Jurnal Linguistik Komputasional
Indonesian Association for Computational Linguistics
JLK
Vol 2, No 1
Maret 2019

Penerapan *Cosine Similarity* dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen

Ade Riyani[#1], Muhammad Zidny Naf'an[#2], Auliya Burhanuddin[#3]

[#] *Program Studi S1 Informatika, Fakultas Teknologi Industri dan Informatika, Institut Teknologi Telkom Purwokerto*
Jl. D.I. Panjaitan, No. 128. Purwokerto.

[1]15102002@st3telkom.ac.id
[2]zidny@ittelkom-pwt.ac.id
[3]auliya@ittelkom-pwt.ac.id

***Abstrak*— Plagiarisme merupakan tindakan mengambil sebagian atau seluruh ide seseorang berupa dokumen maupun teks tanpa mencantumkan sumber pengambilan informasi. Penelitian ini bertujuan untuk mendeteksi kemiripan teks dan nilai plagiarismenya menggunakan algoritma *cosine similarity* dan pembobotan TF-IDF. Corpus dibuat dari kumpulan teks abstrak dalam bahasa Indonesia dari laporan skripsi mahasiswa. Tahapan penelitian yang digunakan yaitu *preprocessing* (terdiri dari *case folding, tokenizing, stopword removal,* dan *stemming),* perhitungan pembobotan TF-IDF, dan perhitungan nilai kemiripan menggunakan *cosine similarity.* Skenario penelitian dengan *stemming* menghasilkan nilai kemiripan rata-rata lebih tinggi 10% daripada tanpa *stemming*. Penelitian ini menghasilkan nilai similaritas diatas 50% untuk dokumen yang tingkat kemiripannya tinggi. Sedangkan untuk dokumen dengan tingkat kemiripan rendah atau tidak berplagiat menghasilkan nilai similaritas dibawah 40%. Berdasarkan hasil percobaan *cosine similarity* dan pembobotan TF-IDF mampu menghasilkan nilai kemiripan dari masing-masing teks pembanding.**

***Kata kunci*— dokumen, *preprocessing*, *cosine similarity*, TF-IDF, plagiarisme**

Abstract- Plagiarism is the act of taking part or all of one's ideas in the form of documents or texts without including sources of information. This study aims to detect the similarity of text and its plagiarism value using the cosine similarity algorithm and weighting TF-IDF. Corpus is collected from a collection of abstract texts in Indonesian from student thesis reports. The stages of research used were preprocessing (consisting of case folding, tokenizing, stopword removal, and stemming), TF-IDF weighting calculation, and calculation of similarity values using cosine similarity. The research scenario with stemming results in an average similarity value of 10% higher than without stemming. This study produces a similarity value above 50% for documents with a high degree of similarity. Whereas for documents with a low level of similarity or not plagiarism produces a similarity value below 40%. Based on the results of the cosine similarity experiment and the weighting of TF-IDF, it can produce similarity values from each comparative text.

Keyword- document, preprocessing, cosine similarity, TF-IDF, plagiarism

I. PENDAHULUAN

Plagiarisme merupakan salah satu perbuatan yang melanggar kode etik dalam dunia penulisan, karena tindakan tersebut telah mengambil karya orang lain dan mengakuinya sebagai karya sendiri. Pada tahun 2010, Direktorat Jenderal Pendidikan Tinggi telah mengeluarkan peraturan tentang cara pencegahan dan penanggulangan plagiarisme *term*asuk sanksi untuk dosen, mahasiswa, dan calon guru besar sekalipun. Praktek ini kerap dihubungkan dalam dunia pendidikan khususnya mahasiswa dalam melaksanakan tugasnya. Merujuk pada Permendiknas No. 17 Tahun 2010 tentang Pencegahan dan Penanggulangan Plagiat di Perguruan Tinggi disebutkan bahwa dalam melaksanakan otonomi keilmuan dan kebebasan akademik mahasiswa, dosen, peneliti, tenaga kependidikan wajib menjunjung tinggi kejujuran dan etika akademik, terutama larangan untuk melakukan plagiat dalam menghasilkan karya ilmiah, sehingga kreativitas dalam bidang akademik dapat tumbuh dan berkembang [1].

Berdasarkan permasalahan tentang plagiarism untuk mengetahui apakah suatu karya ilmiah benar milik sendiri dibutuhkan sebuah sistem pendeteksi kemiripan sebuah dokumen. Beberapa metode yang sudah pernah dilakukan untuk memdeteksi kemiripan dokumen adalah *cosine similarity* [2], *jaccard* [3], TF-IDF [4], *Support Vector Regression* [5], Levenshtein Distance [6], dan Latent Semantic Analysis [7].

Wahyuni, dkk pada [2] menggunakan cosine similarity dan pembobotan TF-IDF untuk mengelompokkan 50 dokumen skripsi, dan didapatkan tingkat akurasi sebesar 98%. Selanjutnya Sugiyamto, dkk pada [3] mencoba membandingkan Jaccard dan *cosine similarity* pada pengujian kesamaan dokumen, dengan hasil penelitian menunjukkan bahwa pengujian kemiripan menggunakan *cosine similarity* memiliki tingkat akurasi lebih tinggi yaitu 0,949808

Submitted 12-02-2019; Revised 26-03-2019; Accepted 29-03-2019

dibandingkan dengan Jaccard sebesar 0,949077. Okfalisa dan Harahap [4] menggunakan bobot TF-IDF pada sistem monitoring pelaksanaan diskusi online. Nasution dkk [5] melakukan penelitian untuk mengukur nilai prediksi data pasangan ayat Al-Quran terjemahan bahasa Inggris berdasarkan *alignment* dan *word2vec* menggunakan *Support Vector Regression* (SVR) yang menghasilkan nilai pearson correlation sebesar 0,81. Pada penelitian lain Levenshtein Distance digunakan oleh Ariyani, dkk [6] untuk mengukur kemiripan dokumen dan menghasilkan nilai kemiripan yang antara 77% hingga 100% untuk dokumen yang tingkat kemiripannya tinggi, dan nilai kemiripan 40% untuk dokumen dengan tingkat kemiripan yang rendah. Dary [7] menggunakan Latent Semantic Analysis (LSA) untuk mencari nilai kesamaan antar kosep pada teks ayat Al-Quran dengan nilai akurasi maksimum yang didapatkan adalah 71% dengan F-measure terbaik adalah 40%.

Setelah penulis melakukan studi literatur, algoritma yang akan digunakan oleh penulis yaitu *cosine similarity* dan menggunakan pembobotan TF-IDF. Pembobotan TF-IDF digunakan untuk mencari representasi nilai tiap dokumen dalam koleksi. Sedangkan *cosine similarity* digunakan untuk menghitung nilai kemiripan antar kalimat dan menjadi salah satu teknik untuk mengukur kemiripan teks yang popular. Kelebihan dari algoritma *cosine similarity* adalah tidak terpengaruh pada panjang pendeknya suatu dokumen dan memiliki tingkat akurasi yang tinggi.

Tujuan dari penelitian ini adalah mengetahui nilai kemiripan sebuah dokumen dengan dokumen lainnya menggunakan metode *cosine similarity* dan pembobotan TF-IDF.

II. TINJAUAN PUSTAKA

A. Term Frequency Inverse Document Frequency (TF-IDF)

Metode TF-IDF merupakan metode untuk menghitung bobot suatu kata (*term*) terhadap dokumen. Metode ini juga terkenal efisien, mudah dan memiliki hasil yang akurat [8]. Metode ini menggabungkan dua konsep untuk perhitungan bobot, yaitu frekuensi kemunculan sebuah kata didalam sebuah dokumen tertentu dan *inverse* frekuensi dokumen yang mengandung kata tersebut [9]. Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukan seberapa penting kata tersebut di dalam dokumen. Metode ini akan menghitung bobot setiap *term* di dokumen dengan rumus [10]:

$$tfidf_{t,d} = tf_{t,d} \times idf_t \quad (1)$$

Dengan idf_t diperoleh dari

$$idf_t = \log(\frac{N}{df_t}) \quad (2)$$

A. *Cosine similarity*

Cosine similarity digunakan untuk melakukan perhitungan kesamaan dari dokumen. Rumus yang digunakan oleh *cosine similarity* adalah :

$$Cos\ a = \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^{n} A_i \times B}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} \quad (3)$$

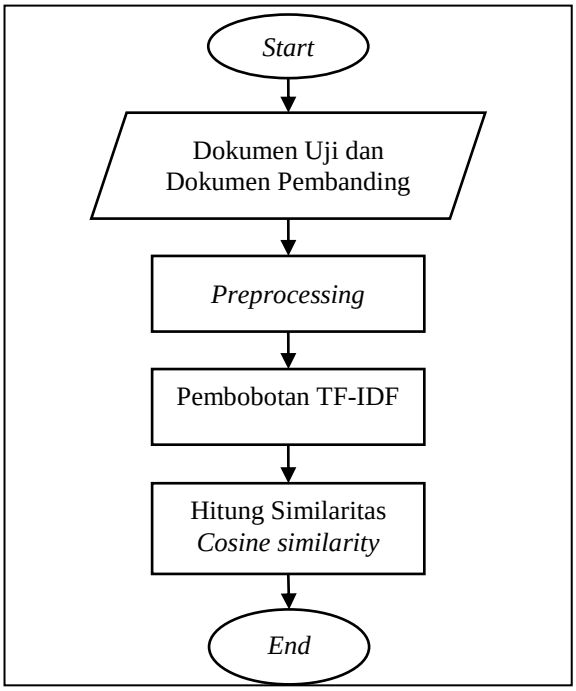
Keterangan :

A = Vektor A, yang akan dibandingkan kemiripan
B = Vektor B, yang akan dibandingkan kemiripan
A.B = dot product antara vektor A dan vektor B
$|A|$ = Panjang vektor A
$|B|$ = Panjang vektor B
$|A||B|$ = cross product antara $|A|\ dan\ |B|$

Eksperimen yang dilakukan pada penelitian ini yaitu membandingkan similaritas antar dokumen. Kemudian mencari nilai similaritas tertinggi antar dokumen. Apabila total similaritas yang didapatkan adalah nol (0) maka dokumen yang diolah tidak memiliki kesamaan dan jika nilai yang didapatkan maksimal adalah 1 maka dokumen tersebut memiliki kemiripan.

III. METODOLOGI PENELITIAN

Gambar 1 merupakan langkah-langkah yang penulis kerjakan untuk penelitian kemiripan dokumen:



Gambar 1. Bagan tahapan penelitian

A. Dokumen Uji dan Dokumen Pembanding

Dokumen uji adalah dokumen yang akan diukur tingkat kemiripannya. Sedangkan dokumen pembanding merupakan dokumen yang digunakan untuk mengukur kemiripan dari dokumen uji.

Dokumen yang digunakan dalam penelitian ini adalah teks abstrak tugas akhir di Institut Teknologi Telkom Purwokerto. Kumpulan dokumen tersebut akan menjadi bahan untuk membandingkan kemiripan antar dokumen.

B. Preprocessing

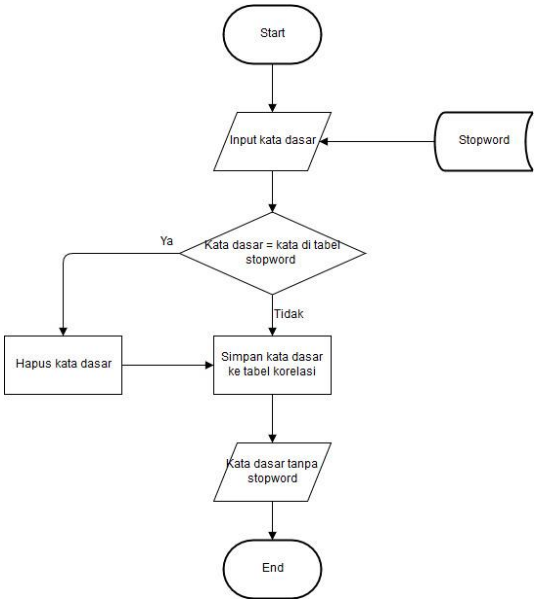
Tahapan yang dilakukan setelah menentukan dokumen uji dan dokumen pembanding selanjutnya harus dilakukan *preprocessing* terlebih dahulu. Berikut merupakan tahapan *preprocessing*:

1. *Case Folding*

Case folding merupakan proses pertama dari rangkaian *preprocessing* dokumen. Dalam proses ini mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf a sampai dengan z yang diterima [11].

2. Stopword removal

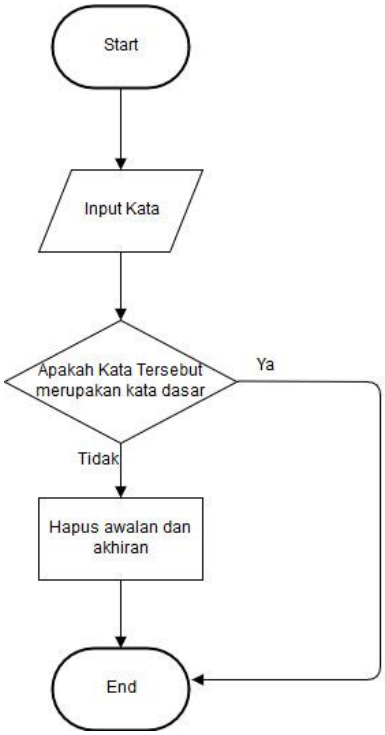
Tahap *stopword removal* adalah tahap menhilangkan kata-kata yang tidak penting dari teks. *Stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-words* [11]. Gambar 2 menunjukan *flowchart stopword removal*. Pada sistem penulis telah menyimpan kata-kata yang dianggap sebagai stopword dalam suatu file txt. Pada modul *stopword removal* ini, setiap kata dasar akan dicek oleh sistem apakah terdapat dalam kamus *stopword* atau tidak. Jika ada maka kata dasar tersebut akan dihapus dari list token.



Gambar 2. Flowchart proses *stopword removal*

3. Stemming

Tahap *stemming* adalah tahap mencari *root* kata dari tiap kata hasil *filtering* [12]. Pada tahap ini dilakukan proses pengembalian berbagai bentukan kata ke dalam suatu representasi yang sama. Pencarian kata dasar dilakukan dengan menghilangkan semua imbuhan dari kata, baik itu awalan, sisipan, maupun akhiran [12]. Gambar 3 menunjukan *flowchart* dari *stemming*. Penulis menggunakan modul sastrawi untuk bahasa python pada proses *stemming*.



Gambar 3. *Flowchart* proses *stemming*

4. Tokenizing

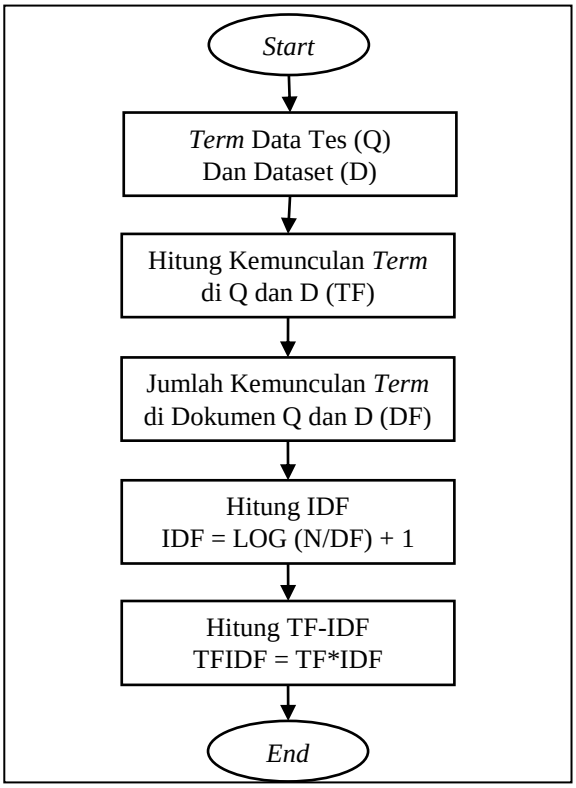
Tahap *Tokenizing* adalah tahap pemotongan string input berdasarkan tiap kata yang menyusunnya [12]. Karakter selain huruf akan dianggap *delimiter* dan akan dihilangkan atau dihapus untuk proses mendapat kata-kata penyusun teks.

C. Pembobotan TF-IDF

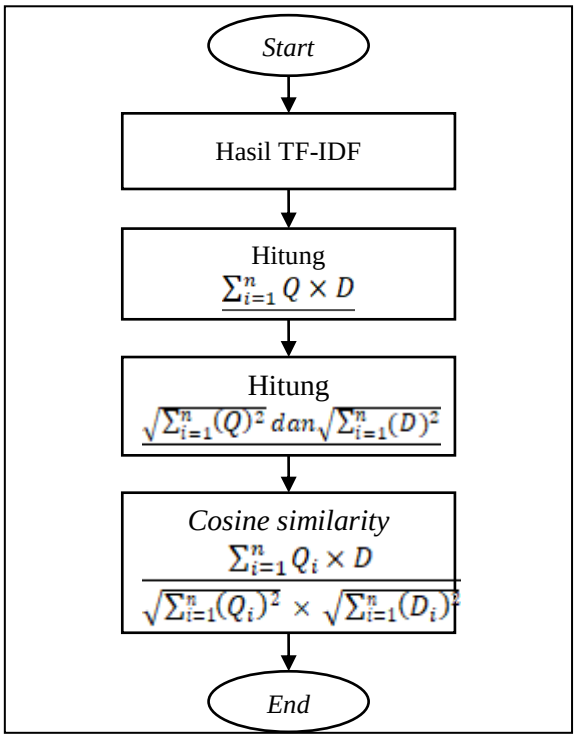
Pada tahapan ini dokumen testing dan dataset dokumen dilakukan pembobotan kata/*term* untuk menghitung frekuensi kemunculan setiap kata dokumen testing dalam masing-masing dokumen yang ada dalam dataset. Proses TF-IDF menggunakan bantuan model dari *sklearn.feature_extraction.text.* Gambar 4 menunjukkan *flowchart* process perhitungan bobot berdasarkan nilai TF-IDF.

D. Hitung Similaritas menggunakan Cosine similarity

Setelah dilakukan pembobotan TF-IDF hasil tersebut digunakan untuk menghitung kemiripan dokumen tes dengan dataset dokumen menggunakan *cosine similarity*. Flowchat *cosine similarity* dapat dilihat pada Gambar 5.



Gambar 4. Tahapan pembobotan TFIDF



Gambar 5. Tahapan Menghitung Similaritas dengan Cosine similarity

Dokumen yang digunakan dalam uji coba ini menggunakan dokumen abstrak yang ada di Institut Teknologi Telkom Purwokerto. 1 dokumen digunakan untuk dokumen uji dan 5 untuk dokumen pembanding.

TABEL I
DOKUMEN UJI (Q) DAN DOKUMEN PEMBANDING (D_N)

Nama	Dokumen	Keterangan
Q	Abstrak 37	Dokumen uji
D1	Abstrak 57	Dokumen Pembanding
D2	Abstrak 27	Dokumen Pembanding
D3	Abstrak 45	Dokumen Pembanding
D4	Abstrak 58	Dokumen Pembanding
D5	Abstrak 47	Dokumen Pembanding

Pada Tabel 1, Abstrak 37 atau yang diberi nama Q merupakan dokuemen uji yang digunakan untuk dicek kemiripannya dengan dokumen yang diberi nama D1, D2, D3, D4, D5.

Selanjutnya yaitu tahapan *preprocessing* dari semua dokumen Q, D1, D2, D3, D4, D5. *Preprocessing* dilakukan 2 kali percobaan yaitu menggunakan *stemming* dan tanpa *stemming* false. Pada tabel 2 merupakan hasil dari *preprocessing* dokumen Q:

TABEL III
HASIL PREPROCESSING

Dengan Stemming	Tanap Stemming
'ada', 'akses', 'alamat', 'aman', 'amat', 'analis', 'antri', 'aplikasi', 'bandwith', 'beberapa', 'beda', 'beri', 'besar', 'bidang', 'bit', 'buah', 'butuh', 'cocok', 'conference', 'conferencing', 'dapat', 'data', 'delay', 'dropped', 'dukung', 'efisiensi', 'enkripsi', 'fair', 'fifo', 'first', 'frekuensi', 'ftp', 'fungsi', 'ghzdengan', 'guna', 'hari', 'hasil', 'hingga', 'host', 'http', 'imbang', 'in', 'internet', 'ip', 'jadi', 'jaring', 'jenis', 'kata', 'kecil', 'keman', 'kembang', 'kerja', 'kirim', 'komunikasi', 'koneksi', 'kunci', 'lain', 'laku', 'layan', 'lebih', 'lihat', 'loss', 'makin', 'mana', 'manajemen', 'masing', 'mbps', 'meski', 'milik', 'mobile', 'mobilitas', 'modeler', 'ms', 'mulai', 'nilai', 'nomor', 'opnet', 'optimal', 'orang', 'out', 'packet', 'paket', 'paling', 'parameter', 'pasti', 'perlu',	'alamat', 'aman', 'antrian', 'aplikasi', 'bandwith', 'beberapa', 'bekerja', 'berada', 'berbeda', 'berpindah', 'bertujuan', 'berupa', 'besar', 'bit', 'cocok', 'conference', 'conferencing', 'data', 'delay', 'diakses', 'diamati', 'dianalisis', 'dibidang', 'digunakan', 'diimbangi', 'dilakukan', 'dimanapun', 'dipastikan', 'diperlukan', 'diterapkan', 'dropped', 'efisiensi', 'enkripsi', 'fair', 'fifo', 'first', 'frekuensi', 'ftp', 'fungsi', 'ghzdengan', 'hari', 'hasil', 'hingga', 'host', 'http', 'in', 'internet', 'ip', 'jaringan', 'jenis', 'kata', 'kebutuhan', 'kecil', 'kemanan', 'ketiga', 'komunikasi', 'kunci', 'lainnya', 'layanan', 'lebih', 'loss', 'mana', 'manajemen', 'masing', 'mbps', 'melakukan', 'memberikan', 'memiliki', 'mempunyai', 'menawarkan',

'pesat', 'pindah', 'pq', 'priority', 'protocol', 'protokol', 'punya', 'queuing', 'rubah', 'rupa', 'salah', 'sama', 'satu', 'semua', 'simulasi', 'sistem', 'skenario', 'software', 'suatu', 'tahu', 'tawar', 'teknologi', 'teliti', 'teori', 'terap', 'terus', 'throughput', 'tiga', 'tinggi', 'tingkat', 'traffic', 'trafik', 'tuju', 'ubah', 'unggul', 'upa', 'user', 'variation', 'video', 'voip', 'vpn', 'wan', 'weighted', 'wfq', 'wifi', 'wighted', 'wireless'	'mendapatkan', 'mendukung', 'mengetahui', 'menggunakan', 'meningkatnya', 'merubah', 'merupakan', 'meskipun', 'mobile', 'mobilitas', 'modeler', 'ms', 'mulai', 'nilai', 'nilainya', 'nomor', 'opnet', 'optimal', 'orang', 'out', 'packet', 'paket', 'paling', 'parameter', 'penelitian', 'pengamatan', 'pengguna', 'pengirim', 'pengiriman', 'perkembangan', 'perpindahan', 'perubahan', 'pesat', 'pq', 'priority', 'protocol', 'protokol', 'queuing', 'salah', 'sama', 'satunya', 'sebesar', 'sebuah', 'semakin', 'semua', 'simulasi', 'sistem', 'skenario', 'software', 'suatu', 'teknologi', 'teori', 'terjadi', 'terkoneksi', 'terlihat', 'terus', 'throughput', 'tinggi', 'traffic', 'trafik', 'tujuan', 'unggul', 'user', 'variation', 'video', 'voip', 'vpn', 'wan', 'weighted', 'wfq', 'wifi', 'wighted', 'wireless'

Hasil dari *preprocessing* adalah *list token* yang sudah bersih dari non-alphabet dan *stopword*. Penulis membuat 2 list token hasil preprocessing, yaitu *list token* dengan tambahan proses *stemming* dan tanpa *stemming*.

Kumpulan term hasil *preprocessing* tersebut digunakan pada proses pembobotaan dengan TF-IDF. Hasil nilai similarity menggunakan Cosine Similarity menggunakan dokumen Q sebagai dokumen uji dan D1,D2,D3,D4,D5 sebagai dokumen pembandinng hasil kemiripannya terdapat pada tabel 3.

TABEL IIIII
PERBANDINGAN NILAI KEMIRIPAN

	Jumlah *Term*	**Nilai Kemiripan (*stemming*)**	**Nilai Kemiripan (Tidak *Stemming*)**
D1	133	0.56700796	0.46381327
D2	138	0.16836027	0.05986167
D3	123	0.51348949	0.35700185
D4	157	0.17296931	0.07973601
D5	127	0.2566103	0.16843501

Berdasarkan Tabel 3, hasil kemiripan jika suatu dokumen menggunakan *stemming* lebih besar tingkat kemiripannya, sedangkan jika tidak dilakukan *stemming* nilai kemiripan dengan dokumen Q lebih rendah. perbedaan *stemming* dan tidak dilakukan *stemming* D1 sebesar 0.10319469,D2 sebesar 0.1084986, D3 sebesar 0.15648764, D4 sebesar 0.0932333, D5 sebesar 0.08817529.

V. KESIMPULAN

Dari hasil pengujian dan analisis maka dapat disimpulkan dalam penelitian ini adalah algoritma cosine similarity dan pembobotan TF-IDF telah berhasil mendeteksi kemiripan suatu dokumen. Proses *stemming* pada *preprocessing* sangat berpengaruh terhadap nilai kemiripan hasil jika dilakukan *stemming* lebih tinggi. Nilai rata-rata perbedaan nilai kemiripan saat dilakukan *stemming* dan tidak dilakukan *stemming* adalah 10 %. Kekurangan jika dilakukan proses *stemming* adalah waktu untuk pemrosesan lebih lama dibandingkan tidak dilakukan proses *stemming*.

REFERENSI

[1] H. Santoso, "Pencegahan dan Penanggulangan Plagiarisme dalam Penulisan Karya Ilmiah di Lingkungan Perpustakaan Perguruan Tinggi," *http://library.um.ac.id*, 2015. .

[2] R. T. Wahyuni, D. Prastiyanto, dan E. Supraptono, "Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi," *J. Tek. Elektro Univ. Negri Semarang*, vol. 9, no. 1, hal. 18–23, 2017.

[3] Sugiyamto, B. Surarso, dan A. Sugiharto, "Analisa Performa Metode Cosine dan Jacard pada Pengujian Kesamaan Dokumen," *J. Masy. Inform.*, vol. 5, no. 10, hal. 1–8, 2014.

[4] Okfalisa dan A. H. Harahap, "Implementasi Metode Terms Frequency–Inverse Document Frequency (Tf-Idf) Dan Maximum Marginal Relevance Untuk Monitoring Diskusi Online," *J. Sains, Teknol. dan Ind.*, vol. 13, no. 2, hal. 6–19, 2016.

[5] A. W. Z. Nasution, M. A. Bijaksana, dan S. Al Faraby, "Analisis dan Implementasi Perhitungan Semantics Similarity Pada Ayat Al-Quran Dengan Pendekatan Word Alignment Berdasarkan Support Vector Regression," in *e-Proceeding of Engineering*, 2017, vol. 4, no. 2, hal. 3156–3165.

[6] N. H. Ariyani, Sutardi, dan R. Ramadhan, "Aplikasi Pendeteksi Kemiripan Isi Teks Dokumen Menggunakan Metode Levenshtein Distance," *semanTIK*, vol. Vol 2, no. 1, hal. 279–286, 2016.

[7] M. I. Dary, "Analisis dan Implementasi Short Text Similarity dengan Metode Latent Semantic Analysis Untuk Mengetahui Kesamaan Ayat al-Quran," *eProceedings Eng.*, vol. 2, no. 3, 2015.

[8] A. A. Maarif, "Penerapan Algoritma Tf-Idf Untuk Pencarian Karya Ilmiah," Semarang, 2015.

[9] M. Nurjannah, Hamdani, dan I. F. Astuti, "Penerapan Algoritma Term Frequency-Inverse Document Frequency (TF-IDF) untuk Text Mining," *J. Inform. Mulawarman*, vol. 8, no. 3, hal. 110–113, 2013.

[10] C. D. Manning, P. Raghavan, dan H. Schütze, *An Introduction to Information Retrieval*. Cambridge: Cambridge University Press, 2009.

[11] R. Feldman dan J. Sanger, *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. New York: Cambridge University Press, 2007.

[12] L. Francis dan M. Flynn, "Text Mining Handbook," *Casualty Actuarial Society E-Forum*. 2010.