PENDETEKSIAN DOKUMEN PLAGIARISME DENGAN MENGGUNAKAN METODE *WEIGHT TREE*

Nurdin[1], Rizal[2], Rizwan[3]

[1,2,3]Program Studi Teknik Informatika
Fakultas Teknik
Universitas Malikussaleh
Email : nurdin@unimal.ac.id, rizal@unimal.ac.id

(Naskah masuk: 11 Januari 2019, diterima untuk diterbitkan: 28 Februari 2019)

ABSTRAK

Sistem pengelolaan dokumen plagiarisme masih ada yang dilakukan secara manual yaitu dengan mengecek satu persatu sehingga membutuhkan waktu yang lama dan kurang efektif. Salah satu algoritma yang dapat digunakan untuk pendeteksian dokumen plagiarisme adalah algoritma *Weight Tree* yaitu sebuah metode untuk melakukan klasifikasi kemiripan dokumen berdasarkan bobot dari dokumen. Tujuan penelitian ini adalah untuk membangun sebuah sistem pendeteksian kemiripan dari dua dokumen teks yang berbeda untuk jenis dokumen teks berbahasa indonesia dengan format file dokumen yaitu: doc, docx, pdf, rtf. Tahapan yang dilakukan pada penelitian ini terdiri dari pengumpulan data, perancangan sistem, pembuatan aplikasi dan pengujian terhadap aplikasi. Hasil pengujian sistem dapat dikategorikan sebagai sistem pendeteksian atau pengetesan kemiripan dokumen. Pada pengujian sistem ini, penulis yang mengkategorikan dokumen tersebut sebagai dokumen plagiat berdasarkan persentase kemiripan. Nilai rata-rata persentase kemiripan dalam pengujian sistem ini adalah 71,60%. Sistem yang di bangun ini berhasil dengan tingkat keakuratan mencapai 90%. Algoritma *Weight Tree* yang diterapkan pada sistem ini terbukti mampu mengidentifikasi dengan baik kemiripan dokumen *plagiarism*e.

Kata Kunci: Deteksi, Dokumen, Weight Tree, Pengujian

ABSTRACT

The plagiarism document management system still exists manually, by checking one by one so that it takes a long time and is less effective. One algorithm that can be used to detect plagiarism documents is the Weight Tree algorithm, which is a method for classifying document similarity based on the weight of the document. The purpose of this study is to build a system of detecting similarities from two different text documents for Indonesian language text documents with document file formats, namely: doc, docx, pdf, rtf. The stages carried out in this study consisted of data collection, system design, application creation and testing of applications. The results of system testing can be categorized as a system of detecting or testing document similarities. In testing this system, the authors categorize these documents as plagiarism documents based on the percentage of similarities. The average value of the percentage of similarities in testing

this system is 71.60%. This built system works with an accuracy of 90%. The Weight Tree algorithm applied to this system has proven to be able to properly identify the similarity of plagiarism documents.

Keywords: Detection, Documents, Weight Tree, Testing

PENDAHULUAN

Perkembangan teknologi informasi yang semakin pesat membawa dampak baik dan buruk bagi kehidupan manusia, dampak buruk yang ditimbulkan salah satunya adalah masalah plagiarisme. Plagiarisme atau plagiat merupakan penjiplakan atau pengambilan karangan, pendapat dan sebagainya dari orang lain dan menjadikan seolah-olah karangan dan pendapat sendiri (Stepchyshyn & Nelson, 2007).

Dalam beberapa tahun terkahir ini dunia pendidikan pernah dikejutkan oleh munculnya pemberitaan tentang masalah yang terkait dengan plagiarisme terhadap sejumlah karya ilmiah atau karya tulis seperti makalah, tugas akhir, tesis, disertasi dan artikel-artikel yang ada pada jurnal ilmiah. Plagiarisme telah menciptakan iklim yang buruk, khususnya terhadap dunia pendidikan. Tindakan ini dapat membunuh ide dan gagasan serta menurunkan tingkat kreativitas seseorang (Afdhal et al, 2014).

Pendeteksian plagiat dapat dilakukan dengan cara manual menggunakan bantuan manusia atau dengan cara otomatis yaitu dengan menggunakan sistem komputer sebagai pendukung pengambilan keputusan (Nurdin & Miranda, 2015), saat ini pendeteksian secara manual merupakan cara yang akurat dalam mendeteksi plagiat, namun cara ini mempunyai kelemahan yaitu banyak menghabiskan waktu dan tenaga serta tidak konsisten karena dipengaruhi faktor emosional manusia. Oleh karena itu, para akademisi berusaha mengembangkan sebuah sistem komputer untuk mendeteksi plagiat dengan tingkat akurasi yang sangat baik.

Pencarian *full text* adalah tipe pencarian dokumen yang dilakukan komputer dengan menelusuri keseluruhan isi sebuah dokumen (Beall, 2008). Cara

kerjanya adalah mencari dokumen yang mengandung kata *query* pengguna. Hasil *query* diurutkan sesuai dengan tingkat kandungan kata dan umumnya frekuensi kandungan kata diurutkan dari tinggi ke rendah. Penelusuran dokumen hanya menggunakan operasi dasar pencocokan kata (*string matching*) tanpa tambahan operasi algoritma lainnya (Yates & Neto, 1999). Dengan metode ini pengguna mengoperasikan dengan mudah, cukup memasukkan kata kunci yang diinginkan. Tampilan antarmuka relatif lebih sederhana tanpa banyak pilihan.

Salah satu metode yang dapat digunakan untuk merancang sebuah mesin pencarian tersebut adalah *weighted tree similarity* yang akan diterapkan untuk melakukan pencarian. Algoritma *weighted tree similarity* memiliki keunikan karena memiliki representasi tree yang berbeda dengan cabang berlabel serta berbobot (Yang et al, 2005). Cabang yang berlabel memberikan pemahaman lebih kepada label nodenya. Begitu pula bobot cabang memungkinkan memberikan tingkat kecenderungan kepada cabang tertentu lebih dari yang lain.

Ada beberapa penelitian yang dilakukan sebelumnya seperti penelitian (Sarno & Rahutomo, 2008) menjelaskan pencarian semantik dengan algoritma *weighted tree similarity* dapat diterapkan pada pencarian dokumen dan pencarian halaman web. Menurut penelitian (Ariyani et al, 2016) membahas masalah pendeteksian kemiripan dokumen teks menggunakan algoritma *lavenshtein distance* dengan mengambil data dari artikel/berita. Algoritma *lavenshtein distance* juga berhasil diterapkan untuk pencarian *E-Boarding House* untuk mempermudah mahasiswa mendapatkan tempat tinggal (Iqbal & Nurdin, 2017).

Salah satu penelitian yang telah dilakukan oleh (Nurdin & Munthoha, 2017), untuk mendeteksi plagiat dari penelitian ini hanya memembahas masalah sistem pendeteksian kemiripan judul skripsi menggunakan algoritma *winnowing*, dan hasilnya sistem ini mampu melakukan proses penentuan persentase pendeteksian kemiripan judul skripsi menjadi lebih cepat dan akurat. Kekurangan dari penelitian ini hanya membahas masalah pendeteksian judul skripsi, tidak membahas masalah pendeteksian dokumen plagiarisme atau kemiripan dokumen.

Dari beberapa penelitian sebelumnya belum membahas masalah bagaimana penerapan metode *weight tree* dalam mencari kemiripan pada dua dokumen teks yang berbeda, sehingga penulis tertarik untuk melanjutkan penelitian yang sudah dilakukan sebelumnya masalah sistem pendeteksian kemiripan judul skripsi (Nurdin & Munthoha, 2017) untuk dikembangkan menjadi sistem pendeteksian dokumen plagiarisme dengan metode *weight tree.* Tujuan penulis membuat penelitian ini adalah untuk membangun sebuah sistem pendeteksian kemiripan dari dua dokumen teks yang berbeda untuk mencegah terjadinyan plagiarisme, dengan membatasi pada jenis dokumen teks berbahasa indonesia dengan format file dokumen yaitu: doc, docx, pdf, rtf.

METODE PENELITIAN

1. Tahapan Penelitian

a. Pengumpulan Data

Untuk penelitian ini, penulis mengumpulkan data berupa data file dokumen dalam bentuk .rtf, .pdf, dan .doc/.docx dengan ukuran yang berbeda-beda. Dokumen ini akan digunakan untuk proses pelatihan dan pengujian setelah sistem dibuat.

b. Analisa Sistem

Pada tahap ini akan dilakukan Preprocessing terhadap dokumen text yang diinput sehingga hanya kata dasar saja yang akan diambil untuk proses *Weighted Tree,* selanjutnya memasukkan dokumen yang diuji adalah dokumen 1 dan dokumen 2 yang berupa berupa document teks, dengan ekstensi .rtf, .pdf, dan .doc/.docx dengan ukuran yang berbeda-beda. Kemudian akan diproses preprocessing, tahap selanjutnya adalah melakukan pembobotan kata dengan menggunakan algoritma *Weighted Tree* berdasarkan data dari hasil preprocessing. Langkah terakhir adalah melihat hasil kemiripan dengan 2 dokumen yang telah diinputkan. Cara untuk mengetahui dokumen tersebut plagiat adalah dengan melihat hasil persentase kemiripan

yang dihasilkan pada 2 dokumen text tersebut dengan menggunakan *cosine similarity* berbentuk presentase. Semakin tinggi nilai presentase akan semakin dekat dokumen tersebut plagiat.

c. Perancangan dan Pembuatan Aplikasi

Pada tahap ini penulis melakukan perancangan dan pembuatan aplikasi sistem pendeteksian dokumen *Plagiarisme* dengan menggunakan metode *Weight Tree* berbasis web yang dijalankan secara online dengan menggunakan bahasa pemrogram PHP dan database MySQL.

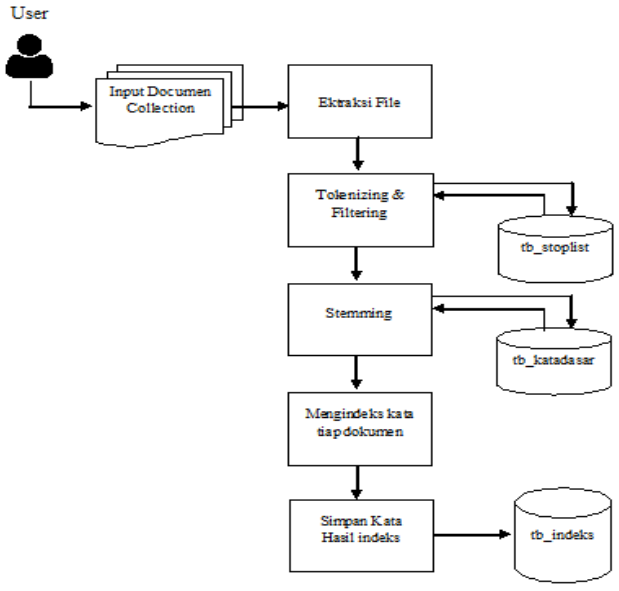
d. Pengujian terhadap Aplikasi

Melakukan pengujian terhadap program yang telah dibuat dengan melakukan beberapa tes terhadap 2 dokumen terutama pada dokumen yang di curigai mengandung unsur *plagiarisme*. Hasil yang didapatkan adalah persentase kemiripan antara dokumen satu dengan dokumen lainnya.

2. Skema Sistem

Untuk membangun sistem tersebut tahapan yang dilakukan yaitu proses pengumpulan koleksi dokumen ke database. Berikut ini merupakan gambaran skema untuk proses tersebut.

a. Skema Input Koleksi Dokumen



Gambar 1. Skema proses input koleksi dokumen

Gambar 1. Menjelaskan, *user* menginput beberapa dokumen, kemudian melakukan proses ekstraksi file selanjutnya dilakukan proses *tokenizing*, *filtering* dan *stemming* yang bertujuan untuk mendapatkan kata dasar dari setiap kata yang sebelumnya terdapat beberapa imbuhan, hal tersebut dilakukan agar dapat menghindari penyimpanan indeks kata yang berlebihan dan hasilnya disimpan dalam database, setelah selesai proses tersebut baru mengindeks kata tiap dokumen, hasil indeks disimpan dalam database tb_indeks.

HASIL DAN PEMBAHASAN

1. Menghitung Pembobotan Kata dengan *Weight Tree*

Pada tahap ini kata pada dokumen yang telah dihitung frekuensinya akan dicari nilai bobot berdasarkan *leafnode* dari *root* masing-masing dokumen. Berikut merupakan contoh perhitungan *weight tree* pada dokumen teks:

Hasil *preprocessing* dokumen 1

abstrak teliti selesai tentu rangking pilih buat model angket guru guru sekolah medan tulis metode profile matching metode ahp selesai kriteria teliti mimpin tanggung intelektual disiplin metode profile matching senjang kriteria calon kriteria profil jabat metode ahp tahap hasil pilih dasar hasil teliti ambil simpul nilai consistens ratio mana calon pilih nilai consistens ratio calon pilik nilai consistens ratio calon pilih nilai consistens ratio ambil simpul calon nilai consistens ratio layak kepala sekolah kunci ahp profile matching consistens ratio rangking model selection using profile matching and ahp abstract this research used solve problems determining ranking the election the manufacture profile matching thod used see the gap tween the criteria each each candidate profiles and positions criteria ahp the last stage the election results based this research concluded from consistens value ratio where candidate has value consistens ratio candidate has value and the ratio candidate consistens consistens ratio has value concluded consistens candidate ratio and value worthy principal ahp profile matching consistens rasio rangking.

Hasil *preprocessing* dokumen 2

abstrak teliti selesai tentu rangking pilih buat model angket guru guru sekolah medan tulis metode profile matching metode ahp selesai kriteria teliti mimpin tanggung intelektual disiplin metode profile matching senjang kriteria calon kriteria profil jabat metode ahp tahap hasil pilih dasar hasil teliti ambil simpul nilai consistens ratio mana calon pilih nilai consistens ratio calon pilih nilai consistens ratio calon pilih nilai consistens ratio ambil simpul calon nilai consistens ratio layak kepala sekolah kunci ahp profile matching consistens ratio rangking.

Setelah *preprocessing* dilakukan maka selanjutnya yaitu tahap pembobotan masing-masing kata dengan menggunakan rumus sebagai berikut:

$$W = TF/TF_{total} \quad (1)$$

Dengan TF_{total} adalah jumlah total TF (*Term Frequency*).

Tabel 1. Pembobotan kata dokumen 1

Kata	*Frekuensi*	*Bobot*
Ahp	6	0.0504202
Ambil	2	0.0168067
And	3	0.0252101
Calon	5	0.0420168
Candidate	5	0.0420168
Concluded	2	0.0168067
Consistens	12	0.10084
Criteria	2	0.0168067
Each	2	0.0168067
Election	2	0.0168067
Guru	2	0.0168067
Has	3	0.0252101
Hasil	2	0.0168067
Kriteria	3	0.0252101
Matching	6	0.0504202
Metode	4	0.0336134
Model	2	0.0168067
Nilai	5	0.0420168
Pilih	4	0.0336134
Profile	6	0.0504202
Rangking	3	0.0252101
Ratio	11	0.092437

Research	2	0.0168067
Sekolah	2	0.0168067
Selesai	2	0.0168067
Simpul	2	0.0168067
Teliti	3	0.0252101
The	7	0.0588235
This	2	0.0168067
Used	2	0.0168067
Value	5	0.0420168
Total	119	1

Tabel 2. Pembobotan kata dokumen 2

Kata	*Frekuensi*	*Bobot*
Ahp	3	0.05
Ambil	2	0.0333333
Calon	5	0.0833333
Consistens	6	0.1
Guru	2	0.0333333
Hasil	2	0.0333333
Kriteria	3	0.05
Matching	3	0.05
Metode	4	0.0666667
Nilai	5	0.0833333
Pilih	5	0.0833333
Profile	3	0.05
Rangking	2	0.0333333
Ratio	6	0.1
Sekolah	2	0.0333333
Selesai	2	0.0333333

Simpul	2	0.0333333
Teliti	3	0.05
Total	60	1

Menghitung Kemiripan dengan *Cosine Similarity*

Setelah proses pembobotan selesai maka selanjutnya adalah mnghitung tingkat kemiripan antara 2 dokumen tersebut dengan menggunakan rumus berikut:

$$\text{Cosine}(q,d) = \frac{\sum_{k=1}^{m} \text{wqk x wdk}}{\sqrt{\sum_{k=1}^{m} (wqk)^2} \cdot \sqrt{\sum_{k=1}^{t} (wdk)^2}} \qquad (2)$$

Dengan:

Wij : bobot term j terhadap dokumen i

q : vektor dokumen 1

d : vektor dokumen 2

m : dimensi vektor 1 dan 2

Berdasarkan nilai bobot masing-masing dokumen pada Tabel 1 dan Tabel 2, maka untuk mencari nilai variabel pada rumus *cosine similarity* dapat dilihat pada Tabel 3 dibawah ini :

Tabel 3. Perhitungan nilai variabel pada rumus *cosine similarity*

Kata	*Bobot*		*q • d*	*(q)²*	*(d)²*
	Q	*D*			
Ahp	0.0504202	0.05	0.00252101	0.002542197	0.0025
Ambil	0.0168067	0.0333333	0.000560223	0.000282465	0.001111109
And	0.0252101	0	0	0.000635549	0
Calon	0.0420168	0.0833333	0.003501399	0.001765411	0.006944439
Candidate	0.0420168	0	0	0.001765411	0
concluded	0.0168067	0	0	0.000282465	0
consistens	0.10084	0.1	0.010084	0.010168706	0.01

Criteria	0.0168067	0	0	0.000282465	0
Each	0.0168067	0	0	0.000282465	0
Election	0.0168067	0	0	0.000282465	0
Guru	0.0168067	0.0333333	0.000560223	0.000282465	0.001111109
Has	0.0252101	0	0	0.000635549	0
Hasil	0.0168067	0.0333333	0.000560223	0.000282465	0.001111109
Kriteria	0.0252101	0.05	0.001260505	0.000635549	0.0025
Matching	0.0504202	0.05	0.00252101	0.002542197	0.0025
Metode	0.0336134	0.0666667	0.002240894	0.001129861	0.004444449
Model	0.0168067	0	0	0.000282465	0
Nilai	0.0420168	0.0833333	0.003501399	0.001765411	0.006944439
Pilih	0.0336134	0.0833333	0.002801116	0.001129861	0.006944439
Profile	0.0504202	0.05	0.00252101	0.002542197	0.0025
Rangking	0.0252101	0.0333333	0.000840336	0.000635549	0.001111109
Ratio	0.092437	0.1	0.0092437	0.008544599	0.01
Research	0.0168067	0	0	0.000282465	0
Sekolah	0.0168067	0.0333333	0.000560223	0.000282465	0.001111109
Selesai	0.0168067	0.0333333	0.000560223	0.000282465	0.001111109
Simpul	0.0168067	0.0333333	0.000560223	0.000282465	0.001111109
Teliti	0.0252101	0.05	0.001260505	0.000635549	0.0025
The	0.0588235	0	0	0.003460204	0
This	0.0168067	0	0	0.000282465	0
Used	0.0168067	0	0	0.000282465	0
Value	0.0420168	0	0	0.001765411	0
Total			0.0457	0.0463	0.0656

Selanjutnya menghitung kemiripan dokumen 1 dengan dokumen 2 sebagai berikut:

$$Cos(q,d) = \frac{0.0457}{\sqrt{0.0463} * \sqrt{0.0656}} = 0.8292 * 100 = 82.92\ \%$$

2. Implementasi Sistem

Untuk memudahkan user menggunakan sistem pendeteksian plagiarisme dengan metode *weighted tree* ini, maka tampilan layar antarmuka aplikasi ini dibuat sederhana namun tetap menarik untuk digunakan.

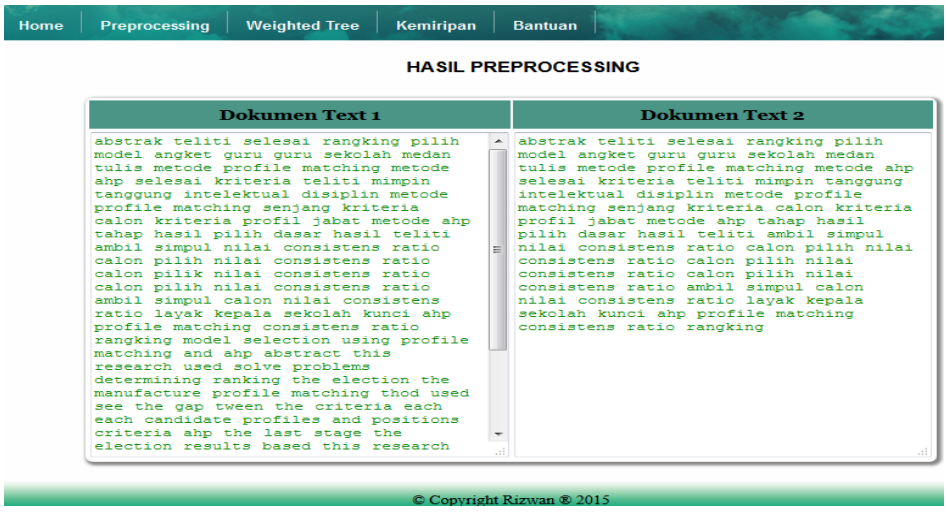
a. Menu Preprocessing

Form menu preprocessing merupakan tahap dokumen text yang diinput sehingga hanya kata dasar saja yang akan diambil untuk proses *weighted tree*, berikut ini tampilan form preprocessing untuk halaman user.

Gambar 3. Menu halaman preprocessing

b. Menu Hasil Preprocessing

Form menu preprocessing merupakan tahap dokumen text yang diinput sehingga hanya kata dasar saja yang akan diambil untuk proses *weighted tree*, berikut ini tampilan form hasil preprocessing.



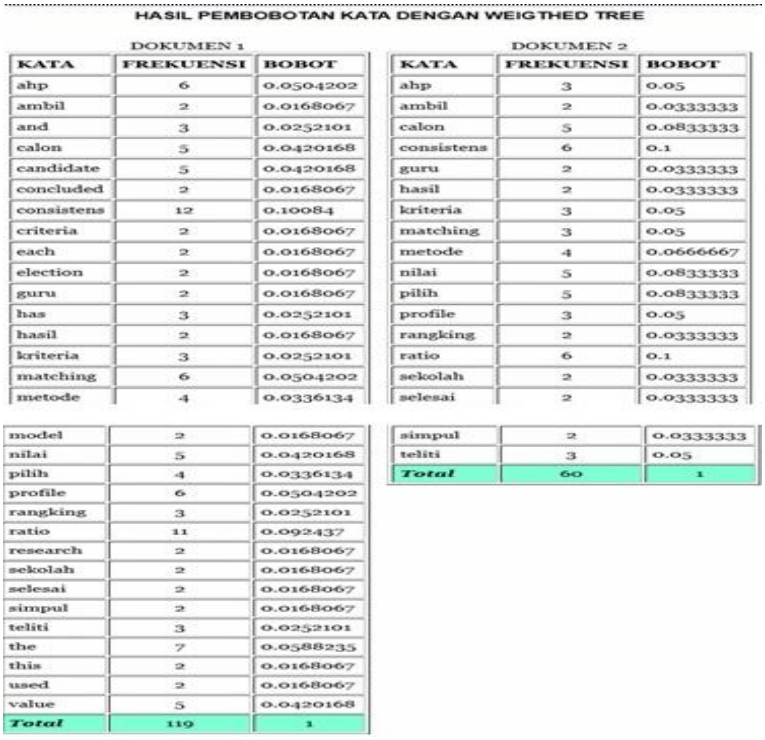
Gambar 4. Menu hasil halaman preprocessing

c. Halaman hasil *Weighted Tree*

Form hasil *weighted tree* merupakan form untuk melakukan pembobotan kata dengan algoritma *weighted tree* berdasarkan data dari hasil preprocessing. Berikut ini tampilan algoritma *weighted tree.*

Gambar 5. Halaman algoritma *weighted tree*

Form hasil pembobotan merupakan form untuk melakukan presentase kemiripan yang akan dihasilkan oleh dua buah dokumen text dengan menggunakan *weighted tree*. Adapun tampilan hasil kemiripannya dapat dilihat pada gambar berikut.



HASIL PEMBOBOTAN KATA DENGAN WEIGTHED TREE

DOKUMEN 1

KATA	FREKUENSI	BOBOT
ahp	6	0.0504202
ambil	2	0.0168067
and	3	0.0252101
calon	5	0.0420168
candidate	5	0.0420168
concluded	2	0.0168067
consistens	12	0.10084
criteria	2	0.0168067
each	2	0.0168067
election	2	0.0168067
guru	2	0.0168067
has	3	0.0252101
hasil	2	0.0168067
kriteria	3	0.0252101
matching	6	0.0504202
metode	4	0.0336134
model	2	0.0168067
nilai	5	0.0420168
pilih	4	0.0336134
profile	6	0.0504202
rangking	3	0.0252101
ratio	11	0.092437
research	2	0.0168067
sekolah	2	0.0168067
selesai	2	0.0168067
simpul	2	0.0168067
teliti	3	0.0252101
the	7	0.0588235
this	2	0.0168067
used	2	0.0168067
value	5	0.0420168
Total	119	1

DOKUMEN 2

KATA	FREKUENSI	BOBOT
ahp	3	0.05
ambil	2	0.0333333
calon	5	0.0833333
consistens	6	0.1
guru	2	0.0333333
hasil	2	0.0333333
kriteria	3	0.05
matching	3	0.05
metode	4	0.0666667
nilai	5	0.0833333
pilih	5	0.0833333
profile	3	0.05
rangking	2	0.0333333
ratio	6	0.1
sekolah	2	0.0333333
selesai	2	0.0333333
simpul	2	0.0333333
teliti	3	0.05
Total	60	1

Gambar 6. Algoritma *weighted tree*

d. Halaman Kemiripan

Form hasil persentase kemiripan yang dihasilkan pada 2 dokumen Text tersebut dengan menggunakan *Cosine Similarity* merupakan form untuk melakukan presentase kemiripan yang akan dihasilkan oleh dua buah dokumen text. Berikut tampilan hasil kemiripannya.

Gambar 7. Menu hasil kemiripan

3. Hasil Pengujian Sistem

Dalam melakukan pengujian sistem ini, penulis mengumpulkan beberapa dokumen yang diperlukan untuk pengujian ini, untuk mengukur persentase keberhasilan sistem ini, dokumen yang di uji sebanyak 100 dokumen yang dikategorikan dalam tiga jenis dokumen yaitu dokumen yang sudah pasti sama, dokumen yang hampir sama (mirip) dan dokumen yang berbeda.

Tabel 4. Pengujian dokumen

Kategori Dokumen	*Jumlah Dokumen*	*tipe*	*Persentase kemiripan (%)*
Dokumen yang Sama	30	Pdf/txt	100
Dokumen yang Hampir Sama	30	Pdf/txt	82.22
Dokumen yang Beda	40	Pdf/txt	31.58
Jumlah	100		
Rata-rata			71.60

Berdasarkan tabel diatas, maka pengujian dokumen yang sama mempunyai nilai rata-rata kemiripan 100 % (seratus persen), nilai rata-rata

pengujian dokumen yang hampir sama mempunyai kemiripan 83,22 %, sedangkan pengujian dokumen yang jelas berbeda mempunyai nilai kemiripan 31,58 % . Maka nilai rata-rata persentase kemiripan adalah 71,60 %.

KESIMPULAN DAN SARAN

Dari hasil penelitian Pendeteksian Plagiarisme dengan Menggunakan Metode *weight tree*, maka penulis dapat mengambil kesimpulan sebagai berikut:

1. Algoritma *weight tree* yang diterapkan pada sistem identifikasi ini terbukti mampu mengidentifikasi dengan baik kemiripan dokumen *plagiarisme* yang diuji dengan membandingkannya pada kumpulan dokumen 1 dan dokumen 2 sampel yang diinput dan dibandingkan dengan metode *weight tree*.
2. Penggunaan rumus yang digunakan pada algoritma *weight tree* ini yaitu rumus *cosine simliarity* terbukti lebih akurat dengan tingkat keberhasilan mencapai 90% pada dokumen pengujian.

Adapun saran dari Penulis untuk pengembangan Penelitian selanjutnya adalah:

1. Untuk kedepan agar dapat mengidentifikasi seluruh ekstensi file dokumen tidak hanya file pdf, doc, docx dan rtf, selanjutnya proses mengidentifikasi seluruh dokumen file, bukan hanya menggunakan dua dokumen uji.
2. Perlu adanya perbaikan struktur data supaya proses pendeteksian dapat dilakukan dengan lebih cepat, karena semakin banyak data maka akan semakin banyak pula waktu yang dibutuhkan untuk menyelesaikan seluruh proses.
3. Sistem dapat dikembangkan dengan menggunakan teknik dan algoritma lainnya seperti *Rabin-Karp* dan *Synonym Recognition*.
4. Dapat terintegrasi langsung ke situs *online* yang memuat karya ilmiah dan dapat dilakukan proses pengidentifikasian yang lebih cepat dari sebelumnya.

DAFTAR PUSTAKA

Afdhal, Chalis, T., & Gani, T.A. (2014). Analisa perbandingan aplikasi pendeteksi plagiat tehadap Karya Ilmiah. Seminar Nasional dan Expo Teknik Elektro 2014, 193-199.

Ariyani, N.H., Sutardi, & Ramadhan, R. (2016). Aplikasi Pendeteksi kemiripan isi teks dokumen menggunakan metode *Levenshtein distance.* Jurnal SemanTIK, 2(1), 279-286.

Beall, J. (2008). The Weaknesses of Full-Text Searching. The Journal of Academic Librarianship.

Iqbal, A., & Nurdin. (2017). Implementasi Aplikasi E-Boarding house di kota Lhokseumawe menggunakan Algoritma *Levenshtein distance*. Jurnal Sistem Informasi, 1(1), 139-167

Nurdin & Munthoha, A. (2017). Sistem Pendeteksian Kemiripan Judul Skripsi Menggunakan Algoritma Winnowing. Jurnal InfoTekJar (Jurnal Nasional Informatika dan Teknologi Jaringan), 2(1), 90-97.

Nurdin & Miranda. (2015). Sistem Pendukung Keputusan Pemilihan PTS di Lhokseumawe Menggunakan Metode Fuzzy AHP Berbasis Web. Jurnal Informatika, 9(2), 1048-1056.

Sarno, R., & Rahutomo, F. (2008). Penerapan Algoritma Weighted Tree Similarity untuk pencarian semantik. Jurnal JUTI, 7(1), 35-42.

Stepchyshyn, V., & Nelson, R. S. (2007). Library plagiarism policies. Assoc. of College & Research Libraries.

Yang, L., Sarker, B.K., Bhavsar, V.C., & Boley, H. (2005). A Weighted Tree Simplicity Algorithm for Similarity Matching of Partial Product Descriptions. Proceeding of ISCA 14th International Conference on Intelligent and Adaptive Systems and Software Engineering (pp 55-60), Toronto.

Yates, R.B., & Neto, B.R. (1999). Modern Information Retrieval. Addison Weasley Longman Limited.