Penerapan Metode Cosine Similarity dalam Mendeteksi Plagiarisme pada Jurnal

[1]Sarima Lumbansiantar, [2]Saruni Dwiasnati, [3]Nenden Siti Fatonah
[1,2]*Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Mercu Buana*[1]
[3]*Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Esa Unggul*

41519210041@*student.mercubuana.ac.id*[1], saruni.dwiasnati@mercubuana.ac.id[2],
nenden.siti@esaunggul.ac.id[3]

ABSTRACT

Abstract - Plagiarism is the behavior of stealing someone else's idea or ideas without mentioning the original source. More and more plagiarism cases seem to have been cultivated in people's lives, especially in the field of scientific writing in academia. The availability of online journals that provide comfort to its readers is the main benefit, but the potential for plagiarism and plagiarism-related behavior is its main drawback. The Cosine Similarity method is one of the methods that can be used to detect plagiarism in text. This method is based on the calculation of the angle between two objects expressed in two vectors. The object used in this study is the abstract of an online journal. A total of 100 online journal data used for this study were obtained from scientific journal publishing sites. The stage carried out before the calculation of Cosine Similarity is pre-processing with the aim of eliminating inconsistencies in the data. Based on the results of research that the cosine similarity method can detect plagiarism in journal documents by calculating the percentage of similarity between repository documents and test documents.

Keywords: Plagiarism, Cosine Similarity, Journals.

ABSTRAK

***Abstrak* – Plagiarisme merupakan perilaku pencurian ide atau gagasan orang lain tanpa menyebutkan sumber aslinya. Kasus plagiarisme makin lama makin banyak nampaknya sudah membudaya dalam kehidupan masyarakat khususnya dibidang karya tulis ilmiah dalam dunia akademisi. Ketersediaan jurnal *online* yang memberikan kenyamanan kepada para pembacanya menjadi manfaat utamanya, akan tetapi potensi plagiarisme dan perilaku terkait plagiarisme adalah kelemahan utamanya. Metode *Cosine Similarity* merupakan salah satu metode yang dapat digunakan untuk mendeteksi plagiarisme pada teks. Metode ini didasarkan pada perhitungan sudut antara dua buah objek yang dinyatakan dalam dua buah vektor. Objek yang digunakan dalam penelitian ini adalah abstrak jurnal *online*. Total sebanyak 100 data jurnal *online* yang digunakan untuk penelitian ini yang diperoleh dari situs-situs penerbit jurnal ilmiah. Tahapan yang dilakukan sebelum perhitungan *Cosine Similarity* adalah *pre-processing* dengan tujuan untuk menghilangkan inkonsistensi pada data. Berdasarkan hasil penelitian bahwa metode *cosine similarity* dapat mendeteksi plagiarisme pada dokumen jurnal dengan menghitung persentase kemiripan antara dokumen repositori dengan dokumen uji.**

Kata kunci: Plagiarisme, *Cosine Similarity*, Jurnal

I. PENDAHULUAN

Plagiarisme berasal dari kata Latin *"plagiarius"* yang berarti penculik, pembajak [1]. Plagiarisme merupakan penggunaan ide atau karya orang lain tanpa menyebutkan sumber aslinya yang seolah-olah ide atau karya tersebut adalah miliknya sendiri.

Plagiarisme bukanlah merupakan masalah yang baru saja muncul akan tetapi sudah ada sejak dahulu yang di mana nampaknya sudah membuadaya di kalangan masyarakat luas.

Menurut Sastroasmoro, plagiarisme merupakan tindakan pencurian atau kebohongan intelektual [2]. Selain masalah moral, tindakan plagiarisme ini juga berujung pada kurang nya kreativitas siswa dalam menuangkan ide dan pendapat pribadi ke dalam konten yang ingin disampaikan.

Plagiarisme sangat mudah terjadi terlebih di kalangan akademik. Maka dari itu seluruh pihak akademisi memberi perhatian besar untuk memberikan imbauan kepada guru dan siswa tentang plagiarisme dan cara menghindarinya [1].

Perkembangan teknologi yang sangat pesat memengaruhi semua bidang kehidupan manusia, tidak terkecuali bidang pendidikan. Teknologi memudahkan akses manusia menelusuri internet untuk mengeksplorasi berbagai bidang ilmu pengetahuan. Akan tetapi sering kali disalahgunakan salah satunya perilaku plagiat.

Plagiarisme tidak selalu disengaja. Ada beberapa kategori plagiarisme yang lebih luas diantaranya:

- Kebetulan: karena kurangnya pengetahuan tentang plagiarisme, dan pemahaman tentang kutipan atau referensi.
- Tidak disengaja: luasnya informasi yang tersedia memengaruhi ide dan pemikiran yang sama dapat keluar melalui ekspresi lisan ataupun tulisan sebagai milik seseorang.

- Disengaja: tindakan yang disengaja menyalin seluruh atau sebagian karya orang lain tanpa memberikan kredit yang layak kepada pencipta asli.
- *Self Plagiarism:* plagiat diri sendiri terjadi saat pengarang menggunakan lagi karyanya sendiri yang telah diterbitkan sebelumnya dan dilindungi hak cipta terbitan selanjutnya.

Plagiarisme yang akan dibahas pada penelitian ini adalah plagiarisme pada jurnal. Jurnal merupakan dokumen digital yang dibutuhkan publik di semua bidang studi. Ada dua versi jurnal: cetak dan digital.

Jurnal *online* merupakan salinan digital jurnal cetak yang sering ditemukan di perpustakaan online. Publikasi serial sama untuk jurnal cetak dan *online*. Satu-satunya perbedaan adalah bahwa jurnal cetak terbuat dari kertas, sedangkan jurnal *online* tidak memerlukan pencetakan dan dapat dilihat langsung secara *online.*

Kenyamanan membaca jurnal *online* tanpa membawa kertas adalah manfaat utamanya, tetapi potensi plagiarisme dan perilaku terkait plagiarisme adalah kelemahan utamanya. Salah satu contoh umum plagiarisme adalah pengeditan salin-dan-tempel artikel jurnal *online*. Jika perilaku plagiarisme ini tidak ditangani secara serius, akan mengakibatkan kurangnya kesadaran publik tentang pelanggaran hak cipta dan kurangnya kreativitas masyarakat dalam menuangkan ide.

Mendeteksi plagiat dapat dialakukan dengan cara manual meskipun tidak efektif karena harus memeriksa dokumen satu per satu dengan teliti dan menafsirkan gaya penulisannya terhadap dokumen yang sudah terbit. Cara yang mudah untuk mendeteksi plagiat dapat dilakukan dengan *search engine* dengan memasukkan kata kunci artikel dan membiarkan mesin pencari menemukan dokumen yang dijiplak. Cara lain untuk mendeteksi plagiat adalah dengan menggunakan Algoritma. Penelitian oleh [3] meneliti tentang Aplikasi Pendeteksi Plagiarisme Tugas Dan Makalah Pada Sekolah Menggunakan Algoritma *Rabin Karb.* Dari hasil yang didapatkan dengan penggujian menggunakan sampel uji sebanyak 50 kali dengan 43 sampel keberhasilan sebesar 14,22%. [4] juga melakukan penelitian tentang Perbandingan Algoritma *Winnowing* Dengan Algoritma *Rabin Karp* Untuk Mendeteksi Plagiarisme Pada Kemiripan Teks Judul Skripsi. Hasil perbandingan dua algoritma adalah kemiripan teks judul skripsi yang terkecil dengan penggunaan pendekatan algoritma *winnowing* yaitu pada ujicoba ke 8 dengan nilai n-gram = 9 dan window = 3 , waktu pemrosesan 0.0257 dengan tingkat kemiripan terkecil yaitu 32.6 %. Pada penelitian selanjutnya yang meneliti tentang Klasifikasi Berita Online dengan *Cosine Similarity* dan pembobotan TF-IDF klasifikasi mampu mengelompokkan berita dengan tingkat akurasi sebesar 91.25% [5]. Metode *Cosine Similarity* banyak digunakan dalam perhitungan kemiripan teks dikarenakan tingkat akurasinya yang tinggi [6].

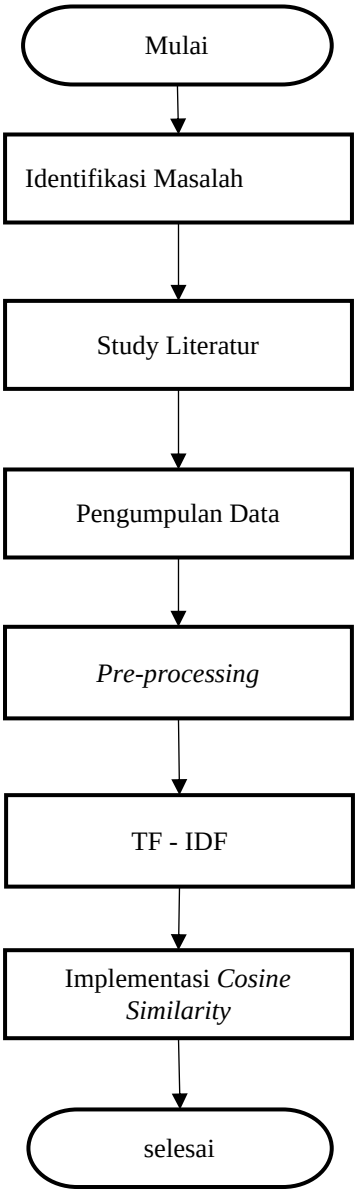
Menurut [2] ambang batas *(threshold)* plagiarisme untuk menentukan suatu dokumen plagiarisme atau tidak, terdapat 3 klasifikasi, yaitu:

1. Plagiarisme ringan : < 30%
2. Plagiarisme sedang : < 30% - 70%
3. Plagiarism berat : > 70%

Akan tetapi batas angka tersebut dapat disesuaikan dengan kebijakan masing-masing lembaga atau perguruan tinggi. Pada penelitian ini akan membahas mengenai metode *Cosine Similarity* dalam menghitung kemiripan dokumen serta penggunaan algoritma TF-IDF untuk pembobotan kata *(term)*. Penelitian ini tidak membuat suatu aplikasi yang sebenarnya diperlukan untuk memudahkan pembaca melakukan pengecekan secara mandiri[7].

II. METODE PENELITIAN

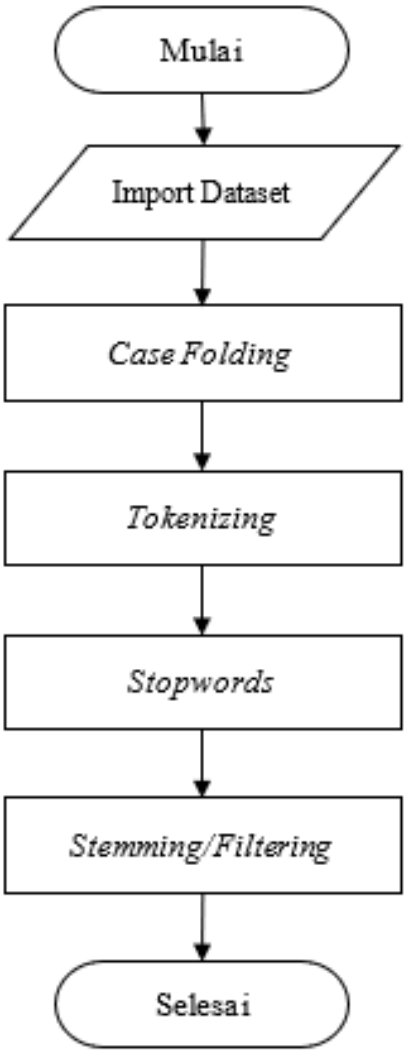
Metode penelitian yang digunakan dalam penelitian ini terdiri dari beberapa tahap yang diawali dengan mengidentifikasi suatu masalah dilanjutkan mencari literatur yang sesuai akan penlitian untuk memperdalam teori, pengumpulan data dan tahap *cleaningsing* dataset, setelah itu dilanjutkan perhitungan kemiripan dokumen dengan metode *Cosine Similarity*. Berikut adalah tahapan penelitian yang dituangkan dalam bagan alir.



Gambar 1. Tahapan Penelitian

Tahapan Metode Penelitian

1. Identifikasi Masalah
   Identifikasi masalah adalah proses hasil dari pengenalan masalah, yaitu dengan kata lain bahwa identifikasi masalah merupakan proses dari penelitian yang boleh dikatakan yang paling penting diantara proses yang lain. Berdasarkan dari hasil penelitian yang akan dilakukan oleh peneliti adalah bagaimana cara mendeteksi plagiasi pada dokumen jurnal online dengan metode *Cosine Similarity*.
2. Studi Literatur
   Studi literatur *(library research)* yaitu mengumpulkan data atau karya tulis ilmiah yang berkaitan dengan objek penelitian atau pengumpulan data yang bersifat kepustakaan. Studi pustaka adalah teknik pengumpulan data dengan mengadakan studi penelaahan terhadap buku-buku, literatur-literatur, catatan-catatan, dan laporan-laporan yang relevan dengan masalah yang ada hubungannya dengan deteksi plagiat pada dokumen dengan metode *Cosine Similarity*.
3. Pengumpulan Data
   Data yang digunakan pada penelitian kali ini merupakan data sekunder, dikarenakan data diperoleh melalui media perantara atau secara tidak langsung dokumen terbitan berupa jurnal online [8]. Pada tahap ini jumlah jurnal yang dikumpulkan sebanyak 100 jurnal. Isi jurnal yang akan diproses pada penelitian ini adalah bagian abstrak dari jurnal.
4. *Pre-processing*
   Dalam data *mining*, ada istilah tahap *pre-processing*, yaitu proses awal yang dilakukan pada pengolahan data untuk mengubah data teks menjadi data *numerik* agar dapat dilakukan proses selanjutnya [9]. Tahapan *pre-processing* dapat dilihat pada Gambar 2.



Gambar 2. Tahapan *Pre-processing*

Berikut adalah tahapan pre-processing yang digunakan dalam penelitian ini antara lain sebagai berikut:

1. *Case Folding*
   Merupakan proses menyeragamkan hurup kapital menjadi huruf kecil. Seluruh karakter "A" – "Z" diubah menjadi karakter "a" – "z". Untuk tanda baca dan angka akan dihilangkan dari data untuk mengurangi *noise*.
2. *Tokenizing*
   Merupakan suatu proses yang dilakukan untuk memotong kalimat menjadi bagian atau kata. Bagian kata yang dipecah disebut sebuah token [10].
3. *Stopwords*
   Merupakan proses menghilangkan kata-kata yang muncul dalam jumlah besar tetapi diangggap tidak ada arti dan tidak ada hubungannya pada pengolahan data untuk mendapatkan hasil yang dituju.
4. *Stemming*
   Merupakan proses untuk mengembalikan kata-kata ke bentuk dasarnya, menghilangkan imbuhan awal (*prefix*) dan imbuhan akhir (*sufix*) sehingga menghasilkan kata dasar yang sesuai dengan KBBI [11]

5. Pembobotan

Pembobotan kata adalah mekanisme untuk memberikan skor pada frekuensi kemunculan kata dalam teks [10]. Salah satu metode populer untuk pembobotan kata-kata adalah TF-IDF (*Term Frequency-Inverse Document Frekuensi).* Metode TF-IDF merupakan metode untuk menghitung bobot suatu kata *(term)* terhadap dokumen. Metode ini juga terkenal efisien, mudah dan memiliki hasil yang akurat. Metode ini menggabungkan dua konsep untuk perhitungan bobot, yaitu frekuensi kemunculan sebuah kata di dalam sebuah dokumen tertentu dan inverse frekuensi dokumen yang mengandung kata tersebut. Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukkan

seberapa penting kata tersebut di dalam dokumen [9]. Bobot kata makin besar jika sering muncul dalam suatu dokumen dan makin kecil jika muncul dalam banyak dokumen [12]

Rumus TF-IDF adalah sebagai berikut [13] :

$$tf = 0{,}5 + 0{,}5\ x\ \frac{tf}{\max(tf)} \qquad (1)$$

$$df_t = log\left(\frac{D}{df_t}\right) \qquad (2)$$

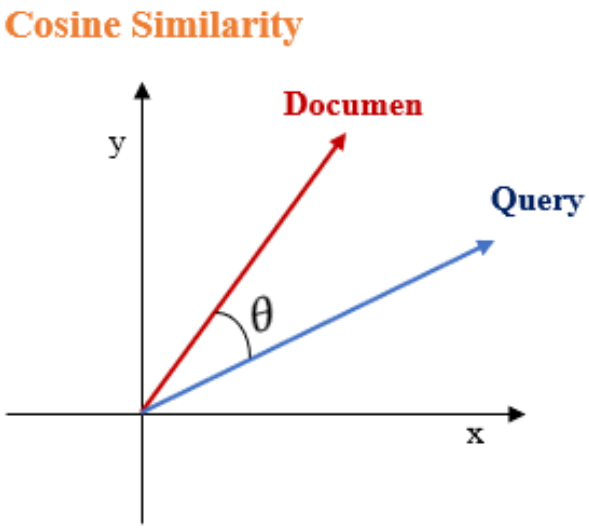
$$W_{d,t} = tf_{d,t}\ \text{x}\ IDF_{d,t} \qquad (3)$$

Keterangan:
D : total dokumen
d : dokumen ke-d
t : kata ke-t dari kata kunci
W : bobot dokumen ke-d terhadap kata ke-t
tf : banyaknya kata yang dicari pada sebuah dokumen
df : banyak dokumen yang mengandung kata yang dicari
IDF : Inversed Document Frequency

Nilai IDF didapatkan dari IDF : log2 (D/df)

6. *Cosine Similarity*

Perhitungan *cosine similarity* merupakan komponen dasar yang banyak digunakan pada aplikasi data *mining* [6]. Secara gais besar metode *cosine similarity* ini didasarkan pada perhitungan sudut antara dua buah objek (contohnya Dokumen 1 dan Dokumen 2) yang dinyatakan dalam dua buah vector dengan menggunakan keywords (kata kunci) dari sebuah dokumen sebagai ukuran [14].



Gambar 3. *Cosine Similarity*

Tingkat *similarity* pada fungsi *cosine similarity* berkisar pada interval 0 (nol) sampai dengan 1 (satu). Dokumen dikatakan mirip apabila sudut yang dibentuk antar dua buah objek adalah 0° nilai kesamaannya adalah 1 (satu) dan dokumen dikatakan tidak mirip apabila sudutnya 90° dengan nilai kesamaannya adalah 0 (nol) [15]. Oleh karena itu makin kecil sudut antara dua buah dokumen maka akan makin besar kemiripannya.
Berikut ini merupakan persamaan *Cosine Similarity* dalam menghitung kemiripan antar dua buah vektor [6].

$$\text{Cos } \alpha = \frac{A \bullet B}{|A||B|} = \frac{\sum_{i=1}^{n} Ai \times Bi}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} \quad (4)$$

Keterangan:
A = Vektor A, yang akan dibandingkan kemiripannya
B = Vektor B, yang akan dibandingkan kemiripannya
A • B = Cross product antara |A| dan |B|
|A| = Panjang vektor A
|B| = Panjang vektor B
|A||B| = Cross product antara |A| dan |B|

III. HASIL DAN PEMBAHASAN

1. Dataset
   - Jurnal *Online*. Obyek yang digunakan dalam penelitian ini adalah jurnal *online*. Tahap pertama ialah mengumpulkan jurnal *online* melalui situs-situs penerbit jurnal ilmiah.
   - Isi jurnal yang akan digunakan dalam perhitungan kemiripan teks adalah bagian abstrak dari setiap jurnal. Sebanyak 100 data jurnal yang digunakan dalam penelitian ini.
   - Database. Tahap akhir, memasukkan data kedalam database dengan format .csv dan terdiri dari satu atribut yaitu abstrak, yang nantinya dataset tersebut melalui tahap pre-processing dilanjutkan dengan metode *Cosine Similarity* untuk mengetahui kemiripan antar teks.
2. *Pre-processing*

   Tahap *pre-processing* merupakan tahap awal pada pengolahan data untuk menghasilkan dataset yang bersih sebelum dilakukan ke tahap yang lebih lanjut.

   1) Case Folding

   Case Folding adalah tahap menyeragamkan seluruh karakter menjadi huruf kecil. Hal tersebut dilakukan agar dua kata yang sama tidak mengandung makna yang berbeda misalnya ”Hari” dengan ”hari” mempunyai makna yang sama tetapi oleh komputer itu sudah berbeda. Berikut adalah hasil *case folding*:

Tabel 1. Hasil Case Folding

	abstrak
0	penyakit hepatitis merupakan penyakit peradang...
1	twitter salah satu situs sosial media yang mem...
2	judul artikel ini adalah pemanfaatan teknologi...
3	corona virus yang saat ini terjadi menjadikan ...

   2) Tokenizing

   Tokenizing Merupakan suatu proses yang dilakukan untuk memotong kalimat menjadi bagian atau kata. Bagian kata yang dipecah disebut sebuah token. Berikut adalah contoh dari hasil tokenisasi:

Tabel 2. Hasil *Tokenizing*

	abstrak
0	[penyakit, hepatitis, merupakan, penyakit, per...
1	[penyakit, degenerative, merupakan, penyakit, ...
2	[hepatitis, merupakan, salah, satu, penyakit, ...
3	[corona, virus, yang, saat, ini, terjadi, menj...

   3) Stopwords

Merupakan proses menghilangkan kata-kata yang muncul dalam jumlah besar tetapi diangggap tidak ada arti dan tidak ada hubungannya pada pengolahan data untuk memperoleh hasil yang di inginkan. Tahapan ini dilakukan setelah melewati tahap tokenisasi. Berikut adalah tampulan hasil *stopwords:*

Tabel 3. Hasil *Stopwords*

	abstrak
0	[penyakit, hepatitis, penyakit, peradangan, se...
1	[twitter, salah, situs, sosial, media, penggun...
2	[judul, artikel, pemanfaatan, teknologi, infor...
3	[corona, virus, menjadikan, perubahan, tatanan...

4) Stemming

Tahapan ini merupakan proses mengembalikan kata ke bentuk dasarnya yaitu dengan menghilangkan imbuhan awal (*prefix*) dan imbuhan akhir (*sufix*). Berikut merupakan contoh dari *stemming:*

Tabel 4. Hasil *Stemming*

	abstrak
0	sakit hepatitis sakit adang sel hati sebab inf...
1	twitter salah situs sosial media guna tulis ha...
2	judul artikel manfaat teknologi informasi pust...
3	corona virus jadi ubah tatanan hidup masyaraka...

3. TF-IDF

Dataset yang telah melalui tahap pre-processing data selanjutnya dilakukan tahap pembobotan data. Tahap ini merupakan tahapan untuk menghitung bilai bobot suatu kata dalam dokumen. Berikut merupaka hasil pembobotan kata pada dokumen:

Tabel 5. Hasil Pembobotan kata

(0, 1656)	3
(0, 741)	6
(0, 178)	1
(0, 1698)	1
(0, 735)	1
(0, 1682)	3
(0, 794)	1
(0, 2050)	2
(0, 305)	1
(0, 1356)	1
(0, 1293)	2
(0, 1956)	1
(0, 987)	1
(0, 214)	1
(0, 1073)	1
(0, 286)	1
(0, 1507)	2
(0, 788)	1
(0, 60)	2
(0, 35)	1
(0, 1940)	4
(0, 892)	1
(0, 311)	2
(0, 1627)	1
(0, 58)	1
:	:
(98, 1899)	1
(98, 475)	2
(98, 505)	2
(98, 936)	1
(98, 2091)	5

(0, 2050)	0.10733497117192455
(0, 2042)	0.17874073704291385
(0, 2025)	0.05125142981331933
(0, 1975)	0.034225708769975174
(0, 1956)	0.07229694707522227
(0, 1940)	0.1463153216893342
(0, 1939)	0.058030961230624165
(0, 1936)	0.07633440414568599
(0, 1918)	0.08153959161637643
(0, 1902)	0.08153959161637643
(0, 1893)	0.1013583949880123
(0, 1882)	0.05015451571226027
(0, 1843)	0.22900321243705796
(0, 1698)	0.07229694707522227
(0, 1682)	0.14164191822856687
(0, 1656)	0.12740062223270324
(0, 1627)	0.08887589131186932
(0, 1534)	0.08887589131186932
(0, 1516)	0.061661804754700206
(0, 1507)	0.17775178262373864
(0, 1498)	0.10483832042709212
(0, 1368)	0.058030961230624165
(0, 1361)	0.24461877484912925
(0, 1356)	0.08887589131186932
(0, 1320)	0.22900321243705796
:	:
(98, 807)	0.09107185988217348
(98, 802)	0.09926579910454472
(98, 748)	0.06143903431732087
(98, 734)	0.025490593414515945
(98, 665)	0.06143903431732087
(98, 659)	0.09107185988217348
(98, 505)	0.14789807429486865

4. Cosine Similarity

Pada tahap ini menggunakan data hasil perhitungan TF-IDF untuk menghitung tingkat kemiripan dari setiap dokumen dan menggunakan persamaan Berikut adalah hasil dari perhitungan Cosine Similarity yang ditampilkan dalam bentuk array:

```
array([[1.        , 0.0440798 , 0.0105683 , 0.13637751, 0.05147174,
        0.32136474, 0.19103343, 0.08743242, 0.1060636 , 0.2630254 ,
        0.14621892, 0.03776003, 0.10809402, 0.38734232, 0.0297519 ,
        0.06932133, 0.21074754, 0.17627011, 0.43137813, 0.06289827,
        0.11023139, 0.04604918, 0.15851834, 0.17415209, 0.05499508,
        0.10743814, 0.02485119, 0.04854089, 0.04361529, 0.05128409,
        0.0376293 , 0.04963766, 0.04158563, 0.01355905, 0.0354927 ,
        0.02423605, 0.01487817, 0.01059508, 0.03094966, 0.01772586,
        0.01642642, 0.0021334 , 0.0021334 , 0.09393885, 0.22545245,
        0.41215318, 0.13274819, 0.46068627, 0.00989635, 0.05199529,
        0.0148089 , 0.04711814, 0.175128  , 0.1179381 , 0.02586311,
        0.0301234 , 0.03507969, 0.00531528, 0.02338296, 0.04347385,
        0.00168811, 0.03353105, 0.04464348, 0.01972923, 0.01468886,
        0.00993218, 0.01258757, 0.01059508, 0.01059508, 0.10899439,
        0.00794625, 0.02937917, 0.01308678, 0.05492541, 0.03068132,
        0.01725554, 0.0216874 , 0.02933112, 0.03234683, 0.04111759,
        0.01581646, 0.00973794, 0.0187137 , 0.03771874, 0.01258842,
        0.08430125, 0.00314642, 0.03936579, 0.02669175, 0.01820447,
        0.18800142, 0.08570367, 0.01835395, 0.05389754, 0.01126355,
        0.02634782, 0.0366661 , 0.02618594, 0.0237398 ]])
```

Gambar 4. Hasil *Cosine Similarity*

Jika di urutkan index dari hasil similaritas mulai dari nilai terkecil hingga yang terbesar maka kita akan menemukan bahwa di index ke-60 merupakan dokumen dengan tingkat similaritas terkecil begitupun index ke-0 merupakan dokumen dengan tingkat similaritas tertinggi. Untuk lebih jelasnya berikut ditampilkan pada Gambar 5.

```
array([[60, 42, 41, 86, 57, 70, 81, 48, 65,  2, 37, 67, 68, 94, 66, 84,
        72, 33, 64, 50, 36, 80, 40, 75, 39, 89, 92, 82, 63, 76, 58, 98,
        35, 26, 54, 97, 95, 88, 77, 71, 14, 55, 74, 38, 78, 61, 56, 34,
        96, 30, 83, 11, 87, 79, 32, 59, 28,  1, 62, 21, 51, 27, 31, 29,
         4, 49, 93, 73, 24, 19, 15, 85, 91,  7, 43,  8, 25, 12, 69, 20,
        53, 46,  3, 10, 22, 23, 52, 17, 90,  6, 16, 44,  9,  5, 13, 45,
        18, 47,  0]], dtype=int64)
```

Gambar 5. Urutan Index Tingkat Plagiarisme

IV. KESIMPULAN

Data yang digunakan dalam penelitian ini berasal dari situs penerbit jurnal online. Jumlah data yang digunakan dalam penelitian ini ialah sebanyak 100 data jurnal dan yang diolah hanya bagian abstrak dari jurnal. Tahap *pre-processing* merupakan tahap awal yang dilakukan untuk menghilangkan inkonsistensi data sebelum dilanjutkan ke tahap berikutnya. Berdasarkan hasil penelitian dan pembahasan tentang mendeteksi plagiarisme pada jurnal menggunakan metode *Cosine Similarity* dapat disimpulkan bahwa algoritma *Cosine Similarity* dapat mendeteksi kemiripan antar dokumen yang ada di dataset, serta dapat diurutkan berdasarkan nilai kemiripan terkecil hingga terbesar.

REFERENSI

[1] H. Maurer and F. Kappe, “Plagiarism - A Survey,” vol. 12, no. 8, pp. 1050–1084, 2006.

[2] S. Sastroasmoro, “Beberapa Catatan tentang Plagiarisme *,” *Maj Kedokt Indon*, vol. Volum: 57, pp. 239–244, 2007.

[3] D. Steveson, H. Agung, and F. Mulia, “Aplikasi Pendeteksi Plagiarisme Tugas Dan Makalah Pada Sekolah Menggunakan Algoritma Rabin Karp,” *J. Algoritm. Log. dan Komputasi*, vol. 1, no. 1, pp. 12–17, 2018, doi: 10.30813/j-alu.v1i1.1104.

[4] F. T. Informasi *et al.*, “PERBANDINGAN ALGORITMA WINNOWING DENGAN ALGORITMA RABIN KARP UNTUK MENDETEKSI,” vol. 8, no. 3, pp. 124–134, 2017.

[5] B. Herwijayanti, D. E. Ratnawati, and L. Muflikhah, “Klasifikasi Berita Online dengan menggunakan Pembobotan TF-IDF dan Cosine Similarity,” *Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 1, pp. 306–312, 2018.

[6] B. Li, “Distance Weighted Cosine Similarity Measure for Text Classification,” no. October 2013, 2020, doi: 10.1007/978-3-642-41278-3.

[7] W. Gunawan and H. D. Wijaya, “An Application of Multimedia for Basic Arabic Learning Using FisherYates Shuffle Algorithm on Android Based,” *Sch. Bull.*, pp. 347–355, 2019.

[8] H. D. Wijaya and S. Dwiasnati, “Implementasi Data Mining dengan Algoritma Naïve Bayes pada Penjualan Obat,” *J. Inform.*, vol. 7, no. 1, pp. 1–7, 2020, doi: 10.31311/ji.v7i1.6203.

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

[10] I. E. Letters, “AND APRIORI ALGORITHM FOR STORE DISPLAY DESIGN,” vol. 15, no. 11, pp. 1221–1226, 2021, doi: 10.24507/icicel.15.11.1221.

[11] A. Bayhaqy, U. N. Mandiri, and S. Sfenrianto, “Sentiment Analysis about E-Commerce from Tweets Using Decision Tree , K-Nearest Neighbor , and Naïve Bayes,” no. November 2019, 2018, doi: 10.1109/ICOT.2018.8705796.

[12] S. Salah, S. Syarat, U. Memperoleh, and R. Melita, “Uin syarif hidayatullah jakarta,” 2018.

[13] M. Nurjannah and I. F. Astuti, “PENERAPAN ALGORITMA TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF) UNTUK TEXT MINING,” vol. 8, no. 3, pp. 110–113, 2013.

[14] A. Sholihin, N. Puspitasari, and M. Wati, “Analisis Penyakit Difteri Berbasis Twitter Menggunakan Algoritma Naïve Bayes,” vol. 1, no. 1, pp. 7–15, 2019.

[15] R. P. Pratama, M. Faisal, and A. Hanani, “Deteksi Plagiarisme pada Dokumen Jurnal Menggunakan Metode Cosine Similarity,” *SMARTICS J.*, vol. 5, no. 1, pp. 22–26, 2019, doi: 10.21067/smartics.v5i1.2848.