Muhammadiyah Bengkulu Jln. Bali, Bengkulu 38119

Email: [yuliadarnita@umb.ac.id](mailto:yuliadarnita@umb.ac.id), [danyadhiwibowo@gmail.com](mailto:danyadhiwibowo@gmail.com), [rozalitoiyib@umb.ac.id](mailto:rozalitoiyib@umb.ac.id)

*Abstract*—Plagiarism or plagiarism is the act of taking ideas, writings, essays or opinions of others without including the source and making it as if the writing, essay or opinion is one's own, in practice plagiarism is very easy to do, namely by copying, sticking and modifying some of the contents document. Rabin Karp's Algorithm. Rabin Karp Algorithm is a string search algorithm discovered by Michael Rabin and Richard Karp. This algorithm uses hashing to find a string in a text. In addition to using the Rabin Karp algorithm, to get a good percentage of results, the need for preprocessing files to be detected, there are many stages of preprocessing including case folding, tokenizing, filtering, stemming, tagging and analyzing. This system will produce a plagiarism detection system using two algorithm, the Stemming Porter algorithm, which functions to convert words in a document into basic word forms and the Rabin Karp algorithm, which functions to compare the similarities of two documents to be tested. The results of testing the Rabin Karp algorithm detect similarities not based on the similarity of each word but by detecting the similarity of each Fingerprint between the two documents, the Rabin Karp algorithm has a faster processing time, whereas the detection of plagiarism using the Stemming Porter algorithm and the Rabin Karp algorithm has a longer processing time.

*Keywords:* plagiarism, robin karp, preprocessing, stemming

*Abstrak*—Plagiat atau plagiarisme adalah perbuatan mengambil ide, tulisan, karangan atau pendapat orang lain tanpa mencantumkan sumber dan menjadikannya seolah-olah tulisan, karangan atau pendapat itu milik sendiri, dalam praktiknya plagiarisme ini sangat mudah dilakukan, yaitu dengan cara menyalin, menempel dan memodifikasi sebagian isi dokumen. Algoritma Rabin Karp. Algoritma Rabin Karp adalah suatu algoritma pencarian string yang ditemukan oleh Michael Rabin dan Richard Karp. Algoritma ini menggunakan hashing untuk menemukan sebuah string dalam sebuah teks. Selain menggunakan algoritma Rabin Karp, untuk mendapatkan hasil presentase yang baik maka diperlukannya preprocessing file yang akan dideteksi, ada banyak tahapan dari preprocessing diantaranya yaitu case folding, tokenizing, filtering, stemming, tagging dan analyzing Sistem ini nantinya akan menghasilkan sebuah sistem deteksi plagiat dengan menggunakan dua algoritma yaitu algoritma Stemming Porter yang berfungsi untuk mengubah kata-kata dalam dokumen kedalam bentuk kata dasar dan algoritma Rabin Karp yang berfungsi untuk membandingkan kemiripan dua dokumen yang akan diuji. Hasil pengujian Algoritma Rabin Karp mendeteksi kemiripan tidak berdasarkan dengan kemiripan setiap kata namun dengan melakukan pendeteksian kemiripan setiap Fingerprint antar kedua dokumen, Algoritma Rabin Karp memiliki waktu proses yang lebih cepat, sedangkan pendeteksian plagiat dengan menggunakan algoritma Stemming Porter dan algoritma Rabin Karp memiliki waktu proses yang lebih lama.

*Kata Kunci :* plagiat, robin karp, preprocessing, stemming

## I. PENDAHULUAN

Plagiat atau plagiarisme adalah perbuatan mengambil ide, tulisan, karangan atau pendapat orang lain tanpa mencantumkan sumber dan menjadikannya seolah-olah tulisan, karangan atau pendapat itu milik sendiri. Dalam praktiknya plagiarisme ini sangat mudah dilakukan, yaitu

dengan cara menyalin, menempel dan memodifikasi sebagian isi dokumen. Plagiarisme ini merupakan hal yang tidak baik karena dapat menghambat kreatifitas seseorang, dan menumbuhkan sifat malas untuk berfikir.

Dalam dunia pendidikan tindakan plagiarisme bukanlah hal baru lagi, apa lagi

dikalangan mahasiswa hampir setiap hari harus mengerjakan tugas-tugas kuliah yang diberikan oleh dosen. Dalam mengerjakan tugas-tugas tersebut praktik plagiarisme tidak dapat dihindari lagi mengingat waktu untuk mengerjakan tugas terbatas dan kurangnya keinginan untuk menyelesaikan tugas dengan kemampuan sendiri. Dalam hal ini banyak pelaku-pelaku plagiarisme yang melakukan plagiat dengan cara mengubah struktur kalimat, merubah kosakata, serta menambah atau mengubah imbuhan yang digunakan. Dengan begitu maka plagiarisme akan sedikit sulit dideteksi.

Maka diperlukan sebuah sistem aplikasi untuk mempermudah dalam pengecekan plagiarisme tersebut. Banyak algoritma yang dapat digunakan untuk menyelesaikan masalah tersebut, salah satunya adalah algoritma *Rabin Karp*. Algoritma *Rabin Karp* adalah suatu algoritma pencarian *string* yang ditemukan oleh Michael Rabin dan Richard Karp. Algoritma ini menggunakan *hashing* untuk menemukan sebuah *string* dalam sebuah teks. Selain menggunakan algoritma *Rabin Karp*, untuk mendapatkan hasil presentase yang baik maka diperlukannya *preprocessing* file yang akan dideteksi, ada banyak tahapan dari *preprocessing* diantaranya yaitu *case folding, tokenizing, filtering, stemming, tagging* dan *analyzing*. *Praprocessing* ini diharapkan dapat membuat proses deteksi plagiat lebih efektif dan akurat, salah satu contohnya adalah tahapan *stemming*. *Stemming* berfungsi untuk mengembalikan kata-kata dalam sebuah kalimat atau dokumen kedalam bentuk kata dasar (*root word*). Ada beberapa algoritma yang dapat digunakan pada proses *stemming* ini, salah satu contohnya adalah algoritma *stemming Porter* bahasa Indonesia. Sistem ini nantinya akan menghasilkan sebuah sistem deteksi plagiat dengan menggunakan dua algoritma yaitu algoritma *Stemming Porter* yang berfungsi untuk mengubah kata-kata dalam dokumen kedalam bentuk kata dasar dan algoritma *Rabin Karp* yang berfungsi untuk membandingkan kemiripan dua dokumeyang akan diuji. Dalam pembangunan sistem ini akan menggunakan bahasa pemrograman PHP dan menggunakan *MySQL* sebagai *database* yang akan digunakan serta akandirancang berbasis web agar pemanfaatan sistem ini lebih meluas.

Pencarian informasi berupa dokumen teks atau yang dikenal dengan istilah Information

Retrieval (IR) merupakan proses pemisahan dokumen-dokumen yang dianggap relevan dari sekumpulan dokumen yang tersedia [1]. *Stemming* merupakan suatu proses yang terdapat dalam sistem IR yang mentransformasi kata-kata yang terdapat dalam suatu dokumen ke kata-kata akarnya (*rootword*) dengan menggunakan aturanaturan tertentu<sup>2</sup>.

Algoritma Rabin-Karp adalah algoritma pencocokan string yang menggunakan fungsi hash sebagai pembanding antara string yang dicari (*m*) dengan substring pada teks(*n*)<sup>3</sup>.

Kata plagiarisme berasal dari kata Latin *plagiarius* yang berarti merampok, membajak. Plagiarisme merupakan tindakan pencurian atau kebohongan intelektual. Plagiaris menurut epik adalah perompak yang suka mencuri atau menculik anak<sup>4</sup>.

## II. LANDASAN TEORI

### A. Algoritma Stemming

Algoritma Stemming diciptakan oleh Michael O. Rabin dan Ricard M.Karp pada tahun 1987 yang menggunakan fungsi hashing untuk menemukan *pattern*<sup>5</sup>. *Stemming* merupakan salah satu tahapan *text pre-processing* pada sistem temu kembali informasi (STKI)<sup>6</sup>.

Algoritma ini memiliki beberapa karakteristik adalah:

- Menggunakan fungsi Hashing merupakan suatu cara untuk mentransformasikan sebuah string menjadi suatu nilai yang unik dengan panjang tertentu yang berfungsi sebagai penanda string.
- K-Gram Merupakan rangkaian term dengan panjang K. Merupakan sebuah metode yang diaplikasikan untuk pembangkitan kata atau karakter. Metode ini digunakan untuk pengambilan potongan karakter huruf sejumlah K dari sebuah kata yang secara kontinuitas dibaca dari teks sumber Berikut contoh K gram dengan K=4: Teks : Saya adalah mahasiswa nusamandiri jurusan sistem informasi Hasil : Saya|ad|al|ah m|ahas|iswa| nus|aman|diri| jur|usan| sis|tem |info|rmas|i.

### B. Algoritma Rabin-Karp

Algoritma Rabin-Karp adalah suatu algoritma pencarian string yang ditemukan oleh

Michael Rabin dan Richard Karp. Algoritma ini menggunakan hashing untuk menemukan sebuah substring dalam sebuah teks<sup>7</sup>. Algoritma Rabin-Karp ini menggunakan fungsi hash. Fungsi hash adalah fungsi yang digunakan untuk mengubah string menjadi untaian integer. Pada algoritma ini untaian string akan diubah menjadi integer berdasarkan bilangan ASCII nya<sup>8</sup>.

### C. Preprocessing Dokumen

Proses peringkasan dokumen adalah sebuah proses untuk melakukan pengurangan volume dokumen menjadi lebih ringkas, dengan cara mengambil inti dokumen dan membuang term yang dianggap tidak penting tanpa mengurangi makna sebuah dokumen, terdapat dua tipe pembuatan suatu ringkasan yang mengambil bagian terpenting dari teks aslinya yaitu abstrak dan ekstrak. Abstrak menghasilkan sebuah interpretasi terhadap teks aslinya, dimana sebuah kalimat akan ditransformasikan menjadi kalimat yang lebih singkat<sup>9</sup>. Terdapat beberapa tahap yang dilakukan dalam preprocessing dokumen, yaitu : ekstraksi term, penghilangan stopwords, stemming, dan seleksi term. Pada tahap awal, hasil dari ekstraksi dokumen dikumpulkan dalam suatu koleksi single word<sup>10</sup>.

### D. Plagiarisme

Istilah plagiarisme atau sering disebut plagiat adalah penjiplakan atau pengambilan karangan, pendapat, dan sebagainya dari orang lain dan menjadikannya seolah karangan dan pendapat sendiri. Sedangkan Permendiknas No. 17 Tahun 2010 tentang Pencegahan dan Penanggulangan Plagiat di Perguruan Tinggi menyebutkan plagiat sebagai perbuatan secara sengaja atau tidak sengaja dalam memperoleh atau mencoba memperoleh angka kredit atau nilai untuk suatu karya ilmiah, dengan mengutip sebagian atau seluruh karya dan/atau karya ilmiah pihak lain yang diakui sebagai karya ilmiahnya, tanpa menyatakan sumber secara tepat dan memadai<sup>11</sup>. Ada dua jenis plagiarisme, yaitu plagiarisme yang dilakukan dengan sengaja (deliberate) dan plagiarisme yang dilakukan secara tanpa disengaja (accidental). Deliberate plagiarism adalah kegiatan yang sengaja dilakukan oleh seseorang untuk membajak karya ilmiah orang lain, contohnya adalah membajak isi buku orang lain, menerjemahkan karya orang lain tanpa meminta izin terlebih dahulu (apalagi jika mengklaimnya sebagai karyanya sendiri), dll.

Sedangkan accidental plagiarism terjadi lebih disebabkan karena ketidaktahuan si penulis tentang kaidah-kaidah penulisan karya ilmiah dan tentang tata cara atau etika menulis artikel ilmiah atau mungkin karena si penulis artikel tidak memiliki akses ke perpustakaan yang diperlukannya tersebut<sup>12</sup>.

### E. Bahasa

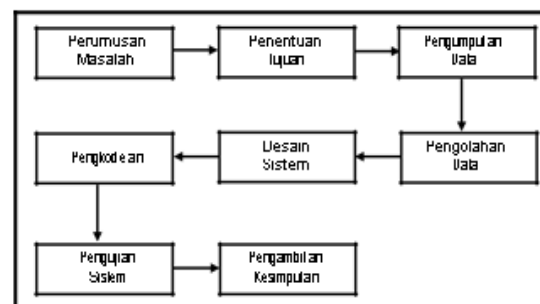
Bahasa adalah salah satu faktor mendasar yang membedakan manusia dengan hewan. Bahasa sebagai anugerah dari Sang Pencipta memungkinkan individu dapat hidup bersama dengan orang lain, membantu memecahkan masalah, dan memposisikan diri sebagai makhluk yang berbudaya<sup>13</sup>. Bahasa dalam bentuk tulisan sebenarnya terdiri atas simbol-simbol satuan yang jika dikombinasikan akan mempunyai arti yang berbeda-beda. Simbol-simbol yang bisa dipergunakan dalam sebuah bahasa tentunya terbatas jumlahnya, yang membentuk sebuah himpunan dan disebut sebagai abjad (alphabet)<sup>14</sup>. Bahasa merupakan suatu sistem komunikasi yang mempergunakan simbol-simbol yang bersifat arbitrer/ manasuka, yang dapat diperkuat dengan gerak-gerak badaniah yang nyata. Simbol adalah tanda yang diberikan makna tertentu, yaitu mengacu pada sesuatu yang dapat diserap oleh pancaindera<sup>15</sup>.

## III. METODE PENELITIAN

### A. Kerangka Kerja

Untuk membantu dalam melakukan penelitian ini, maka perlu adanya kerangka kerja. Kerangka kerja merupakan langkah-langkah yang disusun secara sistematis, logis, jelas dan terstruktur, kerangka kerja ini digunakan peneliti sebagai pedoman dalam melakukan penelitian.

Berikut ini adalah susunan kerangka kerja yang digunakan dalam penelitian ini:



Gambar 1. Kerangka Kerja Penelitian

### B. Flowchart

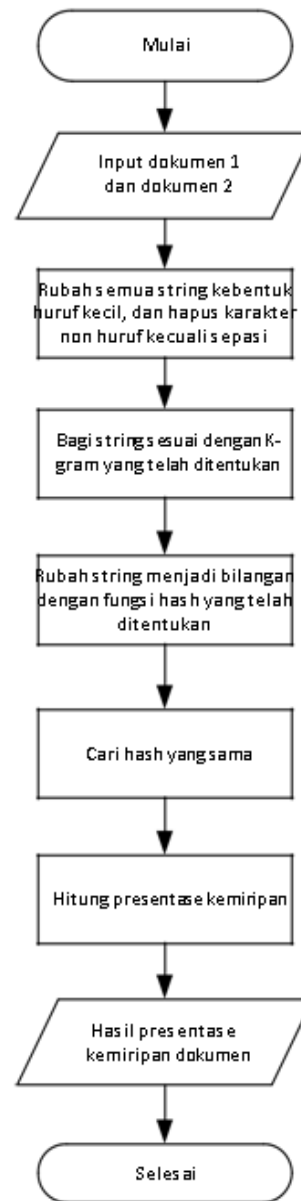
Dalam sistem yang akan dibangun ini memiliki tiga *flowchart* (proses) yaitu *flowchart* proses *stemming Porter* bahasa Indonesia, *flowchart* proses deteksi plagiat dengan algoritma *Rabin Karp*, dan *flowchart* proses deteksi plagiat dengan algoritma *Rabin Karp* dan algoritma *stemming Porter* bahasa Indonesia. Ketiga *flowchart* diatas akan menggambarkan bagaimana proses masing-masing sistem akan dibangun dapat bekerja. Berikut ini adalah gambar dari ketiga *flowchart* tersebut:

1. *Flowchart Stemming Porter* Bahasa Indonesia



Gambar 2. *Flowchart Stemming Porter*.

2. *Flowchart* Deteksi Plagiat Dengan Algoritma *Rabin Karp*



Gambar 3. *Flowchart* Deteksi Plagiat Dengan

3. *Flowchart* Deteksi Plagiat Dengan Algoritma *Stemming Porter* Dan Algoritma *Rabin Karp*



Gambar 4. Flowchart Deteksi Plagiat Dengan Algoritma Rabin Karp dan Stemming Porter.

### C. Analisis Manual

Berikut ini adalah studi kasus dari sebuah proses deteksi plagiat file dokumen dengan algoritma *stemming Porter* dan algoritma *Rabin karp*.

Misalkan sumber *string* dari dokumen 1 adalah “Bacalah 5 Buku.” dan dokumen 2 “Menjual Buku.”.

Tahap pertama yang dilakukan sistem adalah tahap *case folding* yaitu mengubah sumber *string* kedalambentuk huruf kecil dan penghapusan karakter-karakter non huruf selain spasi. Maka sumber *string* yang telah diinput tersebut menjadi:

Dokumen 1 : “bacalah buku”

Dokumen 2 : “menjual buku”

Setelah itu barulah sumber *string* di-*stemming* yaitu proses merubah kata-kata kedalam bentuk kata dasar dengan cara menghapus imbuhan. Sebelum menghapus imbuhan pada sumber *string*, terlebih

dahulu kata dicek apakah ada di *database* kamus kata dasar atau tidak, jika ada maka proses penghapusan imbuhan tidak perlu dilakukan.

Dokumen1: -kata “bacalah” → tidak ada di kamus

-kata “buku” → ada di kamus

Dokumen2:-kata “menjual” → tidak ada di kamus

-kata “buku” → ada di kamus

Setelah dicek pada kamus kata dasar, kata yang tidak ada pada kamus akan diproses untuk menghilangkan imbuhan-imbuhan, jadi kata yang akan diproses adalah kata “bacalah” dan kata “menjual”.

Jadi, sumber *string* yang dihasilkan setelah proses *stemming* adalah:

Dokumen 1 : “baca buku”

Dokumen 2 : “jual buku”

Selanjutnya adalah tahap merubah sumber *string* sesuai dengan *k-gram* yang telah ditentukan. Misalkan pada studi kasus ini menggunakan *k-gram* atau  $k=4$ , maka sumber *string* berubah menjadi beberapa bagian seperti berikut ini:

Dokumen 1: “baca buku”

*k-gram*: {baca} {aca} {ca b} {a bu} { buk}  
{ buku}

Dokumen 2: “jual buku”

*k-gram*: {jual} {ual} {al b} {l bu} { buk}  
{ buku}

setelah sumber *string* dibagi sesuai dengan *k-gram* yang ditentukan, maka proses selanjutnya adalah mengubah setiap sumber *string* tersebut menjadi bilangan yang disebut dengan *hash value* dengan menggunakan rumus sebagai berikut:

$$I^p = P1*d^{(m-1)} + P2*d^{(m-2)} + \dots + P(m-1)*d^1 + P(m-1)*d^0$$

Keterangan :

$I$  = Nilai *hash*  
 $P$  = Nilai *ASCII* karakter (desimal)  
 $m$  = Banyak karakter (indeks karakter)  
 $d$  = Basis Bilangan (nilai harus bilangan prima)

Dengan menggunakan  $b=2$  dan  $k$ -gram ( $k=4$ ). Berikut beberapa contoh penghitungan manual dari sumber *string* diatas:

(nilai ASCII  $b=98, a=97, c=99, a=97$ )

$$\begin{aligned} I &= 98*2^3+97*2^2+99*2^1+97*2^0 \\ &= 784+388+198+97 \\ &= 1467 \end{aligned}$$

Maka nilai *hash* dari *string* { baca } adalah 1467 (nilai ASCII  $a=97, c=99, a=97$  space=32)

$$\begin{aligned} I &= 97*2^3+99*2^2+97*2^1+32*2^0 \\ &= 776+396+194+32 \\ &= 1398 \end{aligned}$$

Maka nilai *hash* dari *string* {aca } adalah 1398 (nilai ASCII  $c=99, a=97$  space=32  $b=98$ )

$$\begin{aligned} I &= 99*2^3+97*2^2+32*2^1+98*2^0 \\ &= 792+388+64+98 \\ &= 1342 \end{aligned}$$

Keterangan:

S = Nilai kemiripan

C = Jumlah *fingerprnt* yang sama dari kedua dokumen

A = Jumlah *fingerprnt* dokumen 1

B = Jumlah *fingerprnt* dokumen 2

Jadi, dengan rumus diatas nilai kemiripan antar dokumen 1 dan 2 adalah sebagai berikut:

Dari dokumen uji sebelumnya maka dapat diketahui

C = 2

A= 6

B= 6

$$S = \frac{2*2}{6+6}$$

$$= \frac{4}{12}$$

$$= 0,333$$

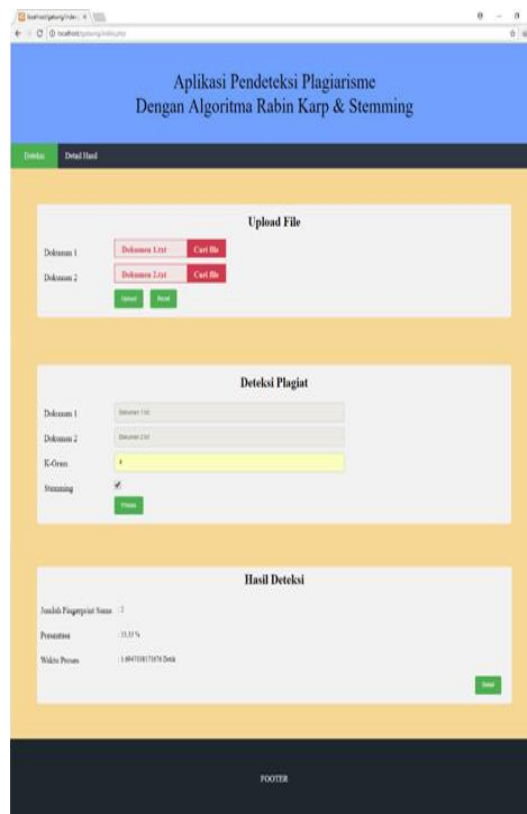
Jadi, nilai kemiripan antara dokumen 1 dan dokumen 2 adalah 33,3%

## HASIL DAN PEMBAHASAN

### A. Hasil Program

#### 1. Tampilan Menu Deteksi Plagiat

Pada tampilan menu deteksi plagiat ini, sistem akan menampilkan 3 bagian, yaitu bagian *upload* file, bagian deteksi plagiat, dan bagian hasil deteksi. Tampilan menu deteksi plagiat dapat dilihat pada gambar 5 berikut.



Gambar 5. Tampilan Deteksi.

Pada bagian *upload* file digunakan untuk memilih file yang akan dideteksi, *upload* disini berfungsi untuk memberitahu sistem, file mana yang akan dideteksi kemiripannya, setelah user memilih file yang akan dideteksi, user harus menekan tombol *upload* agar teks dalam file dapat digunakan dalam menu deteksi plagiat. Tampilan bagian *upload* file tampak pada Gambar 6 berikut.



Gambar 6. Tampilan Upload File.

Bagian selanjutnya adalah bagian deteksi plagiat, disini user akan ditampilkan sebuah *form* yang digunakan untuk menginput data deteksi plagiat yang diinginkan, data deteksi tersebut berupa *input* K-Gram dan *stemming*. *Input* K-Gram disini digunakan untuk memilih berapa nilai K-Gram yang ingin digunakan. Nilai k-gram adalah nilai yang digunakan untuk pemotongan N-karakter yang diambil dari suatu *string*, dan

pemotongan selanjutnya dilakukan dengan cara bergerak satu katekter kedepan. Sedangkan pada *stemming*, *user* akan disajikan dengan *inputan* berupa *checkbox*, *checkbox* disini digunakan untuk memilih apakah sumber *string* akan distemming atau tidak. Apabila *checkbox* diklik maka sumber *string* akan distemming terlebih dahulu sebelum diproses, dan apabila *checkbox* tidak di-klik maka sumber *string* tidak akan distemming dan langsung akan diproses. Tampilan bagian deteksi plagiat ini akan tampak pada gambar 7. berikut :

Gambar 7. Tampilan Form Deteksi Plagiat.

Bagian ketiga adalah bagian hasil deteksi, bagian ini berfungsi untuk menampilkan data-data hasil deteksi plagiat, baik yang menggunakan *stemming* maupun yang tidak menggunakan *stemming*. Data-data yang akan ditampilkan pada hasil deteksi ini ada 3 yaitu jumlah *fingerprint* sama, presentase, dan waktu proses. “Jumlah *Fingerprint* Sama” disini akan menampilkan berapa jumlah *fingerprint* yang sama antara kedua file yang diuji kemiripannya. Bagian “Persentase” akan menampilkan hasil perhitungan persentase dari algoritma *Rabin Karp*, yang akan ditampilkan adalah berapa persen kemiripan antar kedua dokumen. Dan pada bagian terakhir adalah bagian “Waktu Proses”, pada bagian ini akan ditampilkan berapa lama waktu yang dibutuhkan sistem dalam mendeteksi plagiat, waktu yang akan ditampilkan menggunakan satuan *second* atau detik. Tampilan bagian “Hasil Deteksi” dapat dilihat pada gambar 8 berikut ini.

Gambar 8. Tampilan Hasil Deteksi.

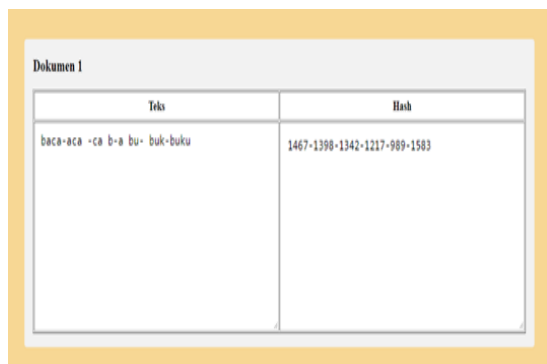
## 2. Tampilan Menu Detail Hasil

Pada tampilan menu “Detail Hasil” ini, sistem akan menampilkan detail dari hasil deteksi plagiat yang telah dilakukan sebelumnya. Pada menu “Detail Hasil” ini ada beberapa bagian antara lain bagian “Dokumen 1”, bagian “Dokumen 2”, dan bagian “*Fingerprint*”. Tampilan dari menu “Detail Hasil” dapat dilihat pada gambar 9 berikut.

Gambar 9. Tampilan Menu Hasil Deteksi

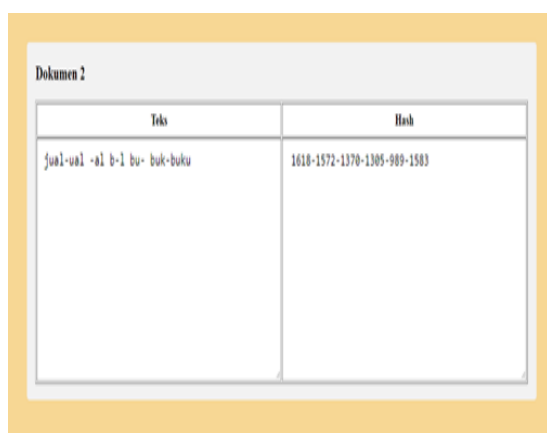
Pada bagian “Dokumen 1” sistem akan menampilkan 2 tabel, yaitu tabel teks dan tabel *hash*. Pada bagian tabel “teks” sistem akan menampilkan hasil *string* yang telah dipecah berdasarkan nilai *K-gram* yang telah diinput sebelumnya, sedangkan pada bagian tabel “*hash*” sistem akan menampilkan nilai *hash* dari *string*

pada tabel teks sebelumnya. Tampilan bagian “Dokumen 1” tampak pada Gambar 10. berikut.



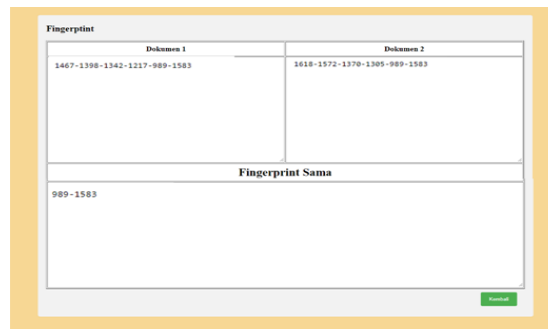
Gambar 10. Tampilan Hasil Deteksi Dokumen 1.

Pada bagian “Dokumen 2” sistem akan menampilkan 2 tabel yang sama seperti pada bagian Dokumen 1, yaitu tabel teks dan tabel hash. Pada bagian tabel “teks” sistem akan menampilkan hasil string dokumen 2 yang telah dipecah berdasarkan nilai K-gram yang telah diinput sebelumnya, sedangkan pada bagian tabel “hash” sistem akan menampilkan nilai hash dari string pada tabel teks sebelumnya. Tampilan bagian Dokumen 1 tampak pada Gambar 10. berikut.



Gambar 10. Tampilan Hasil Deteksi Dokumen 2.

Pada bagian selanjutnya adalah bagian fingerprint, pada bagian ini sistem akan menampilkan nilai unique (tidak ada yang sama) pada masing-masing nilai hash sebelumnya. Selain menampilkan nilai unique dari masing-masing nilai hash dokumen, sistem juga menampilkan fingerprint yang sama antara dokumen 1 dan nilai hash dokumen 2. Tampilan bagian fingerprint dapat dilihat pada gambar 11 berikut ini.



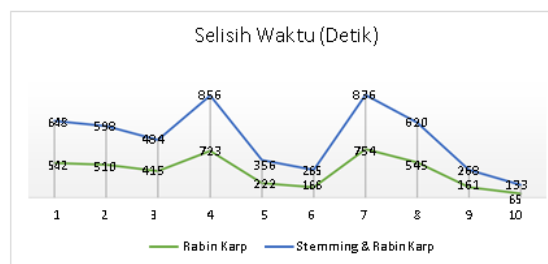
Gambar 11. Tampilan Fingerprint Dokumen 1 dan Dokumen 2.

B. Pembahasan

Dalam penelitian ini semua nilai K-gram yang digunakan adalah 4. Peneliti ini bertujuan untuk mengetahui pengaruh algoritma Stemming Porter bahasa Indonesia terhadap kinerja algoritma Rabin Karp dalam mendeteksi plagiarisme dokumen yang diujikan di atas. Hasil pengujian dapat dilihat dalam tabel dan grafik berikut ini.

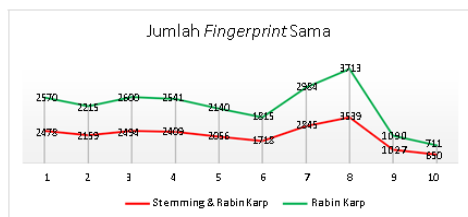
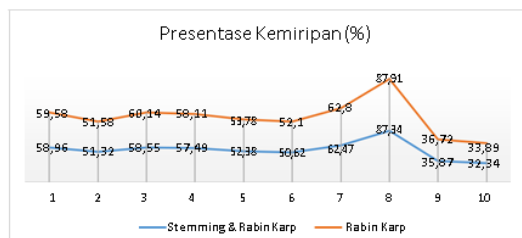
Tabel 1. Hasil Pengeujian Deteksi Plagiat.

Uji Coba Ke	Algoritma	Hasil Deteksi		
		Waktu proses (detik)	Jumlah Fingerprint sama	Persentase kemiripan (%)
1	Stemming-Rabin Karp	648	2478	58,96
	Rabin Karp	548	2570	59,58
2	Stemming-Rabin Karp	598	2159	51,32
	Rabin Karp	510	2215	51,58
3	Stemming-Rabin Karp	484	2494	58,55
	Rabin Karp	415	2600	60,14
4	Stemming-Rabin Karp	856	2409	57,49
	Rabin Karp	723	2541	58,11
5	Stemming-Rabin Karp	356	2056	52,38
	Rabin Karp	222	2140	53,76
6	Stemming-Rabin Karp	265	1718	50,62
	Rabin Karp	166	1815	52,10
7	Stemming-Rabin Karp	836	2845	62,47
	Rabin Karp	754	2964	62,80
8	Stemming-Rabin Karp	620	3539	87,34
	Rabin Karp	545	3713	87,91
9	Stemming-Rabin Karp	268	1027	35,87
	Rabin Karp	161	1090	36,72
10	Stemming-Rabin Karp	133	650	32,34
	Rabin Karp	65	711	33,89



Gambar 12. Perbandingan Waktu Proses.



Gambar 13. Perbandingan Jumlah *Fingerprint* Sama.

Gambar 14 Perbandingan Presentase Kemiripan

Pada gambar 12 menunjukkan selisih waktu yang diperlukan dalam mendeteksi plagiat. Seperti yang terlihat pada grafik diatas, penggunaan algoritma *stemming* membutuhkan waktu yang lebih lama, rata-rata selisih waktunya adalah 98 detik lebih lama. Hal ini dikarenakan sebelum dideteksi, dokumen harus memasuki proses *stemming* terlebih dahulu baru dilakukannya proses deteksi dengan algoritma *Rabin Karp*.

Sedangkan pada gambar 13 di atas, terlihat grafik perbandingan jumlah *fingerprint* sama, pengujian yang hanya menggunakan algoritma *Rabin Karp* memiliki jumlah *fingerprint* sama lebih banyak daripada pengujian yang dilakukan dengan menggunakan algoritma *stemming-Rabin Karp*, hal ini disebabkan karena ketika pengujian menggunakan algoritma *stemming-RabinKarp* dokumen yang diuji harus memasuki proses *stemming* terlebih dahulu, dan hal ini menyebabkan beberapa *string* dalam dokumen akan dihapus, dengan begitu maka jumlah *string* pun akan berkurang.

Gambar 14 menunjukkan perbandingan presentase hasil deteksi kemiripan dua dokumen antara algoritma *stemming-RabinKarp* dan algoritma *Rabin Karp*. Dari gambar grafik diatas dapat dilihat deteksi menggunakan algoritma *stemming-Rabin Karp* memiliki presentase yang lebih rendah, hal ini dikarenakan jumlah *Fingerprint* sama yang berkurang, sehingga presentase yang dihasilkan juga ikut berkurang.

## KESIMPULAN

Dari hasil pembahasan, perancangan serta pembuatan Aplikasi Deteksi Plagiat dengan algoritma *Stemming Porter* dan *Rabin Karp*, maka penulis menarik kesimpulan sebagai berikut:

1. Sistem yang dibangun telah berhasil menerapkan algoritma *Stemming Porter* dan algoritma *Rabin Karp* untuk mendeteksi plagiat.
2. Algoritma *Rabin Karp* mendeteksi kemiripan tidak berdasarkan dengan kemiripan setiap kata namun dengan melakukan pendeteksian kemiripan setiap *Fingerprint* antar kedua dokumen yang merupakan hasil dari proses *Hashing* dan *K-Gram*.
3. Berdasarkan pengujian sebelumnya, pendeteksian plagiat dengan algoritma *Rabin Karp* memiliki presentase lebih tinggi dan jumlah *fingerprint* yang lebih banyak ketimbang pendeteksian menggunakan dua algoritma *Stemming Porter* dan *Rabin Karp*.
4. Pendeteksian plagiat menggunakan algoritma *Rabin Karp* memiliki waktu proses yang lebih cepat, sedangkan pendeteksian plagiat dengan menggunakan algoritma *Stemming Porter* dan algoritma *Rabin Karp* memiliki waktu proses yang lebih lama.

## Referensi

- [1] Agusta, L. (2009). Perbandingan algoritma stemming Porter dengan algoritma Nazief & Adriani untuk stemming dokumen teks bahasa Indonesia. *Konferensi Nasional Sistem dan Informatika, 2009*, 196-201.
- [2] Pramudita, H. R. (2014). Penerapan Algoritma Stemming Nazief & Adriani dan Similarity pada Penerimaan Judul Thesis. *Data Manajemen dan Teknologi Informasi (DASI)*, 15(4), 15
- [3] Surahman, A. M. (2013). Perancangan Sistem Penentuan Similarity Kode Program Pada Bahasa C Dan Pascal Dengan Menggunakan Algoritma Rabin-Karp. *Jurnal Sistem dan Teknologi Informasi (JustIN)*, 1(1), 50-55.
- [4] Sastroasmoro, S. (2007). Beberapa Catatan tentang Plagiarisme. *Majalah Kedokteran Indonesia*, 57(8), 239-244.
- [5] Novitasari, D. (2017). Perbandingan Algoritma Stemming Porter dengan Arifin Setiono untuk Menentukan Tingkat Ketepatan Kata Dasar. *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, 1(2), 120-129.
- [6] Simarankir, M. S. H. (2017). Studi Perbandingan Algoritma-Algoritma Stemming Untuk Dokumen Teks Bahasa Indonesia. *Jurnal Inkofar*, 1
- [7] Oktaviani, P. (2016). Perbandingan Algoritma Boyer-Moore Dan Algoritma Rabin Karp Pada Pencarian Teks Dalam Undang Undang Perlindungan Anak.
- [8] Wirawan, T. P. (2008). Penggunaan Algoritma Rabin-Karp dalam Pencocokan String.
- [9] Luthfiarta, A., Zeniarja, J., & Salam, A. (2013, November). Algoritma Latent Semantic Analysis (LSA) pada peringkasan dokumen otomatis untuk proses clustering

- dokumen. In *Seminar nasional teknologi informasi & komunikasi terapan*.
- [10] Rozi, F., & Kartadie, R. (2017). Clustering Dokumen dengan Semantic Word Holonim dan Fuzzy Association Rule Mining. *Semnasteknomedia Online*, 5(1), 2-1.
- [11] Suganda, T. (2006). Perihal Plagiarisme dalam Artikel Ilmiah. *Agrikultura*, 17 (3), 161-164
- [12] Penggunaan software Turn It In untuk mendeteksi tindakan plagiarisme. *AL-MAKTABAH*, 16(1).
- [13] Dhieni, N., Fridani, L., Muis, A., & Yarmi, G. (2014). Metode pengembangan bahasa.
- [14] Basuki, T. A. (2000). Pengenalan suku kata bahasa Indonesia menggunakan finite-state automata. *Bandung: Universitas Khatolik Parahyangan*.
- [15] Mularsih, H. (2010). Strategi pembelajaran, tipe kepribadian dan hasil belajar Bahasa Indonesia pada siswa sekolah menengah pertama. *Makara, Sosial Humaniora*, 14(1), 65-79.