

# STUDI DAN ANALISIS ALGORITMAKEMIRIPAN COSINE SIMILARITY UNTUK MENGIDENTIFIKASIKESAMAAN JUDUL SKRIPSI

Harry Eko Ujiantoro Adi Basuki<sup>1</sup>, Kusrini<sup>2</sup>, Asro Nasiri<sup>3</sup>

Magister Teknik Informatika, Universitas AMIKOM Yogyakarta

e-mail: harryekouab@students.amikom.ac.id, kusrini@gmail.com, asro@amikom.ac.id

## Abstrak

Langkah awal dalam proses pengajuan skripsi adalah proses validasi oleh pembimbing skripsi. Masalah utama yang dihadapi dosen pembimbing skripsi di Fakultas Teknologi Informasi Universitas Islam Balitar ini adalah banyaknya proses validasi pengajuan skripsi yang dilakukan oleh mahasiswa. Hal ini disebabkan belum adanya sistem validasi pengajuan judul, baik yang sudah pernah diajukan maupun belum. Pada penelitian ini, penulis melakukan penelitian untuk membuat sistem microframework berbasis Python yang memiliki kemiripan dengan mesin pencari informasi. Sistem tersebut bertujuan untuk mendeteksi kesamaan judul skripsi yang diajukan oleh mahasiswa Fakultas Teknologi Informasi Universitas Islam Balitar. Metode yang digunakan dalam sistem ini adalah cosine similarity yang merupakan salah satu metode yang digunakan untuk mendeteksi kesamaan pada suatu kata. Hasil dari penelitian ini adalah sebuah sistem yang dapat digunakan sebagai alat untuk pencarian informasi. Sistem menghasilkan informasi berupa persentase kesamaan judul skripsi, yang dapat digunakan sebagai alat bantu validasi judul skripsi di Fakultas Teknologi Informasi Universitas Islam Balitar.

**Kata kunci**— Cosine Similarity, Flask, Similarity, Micro Framework, Judul Skripsi.

## Abstract

The first step in the thesis submission process is the validation process by the thesis supervisor. The main problem faced by thesis supervisors at the Faculty of Information Technology, Balitar Islamic University is the large number of validation processes for thesis submissions made by students. This is due to the lack of a system to validate the thesis submission, whether it has been submitted or not. In this study, the authors conducted research to create a Python-based microframework system that has similarities with information search engines. The system aims to detect the similarity of thesis titles submitted by students of the Faculty of Information Technology, Balitar Islamic University. The method used in this system is cosine similarity, which is one of the methods used to detect similarity in a word. The result of this research is a system that can be used as a tool for information retrieval. The system produces information in the form of a percentage of similarity of dissertation titles, which can be used as a tool for validating dissertation titles at the Faculty of Information Technology, Balitar Islamic University.

**Keywords**— Cosine Similarity, Flask, Similarity, MicroFramework, ThesisTitle.

## 1. Pendahuluan

Kemajuan teknologi saat ini memudahkan kita dalam melakukan pendeteksian kecurangan. salah satu tindak kecurangan yang sering dilakukan mahasiswa adalah tindakan plagiarisme. Plagiarisme merupakan tindakan untuk melakukan penjiplakan atau mengambil pendapat orang lain dan menjadikannya seolah-olah karangan dan pendapat sendiri[1].

Mahasiswa sarjana, magister, dan doktor memiliki Standar untuk memiliki publikasi di jurnal nasional atau internasional yang terakreditasi. Bagi dosen wajib melampirkan surat pernyataan pengesahan karya ilmiah yang akan dipromosikan atau diposisikan dengan melampirkan hasil pemeriksaan terkait orisinalitas. Hal ini sesuai dengan Peraturan Menteri Riset, Teknologi, dan Pendidikan Tinggi (Permenristekdikti) Nomor 20 Tahun 2017, diwajibkan memiliki publikasi ilmiah pada jurnal terakreditasi. Peraturan ini diperkuat dengan Surat Direktur Jenderal Sumber Daya Iptek dan Pendidikan Tinggi Kementerian Riset, Teknologi, dan Pendidikan Tinggi Nomor 1753/D2/KP/2016 tanggal 12 Juli 2016. Peraturan ini bertujuan untuk menjaga kualitas karya ilmiah dari dosen. Dengan

dilakukan pengecekan kemiripan atau orisinalitas, diharapkan karya dosen memiliki tingkat kemiripan yang rendah dan banyak bermunculan ide dan gagasan baru yang memiliki tingkat orisinalitas yang tinggi.

Tujuan dari penelitian ini adalah memberikan alternatif solusi untuk mengatasi plagiarisme terkait pengajuan judul skripsi pada Program Studi Teknik Informatika Universitas Islam Balitar.

### 2.1 Penelitian Terdahulu

Menurut penelitian yang dilakukan oleh Bambang Krismono Triwijoyo, Kartarina dalam artikel *Analysis of Documents Clustering Based Cosine Similarity dan K-Main Algorithm* menyatakan bahwa hasil penelitian menunjukkan bahwa metode ini menghasilkan akurasi sebesar 84,3%. Dalam menggunakan metode standar K-means [3].

Menurut Dhamayanti Lidia Permata Sari dalam artikel *Aplikasi Deteksi Plagiarisme Berbasis Web di Universitas Indo Global Mandiri Dengan dibangunnya aplikasi pendeteksi plagiarisme ini dapat berguna untuk mencari kesamaan data skripsi mahasiswa baru dengan data script mahasiswa lama yang sudah tersimpan di database.*

Lebih lanjut Menurut Yisti Vita Via, Retno dalam jurnal yang berjudul *Deteksi Kemiripan Dokumen Publikasi Skripsi Mahasiswa Menggunakan Algoritma Modifikasi Cosine Pada Aplikasi Sistem Kemiripan dapat digunakan oleh mahasiswa Untuk membantu pengecekan naskah publikasi jurnal ada indikasi kesamaan dengan dokumen jurnal mahasiswa lain dengan prosentase kemiripan tertentu. Selain itu Algoritma Levenshtein Distance dan Cosine Similarity mampu mendeteksi dan mengukur tingkat kemiripan dokumen publikasi jurnal skripsi mahasiswa yang diunggah ke dalam aplikasi.*

Dari penelitian sebelumnya, penulis memiliki ide untuk mengimplementasikan sistem deteksi kesamaan tesis menggunakan metode kesamaan kosinus dengan tiga variasi bobot yakni 0,7 – 0,9. Penentuan bobot ini bertujuan untuk mencari bobot yang sesuai untuk mencapai hasil yang akurat dalam mendeteksi plagiarisme pada judul skripsi di lingkungan program studi Teknik Informatika Universitas Islam Balitar.

### 2.2 Preprocessing Text

Preprocessing merupakan tahap awal pada proses Text Mining. Berikut langkah-langkah pada preprocessing teks [4]:

1. Tokenizing

Pada prinsipnya proses ini memisahkan setiap kata yang menyusun suatu dokumen. [5]

2. Case Folding

Case Folding merupakan proses untuk mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf 'a' hingga 'z' yang diterima oleh karakter lain dan tanda baca selain huruf juga dihapus. Pada beberapa studi kasus Tidak semua dokumen teks konsisten dalam penggunaan huruf kapital. [6]

3. Filtering

Pada Tahap filtering merupakan tahap pengambilan kata-kata penting dari proses tokenizing. Pada tahap ini, kata akan dibersihkan dari karakter khusus, tautan URL, nama pengguna, maupun emotikon [5]

4. Stemming

Proses stemming berguna untuk menemukan kata dasar dari setiap kata yang disaring. Data yang dihasilkan dari stopword akan digunakan untuk melakukan proses penghitungan Term Frequency / Penghitungan banyaknya kemiripan kata dasar. [7]

### 2.3 Cosine Similarity

Cosine Similarity adalah algoritma yang dapat digunakan untuk melakukan perhitungan kesamaan dari suatu dokumen. Untuk notasi himpunan pada cosine similarity digunakan rumus (Waveni, Prastiyanto, & Suprpto, 2017)

### 2.4 Pembobotan TF

Term Frequency merupakan proses penghitungan kemunculan kata dasar yang telah disimpan pada dataset dengan data yang akan di test pada system. Setelah proses pembobotan ini dilakukan proses pencarian kemiripan antar dokumen teks. Sistem akan menghitung kata dasar kemiripan dokumen A dan dokumen B sehingga pengguna dapat melihat berapa banyak dokumen yang memiliki kemiripan

### 2.5 Confusion Matrix

Confusion matrix adalah sebuah tabel yang menyatakan jumlah data uji yang benar diklasifikasikan dan jumlah data uji yang salah diklasifikasikan. Contoh dari Confusion matrix ditunjukkan pada tabel 1.

Tabel 1. Tabel Confussion Matrix

		Kelas Prediksi	
		1	0
Kelas Aktual	1	TP	FN
	0	FP	TN

## 2.6 Framework Flask

*Framework* Flask adalah sebuah web *framework* yang ditulis dengan bahasa Python dan tergolong ke dalam jenis *microframework*. Dengan menggunakan Flask dan bahasa Python, developer dapat membuat sebuah web yang terstruktur dan dapat diakses serta mengatur perilaku sebuah web dengan lebih mudah. Flask termasuk ke dalam jenis *microframework* karena tidak membutuhkan alat bantu atau referensi tertentu dalam penggunaannya. Performa menggunakan *framework* sangat ringan karena core dengan desain modular Super simple.

## 2. Metode Penelitian

Pelaksanaan penelitian agar dapat dilakukan secara efektif dan efisien serta sesuai dengan tujuan maka langkah awal pelaksanaan penelitian yang kami lakukan antara lain :

1. Melakukan identifikasi masalah yakni adanya potensi maraknya plagiarisme di dunia akademik khususnya pada naskah-naskah akademik seperti skripsi, tesis dan penelitian
2. Menentukan tujuan penelitian tujuan penelitian ini adalah melakukan pencegahan plagiarisme skripsi di program studi Teknik Informatika Universitas Islam Balitar
3. Membaca serta mencari acuan studi yang mendukung pelaksanaan penelitian
4. Merumuskan sistem atau metoda langkah-langkah penelitian termasuk mekanisme mendapatkan data, memproses data serta melakukan penilaian dan kajian atas keluaran data yang dihasilkan

Metoda penelitian diatas secara detail dapat kami jelaskan dengan tahapan-tahapan sebagai berikut :

### 1. Identifikasi masalah

Pada tahap ini dilakukan identifikasi masalah dari penelitian yang akan dibuat berdasarkan informasi yang didapat di obyek penelitian yakni program studi Teknik Informatika Universitas Islam Balitar

2. Perumusan masalah dilakukan setelah melakukan identifikasi masalah dengan memperhatikan data-data dari lokasi obyek penelitian, khususnya jumlah mahasiswa serta skripsi yang sudah di hasilkan pada periode sebelum penelitian dilaksanakan. Tentunya perumusan masalah ini menghasilkan tujuan serta manfaat penelitian yang diharapkan.
3. Referensi Studi pada tahap ini dilakukan pengumpulan informasi yang berkaitan dengan topik atau masalah yang diteliti. Referensi dapat diperoleh melalui artikel , buku , jurnal dan tesis terkait dengan Sistem deteksi plagiarisme di Skripsi Jurusan Teknik Informatika Universitas Islam Balitar. Selanjutnya hasil membaca serta mengumpulkan literasi atas teori-teori penelitian sebelumnya termasuk algoritma yang sudah pernah digunakan untuk menjalankan proses penilaian uji kemiripan dokumen.
4. Hasil dari literasi materi peneliti sebelumnya sebagian hasil penelitian diadopsi parameter dan variabel yang digunakan sebelumnya untuk dijadikan parameter serta penentuan metoda yang dapat digunakan pada penelitian ini yakni menggunakan Algoritma *Cosine Similarity*..
5. Dengan menggunakan metoda pengembangan aplikasi *waterfall*, peneliti membuat program yang menerjemahkan *Algoritma Cosine Similarity* dalam bahasa pemrograman Phyton. Data yang dihasilkan dari proses pengolahan data tersebut ditampilkan dengan Flask *website framework*. Aplikasi dibangun pada perangkat laptop peneliti. Sehingga parameter dan spesifikasi laptop peneliti yakni laptop Intel Pentium Core i5, ram 4GB dan HD 500GB yang berjalan dengan *Operating System Windows 10* tentunya sangat mempengaruhi kecepatan akses dan pengolahan data uji kemiripan dokumen.
6. Data yang diperoleh dari program studi Teknik Informatika Universitas Islam Balitar selanjutnya dipisah menjadi 2 kategori yakni ada yang dijadikan data set training dan sebagian akan dijadikan data testing untuk diuji dengan aplikasi yang terpasang pada laptop peneliti. Tentunya jumlah data

training jauh lebih banyak dari data testing karena data testing mewakili judul tesis atau skripsi mahasiswa yang diuji saat tertentu,

Untuk metode yang digunakan dalam membangun sistem ini menggunakan *Waterfall SDLC (Software Developing Life Cycles)* yang terdiri dari :

### 3.1. Analisis

Pada tahap awal pengembangan sistem dilakukan analisis kebutuhan sistem. Tahap ini sangat penting karena setiap kesalahan dalam analisis dapat menyebabkan masalah pada tahap selanjutnya. Analisis kebutuhan sistem meliputi perhitungan cosinus similarity, flowchart, dan faktor lain yang relevan. Berikut rumus kesamaan Cosinus :

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

Keterangan :

A : Vektor A

B : Vektor B

$A_i$  : Suku Bobot  $i$   $A_i$

$B_i$  : Bobot Suku  $i$   $B_i$

$AB$  = perkalian titik vektor A & B

$|A|$  = Panjang Vektor A

$|B|$  = Panjang Vektor B

$|A||B|$  = Vektor Perkalian Silang  $|A||B|$

$i$  = Suku Hitung

$n$  = Hitung Vektor

### 3.2. Desain

Untuk Merancang perangkat lunak, tahap pertama melibatkan pembuatan modul dari kebutuhan spesifikasi perangkat lunak perangkat. Tahap ini sangat penting karena membantu memperkirakan kualitas presentasi sistem sebelum tahap pengkodean. Alat-alat berikut akan digunakan dalam tahap ini: *UML (Unified Modeling Language)*, desain antara advance, dan file spesifikasi. Studi ini juga akan memanfaatkan kasus penggunaan untuk memastikan kualitas sistem perangkat lunak.

### 3.3 Coding

Pada tahap ini dilakukan proses implementasi sesuai dengan desain system yang dibuat. Pada penelitian ini implementasi menggunakan *Framework Flask* dengan menggunakan dataset berbasis CSV.

### 3.4 Pengujian

Pada tahap ini dilakukan pengujian terhadap sistem yang telah dibuat. Pada pengujian ini dilakukan untuk melakukan perbandingan hasil ketelitian dengan variasi dari 3 bobot ambang yaitu 0,7 – 0,9.

## 3. Hasil dan Pembahasan

Berikut tahapan dari hasil perancangan system deteksi kemiripan judul skripsi pada Jurusan Teknik Informatika Fakultas Teknologi Informasi Universitas Islam Balitar :

#### A. Pengumpulan Data

Proses pengumpulan data diambil dari data skripsi yang terdapat pada Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Islam Balitar sejumlah 112 judul skripsi.

## B. Tahap Pre-Processing

Pada tahap ini merupakan awal proses data teks akan diolah menjadi data berbasis numerik. Menurut Wahyuni(2017), menjelaskan bahwa proses *Preprocessing* merupakan tahapan dimana deskripsi akan diolah dan siap diproses pada tahap teks *mining*. Setelah data terkumpul proses selanjutnya terdiri :

- **Data Integrasi**  
Pada penelitian ini data judul skripsi akan dijadikan 1 dalam bentuk file csv.
- **Case Folding**  
Pada proses *case folding* judul skripsi akan dilakukan konversi semua judul menjadi *lowercase* agar memudahkan proses deteksi kemiripan judul skripsi menggunakan *cosine similarity*.
- **Tokenizing**  
Pada proses *Tokenizing*, data yang sebelumnya dilakukan konversi pada *case folding* selanjutnya akan dilakukan pengecekan terkait tanda baca , karakter khusus akan dihapus.
- **Stopword / Filtering**  
Proses *stopword* digunakan untuk menghapus pengulangan kata yang mungkin terjadi pada saat penginputan dataset judul.
- **Stemming**  
Proses *stemming* berguna untuk menemukan kata dasar dari setiap kata yang disaring. Data yang dihasilkan dari *stopword* akan digunakan untuk melakukan proses penghitungan Term Frequency / Penghitungan banyaknya kemiripan kata dasar.

## C. Proses Penghitungan Cosine Similarity

Data *testing* judul skripsi mahasiswa dimasukkan kedalam Aplikasi yang selanjutnya diproses (*di-running*) menghasilkan angka prosentase kemiripan dan kecepatan proses yang berbeda beda. Selanjutnya hasil uji beberapa judul skripsi tersebut dimasukkan ke dalam tabel untuk dianalisa untuk menentukan skenario mana setelah data testing di *treatment* menghasilkan metoda uji kemiripan judul skripsi yang terbaik (prosentase kemiripannya) dan tingkat sensitifitasnya dalam mendeteksi kemiripan judul skripsi yang diuji.

Adapun script untuk melakukan pengujian kemiripan judul skripsi dengan metoda algoritma *Cosine Similarity* sesuai dalam gambar 1. bawah ini.

```
def count_cosine_similarity (doc1, doc2):
# Tentukan bobot untuk setiap istilah bobot = {'judul': 0.9}

# Konversi dokumen ke vektor menggunakan TF-IDF
vectorizer = TfidfVectorizer( stop_words = 'ind ')
vectors = vectorizer.fit_transform ([doc1, doc2]).
toarray ()

# Masukkan bobot threshold Cosine
weights_array = np.array ([bobot['judul'],
bobot['judul']])
weighted_vectors = vektor * np.expand_dims (
weights_array , axis=1)

# Hitung Cosine
similarity = cosine_similarity ( weighted_vectors
[0].reshape(1, -1), weighted_vectors [1].reshape(1, -1))
return similarity[0][0]
```

Gambar 1. Script Pengujian Kemiripan Judul Skripsi Dengan Metoda Algoritma *Cosine Similarity*

Berikut hasil proses penghitungan dari cosine similarity dalam melakukan proses pendeteksian kemiripan judul skripsi.

Tabel 2. Hasil *Cosine Similarity*

<i>Term</i>	D1
Perancangan	0
Sistem	1
Informasi	1
Penjualan	1
<i>Online</i>	1
Dengan	1
Metode	1
<i>Ecommerce</i>	1
<i>TF</i>	7
IDF	0,0579919
TF*IDF	0,4059436
Vector Q > D ( Test Data > Training Dataset)	2,0102119
Vector Q	2,6457513
Similarity	0,537251722

Tabel 3. Test Kemiripan Menggunakan Judul Skripsi

<i>Term</i>	D1
Perancangan	0
Sistem	0
Informasi	0
Penjualan	1
Online	1
Dengan	0
Metode	0
<i>Ecommerce</i>	0
<i>TF</i>	2
IDF	0,1760913
TF*IDF	0,3521825
Vector Q -> D	2,0310081
Vector Q	1,4142136
Similarity	0,089758726 atau 8%

#### D. Prototipe Sistem

Pada Gambar 1, merupakan halaman utama sistem deteksi plagiarisme menggunakan *microframework* Flask. pengguna akan diarahkan untuk memasukkan judul naskah yang ingin diperiksa plagiarismenya. Dataset skripsi yang digunakan dalam sistem ini diperoleh dari Fakultas Teknologi Informasi Universitas Islam Balitar yang terdiri dari total 112 judul skripsi.



Gambar 2. Tampilan Sistem Deteksi Plagiarisme

Pada gambar 2. menunjukkan hasil pendeteksian plagiarisme judul skripsi menggunakan metode *cosine similarity*. Sistem akan menampilkan persentase hasil plagiarisme dari judul yang telah diajukan mahasiswa pada sistem. Selain hasil persentase sistem ini juga menampilkan perangkingan 5 data yang memiliki kemiripan.



Pada gambar 1. sebelumnya menampilkan kode untuk memproses *cosine similarity* menggunakan *framework* Flask. kode tersebut menjelaskan bagaimana proses *cosine similarity* digunakan untuk mendeteksi plagiarisme pada judul skripsi di Fakultas Teknologi Informasi Universitas Islam Balitar. Setelah mahasiswa menginput judul yang ingin diperiksa plagiatnya, sistem melakukan proses pencocokan untuk membandingkan kemiripan judul dengan dataset skripsi yang ada di fakultas. Proses pencocokan mendefinisikan doc1 sebagai input judul oleh siswa dan doc2 sebagai dataset dokumen yang akan diuji kesamaannya. Untuk membedakan dengan penelitian lain, penulis menguji *cosinus similarity* dengan variasi tiga bobot yaitu 0,7, 0,8, dan 0,9. Bobot 0,9 memiliki sensitivitas paling tinggi untuk metode *cosine similarity*.

E. Pengujian

Tahap ini merupakan proses evaluasi terhadap sistem yang telah dirancang sebelumnya. Berikut adalah skenario pengujian dalam penelitian ini

- Pengujian akurasi menggunakan Turnitin sebagai perbandingan sistem cek plagiarisme
- Pengujian dataset menggunakan data script sebanyak 112 data di Fakultas Teknologi Informasi

Berikut hasil pengujian yang telah dilakukan ditunjukkan pada tabel :

Tabel 2. Pengujian Dengan Threshold dan Akurasi

Bobot	Kecepatan (detik)			Akurasi		
	0,7	0,75	0,74	0,73	38%	37%
0,8	0,70	0,71	0,70	43%	42%	43%
0,9	0,74	0,73	0,73	66%	67%	66%

Tabel 3. Nilai Rata-rata dari tabel 2.

Bobot	Kecepatan	Akurasi
0,7	1,73 dtk	38%
0,8	1,64 dtk	43%
0,9	1,71 Detik	67%

#### 4. Kesimpulan

Berikut kesimpulan dari implementasi yang telah dilakukan :

1. Sistem yang dibuat dapat membantu mahasiswa dalam melakukan pengecekan tingkat plagiarisme judul skripsi di Jurusan Teknik Informatika Unisba Balitar .
2. Penggunaan Database Berbasis CSV pada sistem pendeteksian Hal ini memudahkan dalam pengelolaan dataset skripsi jika ingin melakukan penambahan maupun pengurangan data.
3. Pada pengujian yang telah dilakukan konfigurasi yang tepat untuk sistem deteksi plagiarisme adalah dengan setting Bobot Threshold 0,9. Dengan hasil akurasi sebesar 66% dengan kecepatan akses 0,74 detik
4. Untuk penelitian selanjutnya, teknik deteksi plagiarisme ini akan dibandingkan dengan metode Vector Space Model (VSM) dalam mendeteksi kesamaan pada skripsi. Perbandingan akan melibatkan ambang bobot yang bervariasi berdasarkan tingkat sensitivitas metode VSM.
5. Penelitian selanjutnya akan membandingkan hasil similarity dan kecepatan pemrosesan antara algoritma Cosine Similarity dan VSM terhadap standar aplikasi uji kemiripan Turnitin yang sudah jamak digunakan oleh lembaga pendidikan dan penelitian termasuk perguruan tinggi untuk menentukan algoritma mana yang lebih baik.

#### 5. Saran

Dengan hasil penelitian eksperimental ini selanjutnya disarankan ada penelitian lanjutan dengan melakukan proses uji similarity mengkombinasikan dua algoritma uji plagiarisme Cosine Similarity dan VSM berurutan atau diperbandingkan dengan membalik urutan algoritma tersebut sebagai series pengujian. Selanjutnya diharapkan dengan penelitian lanjutan tersebut akan mendapatkan hasil perbandingan terbaik yang dapat menentukan urutan kombinasi algoritma mana yang lebih baik dari yang lain dan apakah metoda melakukan lebih dari satu algoritma bisa meningkatkan keakuratan hasil pengujian atau menurunkan kecepatan rata-rata pemrosesan secara signifikan.

#### Daftar Pustaka

- [1] [1] Zuliarso, Eri. 2010. "Aplikasi Web Crawler Berbasis Breadth First Search Dan Back-Link." Fakultas Teknologi Informasi Universitas Stikubank Semarang XV(1): 52–56.
- [2] Sugiyamta. 2015. "Sistem Deteksi Kesamaan Dokumen Dengan Algoritma Cosine Similarity dan Single Pass Clustering." *Dinamika Informatika* 7(2): 7.
- [3] Nurdiana, Ogie, Jumadi, dan Dian Nursantika. 2016. "Perbandingan Metode Cosine Similarity dengan Metode Jaccard Similarity pada Aplikasi Pencarian Terjemah Al-Qur'an Bahasa Indonesia." *Jurnal Online Informatika (GABUNG)* 1(1): 59–63.
- [4] Santos, Hari. 2015. "Pencegahan dan Pengendalian Plagiat dalam Penulisan Karya Ilmiah di Perpustakaan Perguruan Tinggi Oleh: Drs. Hari Santoso, S.Sos. 1." Perpustakaan UM Malang (1): 1–23.
- [5] Kumar, Manish, Ankit Bindal, Robin Gautam, dan Rajesh Bhatia. 2018. "Perayap Web Berfokus Kueri Berbasis Kata Kunci." *Ilmu Komputer Procedia* 125:584–90. <http://linkinghub.elsevier.com/retrieve/pii/S1877050917328399>.



- [6] Rungsawang, Arnon, dan Niran Angkawattanawit. 2005. "Perayap Web Khusus Topik yang Dapat Dipelajari." *Jurnal Aplikasi Jaringan dan Komputer* 28(2): 97–114.
- [7] Pahlevi, Irfan, Moch Arief Bijaksana, dan M Ir Tech. "Perhitungan Kemiripan Dokumen Berbahasa Indonesia Menggunakan Metode Cosine Similarity (Studi Kasus: Abstrak Tugas Akhir Fakultas Informatika Universitas Telkom)."
- [8] Hutabarat, Suetha Ronarumata. (2016). "Tingkat Plagiat Pada Skripsi Mahasiswa Lulusan Program Studi Ilmu Perpustakaan Tahun 2015 Berdasarkan Plagiarism Checker X Scanner." Medan: Program Studi Ilmu Perpustakaan dan Informasi Fakultas Ilmu Budaya Universitas Sumatera Utara.
- [9] Triana, A., Saptono, R., & Sulistyono, ME 2014. Pemanfaatan Metode Vector Space Model dan Metode Cosine Similarity pada Fitur Deteksi Hama dan Penyakit Tanaman Padi. *ITSMART: Jurnal Teknologi dan Informasi*, 3(2), hlm .90-95.
- [10] Rizki, Dhidik, dan Eko Suprpto. 2017. Pengaplikasian Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen. Skripsi. Jurusan Teknik Elektro. Fakultas Teknik. Universitas Negeri Semarang Kampus Sekaran, Gunungpati, Semarang
- [11] Ma'arif, Abdul Aziz. 2015. Pengaplikasian Algoritma Tf-Idf Untuk Pencarian Karya Ilmiah. *Jurnal Jurusan Teknik Informatika. Fakultas Ilmu Komputer. Universitas Dian Nuswantoro Semarang.*
- [12] Jiffriya, MAC, Jahan, MA, & Ragel, RG 2014. Deteksi plagiarisme pada penugasan berbasis teks elektronik menggunakan model ruang vektor. Dalam Konferensi Internasional ke-7 tentang Informasi dan Otomasi untuk Keberlanjutan (hlm. 1-5). IEEE.
- [13] Rozaq, A., Hardinto, RK, Yunida, R., & Faslah, R. (2018). Sistem Informasi Penelitian dan Pengabdian Masyarakat Guna Meningkatkan Kinerja P3M Politeknik Negeri Banjarmasin. *Jurnal Sains Terapan Teknologi Informasi*, 10(2), 36-41.
- [14] Wahyuni, RT, Prastiyanto, D., & Suprpto, E. (2017). Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi. *Jurnal Teknik Elektro Vol. 9 No.1*, 18-23.
- [15] Pratama, R. P., Faisal, M., & Hanani, A. (2019). Deteksi Plagiarisme pada Artikel Jurnal Menggunakan Metode Cosine Similarity. *SMARTICS Journal*, Vol.5 No. 1 2019, 22-26.