

Penggunaan *Information Retrieval* untuk Mendeteksi Kesamaan Judul Skripsi dengan *Modified Cosine Similarity*

^{1,2,3}Sri Wahyuni, ^{2*}Asrul Abdullah, ³Sucipto

^{1,2,3}Teknik Informatika, Universitas Muhammadiyah Pontianak, Pontianak, Indonesia

Email : riwahyuni.spy88@gmail.com, asrul.abdullah@unmuhpnk.ac.id, sucipto@unmuhpnk.ac.id

ARTICLE INFO

Article History:

Received : 17-03-2025
Revised : 30-05-2025
Accepted : 04-06-2025
Online : 10-06-2025

Keywords:

Information Retrieval
TF-IDF
Cosine Similarity
Plagiarism
Thesis Title



ABSTRACT

Abstract: *This research develops a web-based system to detect the similarity of thesis titles using the Modified Cosine Similarity method and TF-IDF weighting. This system helps students in evaluating the similarity of titles automatically, so as to avoid plagiarism and increase the originality of research. The process used includes text preprocessing (case folding, tokenizing, stopword removal, stemming), TF-IDF calculation for word weighting, and the use of Modified Cosine Similarity to measure the level of similarity between titles. The test results show that the system is able to identify the similarity of titles with a 100% recall rate, where titles with >70% similarity need to be revised, 31%-70% similarity can be clarified by adding words, and ≤30% similarity is potentially accepted as an original title. With this implementation, it is expected that students can more easily determine a thesis title that is unique and in accordance with academic standards.*

Abstrak: Penelitian ini mengembangkan sistem berbasis web untuk mendeteksi kemiripan judul skripsi menggunakan metode *Modified Cosine Similarity* dan pembobotan TF-IDF. Sistem ini membantu mahasiswa dalam mengevaluasi kemiripan judul secara otomatis, sehingga dapat menghindari *plagiarisme* dan meningkatkan orisinalitas penelitian. Proses yang digunakan meliputi *text preprocessing* (*case folding, tokenizing, stopword removal, stemming*), perhitungan TF-IDF untuk pembobotan kata, dan penggunaan *Modified Cosine Similarity* untuk mengukur tingkat kemiripan antar judul. Hasil pengujian menunjukkan bahwa sistem mampu mengidentifikasi kemiripan judul dengan tingkat *recall* 100%, di mana judul dengan kemiripan >70% perlu direvisi, kemiripan 31%-70% dapat diperjelas dengan penambahan kata, dan kemiripan ≤30% berpotensi diterima sebagai judul orisinal. Dengan implementasi ini, diharapkan mahasiswa dapat lebih mudah menentukan judul skripsi yang unik dan sesuai dengan standar akademik.



<https://doi.org/10.31764/justek.vXIY.ZZZ>



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license

A. LATAR BELAKANG

Perkembangan teknologi informasi telah membawa dampak signifikan di berbagai bidang, termasuk pendidikan (Septiani & Isabela, 2022). Salah satu produk akademik yang dihasilkan mahasiswa adalah skripsi, yang menjadi indikator kelulusan. Penentuan judul merupakan tahap awal yang penting (Herlambang et al., 2021). di mana referensi penelitian sebelumnya sering dimanfaatkan untuk memperkuat konteks topik (Ahmad et al., 2021).

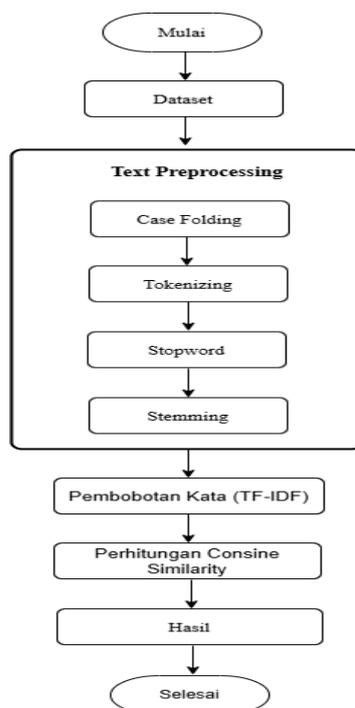
Namun, tidak sedikit mahasiswa yang kesulitan menentukan judul dan berpotensi menggunakan karya orang lain secara sengaja ataupun tidak sengaja (Kurnia Aini, 2022). Di sisi lain, hal ini dapat memicu tingkat kemiripan judul yang tinggi dan menyebabkan kurangnya keragaman tema atau judul penelitian. Permasalahan ini penting untuk segera diatasi karena jika dibiarkan, dapat mengurangi nilai orisinalitas karya ilmiah, memicu isu etika akademik, serta mempersempit keberagaman penelitian yang ada di lingkungan perguruan tinggi. Oleh karena itu, dibutuhkan sistem pendeteksi kesamaan judul skripsi yang efektif, cepat, dan akurat untuk membantu mahasiswa maupun pihak kampus dalam melakukan pengecekan.

Dalam dunia akademik, sistem temu kembali informasi telah dimanfaatkan untuk mendeteksi kemiripan teks, termasuk judul skripsi (Ihsan Maulana & Abdillah, 2022). Beberapa metode yang biasanya dimanfaatkan dalam mencari kesamaan dokumen antara lain *cosine similarity* dan *Rabin-Karp*. *Cosine similarity* bekerja dengan mengukur sudut antara dua vektor dalam ruang multidimensi (Via & Mumpuni, 2019). Untuk meningkatkan akurasi dalam mendeteksi kesamaan dokumen, perlu dukungan metode TF-IDF yang membantu menekankan kata-kata penting dalam teks (Putri Gabriella, 2023). Beberapa penelitian sebelumnya telah menerapkan metode *cosine similarity* tersebut dalam berbagai konteks, salah satunya penelitian oleh Suzanti *et al.* (2023) yang memanfaatkan *Modified Cosine Similarity* dengan TF-IDF untuk meningkatkan relevansi pencarian informasi di bidang pariwisata. Hasil penelitian tersebut menunjukkan efektivitas metode ini dalam meningkatkan presisi, *recall*, dan F-measure pencarian. Namun, hingga saat ini belum ditemukan penelitian serupa yang menerapkan metode *Modified Cosine Similarity* khusus untuk mendeteksi kemiripan judul skripsi mahasiswa. Hal ini menunjukkan adanya kesenjangan (*gap analysis*) dalam penelitian sebelumnya yang masih terbatas pada konteks non-akademik atau dokumen panjang, sedangkan skripsi memiliki karakteristik judul yang lebih pendek dan minim konteks (Mirs, 2023).

Berdasarkan permasalahan tersebut, kebaruan (*novelty*) dari penelitian ini terletak pada pengembangan aplikasi berbasis web yang mengimplementasikan metode *Modified Cosine Similarity* dikombinasikan dengan TF-IDF untuk mendeteksi kesamaan judul skripsi (Alamsyah *et al.*, 2019). Kebaruan ini terletak pada konteks penerapannya di lingkungan akademik khususnya perguruan tinggi, dengan sistem yang mampu melakukan pengecekan judul skripsi secara otomatis, cepat, dan akurat. Dengan adanya sistem ini, diharapkan membantu mahasiswa dan pihak akademik dapat lebih mudah menemukan judul skripsi yang unik serta sesuai dengan standar akademik, sehingga dapat menghindari plagiarisme dan meningkatkan kualitas penelitian (Nuzul Hikmah *et al.*, 2022).

B. METODE PENELITIAN

Tahapan dalam penelitian ini memanfaatkan *Information Retrieval* untuk mendeteksi kemiripan judul skripsi berbasis teks. Tahapan penelitian yang dilakukan terdiri dari beberapa langkah sebagai berikut (Dara Jayanti & Masruri, 2024).



Gambar 1. Diagram alir tahapan sistem

1. Dataset

Mempersiapkan dataset adalah salah satu prosedur yang diperlukan. Data yang dikumpulkan dan disajikan dalam ekstensi .csv. File CSV ini dipilih karena memiliki banyak keistimewaan, seperti ringan, ukuran relatif kecil, dan sederhana (Ahmad, Nuraini, Arienda, Addis Prasetyo, 2021).

2. Text Preprocessing

Dalam skripsi ini akan dilakukan tiga tahap yaitu *case folding*, *text tokenizing*, *stopword removal* dan *stemming* (Mi'andri et al., 2022).

a. Case Folding

Setiap dokumen atau kumpulan data teks akan melalui tahap normalisasi, yaitu mengonversi seluruh huruf menjadi huruf kecil dan mengabaikan simbol, tanda baca, serta spasi. (Al Rasyid & Ningsih, 2024).

b. Text tokenizing

Tokenisasi adalah proses mengubah teks atau input string menjadi fragment bagian yang lebih kecil, seperti kata atau frasa. Tokenisasi sangat penting dalam analisis data karena membuat analisis data menjadi lebih efektif dan efisien (Afandi et al., 2024).

c. Text Filtering

Text filtering adalah proses menghapus kata-kata yang tidak perlu atau kata-kata yang muncul dalam stopwords, yaitu kata-kata yang sering muncul dan diberikan agar dapat digunakan secara efektif dalam penulisan (Nurmalasari & Nengsih, 2021).

d. stemming

Stemming digunakan untuk menghapus afiks dari sehingga diperoleh kata dasar (Hidayatulloh, 2020).

3. TF-IDF (*Term Frequency-Inverse Document Frequency*)

Setelah langkah *preprocessing*, data harus direpresentasikan secara numerik. Gunakan metode pembobotan TF-IDF untuk mengubah data di atas ke dalam format numerik (Deolika et al., 2019). Metode TF-IDF merupakan metode umum untuk menghitung bobot kata dalam sistem temu kembali informasi dengan menggabungkan frekuensi kemunculan term dalam dokumen dan frekuensi *invers* dokumen yang memuat *term* tersebut. (Al Rasyid & Ningsih, 2024).

Metode TF-IDF memperhitungkan dua faktor penting sebagai berikut:

- a. **Term Frequency (TF)** yaitu mengukur frekuensi kemunculan suatu kata dalam sebuah dokumen. Dihitung dengan rumus sebagai berikut.

$$TF(t, d) = \frac{n(t,d)}{N(d)} \quad (1)$$

di mana $n(t, d)$ adalah jumlah kemunculan kata kunci t dalam dokumen d , dan $N(d)$ adalah total jumlah kata dalam dokumen tersebut.

- b. **Inverse Document Frequency (IDF)** untuk mengurangi bias akibat kata yang terlalu umum, *invers document frequency* menilai seberapa unik koleksi kata dalam sebuah dokumen (Kosasih & Alberto, 2021). Rumus IDF sebagai berikut.

$$IDF(t) = \log \frac{N}{df(t) + 1} \quad (2)$$

di mana N adalah jumlah dokumen dalam koleksi secara keseluruhan dan $df(t)$ dengan kata kunci t .

Nilai TF-IDF dikalikan untuk menghasilkan bobot kata yang mencerminkan tingkat kepentingannya dalam suatu dokumen dibandingkan dengan koleksi dokumen lainnya (Priandono et al., 2020). Dengan rumus sebagai berikut.

$$W = TF(t, d) \times IDF(t) \quad (3)$$

Bobot ini membantu dalam analisis teks, seperti pencarian informasi dan deteksi kemiripan dokumen

4. Cosine Similarity

Untuk menentukan seberapa mirip dua dokumen, *cosine similarity* adalah salah satu metode yang sering digunakan dalam analisis teks dan text mining, yang tidak terpengaruh oleh panjang atau pendeknya dokumen, serta memiliki tingkat akurasi yang tinggi. Metode ini bergantung pada korelasi terbalik yang ada antara dua *vector* dalam ruang multidimensi (Afandi et al., 2024). Rumus *Cosine Similarity* memiliki rumus sebagai berikut.

$$Similarity = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (4)$$

Keterangan	:
A	= Vektor A
B	= Vektor B
A _i	= Bobot <i>term</i> i dalam blok A _i
B _i	= Bobot <i>term</i> i dalam blok B _i
i	= indeks elemen dalam vector
n	= jumlah elemen vector

C. HASIL DAN PEMBAHASAN

Sampel dokumen berikut diproses menggunakan *cosine similarity* yang ada pada sistem dengan tahapan penelitian berikut.

1. Dataset

Data berupa 241 judul skripsi mahasiswa angkatan 2019–2024 disimpan dalam Microsoft Excel dan diinput ke aplikasi deteksi similarity. Sampel perhitungan diambil dari 5 dokumen dalam format csv.

Query = “Perbandingan Algoritma Svm Dan *Naive Bayes* Untuk Klasifikasi Kebakaran Hutan Dan Lahan”

D1 = “Perbandingan Akurasi Algoritma *Naive Bayes* Dan C4.5 Untuk Identifikasi Penyakit Kanker Paru-Paru”

D2 = “Klasifikasi Tingkat Rawan Kebakaran Hutan Berdasarkan Penyebaran (Hotspot) Di Kalimantan Barat Menggunakan Metode *Scoring* Berbasis Webgis”

D3 = “Klasifikasi Algoritma *K-Nearest Neighbor* Dan *Naive Bayes* Untuk Kelayakan Pemberian Kredit Perbankan”

D4 = “Klasifikasi Penyakit Infeksi Saluran Pernapasan Akut (Ispa) Menggunakan Metode *Naive Bayes* Pada Puskesmas Siduk Kayong Utara”

D5 = “Perbandingan Model Arsitektur Cnn Dengan Metode *Transfer Learning* Untuk Klasifikasi Spesies Ikan Laut”

2. Hasil *Text preprocessing*

Dataset sampel selanjutnya disaring untuk memastikan bahwa bidang deskripsi/konten dapat digunakan untuk melakukan proses kemiripan. Proses *preprocessing* mencakup *case folding*, *stopword removal*, *tokenizing*, dan *stemming*. Di bawah ini contoh input *query* serta dokumen sasaran dalam tahap pra-pemrosesan.

Tabel 1. Hasil *text preprocessing*

Database	Kata Kunci
Query	'banding', 'algoritma', 'svm', 'naive', 'bayes', 'klasifikasi', 'bakar', 'hutan', 'lahan'
Doc 1	'banding', 'akurasi', 'algoritma', 'naive', 'bayes', 'c4.5', 'identifikasi', 'penyakit', 'kanker', 'paru-paru'
Doc 2	'klasifikasi', 'tingkat', 'rawan', 'bakar', 'hutan', 'sebar', 'hotspot', 'kalimantan', 'barat', 'metode', 'scoring', 'basis', 'webgis'
Doc3	'klasifikasi', 'algoritma', 'k-nearest', 'neighbor', 'naive', 'bayes', 'layak', 'beri', 'kredit', 'perban'

Doc 4	'klasifikasi', 'penyakit', 'infeksi', 'salur', 'napas', 'akut', 'ispa', 'metode', 'naive', 'bayes', 'puskesmas', 'siduk', 'kayong', 'utara'
Doc 5	'banding', 'model', 'arsitektur', 'cnn', 'metode', 'transfer', 'learning', 'klasifikasi', 'spesies', 'ikan', 'laut'

3. Hasil TF-IDF

Setelah dokumen melalui tahap preprocessing, langkah berikutnya adalah menerapkan bobot vektor TF-IDF berdasarkan kata kunci dalam kueri. Berikut ini adalah hasil pembobotan TF-IDF pada contoh dokumen seperti yang terlihat pada gambar 2 dan tabel 2.

Term (t)	Q	TF(t,d) = n(t,d) / N(d)					df	N	IDF = N/df
		D1	D2	D3	D4	D5			
banding	0.1111	0.1	0	0	0	0.0909	4	85	1,2304
Algoritma	0.1111	0.1	0	0.1	0	0	14	85	0,7533
Svm	0.1111	0	0	0	0	0	0	85	0
Naive	0.1111	0.1	0	0.1	0.0714	0	3	85	1,3274
Bayes	0.1111	0.1	0	0.1	0.0714	0	3	85	1,3274
Klasifikasi	0.1111	0	0,0769	0.1	0,0714	0,0909	10	85	0,8888
Bakar	0.1111	0	0,0769	0	0	0	2	85	1,4524
Hutan	0.1111	0	0,0769	0	0	0	1	85	1,6284
Lahan	0.1111	0	0	0	0	0	0	85	0

Gambar 2. Proses pencarian nilai TF dan IDF

Setelah mendapatkan masing-masing nilai hasil dari TF dan IDF, selanjutnya menghitung proses pencarian nilai bobot dengan mengalikan nilai TF dan IDF seperti yang terlihat pada tabel 2.

Tabel 2. Proses pencarian nilai bobot (W)

Term (t)	W = TF(t, d) x IDF(t)					
	Q	D1	D2	D3	D4	D5
banding	0,1367	0,123	0	0	0	0,1119
Algoritma	0,0837	0,0753	0	0,0753	0	0
Svm	0	0	0	0	0	0
Naive	0,1475	0,1327	0	0,1327	0,0948	0
Bayes	0,1475	0,1327	0	0,1327	0,0948	0
Klasifikasi	0,0987	0	0,0683	0,0888	0,0634	0,0807
Bakar	0,1614	0	0,1117	0	0	0
Hutan	0,1809	0	0,1253	0	0	0
Lahan	0	0	0	0	0	0
Dot Product vector 1 & vector 2 = Ai . Bi						
Dot Product sum (Q*D)	0,0623	0,0474	0,0543	0,0341	0,0233	

Keterangan:

Term = Kata

Q = Query

N = Jumlah dokumen

TF = Nilai berdasar banyaknya term yang muncul pada dokumen

DF = Jumlah term yang dalam satu dokumen

IDF = Jumlah *term* pada seluruh dokumen

W = Bobot *term* pada dokumen

4. Hasil Cosine Similarity

Contoh perhitungan kemiripan dilakukan dengan membandingkan D1 terhadap D2–D5. Langkah pertama adalah menghitung panjang vektor masing-masing dokumen menggunakan hasil pembobotan kata, dilanjutkan dengan mengalikan bobot antar dokumen, menjumlahkan hasilnya, mengkuadratkan bobot setiap term, menjumlahkan nilai kuadrat, lalu mengambil akarnya.

Tabel 3. Perhitungan skalar vector

Term (t)	Magnitude vector = $(W(Q, D))^2$							
	Q^2	$D1^2$	$D2^2$	$D3^2$	$D4^2$	$D5^2$		
banding	0,0187	0,0151	0	0	0	0,0125		
Algoritma	0,007	0,0057	0	0,0057	0	0		
Svm	0	0	0	0	0	0		
Naive	0,0218	0,0176	0	0,0176	0,0089	0		
Bayes	0,0218	0,0176	0	0,0176	0,0089	0		
Klasifikasi	0,0097	0	0,0047	0,0079	0,004	0,0065		
Bakar	0,026	0	0,0125	0	0	0		
Hutan	0,0327	0	0,0157	0	0	0		
Lahan	0	0	0	0	0	0		
SUM	0,1377	0,056	0,0329	0,0488	0,0218	0,019		
	$\sqrt{\sum_{i=1}^n A_i^2}$	$\sqrt{\sum_{i=1}^n B_i^2}$	0,3711	0,2366	0,1813	0,2209	0,1476	0,1378

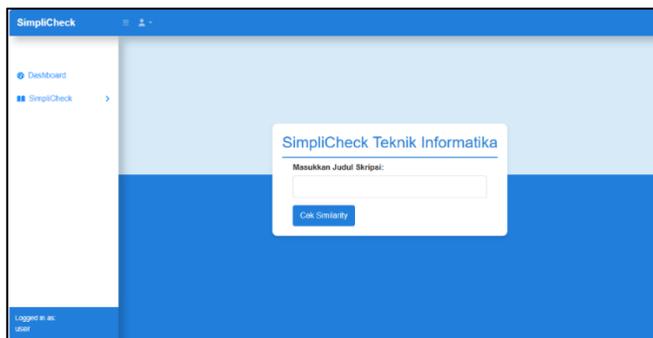
Setelah mendapatkan nilai perhitungan skalar *vector term* dan dokumen, langkah selanjutnya menghitung kemiripan dari suatu dokumen tersebut dengan *query*. Perhitungan kemiripan dapat dilihat pada tabel 4.

Tabel 4. Perhitungan similarity

Kemiripan	Similarity		Persentase
	= $\frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$		
Sim (D1, Q)	$\frac{0,0623}{0,3711 * 0,2366}$	$= \frac{0,0623}{0,0878} = 0,7088$	70,88%
Sim (D2, Q)	$\frac{0,0474}{0,3711 * 0,1813}$	$= \frac{0,0474}{0,0673} = 0,7043$	70,43%
Sim (D3, Q)	$\frac{0,0543}{0,3711 * 0,2209}$	$= \frac{0,0543}{0,0819} = 0,6630$	66,30%
Sim (D4, Q)	$\frac{0,0341}{0,3711 * 0,1476}$	$= \frac{0,0341}{0,0548} = 0,6223$	62,23%
Sim (D5, Q)	$\frac{0,0233}{0,3711 * 0,1378}$	$= \frac{0,0233}{0,0511} = 0,4560$	45,60%

5. Hasil Pengujian sistem

Pengujian ini memanfaatkan sampel data yang serupa dengan data yang terdapat dalam dataset dokumen. Hasilnya adalah sebagai berikut.



Gambar 3. Implementasi halaman *simplicheck*

Di halaman ini, mahasiswa cukup memasukkan judul skripsi ke dalam kolom teks "Judul Skripsi" dan menekan tombol "Cek *Similarity*" untuk menganalisis serta membandingkan kemiripan kueri dengan judul skripsi referensi yang tersimpan dalam database.

Judul yang dimasukkan:
Perbandingan Algoritma Svm Dan Naive Bayes Untuk Klasifikasi Kebakaran Hutan Dan Lahan

Hasil Recall:

Jumlah dokumen relevan yang ditemukan: 0
Total dokumen relevan dalam database: 3
Hasil recall adalah: 0
Recall: 0%

Keterangan:

- Merah: Kemiripan > 70% (Mip)
- Oranye: Kemiripan 30% - 70% (Sedang)
- Hijau: Kemiripan < 30% (Tidak Mip)

Hasil Pengecekan SimpliCheck:

Nama	NIM	Profil	Tahun	Judul Skripsi	Persentase Kemiripan	Relevansi (Ground Truth)	Relevansi (Terdeteksi)
Rendi Kurniawan	181220054	Teknik Informatika	2023	Perbandingan Akurasi Algoritma Naive Bayes Dan C4.5 Untuk Identifikasi Penyakit Kanker Papan Paras	79.89%	Tidak Relevan	Relevan
Yulham Elia Saputra	181220055	Teknik Informatika	2023	Klasifikasi Tingkat Rawan Kebakaran Hutan Berdasarkan Penyebaran (Hotspot) Di Kalimantan Barat Menggunakan Metode Scoring Berbasis Webgis	78.63%	Tidak Relevan	Relevan
Elma Ayyari	191220063	Teknik Informatika	2023	Klasifikasi Algoritma K-Nearest Neighbor Dan Naive Bayes Untuk Kelayakan Pemberian Kredit Perbankan	66.14%	Tidak Relevan	Relevan
Oktaviani	181220059	Teknik Informatika	2023	Klasifikasi Penyakit Infeksi Saluran Pernapasan Akut (Ispa) Menggunakan Metode Naive Bayes Pada Puskesmas Sialak Kuyung Utara	62.16%	Tidak Relevan	Relevan
Iham Anugrah	181220066	Teknik Informatika	2023	Perbandingan Model Analektor Cnn Dengan Metode Transfer Learning Untuk Klasifikasi Spesies Ikan Laut	46.47%	Tidak Relevan	Tidak Relevan
Dzul Zulfahid	171220274	Teknik Informatika	2022	Pemanfaatan Iot Sebagai Sistem Deteksi Dini Kebakaran Dengan Sensor Api Dan Sensor Suhu Berbasis Arduino	43.48%	Relevan	Tidak Relevan

Gambar 4. Implementasi halaman hasil *simplicheck*

Setelah proses perhitungan kemiripan selesai, aplikasi akan menampilkan daftar dokumen yang memiliki tingkat kesamaan dengan kueri yang diinput. Hasil ini menunjukkan persentase kemiripan similarity, sebagaimana ditampilkan pada Gambar 4. Sistem ini, diuji dalam dua skenario. Situasi pertama melibatkan membandingkan hasil dari dokumen yang sama dengan dokumen yang ada di sistem dan hasil perhitungan. Dalam uji coba ini, sistem menunjukkan persentase recall 100% untuk dokumen yang identik dan kurang dari 100% untuk dokumen yang memiliki kemiripan dengan ambang batas tertentu. Hasil dari kedua skenario tersebut menunjukkan bahwa aplikasi dapat berjalan dengan baik dengan tingkan aktivitas pengguna dan fungsinya. Berdasarkan hasil uji coba aplikasi dapat berjalan dengan baik dengan fungsi yang sesuai.

D. SIMPULAN DAN SARAN

Penelitian ini mengembangkan sistem berbasis web untuk mendeteksi kemiripan judul skripsi menggunakan TF-IDF dan cosine similarity. Proses preprocessing seperti stemming dan stopword removal meningkatkan akurasi. Sistem mengklasifikasikan tingkat kemiripan menjadi tinggi (>70%), sedang (31%–70%), dan rendah (≤30%). Uji black box menunjukkan seluruh fungsi berjalan optimal dengan recall 100%, membuktikan sistem ini akurat dan andal dalam mendeteksi kesamaan judul.

Saran untuk penelitian yang akan datang adalah dapat meningkatkan dengan fitur tambahan seperti pengecekan abstrak dan pengujian pada dataset lebih besar. Penggunaan *Word2Vec* atau *BERT* dapat meningkatkan pemahaman konteks kata, sementara integrasi dengan *thesaurus* atau *ontologi* akademik membantu mengenali sinonim. Kombinasi *cosine similarity* dengan metode lain seperti *Jaccard Similarity* atau *LSA* juga berpotensi meningkatkan akurasi deteksi kemiripan.

UCAPAN TERIMA KASIH

Penulis mengucapkan puji syukur kepada Tuhan Yang Maha Esa yang telah memberikan rahmat-Nya sehingga penulis dapat menulis karya ilmiah ini dengan baik. Ucapan terima kasih juga disampaikan kepada Orang Tua penulis yang telah memberikan ketenangan dan bimbingan sehingga selalu dalam keadaan sehat dan diberikan keselamatan dalam setiap aktivitasnya. Ucapan terima kasih juga penulis sampaikan kepada semua pihak yang terlibat sehingga penelitian ini dapat berjalan dengan baik, terutama kepada Bapak Asrul Abdullah, S.Kom., M.Cs. dan Bapak Sucipto, M.Kom. Atas saran, serta masukan dalam penyelesaian penelitian ini.

REFERENSI

- Afandi, M. D., Homaidi, A., Ghofur, A., & Zubairi, A. (2024). Penerapan Information Retrieval dalam Sistem Analisis Kemiripan Proposal Skripsi menggunakan Cosine Similarity. *Swabumi*, 12(1), 39–46. <https://doi.org/10.31294/swabumi.v12i1.17087>
- Ahmad, Nuraini, Arienda, Addis Prasetyo, A. M. (2021). Penerapan Information Retrieval Pada Search Engine. *KNOWLEDGE : Jurnal Inovasi Hasil Penelitian Dan Pengembangan*, 1(no.1), 15–23.
- Ahmad, I., Indra Borman, R., Caksana, G. G., & Fakhrurozi, J. (2021). *SINTECH Journal | 53 Implementasi String Matching Dengan Algoritma Boyer-Moore Untuk Menentukan Tingkat Kemiripan Pada Pengajuan Judul Skripsi/Ta Mahasiswa (Studi Kasus: Universitas XYZ)*. <https://doi.org/10.31598>
- Al Rasyid, R., & Ningsih, D. H. U. (2024). Penerapan Algoritma TF-IDF dan Cosine Similarity untuk Query Pencarian Pada Dataset Destinasi Wisata. *Jurnal JTik (Jurnal Teknologi Informasi Dan Komunikasi)*, 8(1), 170–178. <https://doi.org/10.35870/jtik.v8i1.1416>
- Alamsyah, N., Rasyidan, M., Kalimantan, I., Al, M. A., & Banjarmasin, B. (2019). Deteksi Plagiarisme Tingkat Kemiripan Judul Skripsi Pada Fakultas Teknologi Informasi Menggunakan Algoritma Winnowing. In *Technologia* (Vol. 10, Issue 4). Oktober-Desember.
- Deolika, A., Kusriani, K., & Luthfi, E. T. (2019). Analisis Pembobotan Kata Pada Klasifikasi Text Mining. *Jurnal Teknologi Informasi*, 3(2), 179. <https://doi.org/10.36294/jurti.v3i2.1077>
- Hidayatulloh, M. S. (2020). Sistem Temu Kembali Informasi Dinas Pemuda Olahraga dan Pariwisata Kabupaten Purbalingga dengan Metode Vector Space Model. *Journal of Informatics, Information System, Software Engineering and Applications (INISTA)*, 2(2), 1–10. <https://doi.org/10.20895/inista.v2i2.101>
- H. Herlambang, J. Suwita, and B.Tiara, "Analisa Dan Perancangan Sistem Pendeteksi Plagiarisme Skripsi Pada STMIK Insan Pembangunan Menggunakan Metode Cosine Similarity," *Jurnal IPSIKOM*, vol. 9, no. 1, 2021.
- Ihsan Maulana, T., & Abdillah, R. (2022). *SINTESIA: Jurnal Sistem dan Teknologi Informasi Indonesia Pemanfaatan Sistem Temu Kembali Informasi Dalam Pencarian Dokumen Menggunakan Vector Space Model*.
- Kosasih, R., & Alberto, A. (2021). Analisis Sentimen Produk Permainan Menggunakan Metode TF-IDF Dan Algoritma K-Nearest Neighbor. 6(1). <https://doi.org/10.30743/infotekjar.v6i1.3893>
- Kurnia Aini, S. (2022). Perancangan Sistem Pendukung Keputusan Terhadap Pendeteksi

- Plagiarisme Judul Skripsi. In *Teknologipintar.Org* (Vol. 2, Issue 2).
- L. Dara Jayanti And A. Masruri. (2024). "Temu Kembali Informasi Dengan Menggunakan Eskripsi Di Perpustakaan Universitas Respati Yogyakarta," *Jurnal Pustaka Budaya*, vol. 11, no. 1, 2024. <https://journal.unilak.ac.id/index.php/pb/>.
- Mi'andri, M., Siregar, A. C., & Utami, P. Y. (2022). Sistem Penilaian Ujian Otomatis Untuk Soal Esai Menggunakan Metode Vector Space Model. *JUTECH: Journal Education and Technology*, 2(2), 1-15. <https://doi.org/10.31932/jutech.v2i2.1273>
- Mirs, E. (2023). v o L. *Romanian Journal Ofapplied Science and Technology*, XIII(3), 254-260.
- Nurmalasari, D., & Nengsih, W. (2021). Sistem Rekomendasi Referensi Jurnal Ilmiah dengan Metode Frequent Itemset. *Journal of Applied Informatics and Computing*, 5(2), 176-181.
- Nuzul Hikmah, Dyah Ariyanti, & Ferry Agus Pratama. (2022). Implementasi Chatbot Sebagai Virtual Assistant di Universitas Panca Marga Probolinggo menggunakan Metode TF-IDF. *JTIM: Jurnal Teknologi Informasi Dan Multimedia*, 4(2), 133-148. <https://doi.org/10.35746/jtim.v4i2.225>
- Priandono, I. R., Rozi, N. F., & Hakimah, M. (2020). Implementasi Vector Space Model Dengan Pembobotan Berbasis Kelas Pada Mesin Pencari Dokumen Skripsi. *Jurnal Informatika: Jurnal Pengembangan IT*, 5(2), 54-58. <https://doi.org/10.30591/jpit.v5i2.2079>
- Putri Gabriella, Y. A. (2023). Optimasi Penerimaan Siswa Baru Dengan Penerapan Algoritma Text Mining Dan Tf-Idf. *Journal of Computing and Informatics Research*, 2(3), 110-117. <https://doi.org/10.47065/comforch.v2i3.941>
- Septiani, D., & Isabela, I. (2022). *SINTESIA: Jurnal Sistem dan Teknologi Informasi Indonesia Analisis Term Frequency Inverse Document Frequency (TF-IDF) Dalam Temu Kembali Informasi Pada Dokumen Teks*.
- Via, Y. vita, & Mumpuni, R. (2019). Deteksi Kemiripan Dokumen Publikasi Skripsi Mahasiswa Menggunakan Algoritma Modifikasi Cosine Similarity. *Journal of Information Engineering and Educational Technology*, 3(2), 57-61. <https://doi.org/10.26740/jieet.v3n2.p57-61>