MENDETEKSI PLAGIARISM PADA DOKUMEN PROPOSAL SKRIPSI MENGGUNAKAN ALGORITMA JARO WINKLER DISTANCE

Sherly Christina 1, Enny Dwi Oktaviyani 2 dan Buyung Famungkas 3
1,2,3)Program Studi Teknik Informatika Universitas Palangka Raya
Kampus Unpar Tunjung Nyaho Jl. Yos Sudarso Palangka Raya
Email: sherly.christina.upr@gmail.com, enny.obrien@gmail.com, buyungfamungkas@gmail.com

ABSTRACT

The practice of plagiarism among students in working on their final project document might happen. Plagiarism is done considering to the limited workmanship and lack of motivation to try with their own ability. It is necessary to provide a tool to prevent the act of plagiarism among the students. In this research, we built an application to detect the plagiarism on the final project proposal document.

The application applies the Jaro Winkler Distance algorithm to detect the similarity from several documents. The first phase to detect the similarity of the final project proposals is the text preprocessing phase of the document such as case folding, tokenizing, stopwords removal and stemming. Precision and Recall formula will be used to analyze the performance of the method in application.

The test result of the data set shows the application could find the 80% relevant data that indicated the similarity. It means the application could contribute to detect the plagiarism on the final project proposal document.

***Keyword** : Jaro Winkler Distance algorithm , plagiarism, document.*

PENDAHULUAN

Saat ini perkembangan teknologi dan kemudahan memperoleh akses internet, menyebabkan semakin banyaknya informasi yang tersedia bebas. Kemudahan ini dapat memberikan kesempatan bagi pengguna teknologi yang tidak bertanggung jawab melakukan pelanggaran terhadap Hak Asasi Kekayaan Intelektual (HAKI). Contoh pelanggaran HAKI yang sering terjadi dalam lingkungan akademi Perguruan Tinggi adalah penjiplakan suatu karya ilmiah.

Penjiplakan menurut Kamus Besar Bahasa Indonesia (KBBI, 2016) adalah menggambar atau menulis garis-garis gambaran atau tulisan yang telah tersedia (dengan menempelkan kertas kosong pada gambar atau tulisan yang akan ditiru), mencontoh atau meniru tulisan, pekerjaan orang lain, mencuri karangan orang lain dan mengakui sebagai karangan sendiri, mengutip

karangan orang lain tanpa seizin penulisnya. Menurut peraturan Menteri Pendidikan Nasional RI No. 17 Tahun 2010 tentang pencegahan dan penanggulangan plagiat di perguruan tinggi, Bab I pasal I ayat I: Plagiat adalah perbuatan secara sengaja atau tidak sengaja dalam memperoleh atau mencoba memperoleh 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.

Praktek plagiat kadang sulit dihindari, misalnya praktek plagiat yang terjadi di kalangan mahasiswa yang sedang menyusun Tugas Akhir atau Skripsi. Praktek plagiat dapat dilakukan sebagian mahasiswa dengan alasan-alasan seperti waktu pengerjaan skripsi yang terbatas dan tidak adanya motivasi atau kepercayaan diri untuk berusaha menyelesaikan skripsi dengan kreatifitas dan kemampuan sendiri. Oleh karena itu Universitas perlu mengambil tindakan untuk menangani praktek plagiat, demi membangun karakter mahasiswa agar menjadi Sumber Daya Manusia yang berkualitas, bermoral Pancasila dan berdaya saing tinggi. Prinsip anti terhadap praktek plagiat akan mendidik mahasiswa untuk jujur dan percaya diri dalam menuangkan ide dan pokok pikirannya ke dalam karya tulis tanpa melakukan plagiat.

Penelitian ini bertujuan mendeteksi praktek plagiat yang mungkin terjadi dalam proses penyusunan Skripsi mahasiswa. Penelitian ini membangun aplikasi berbasis web untuk mendeteksi dokumen proposal Skripsi mahasiswa yang plagiat terhadap dokumen Skripsi mahasiswa lain.

Metode yang digunakan untuk mendeteksi plagiat adalah mengukur tingkat kesamaan teks dari beberapa dokumen proposal Skripsi yang diajukan oleh mahasiswa. Algoritma yang digunakan dalam penelitian ini adalah *Jaro-Winkler Distance*. Algoritma *Jaro-Winkler Distance* adalah sebuah algoritma untuk mengukur kesamaan antara dua string (Rinusantoro, 2014). Algoritma Jaro-Winkler Distance dapat digunakan untuk mendeteksi *plagiarism* pada dokumen teks(Kurniawati, 2010), (Rinusantoro, S., 2014).

Penelitian ini menggunakan data set yang diperoleh dari beberapa dokumen proposal Skripsi mahasiswa.

Kemudian pengujian untuk mengetahui performa dari aplikasi akan diukur menggunakan formula *Precision* dan *Recall.*

METODE

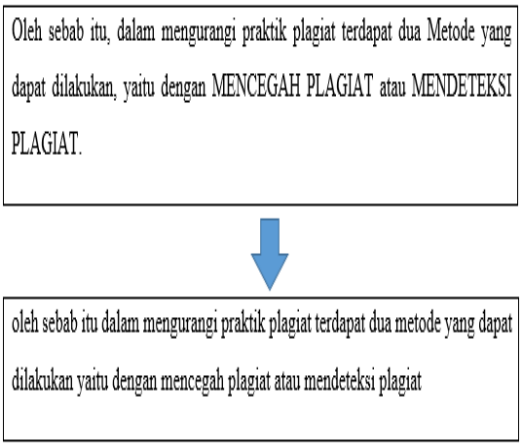
Metode untuk mendeteksi dokumen skripsi yang terindikasi plagiat menggunakan beberapa tahapan berikut.

1. *Text Prepocessing*

Tahap awal adalah *Text Preprocessing*, yang bertujuan mempersiapkan teks menjadi data yang akan diolah lebih lanjut. Sekumpulan karakter dalam dokumen teks harus dipecah-pecah menjadi unsur yang lebih berarti. *Text preprocessing* pada penelitian ini melewati tiga proses, yaitu:

a.*Case Folding*

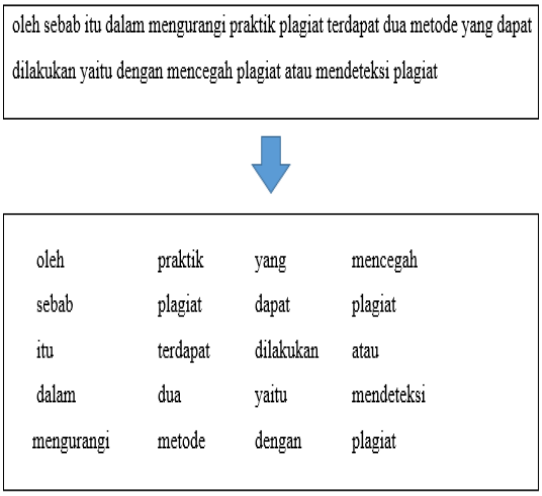
Merupakan proses mengubah dokumen/kalimat menjadi huruf kecil, hanya huruf "a" sampai dengan "z" yang diterima, namun yang dirubah hanya huruf saja bukan angka maupun simbol. Gambar 1 adalah contoh Case Folding pada teks.



Gambar 1. Contoh Case Folding pada Teks

b. Tokenizing

Tahap *Tokenizing* adalah proses pemotongan teks atau kalimat berdasarkan kata-kata penyusunnya. Gambar 2 menunjukkan contoh Tokenizing pada teks.

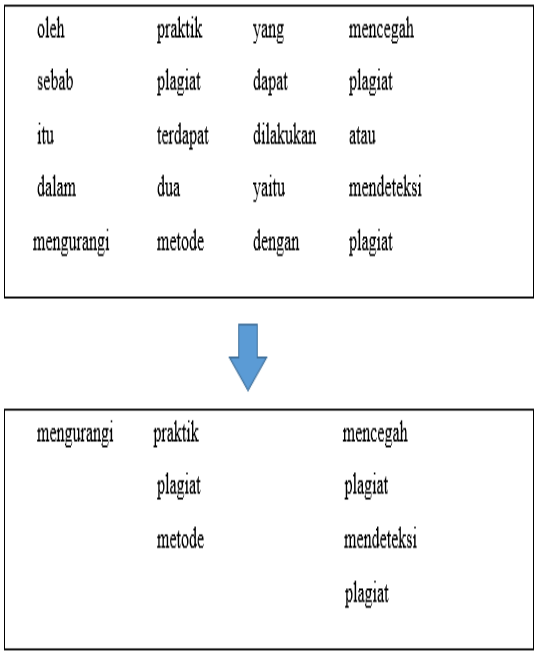


Gambar 2. Contoh Tokenizing pada teks

c. *Stopwords Removal*

Tahap Stopwords Removal adalah proses mengeliminasi kata-kata yang dianggap tidak memiliki relevansi yang erat dengan isi dari dokumen.

Contohnya kata: “dan”, “yang”, “supaya”. Gambar 3 menunjukkan contoh Stopwords Removal.



Gambar 3. Contoh *Stopwords Removal* pada teks.

d. *Stemming*

Tahap *stemming* adalah tahap terakhir dari *Text Preprocessing* yang bertujuan menghasilkan kata dasar dari setiap kata hasil proses *Tokenizing* dan *Stopword Removal*. Pada tahap ini dilakukan proses pengembalian berbagai bentuk kata ke dalam suatu representasi yang sama. Algoritma *stemming* yang digunakan pada penelitian ini adalah *stemming* teks berbahasa Indonesia yang dibuat oleh Arifin-Setiono (Arifin dkk, 2009).

Input dari algoritma ini sebuah kata yang kemudian dilakukan :

1. Pemeriksaan semua kemungkinan bentuk kata. Setiap kata diasumsikan memiliki 2 awalan (prefiks) dan 3 akhiran (sufiks) sehingga bentuknya menjadi :
   Prefiks 1 + Prefiks 2 + Kata Dasar + Sufiks 1 + Sufiks 2 + Sufiks 3
2. Pemotongan dilakukan secara berurutan sebagai berikut :
   AW : AW (Awalan), AK : AK (Akhiran), KD : KD (Kata Dasar)
   a. AW I, Hasilnya disimpan pada p1 (prefiks 1)
   b. AW II, Hasilnya disimpan pada p2 (prefiks 2)
   c. AK I, Hasilnya disimpan pada s1 (sufiks 1)
   d. AK II, Hasilnya disimpan pada s2 (sufiks 2)
   e. AK III, Hasilnya disimpan pada s3 (sufiks 3)

   Pada setiap tahap pemotongan di atas diikuti dengan pemeriksaan pada kamus untuk memastikan hasil pemotongan itu sudah berada dalam bentuk dasar. Bila pemeriksaan berhasil maka proses dinyatakan selesai dan tidak perlu melanjutkan proses pemotongan imbuhan lainnya.
3. Namun jika sampai pada pemotongan AK III, belum juga

ditemukan pada kamus, maka dilakukan proses kombinasi. KD yang dihasilkan dikombinasikan dengan imbuhan-imbuhannya dalam 12 konfigurasi berikut:

a. KD
b. KD + AK III
c. KD + AK III + AK II
d. KD + AK III + AK II + AK I
e. AW I + AW II + KD
f. AW I + AW II + KD + AK III
g. AW I + AW II + KD + AK III + AK II
h. AW I + AW II + KD + AK III + AK II + AK I
i. AW II + KD
j. AW II + KD + AK III
k. AW II + KD + AK III + AK II
l. AW II + KD + AK III + AK II + AK I

Sebenarnya kombinasi a, b, c, d, h, dan I sudah diperiksa pada tahap sebelumnya, karena kombinasi ini adalah hasil pemotongan bertahap tersebut. Dengan demikian, kombinasi yang masih perlu dilakukan tinggal 6 yakni pada kombinasi-kombinasi yang belum dilakukan (e, f, g, I, j, dan k). tentunya bila hasil pemeriksaan suatu kombinasi adalah ‘ada’, maka pemeriksaan pada kombinasi lainnya sudah tidak diperlukan lagi. Pemeriksaaan 12 kombinasi ini diperlukan, karena adanya fenomena *overstemming* pada algoritma pemotongan imbuhan. Kelemahan ini berakibat pada pemotongan bagian kata yang sebenarnya adalah milik Kata Dasar itu sendiri yang kebetulan mirip dengan salah satu jenis imbuhan yang ada. Dengan 12 kombinasi, pemotongan yang sudah terlanjut tersebut dapat dikembalikan sesuai posisinya.

2. Menghitung Kemiripan Dokumen

Setelah melakukan *Text Prepocessing*, tahapan berikutnya adalah menghitung Kesamaan Dokumen. Algoritma *Jaro-Winkler* merupakan varian dari Jaro *Distance* metrik sebuah algoritma untuk mengukur kesamaan antara dua *string*, algoritma ini dapat digunakan di dalam pendeteksian duplikat. Semakin tinggi nilai Jaro-Winkler *distance* untuk dua *string*, semakin mirip kedua *string* tersebut. Skor normalnya adalah nilai 0 menandakan tidak ada kesamaan, dan 1 adalah sama persis (Winkler, 2002).

Dasar dari algoritma ini memiliki tiga bagian (Wilyanto, 2015):

1. Menghitung panjang string
2. Menemukan jumlah karakter yang sama di dalam dua string
3. Menemukan jumlah transposisi

Pada algoritma *Jaro* digunakan rumus untuk menghitung jarak (*dj*) antara dua *string* yaitu *s1* dan *s2* adalah

$$Dj = \frac{1}{3} \times \left(\frac{m}{|s1|} + \frac{m}{|s2|} + \frac{m-t}{m}\right) \quad (1)$$

Dimana :

m = jumlah karakter yang sama persis

|s1| = panjang *string* 1

|s2| = panjang *string* 2

t = jumlah transposisi

Jarak teoritis dua buah karakter yang dikatakan sama dapat dibenarkan jika tidak melebihi:

$$\left(\frac{\max(|s1|,|s2|)}{s}\right) < -1 \quad (2)$$

Akan tetapi bila mengacu kepada nilai yang akan dihasilkan oleh algoritma *Jaro Winkler* maka nilai jarak maksimalnya adalah 1 yang menandakan kesamaan string yang dibandingkan mencapai seratus persen atau sama persis. Biasanya *s1* digunakan sebagai acuan untuk urutan di dalam mencari transposisi. Yang dimaksud transposisi disini adalah karakter yang sama dari *string* yang dibandingkan akan tetapi tertukar urutannya.

Sebagai contoh, dalam membandingkan kata CRATE dengan TRACE, bila dilihat seksama maka dapat dikatakan semua karakter yang ada di *s1* ada di *s2* dan sama dengan karakter yang ada di *s2*, tetapi dengan urutan yang berbeda. Dengan mengganti C dan T, dapat dilihat perubahan kata CRATE menjadi TRACE, pertukaran dua elemen *string* inilah adalah contoh nyata dari transposisi yang dijelaskan.

Jaro Winkler Distance menggunakan *prefix scale (p)* yang memberikan tingkat penilaian yang lebih, dan *prefix length (l)* yang menyatakan panjang awalan yaitu panjang karakter yang sama dari *string* yang dibandingkan sampai ditemukannya ketidaksamaan, maka Jaro Winkler *distance*nya (d_w) adalah :

$$d_w = d_j + (l_p\ (1 - d_w)) \quad (3)$$

Dimana:

d_j = Jaro *Distance* untuk string *s1* dan *s2*

l = panjang prefiks umum diawal *string* nilai maksimalnya 4 karakter (panjang

karakter yang sama sebelum ditemukan ketidaksamaan max 4).

p = konstanta *scalling factor*. Nilai standar untuk konstanta ini menurut Winkler adalah *p* = 0,1.

Dalam beberapa implementasi dari Jaro Winkler, prefiks $l_p(1- d_j)$ hanya ditambahkan bila perbandingan *string* Jaro *Distance* dibawah "*boost threshold*" atau ambang batas. Ambang batas atas dari implementasi Winkler adalah 0,7.

Berikut ini adalah contoh perhitungan Jaro Winkler *Distance*. Jika *string* :

S1 = B A R T H A dan

S2= B A R H T A maka :

m = 6

s1 = 6

s2 = 6

karakter yang tertukar hanyalah T dan H, maka

t = 1

maka nilai *Jaro Distance* adalah :

d_j=

$$\frac{1}{3}x\left(\frac{6}{6}+\frac{6}{6}+\frac{6-1}{6}\right)=0,944$$

$$\frac{1}{3}x\left(\frac{6}{6}+\frac{6}{6}+\frac{6-1}{6}\right)=0,944$$

Kemudian bila diperhatikan susunan s1 dan s2 dapat diketahui nilai l = 3, dan dengan nilai konstan p = 0,1 maka nilai Jaro Winkler *distance* adalah :

$$d_j=\frac{1}{3}x\left(\frac{6}{6}+\frac{6}{6}+\frac{6-1}{6}\right)=0,944$$

$$d_w=0,944+\left(3x0,1(1-0,944)\right)=0,961$$

$$d_w=0,944+\left(3x0,1(1-0,944)\right)=0,961$$

Pada penelitian ini, nilai Jaro Winkler distance adalah nilai proximity sehingga untuk mendeteksi kesamaan dokumen proposal Skripsi, nilai *Jaro Winkler Distance* perlu dikurangi 1, menjadi:

$$Nilai\ Kesamaan\ Dokumen = 1 - d_w$$

$$Nilai\ Kesamaan\ Dokumen = 1 - d_w \quad (4)$$

Sehingga pada *range* nilai 0 sampai 1, nilai 0 adalah nilai yang menunjukan kesamaan penuh, sedangkan nilai 1 menunjukkan tidak ada kesamaan sama sekali.

Pada penelitian ini toleransi kesamaan dokumen adalah 70%. Sehingga untuk Nilai Kesamaan Dokumen ≥0,7 dianggap tidak plagiat, sedang untuk range Nilai Kesamaan Dokumen diantara 0,69 - 0,3 dianggap plagiasi sedang, dan Nilai Kesamaan Dokumen = 0 dianggap plagiasi berat.

Aplikasi berbasis web yang dibangun pada penelitian ini

menggunakan beberapa tahapan *Waterfall* (Pressman, 2010), yaitu : *Communication, Planning, Modelling* dan *Construction*.

Communication adalah tahap analisis terhadap kebutuhan aplikasi, dan tahap untuk mengadakan pengumpulan data yang diperlukan. Hasil pada tahap komunikasi yaitu data-data yang dibutuhkan dalam pembuatan aplikasi.

Planning adalah lanjutan dari proses *communication*. Tahapan ini akan menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan aplikasi, termasuk rencana yang akan dilakukan.

Modelling adalah tahap menerjemahkan kebutuhan sistem ke sebuah perancangan *software* yang dapat diperkirakan sebelum dibuat *coding*. Proses ini terdiri atas proses analisis dan desain.

Construction adalah proses membuat kode program (*coding*). *Coding* atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. *Programmer* akan menerjemahkan transaksi yang diminta oleh *user*.

Mekanisme untuk mendeteksi praktek plagiat pada dokumen proposal Skripsi mahasiswa pada aplikasi ini, terdiri atas beberapa langkah seperti berikut:

1. Input data proposal Skripsi Mahasiswa yang akan dicek.
2. Aplikasi melakukan Text Preprocessing terhadap data inputan.
3. Aplikasi mengukur kesamaan isi dokumen proposal dengan cara membandingkan dengan seluruh koleksi data Skripsi yang telah tersimpan di dalam database sistem.
4. Aplikasi menampilkan hasil perhitungan algoritma, berupa daftar dokumen yang memiliki kesamaan dengan data inputan.

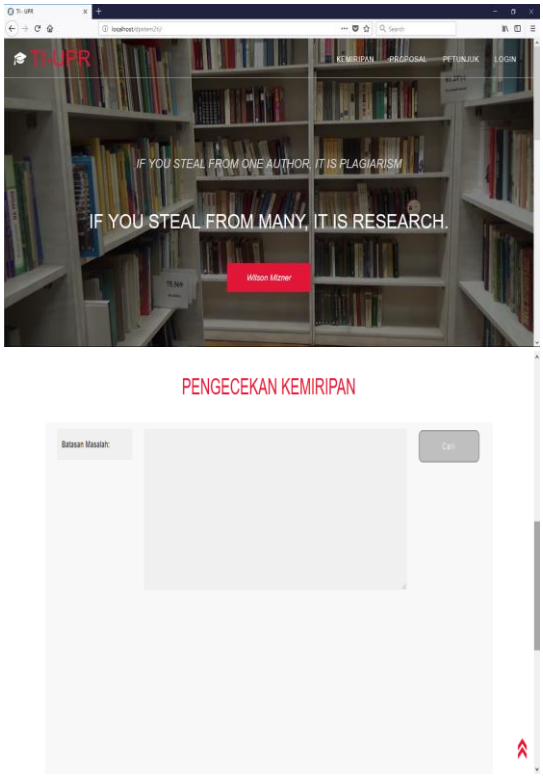
Kemudian pengujian untuk mengetahui performa sistem menggunakan formula *Recall* dan *Precision*. *Recall* adalah proporsi jumlah dokumen yang ditemukan kembali oleh sebuah proses pencarian dalam sistem temu kembali informasi. *Precision* adalah proporsi jumlah dokumen yang ditemukan dan dianggap relevan untuk kebutuhan pencari informasi (Shepperd dkk, 2014).

Ukuran *recall* dan *precision* ini juga bergantung pada apa yang sesungguhnya dimaksud dengan

"dokumen yang relevan" itu dan bagaimana memastikan relevan tidaknya dokumen. Relevansi adalah kecocokan antara apa yang dicari dengan apa yang ditemukan. Sebuah dokumen dianggap relevan jika isinya dianggap cocok dengan apa yang diharapkan oleh pencarinya, (Pendit, 2008).

HASIL DAN PEMBAHASAN

Fasilitas untuk mendeteksi dokumen proposal skripsi yang plagiat dibangun dalam sebuah aplikasi berbasis Web. Gambar 4 menunjukkan *screenshoot* dari halaman Web yang menjadi antarmuka untuk mengecek kesamaan dokumen-dokumen proposal.



Gambar 4. Antarmuka halaman Kemiripan, halaman untuk mengecek kemiripan dokumen.

Skenario pengujian performa aplikasi dilakukan pada bagian Sub Bab Batasan Masalah dan Ruang Lingkup dari dokumen proposal Skripsi. Peneliti menyiapkan 2 hasil perhitungan dari *data set* pengujian, yaitu hasil perhitungan dari sistem dan hasil perhitungan dengan algoritma *Jaro-Winkler* yang dilakukan secara manual. Tabel 1 menunjukkan Hasil Perhitungan *Jaro-Winkler Distance* oleh Sistem dan secara Manual.

Tabel 1. Hasil Perhitungan Jaro-Winkler Distance oleh Sistem dan Manual

Proposal	Skripsi	Hasil Sistem	Hasil Manual
A1	B1	0,56	0,56
A2	B2	0.59	0.59
A3	B3	0.34	0.33
A4	B4	0.36	0.36
A5	B5	0.44	0.44
A6	B6	0.53	0.53
A7	B7	0.52	0.52
A8	B8	0.58	0.58
A9	B9	0.43	0.44
A10	B10	0.43	0.43
Nilai Rata-Rata		0.47	0.47

Pada Tabel 1, dilakukan pengujian terhadap tiap data proposal yang dibandingkan dengan seluruh dokumen skripsi yang disimpan di dalam database. A1,..., A10 adalah data proposal 1 sampai 10. Kemudian B1,..., B10 adalah data skripsi yang terdeteksi

memiliki nilai kesamaan atau plagiasi paling tinggi. Tabel 2 menunjukkan penelusuran terhadap hasil perhitungan yang relevan dan tidak relevan.

Tabel 2. Hasil Penelusuran

Data yang dibandingkan	Relevan	Tidak Relevan	Di Temukan
10 Proposal	8	2	10

Hasil perhitungan *recall* dan *precision*, dari data pada Tabel 1 dan Tabel 2 ditunjukkan pada Tabel 3.

Tabel 3. Analisis Terhadap Hasil Penelusuran

Parameter	**Nilai**
Relevan (a)	8
Tidak Relevan (b)	2
Total(a+b)	10
Tidak Ditemukan (c)	0
Total (a+c)	8
Recall [a/(a+c)] x 100%	100%
Precision [a/(a+b)] x 100%	80%

Nilai *Precision* dari hasil penelusuran adalah sebesar 80% dan nilai *Recall* adalah 100% dari skala 0% - 100%, sehingga dapat diketahui bahwa nilai *precision* lebih rendah dari pada nilai *recall.* Walaupun nilai *precision* lebih rendah dari pada nilai *recall*, tingkat keefektifan dari informasi dibedakan menjadi efektif jika nilai di atas 50% dan tidak efektif jika nilai dibawah 50%.

Nilai *recall* dan *precision* tidak seimbang disebabkan oleh beberapa faktor yang diantaranya adalah terbatasnya koleksi dokumen proposal dan dokumen skripsi yang dimiliki, dokumen uji memiliki kata-kata yang berbeda meskipun memiliki arti yang sama dan masih banyak faktor lainnya. Secara perhitungan persentase diperoleh hasil relevan dari aplikasi ini adalah 80%. Hasil perhitungan pengujian tersebut akan lebih akurat apabila data uji yang digunakan pada pengujian bisa lebih banyak. Kemudian untuk rata-rata persentase kemiripan dokumen yang di temukan oleh sistem adalah 0.47 dan hasil rata-rata persentase yang dihitung secara manual adalah 0.47.

KESIMPULAN

Hasil Pengujian terhadap performa algoritma *Jaro-Winkler* pada aplikasi yang dilakukan terhadap 10 dokumen proposal dibandingkan dengan 10 dokumen skripsi menunjukkan hasil yang baik, dengan ditemukannya 80% data yang relevan. Pada penelitian lebih lanjut, perlu dilakukan pengujian terhadap koleksi data skripsi dan data proposal yang lebih banyak agar lebih mendekati ukuran tingkat kemiripan sebenarnya pada sistem temu kembali yang ada.

DAFTAR PUSTAKA

Arifin, A.Z., Mahendra, I., A., Ciptaningtyas H.,T., 2009,

Enhanced Confix Stripping Stemmer and Ants Algorithm For Classifying News Document In Indonesian Language, The 5th International Conference on Information & Communication Technology and System:149-158.

KBBI, 2016, Kata Dasar, https://www.dropbox.com/s/i11nom7r7s54p5a/kbbi_all_words.zip?dl=0. (retrieved . 4 April 2017)

KBBI, 2016, Penjiplakan, http://kbbi.web.id/jiplak>. (retrieved 1 May 2017)

Kurniawati A, Puspitodjati S, Rahman S., 2010, *Implementasi Algoritma Jaro-Winkler Distance untuk Membandingkan Kesamaan Dokumen Berbahasa Indonesia,* Proceedings of Seminar Ilmiah Nasional KOMMIT 2010, Bali

Pendit, P., L., 2008, Perpustakaan Digital dari A sampai Z, Cita Karya Karsa Mandiri, Jakarta, ISBN:9791695210

Peraturan Menteri, 2010, Pencegahan dan Penanggulangan Plagiat di Perguruan Tinggi. Peraturan Menteri Pendidikan Nasional Nomor 17 Tahun 2010. 16 Agustus 2010, https://luk.staff.ugm.ac.id/atur/Permen17-2010.pdf, (retrieved 2 May2017)

Pressman, S.S., Ph.D, 2010, *Software Engineering Seventh Edition.* New York: McGraw-Hill.

Rinusantoro, S., 2014, *Aplikasi Deteksi Kemiripan Dokumen Teks.* Yogyakarta: Universitas Gadjah Mada Yogyakarta.

Shepperd, M., Bowes, D., Hall, T., 2014, Researcher Bias: The Use of Machine Learning in Software Defect Prediction, IEEE Transaction on Software Engineering, Vol. 40 Issue 6, June 1 2014

Wilyanto, T., 2015, Sistem Deteksi plagiat Dokumen Teks Menggunakan Algoritma *Jaro-Winkler Distance*".