

Pendeteksi Untuk Tingkat Kemiripan Kata Pada Karya Tulis Ilmiah dengan Menggunakan Metode Nazief Adriani

Arni Damayanti

Fakultas Ilmu Komputer dan Teknologi Informasi, Prodi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia

Email: 'arnidamayanti1997@gmail.com

Email Penulis Korespondensi: arnidamayanti1997@gmail.com

Abstrak—Karya tulis ilmiah merupakan suatu karya tulis yang digunakan untuk memecahkan permasalahan dengan metode-metode ilmiah yang didalamnya memaparkan pendapat, hasil pengamatan, fakta, data, yang akan menghasilkan suatu laporan tertulis yang dilakukan oleh seseorang ataupun secara tim dengan memenuhi etika kaidah dan keilmuan secara baik dan benar. Suatu karangan atau tulisan yang diperoleh sesuai dengan sifat keilmuannya dan didasari oleh hasil pengamatan, peninjauan, penelitian dalam bidang tertentu, disusun menurut metode tertentu dengan sistematika penulisan yang bersantun bahasa dan isinya dapat dipertanggung jawabkan kebenarannya/ keilmiahannya. Dengan perkembangannya teknologi komunikasi dan informasi yang semakin pesat menyebabkan pencarian informasi dan data sangatlah mudah dan cepat. Namun dengan perkembangan teknologi ini juga menimbulkan berbagai dampak negatif seperti penjiplakan kata. Penelitian ini juga bertujuan untuk mencari kesamaan kata atau similarity untuk memecah kata antara kata inputan dengan kata sumber. kemiripan tersebut bukanlah sinonim, melainkan kata yang mendekati kata sebenarnya, Simmilarity merupakan suatu isu yang sangat penting hampir pada setiap ilmu bidang pengetahuan. Fungsi dari Simmilarity adalah mencari kesamaan kata dan ketidaksamaan kata antara dua buah objek yang diobservasi. Maka dengan adanya algoritma Nazief-Adriani yaitu digunakan untuk mencari kata dasar pada suatu kalimat untuk menjadi lebih baik, algoritma stemming Nazief-Adriani menghasilkan akurasi yang lebih baik jika dibandingkan dengan algoritma yang lain karena disebabkan berbagai macam variasi kata serta kombinasi imbuhan yang harus dihapus demi untuk mendapatkan kata dasarnya.

Kata Kunci: Karya Ilmiah; Plagiat; Simmilarity; Nazief Adriani

Abstract—Scientific writing is a written work that is used to solve problems with scientific methods in which it presents opinions, observations, facts, data, which will produce a written report carried out by a person or in a team by meeting ethical and scientific rules properly. and right. An essay or writing obtained in accordance with its scientific nature and based on the results of observation, review, research in a particular field, compiled according to a certain method with systematic writing in a polite language and its contents can be accounted for for its truth/scientific nature. With the development of communication and information technology that is increasingly rapidly causing the search for information and data is very easy and fast. However, with the development of this technology also causes various negative impacts such as word plagiarism. This study also aims to find the similarity of words or similarity to break words between the input word and the source word. The resemblance is not a synonym, but a word that is close to the actual word, Simmilarity is a very important issue in almost every field of knowledge. The function of Simmilarity is to look for word similarity and word dissimilarity between two observed objects. So with the Nazief-Adriani algorithm, which is used to find basic words in a sentence to be better, the Nazief-Adriani stemming algorithm produces better accuracy when compared to other algorithms because it is caused by various kinds of word variations and combinations of affixes that must be removed for the sake of to get the root word.

Keywords: Scientific Work; Plagiarism; Simmilarity; Nazief Adriani

1. PENDAHULUAN

Dengan perkembangan teknologi dan komunikasi yang saat ini semakin pesat yang menyebabkan suatu pencarian informasi dan data sangatlah mudah dan cepat melalui internet. Perkembangansuatu data teks saat ini sudah mencakup jumlah yang sangat besar yang disebabkan oleh perkembangan dunia teknologi dan informasi dengan berbagai macam media *online* yaitu situs berita *online*, blog, dari sumber-sumber tersebut data teks menjadi sumber yang sangat potensial untuk digali lebih dalam.

Pemanfaatan teknologi dan informasi sekarang ini telah menjadi kebutuhan terpenting saat ini di era yang modern yang didalamnya terdapat komponen-komponen salah satunya yaitu dokumen teks, dokumen teks dalam bentuk digital ini memudahkan untuk melakukan penyimpanan sehingga mudah untuk terjadinya plagiat atau *plagiarism*, hal ini menjadi sorotan utama dikalangan intelektual dan akademis.

Maka dengan perkembangan teknologi yang semakin pesat menyebabkan berbagai dampak negatif yang timbul salah satunya seperti sering terjadi suatu penjiplakan yang disebut juga sebagai plagiat. Penjiplakan atau plagiat merupakan suatu tindak pidana yang merugikan seseorang dengan cara melakukan pengambilan ide ataupun pendapat yang dibuat oleh orang lain, secara tidak sengaja plagiat juga dapat terjadi pada pembuatan karya tulis yang terkadang lalai untuk mencantumkan sumber pustaka yang lengkap dan cermat sehingga akan mengakibatkan kerugian bagi seseorang serta tidak mencerminkan sikap yang kreatif dikalangan intelektual. Tanpa disadari juga suatu karya tulis ilmiah, gagasan ataupun ide seseorang bisa terjiplak oleh seseorang dengan secara sengaja yaitu dengan membuat tulisan yang mirip tanpa menuliskan asal usulnya ataupun bisa juga secara tidak sengaja yaitu dengan memparafrasekan dan meringkas tetapi menuliskan sumbernya, plagiat ini merupakan tindakan yang sangat menyalahgunakan hak kekayaan suatu kreatifitas intelektual seseorang untuk dijadikan kreatifitas orang lain yang diakui menjadi miliknya sendiri.

Algoritma *Nazief-Adriani* adalah cara yang digunakan untuk meningkatkan performa *IR (Information Retrieval)* untuk mentransformasikan kata-kata pada sebuah dokumen teks ke bentuk dasarnya. Proses stemming digunakan untuk menghilangkan atau mengubah kata-kata yang memiliki makna seperti prefik (awalan), sufiks (akhiran) dan konfiks (gabungan awalan dan akhiran).

Algoritma *Nazief-Adrianidi* merupakan aturan morfologi bahasa Indonesia yang sangat luas, yang telah dikumpulkan menjadi satu *group* serta dienkapsulasi pada imbuhan *affixes* yang telah diperbolehkan menjadi *allowed affixes* dan yang tidak diperbolehkan yaitu *disallowed affixes*. Algoritma *Nazief-Adrianidi* ini menggunakan kamus kata dasar dengan mendukung *recording*, yaitu menyusun kembali kata yang telah mengalami *stemming*.

Algoritma *Nazief-Adriani* digunakan untuk mencari kata dasar pada suatu kalimat untuk menjadi lebih baik, algoritma *Nazief-Adriani* menghasilkan akurasi yang lebih baik jika dibandingkan dengan algoritma yang lain karena disebabkan berbagai macam variasi kata serta kombinasi imbuhan yang harus dihapus demi untuk mendapatkan kata dasarnya.

Penerapan pada algoritma *Nazief-Adriani* dengan metode *cosive similaryti* yaitu untuk mencari kesamaan suatu dokumen dengan membandingkan penggunaan kalimat yang aktif dan pasif serta dapat menghasilkan tingkat optimal yang besar mencapai 93,81% jika dibandingkan dengan menggunakan algoritma yang lain.[1]

Kemiripan suatu dokumen atau kesamaan suatu kata pada karya tulis ilmiah tidak disebut plagiat jika informasi tersebut merupakan fakta umum, dengan mengubah *paraphrase* atau kalimat opini orang lain dengan menggunakan informasi yang jelas, memberikan tanda batas dengan jelas pada kutipan tulisan orang lain serta menuliskan secara lengkap sumbernya. Tindakan plagiat ini harus dicegah serta dihilangkan dengan menerapkan metode *Nazief-Adrianidi* yaitu untuk meningkatkan performa *IR* dengan mentransformasikan kata pada suatu dokumen teks ke dalam bentuk kata dasarnya dengan cara menghilangkan kata-kata berimbuhan yang memiliki arti morfologi yang berbeda-beda tetapi memiliki interpretasi semantik yang sama.[2]

2. METODOLOGI PENELITIAN

2.1 Natural Language Processing

Natural Language Processing merupakan pendekatan yang terkomputerisasi dalam menganalisa teks berdasarkan teknologi dan aspek teori. Natural Language Processing (NLP) dapat didefinisikan sebagai bidang teoritis yang dapat digunakan untuk mempresentasikan serta menganalisa teks yang ditulis secara (bahasa manusia) ataupun natural, pada satu level atau lebih analisis *linguistic* yang bertujuan untuk memperoleh *human-like language processing* yang dapat digunakan untuk mengimplementasikan dalam berbagai bidang.

2.2 Algoritma Nazief Adriani

Algoritma *Nazief-Adriani* merupakan algoritma yang digunakan pada tahap *stemming* yaitu salah satu cara yang digunakan untuk meningkatkan performa *IR* (*Information Retrieval*) untuk mentransformasikan kata-kata pada sebuah dokumen teks ke bentuk dasarnya dengan mengubah kata-kata yang memiliki makna seperti *prefiks* (awalan), *sufiks* (akhiran) dan *konfiks* (gabungan awalan dan akhiran).

Algoritma ini digunakan untuk penelitian sebagai algoritma pendukung pada proses penentuan nilai-nilai kemiripan pada teks dokumen, metode ini dilakukan untuk menentukan kemiripan dua buah objek. Algoritma *Nazief-Adrianidi* bisa mengenal kesamaan makna dari suatu kata dimana kata tersebut memiliki makna *prefiks*, *sufiks* dan *konfiks* diubah menjadi bentuk dasar.

Proses *Nazief-Adrianidi* cukup rumit dilakukan untuk teks berbahasa Indonesia sebab terdapat begitu banyak variasi imbuhan yang harus dibuang untuk mendapat suatu *root word* (kata dasar) pada sebuah kata. Secara umum kata dasar bahasa Indonesia terdiri atas *prefix 1 + prefix 2 + kata dasar + sufiks 3 + sufiks 2 dan sufiks 1*, algoritma ini menggunakan kamus kata dasar dan mendukung *recording* yaitu melakukan penyusunan kembali terhadap kata-kata yang mengalami proses *stemming* yang berlebihan.

Algoritma *Nazief-Adriani* ini pertama kali dikembangkan oleh Bobby Nazief dan juga Mirna Adriani. Algoritma ini didasari oleh aturan-aturan *morfologi* bahasa Indonesia yang sangat luas, yang dikumpulkan untuk menjadi satu *group* serta dienkapsulasi menjadi imbuhan yang diperbolehkan dan imbuhan tidak diperbolehkan. Algoritma ini menggunakan kamus kata dasar serta mendukung *recording* yaitu penyusunan kembali suatu kata yang mengalami proses *stemming* berlebihan. Adapun tahapannya sebagai berikut ini:

1. Carilah kata yang akan di *stemming* pada kamus, jika ditemukan maka diasumsikan bahwa kata tersebut adalah *root word*, maka algoritma dihentikan.
2. Menghilangkan *inflection suffixes* (“lah”, “kah”, “ku”, “mu”, atau “nya”) jika berupa *particles* (“lah”, “kah”, “tah”, “pun”) maka langkah tersebut diulangi lagi untuk menghapus *Possesive Promous* (“ku”, “mu”, “nya”) jika ditemukan algoritma berhenti, jika tidak dijumpai dalam kamus maka dilanjutkan oleh selanjutnya.
3. Hapus *Derivation Suffixes* (“i”, “an”, “kan”) jika kata ada pada kamus maka algoritma berhenti, jika tidak lakukan langkah “3a”.
 - a. Jika “an” sudah dihapus dan huruf terakhir dari kata tersebut adalah “k” maka “k” juga dihapus. Jika kata tersebut dijumpai dalam kamus maka algoritma berhenti, jika tidak lakukan langkah berikutnya ke “3b”.
 - b. Akhiran yang dihapus (“i”, “an”, “kan”) dikembalikan lalu lanjut ke langkah 4.
4. Hapus *Derivation Prefiks*, jika pada langkah 3 ada *suffiks* yang sudah dihapus maka lanjut ke langkah 4a, jika tidak lanjut ke langkah 4b.
 - a. Periksa tabel kombinasi awalan dan akhiran yang tidak diijinkan jika ditemukan algoritma berhenti, jika tidak ditemukan lanjut ke langkah 4b.

Tabel 1. kombinasi awalan dan akhiran yang tidak diijinkan

awalan	Akhiran yang tidak diijinkan
be	I
di	An
ke	i, kan
me	An
se	i, kan

- b. Menentukan tipe awalan lalu hapus awalan jika awalan kedua sama dengan awalan pertama maka algoritma berhenti.
 - c. Jika root word belum ditemukan lakukan langkah ke 5, jika sudah ditemukan maka algoritma berhenti.
5. Jika semua langkah selesai dilakukan tetapi tidak ditemukan atau tidak berhasil maka kata awal dianggap sebagai root word, maka proses selesai.[7]

3. HASIL DAN PEMBAHASAN

Dalam penelitian ini data yang digunakan adalah data yang berhubungan dengan karya tulis ilmiah, penelitian ini akan mendeteksi tingkat kemiripan dua dokumen teks. Dokumen teks tersebut akan melewati tahap *preprocessing* dengan menggunakan text mining, pada tahap ini akan dilakukan *case folding* (mengubah isi dokumen menjadi huruf kecil), lalu *filtering* (menghilangkan kata yg tidak penting), tahap selanjutnya yaitu *tokenizing* (pembentukan rangkaian k-gram). Setelah proses *preprocessing* selesai dilakukan maka dokumen akan di proses untuk mencari kesamaan atau kecocokan antara dua dokumen dengan menggunakan algoritma Nazief-Adriani. Hasil dari pencocokan dokumen tersebut kemudian dihitung similarity dengan indeks dengan nilai 0 sampai 1, jika nilai yang dihitung mendekati angka 1 maka dapat disimpulkan bahwa tingkat kesamaan dokumen tersebut semakin besar.

3.1 Penerapan Metode Nazief-Adriani

Algoritma *Nazief-Adriani* merupakan algoritma yang digunakan pada tahap stemming yaitu salah satu cara yang digunakan untuk meningkatkan performa *IR (Information Retrieval)* untuk mentransformasikan kata-kata pada sebuah dokumen teks ke bentuk dasarnya dengan mengubah kata-kata yang memiliki makna seperti prefiks (awalan), sufiks (akhiran) dan konfiks (gabungan awalan dan akhiran).

Algoritma ini digunakan untuk penelitian sebagai algoritma pendukung pada proses penentuan nilai-nilai kemiripan pada teks dokumen, metode ini dilakukan untuk menentukan kemiripan dua buah objek. Algoritma *Nazief-Adriani* bisa mengenal kesamaan makna dari suatu kata dimana kata tersebut memiliki makna prefix, sufiks dan konfiks diubah menjadi bentuk dasar. Langkah pertama yang akan dilakukan yaitu dengan melakukan tahap preprocessing pada dokumen :

1. Case folding

Pada proses ini adalah untuk mengubah semua huruf menjadi huruf kecil.

Abstraksi 1

algoritma nazief dan adriani merupakan algoritma untuk mengubah kata yang memiliki sufiks prefiks dan konfiks menjadi bentuk kata dasar algoritma nazief dan adriani digunakan dalam penelitian ini sebagai algoritma pendukung dalam proses penentuan kata dasar dari kata berimbuhan dalam sebuah kalimat untuk menentukan kata dasar dari kata berimbuhan ini dengan menghilangkan semua imbuhan afiks baik yang terdiri dari awalan prefiks sisipan infiks akhiran sufiks dan awalan akhiran konfiks pada kata berimbuhan dengan algoritma stemming nazief dan adriani serta dapat mengetahui kelas kata dari kata dasar yang berimbuhan tersebut

Anstraksi 2

penelitian ini mengimplementasikan algoritma *stemming* nazief-adriani untuk mencari kesamaan judul penulisan berbahasa indonesia dengan mencari kesamaan judul penulisan tersebut dapat membantu mahasiswa maupun dosen dalam menentukan judul penulisan yang ingin dibuat atau dapat menjadi referensi dalam pembuatan penulisan sehingga dapat menghindari tindakan *plagiarisme* judul penulisan penelitian ini juga membangun aplikasi yang dapat menambahkan data judul penulisan ke dalam *database* dan mencari kesamaan judul penulisan dari judul penulisan yang terdapat dalam *database*

2. Filtering atau pembuangan stopword

Langkah yang kedua yaitu menghilangkan kata-kata yang tidak penting seperti kata dan, dari dan lain sebagainya. Proses pembuangan stopword merupakan proses pembuangan *term* yang tidak memiliki arti atau relevan. *Term* tersebut di peroleh setelah tahap *case folding* dilakukan, kemudian dicek ke dalam daftar *stopword*, jika kata tersebut masuk ke dalam

daftar *stopword* maka kata tersebut tidak akan diproses lebih lanjut. Sedangkan jika sebuah kata tidak termasuk ke dalam daftar *stopword* maka kata tersebut akan masuk ke proses berikutnya. Berikut ini merupakan daftar *Stopword list* yang diperoleh setelah proses *case folding*. Dalam penelitian ini daftar *stopword* yang digunakan adalah dari (Tala).

Tabel 1. *Stopword list*

Dokumen 1		
Dan	Dalam	serta
merupakan	Ini	terdiri
Untuk	Sebagai	pada
Kata	Dari	tersebut
Yang	Sebuah	dapat
Menjadi	Dengan	semua
digunakan	Baik	mengatahui

Dokumen 2		
Ini	Yang	juga
Untuk	Ingin	dari
Dengan	dibuat	terdapat
Tersebut	Atau	dalam
Dapat	Dan	sehingga
Maupun	Menjadi	

Tabel *stopword list* yang diperoleh diatas disebut sebagai daftar *stopword* tala yang merupakan sebagai hasil *filtering* dari proses *stopword* yang akan diproses ke tahap selanjutnya.

Abstraksi 1

algoritma nazief adrian algoritma mengubah memiliki sufiks prefix konfiks bentuk dasar algoritma nazief adriani penelitian algoritma pendukung proses penentuan dasar berimbunan kalimat menentukan dasar berimbunan menghilangkan imbunan afiks awalan prefix sisipan infiks akhiran sufiks awalan akhiran konfiks berimbunan algoritma stemming nazief adriani kelas dasar berimbunan

Abstraksi 2

penelitian mengimplementasikan algoritma *stemming* nazief adrian mencari kesamaan judul penulisan berbahasa indonesia. mencari kesamaan judul penulisan membantu mahasiswa dosen menentukan judul penulisan referensi pembuatan penulisan menghindari tindakan *plagiarisme* judul penulisan penelitian membangun aplikasi data judul penulisan *database* mencari kesamaan judul penulisan judul penulisan *database*

3. Tokenizing

Dalam proses *Tokenizing* dokumen yang akan dibaca berupa teks yang kemudian akan dilakukan proses pemotongan *string* input berdasarkan setiap kata yang menyusunnya. Secara umum setiap kata akan terpisahkan dengan kata yang lain oleh karakter spasi, sehingga proses tokenisasi mengandalkan karakter spasi pada dokumen teks tersebut untuk melakukan pemisahan kata. *Tokenizing* merupakan langkah awal dalam penguraian deskripsi yang berupa suatu kalimat menjadi potongan kata. Berikut proses *tokenizing* pada kalimat-kalimat yang terdapat didalam abstrak.

Tabel 2. dua dokumen abtraksi setelah selesai tahap *Tokenizing*

Abstraksi 1					
algoritma	Nazief	Adrian	algoritma	mengubah	Memiliki
Sufiks	Prefix	Konfiks	bentuk	dasar	Algoritma
Nazief	Adriani	Penelitian	algoritma	pendukung	Proses
penentuan	Dasar	berimbunan	kalimat	menentukan	Dasar
Berimbunan	Menghilangkan	Imbunan	afiks	awalan	Prefix
Sisipan	Infiks	Akhiran	sufiks	awalan	Akhiran
akhiran	berimbunan	Algoritma	stemming	nazief	Adriani
Kelas	Dasar	berimbunan			

Abstraksi 2					
penelitian	mengimplementasika	algoritma	<i>stemming</i>	nazief	Adrian
mencari	kesamaan	penulisan	berbahasa	indonesia	Mencari
kesamaan	Judul	penulisan	membantu	mahasiswa	Dosen
menentukan	Judul	penulisan	referensi	pembuatan	Penulisan

Menghindari	tindakan	<i>plagiarisme</i>	judul	penulisan	Penelitian
membangun	aplikasi	Data	judul	penulisan	<i>Database</i>
mencari	kesamaan	Judul	penulisan	judul	Penulisan
<i>database</i>					

4. Stemming

Proses *Stemming* bertujuan untuk menghasilkan bentuk dasar dari sebuah *term* atau kata. *Term* yang sudah melewati proses pembuangan *stopword* yang akan dijadikan sebagai input dalam proses ini. Algoritma yang dipakai dalam proses *stemming* ini adalah Algoritma Nazief & Andriani. Algoritma ini digunakan karena algoritma ini memiliki keakuratan yang baik dalam proses *stemming* Bahasa Indonesia. Tahap *stemming* merupakan tahap mengubah kata imbuhan menjadi kata dasar atau dengan kata lain adalah mengembalikan kata ke dalam bentuk aslinya.

Tabel 3. dua dokumen abstraksi yang menghasilkan stemming

Abstraksi 1					
algoritme	Nazief	adriani	algoritme	ubah	Milik
sufiks	Prefix	konfiks	bentuk	dasar	Algoritme
nazief	adriani	teliti	algoritme	dukung	Proses
Tentu	Dasar	imbuh	kalimat	tentu	Dasar
imbuh	Hilang	imbuh	afiks	awal	Prefix
Sisip	Infiks	akhir	sufiks	awal	Akhir
Akhir	Imbuh	algoritme	stemming	nazief	Adriani
Kelas	Dasar	imbuh			

Abstraksi 2					
Teliti	implementasi	algoritme	<i>stemming</i>	nazief	Adrian
Cari	Sama	tulis	bahasa	indonesia	Cari
Sama	Judul	tulis	bantu	mahasiswa	Dosen
Tentu	Judul	tulis	referensi	buat	Tulis
hindar	Tindak	<i>plagiarisme</i>	judul	tulis	Teliti
bangun	Aplikasi	data	judul	tulis	<i>Database</i>
Cari	Sama	judul	tulis	judul	Ulis
<i>database</i>					

Setelah tahapan pada proses *stemming* selesai dilakukan. Tahap selanjutnya yaitu untuk mengetahui tingkat kemiripan kata dilakukan perhitungan pada setiap pembobotan *term*.

3.1.2 Menentukan bobot untuk setiap term.

Proses pembobotan kata adalah proses pemberian nilai atau bobot ke sebuah kata berdasarkan kemunculannya pada suatu dokumen teks. Pada proses sebelumnya atau proses *Text Processing* akan didapat kumpulan kata atau *term* yang akan dipresentasikan kedalam sebuah *terms*. Tujuan penghitungan *IDF* adalah untuk mencari kata-kata yang benar-benar mempresentasikan dokumen teks pada suatu koleksi. Metode ini merupakan metode paling baik dalam perolehan informasi. Adapun rumus dari *TF-IDF* dapat dilihat pada persamaan berikut.

$$tfidf(i,j) = tf(i,j) \times \log\left(\frac{N}{df(j)}\right) \tag{1}$$

Dengan $tf(i,j)$ adalah frekuensi kemunculan *term* j pada dokumen teks di $\in D^*$, dimana $i = 1,2,3,\dots,N$, $df(j)$ adalah frekuensi dokumen yang mengandung *term* j dari semua koleksi dokumen, dan N adalah jumlah seluruh dokumen yang ada di koleksi dokumen. Berdasarkan rumus diatas berapapun besarnya nilai $tf(i,j)$, apabila $N = df(j)$ maka akan didapatkan hasil 0 (nol) untuk perhitungan *idf*. Untuk itu dapat ditambahkan nilai 1 pada sisi *idf*, sehingga perhitungan untuk pembobotan dapat dilihat padarumus (2) berikut :

$$tfidf(i,j) = tf(i,j) \times \left(\log\left(\frac{N}{df(j)}\right) + 1\right) \tag{2}$$

pada penelitian ini proses pembobotan dilakukan untuk menghitung nilai pada tingkat kemiripan kata dengan menjumlahkan bobot pada setiap kata yang sama. Hasil pembobotan kata dapat dilihat pada tabel 4. di bawah ini.

Tabel 4. hasil pembobotan kata pada dua dokumen abstraksi

Abstraksi 1						
Term	tf	df	N	$\log(n/df)+1=idf$	$tf*idf=tf idf$	Tf idf
algoritme	5	2	3	$\log(3/2)+1= 1.1760$	$5*1.1760= 5.88$	5.88
sufiks	2	1	3	$\log(3/1)+1= 1.4771$	$2*1.4771= 2.9542$	2.9542
nazief	3	1	3	$\log(3/1)+1= 1.4771$	$3*1.4771= 4.4313$	4.4313

Abstraksi 1						
Term	tf	df	N	$\log(n/df)+1=idf$	$tf*idf=tf idf$	Tf idf
Tentu	2	1	3	$\log(3/1)+1= 1.4771$	$2*1.4771= 2.9542$	2.9542
imbuh	5	4	3	$\log(3/4)+1= 0.8750$	$5*0.8750= 4.375$	4.375
sisip	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
akhir	3	2	3	$\log(3/2)+1= 1.1760$	$3*1.1760= 3.528$	3.528
kelas	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
nazief	3	1	3	$\log(3/1)+1= 1.4771$	$3*1.4771= 4.4313$	4.4313
prefix	2	1	3	$\log(3/1)+1= 1.4771$	$2*1.4771= 2.9542$	2.9542
adriani	3	1	3	$\log(3/1)+1= 1.4771$	$3*1.4771= 4.4313$	4.4313
dasar	4	1	3	$\log(3/1)+1= 1.4771$	$4*1.4771= 5.9084$	5.9084
hilang	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
infiks	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
imbuh	5	4	3	$\log(3/4)+1= 0.8750$	$5*0.8750= 4.375$	4.375
dasar	4	2	3	$\log(3/2)+1= 1.1760$	$4*1.1760= 4.704$	4.704
adriani	3	1	3	$\log(3/1)+1= 1.4771$	$3*1.4771= 4.4313$	4.4313
konfiks	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771=1.4771$	1.4771
teliti	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
imbuh	5	4	3	$\log(3/4)+1= 0.8750$	$5*0.8750= 4.375$	4.375
imbuh	5	4	3	$\log(3/4)+1=0.8750$	$5*0.8750= 4.375$	4.375
akhir	3	3	3	$\log(3/3)+1= 1$	$3*1= 3$	3
algoritme	5	2	3	$\log(3/2)+1= 1.1760$	$5*1.1760= 5.88$	5.88
imbuh	5	4	3	$\log(3/4)+1=0.8750$	$5*0.8750= 4.375$	4.375
algoritme	5	2	3	$\log(3/2)+1= 1.1760$	$5*1.1760= 5.88$	5.88

Tabel 5. Hasil Pembobotan Kata Pada Dua Dokumen Abstraksi Lanjutan

Term	tf	df	N	$\log(n/df)+1=idf$	$tf*idf=tf idf$	Tf idf
bentuk	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
algoritme	5	2	3	$\log(3/2)+1= 1.1760$	$5*1.1760= 5.88$	5.88
kalimat	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
Afiks	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
sufiks	2	1	3	$\log(3/1)+1= 1.4771$	$2*1.4771= 2.9542$	2.9542
stemming	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
Ubah	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
Dasar	2	1	3	$\log(3/1)+1= 1.4771$	$2*1.4771= 2.9542$	2.9542
dukung	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
Tentu	2	1	3	$\log(3/1)+1= 1.4771$	$2*1.4771= 2.9542$	2.9542
Awal	2	2	3	$\log(3/2)+1= 1.1760$	$2*1.1760= 2.352$	2.352
Awal	2	2	3	$\log(3/2)+1= 1.1760$	$2*1.1760= 2.352$	2.352
nazief	3	2	3	$\log(3/2)+1= 1.1760$	$3*1.1760= 3.528$	3.528
Milik	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
algoritme	5	2	3	$\log(3/2)+1= 1.1760$	$5*1.1760= 5.88$	5.88
proses	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
Dasar	2	1	3	$\log(3/1)+1= 1.4771$	$2*1.4771= 2.9542$	2.9542
prefix	2	1	3	$\log(3/1)+1= 1.4771$	$2*1.4771= 2.9542$	2.9542
Akhir	3	2	3	$\log(3/2)+1= 1.1760$	$3*1.1760= 3.528$	3.528
adriani	3	1	3	$\log(3/1)+1= 1.4771$	$3*1.4771= 4.4313$	4.4313

Keterangan :

Term : merupakan kata-kata.

Tf : bayaknya kata pada suatu dokumen/artikel.

Df : total artikel pada suatu kata.

N : bayaknya dokumen/artikel.

Idf : hasil perhitungan.

Tabel 6. hasil pembobotan kata pada dua dokumen abstraksi

Abstraksi 2						
Term	Tf	df	n	$\log(n/df)+1=idf$	$tf*idf=tf idf$	Tf idf
teliti	2	2	3	$\log(3/2)+1= 1.1760$	$2* 1.1760= 2.352$	2.352
cari	3	3	3	$\log(3/3)+1= 1$	$3*1= 3$	3
sama	3	3	3	$\log(3/3)+1= 1$	$3*1= 3$	3
tentu	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771

Abstraksi 2						
Term	Tf	df	n	$\log(n/df)+1=idf$	$tf*idf=tf idf$	Tf idf
hindar	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771=1.4771$	1.4771
bangun	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
cari	3	3	3	$\log(3/3)+1= 1$	$3*1= 3$	3
database	2	1	3	$\log(3/1)+1= 1.4771$	$2*1.4771= 2.9542$	2.9542
implementasi	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
sama	3	3	3	$\log(3/3)+1= 1$	$3*1= 3$	3
judul	6	3	3	$\log(3/3)+1= 1$	$6*1= 6$	6
judul	6	3	3	$\log(3/3)+1= 1$	$6*1= 6$	6
tindak	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
aplikasi	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
sama	3	3	3	$\log(3/3)+1= 1$	$3*1= 3$	3
algoritme	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
tulis	6	3	3	$\log(3/3)+1= 1$	$6*1= 6$	6
tulis	6	3	3	$\log(3/3)+1= 1$	$6*1= 6$	6
tulis	6	3	3	$\log(3/3)+1= 1$	$6*1= 6$	6
plagiarisme	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
data	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
judul	6	3	3	$\log(3/3)+1= 1$	$6*1= 6$	6

Tabel 7. hasil pembobotan kata pada dua dokumen abstraksi lanjutan

Term	tf	df	n	$\log(n/df)+1=idf$	$tf*idf=tf idf$	Tf idf
stemming	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
bahasa	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
Bantu	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
referensi	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
Judul	6	3	3	$\log(3/3)+1= 1$	$6*1= 6$	6
Judul	6	3	3	$\log(3/3)+1= 1$	$6*1= 6$	6
Tulis	6	3	3	$\log(3/3)+1= 1$	$6*1= 6$	6
nazief	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
indonesia	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
mahasiswa	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
Buat	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
Tulis	6	3	3	$\log(3/3)+1= 1$	$6*1= 6$	6
Tulis	6	3	3	$\log(3/3)+1= 1$	$6*1= 6$	6
Judul	6	3	3	$\log(3/3)+1= 1$	$6*1= 6$	6
adrian	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
Cari	3	3	3	$\log(3/3)+1= 1$	$3*1= 3$	3
dosen	1	1	3	$\log(3/1)+1= 1.4771$	$1*1.4771= 1.4771$	1.4771
Tulis	6	3	3	$\log(3/3)+1= 1$	$6*1= 6$	6
Teliti	2	1	3	$\log(3/1)+1= 1.4771$	$2*1.4771= 2.9542$	2.9542
database	2	1	3	$\log(3/1)+1= 1.4771$	$2*1.4771= 2.9542$	2.9542
Tulis	6	3	3	$\log(3/3)+1= 1$	$6*1= 6$	6

Setelah proses untuk nilai pembobotan pada setiap term diketahui maka untuk mengetahui hasil dari nilai keseluruhan dari dokumen antraksi 1 dengan dokumen abstraksi 2 yaitu sebagai berikut :

Tabel 8. Dua Dokumen Setelah Melalui Proses Pembobotan

abstraksi 1					
5.88	2.9542	1.4771	1.4771	2.352	
2.9542	4.4313	4.375	1.4771	3.528	
4.4313	5.9084	4.375	2.9542	1.4771	
2.9542	1.4771	3	1.4771	5.88	
4.375	1.4771	5.88	1.4771	1.4771	
1.4771	4.375	4.375	2.9542	2.9542	
3.528	4.704	5.88	1.4771	2.9542	
1.4771	4.4313	1.4771	2.9542	3.528	
4.4313	1.4771	5.88	2.352	4.4313	
Jumlah hash = 42					
abstraksi 2					

2.352	3	6	6	1.4771
3	6	1.4771	6	3
3	6	1.4771	1.4771	1.4771
1.4771	1.4771	6	1.4771	6
1.4771	1.4771	1.4771	1.4771	2.9542
1.4771	3	1.4771	1.4771	2.9542
3	1.4771	1.4771	6	6
2.9542	6	1.4771	6	
1.4771	6	6	6	
Jumlah hash = 42				
abstraksi 1 dan abstraksi 2				
5.88	2.9542	1.4771	1.4771	2.352
2.9542	4.4313	4.375	1.4771	3.528

Lanjutan Tabel 8. dua dokumen setelah melalui proses pembobotan

abstraksi 1 dan abstraksi 2				
4.4313	5.9084	4.375	2.9542	1.4771
2.9542	1.4771	3	1.4771	5.88
4.375	1.4771	5.88	1.4771	1.4771
1.4771	4.375	4.375	2.9542	2.9542
3.528	4.704	5.88	1.4771	2.9542
1.4771	4.4313	1.4771	2.9542	3.528
4.4313	1.4771	5.88	2.352	4.4313
2.352	3	6	6	1.4771
3	6	1.4771	6	3
3	6	1.4771	1.4771	1.4771
1.4771	1.4771	6	1.4771	6
1.4771	1.4771	1.4771	1.4771	2.9542
1.4771	3	1.4771	1.4771	2.9542
3	1.4771	1.4771	6	6
2.9542	6	1.4771	6	1.4771
6	6	6	Jumlah hash = 55	

Nilai *hash* merupakan nilai yang didapat setelah proses mengubah setiap string menjadi angka. Keterangan warna hijau merupakan nilai yang tidak memiliki kesamaan terhadap nilai yang lainnya atau bisa disebut juga sebagai nilai tunggal. Dari tabel 8. diatas dapat diketahui jumlah *hash* yang sama pada kedua abstraksi tersebut, setelah jumlah *hash* pada abtraksi 1 dan jumlah *hash* pada abstraksi 2 deketahui maka proses selanjutnya adalah menghitung nilai *similarity*. Untuk melakukan uji *similarity* pada kedua abstraksi diatas dapat dilakukan dengan cara menggunakan indeks *sørensen* dengan persamaan sebagai berikut. Persamaan tersebut dapat di lihat pada rumus dibawah ini.

$$S = \frac{2n_t}{n_x+n_y} = \frac{2*55}{42+42} = 1,30952 = 0,01\%$$

Diketahui uji *similarity* dari dua abstraksi diatas yaitu menghasilkan nilai sebesar 1,30952 yang berarti kedua dokumen abstraksi tersebut memiliki kemiripan kata yang besar 0,01% sehingga tidak dapat dikatakan *plagiarism*.

4. KESIMPULAN

Berdasarkan hasil studi literatur, analisis, perancangan, implementasi, dan pengujian sistem ini, maka kesimpulan yang didapat adalah Mengidentifikasi kemiripan karya tulis ilmiah dengan algoritma algoritma *stemming Nazief-Adriani*. Algoritma *stemming Nazief-Adriani* dapat diterapkan dengan menyusun kembali kata yang telah mengalami *stemming* terhadap karya tulis ilmiah.

REFERENCES

- [1] Azhar F, E. (2014). Aplikasi Pendeteksi Kemiripan Pada Dokumen Teks Menggunakan Algoritma Nazief & Adriani dan Metode Cosine Similarity. Jurnal Teknologi Informasi .
- [2] Manning, C. D. (2009). An Introduction to Information Retrieval, Cambridge University.
- [3] Alsamadi, Izzat. Saleh, Zakaria Issa, Documents Similarities Algorithms for Research Papers Authenticity. IT Faculty, Yarmouk University. Jordan.
- [4] R. B. K. Susilo, Eko. Madyo, dasar-dasar pendidikan. semarang: Effhar Publishing, 1998.
- [5] et al. Knuth, Donald. E, Fundamental of Algorithm, Volume 1 of The Art of Computer Programming. Addison-Wesley.
- [6] Liddy, E.D.Natural Language Processing. New York: Syracuse University. 2001.

- [7] M. Nazief, B. dan Adriani, *Confix Stripping: Approach to Stemming Algorithm for Bahasa Indonesia*. Jakarta: University of Indonesia, 1996.
- [8] S. Breitinger, Corinna; Gipp, Bela; Langer, ““Researchpaper recommender systems: a literature survey’.” *Int. J. Digit. Libr.*, 2015.
- [9] budi sutedjo dharma Oetomo, *Perancangan dan pengembangan sistem informasi*. Yogyakarta: Andi, 2002.
- [10] A. Kadir, *Pengertian MySQL*. Tersedia dalam : *Buku Pintar Programmer Pemula PHP*. Yogyakarta: Mediakom, 2013.
- [11] A. Kadir, *pemrograman database mysql untuk pemula*. Yogyakarta: mediakom, 2013.