

## ANALISIS PERBANDINGAN COSINE SIMILARITY DAN JACCARD SIMILARITY PADA SISTEM REKOMENDASI ARTIKEL ILMIAH BERBASIS TF-IDF

Zeva Muhammad Calvine, Eko Purwanto, Rudi Susanto

Teknik Informatika, Universitas Duta Bangsa Surakarta  
Jl. Bhayangkara No 55-57 Tipes, Serengan, Kota Surakarta, Jawa Tengah  
zevamuhammad1@gmail.com

### ABSTRAK

Peningkatan jumlah publikasi ilmiah saat ini sering kali memicu fenomena kelebihan informasi (*information overload*). Kondisi ini menimbulkan permasalahan bagi para peneliti, terutama mahasiswa tingkat akhir, berupa kesulitan dalam menemukan literatur yang relevan. Oleh karena itu, penelitian ini bertujuan untuk membangun sebuah sistem rekomendasi artikel ilmiah berbasis *Content-Based Filtering* guna mempermudah proses pencarian referensi yang tepat sasaran. Penelitian ini menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF) sebagai bentuk representasi dokumen teks. Sebelum proses pembobotan, data teks diekstraksi melalui tahapan prapemrosesan yang meliputi pembersihan (*cleaning*), penyeragaman huruf (*case folding*), penghapusan kata umum (*stopword removal*), dan pemotongan imbuhan (*stemming*). Selanjutnya, pengukuran kemiripan antarartikel diuji menggunakan dua metode komparasi, yaitu *Cosine Similarity* dan *Jaccard Similarity*. Hasil penelitian menunjukkan bahwa algoritma *Cosine Similarity* memberikan performa yang lebih unggul dibandingkan *Jaccard Similarity*. Berdasarkan hasil evaluasi, *Cosine Similarity* memperoleh nilai presisi (*precision*) dan *recall* yang lebih tinggi, sehingga terbukti lebih efektif untuk diimplementasikan pada sistem rekomendasi literatur berbasis teks.

**Kata kunci :** Sistem rekomendasi, TF-IDF, Cosine Similarity, Jaccard Similarity, Text Mining.

### 1. PENDAHULUAN

Pertumbuhan pesat publikasi ilmiah pada era digital telah memunculkan fenomena kelebihan informasi (*information overload*), yakni sebuah situasi di mana volume informasi melampaui kapasitas manusia untuk menyaring dan memprosesnya secara efektif [1].

Lonjakan kuantitas artikel ilmiah dari tahun ke tahun memperumit proses penelusuran literatur, khususnya bagi peneliti pemula seperti mahasiswa tingkat akhir. Kelompok ini umumnya masih menghadapi kendala dalam merumuskan kata kunci yang akurat serta menyeleksi rujukan yang benar-benar relevan [2].

Penelusuran literatur yang sekadar bergantung pada kata kunci sederhana kerap kali mengembalikan dokumen dengan tingkat keterkaitan yang bervariasi. Dampaknya, proses pencarian menjadi tidak efisien, menyita banyak waktu, dan berisiko mendegradasi mutu penelitian [3], [4].

Di samping itu, ketiadaan mekanisme penyaringan otomatis yang sanggup menyajikan rekomendasi artikel ilmiah secara presisi dan relevan menjadi pokok permasalahan utama dalam penelitian ini.

Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sebuah sistem rekomendasi artikel ilmiah dengan pendekatan *Content-Based Filtering*. Sistem ini dirancang agar mampu menyuguhkan hasil penelusuran yang lebih relevan berdasarkan tingkat kemiripan konten antar-dokumen.

Sebagai upaya penyelesaian masalah, penelitian ini mengimplementasikan metode *Term Frequency-Inverse Document Frequency* (TF-IDF) untuk

representasi dokumen, yang dipadukan dengan pengukuran kemiripan melalui algoritma *Cosine Similarity* dan *Jaccard Similarity*. Pengolahan teks juga melibatkan tahapan prapemrosesan (*preprocessing*) yang mencakup pembersihan data (*cleaning*), penyeragaman ukuran huruf (*case folding*), pembuangan kata umum yang tidak bermakna (*stopword removal*), dan penemuan kata dasar (*stemming*). Selanjutnya, kinerja sistem akan dievaluasi menggunakan metrik pengukuran presisi (*precision*) dan *recall*.

Perbedaan karakteristik dasar antara *Cosine Similarity* dan *Jaccard Similarity* dalam mengalkulasi kedekatan dokumen berpotensi menghasilkan keluaran rekomendasi yang berbeda. Atas dasar tersebut, penelitian ini juga mengkaji komparasi kinerja antara kedua metode tersebut, khususnya dalam lingkup sistem rekomendasi artikel ilmiah yang berbasis pada teks abstrak.

### 2. TINJAUAN PUSTAKA

#### 2.1. Sistem Rekomendasi

Dengan sasaran untuk membantu pengguna dalam menemukan hal-hal yang paling relevan (item yang diminati) dari sejumlah besar datanya, sistem rekomendasi ialah sistem perangkat lunak yang mencoba meramalkan preferensi ataupun peringkat yang akan diberikan pengguna terhadap suatu item [5].

Dalam konteks akademik, sistem ini berfungsi untuk menyaring literatur ilmiah, artikel, atau abstrak, dari repositori yang luas, sehingga memitigasi masalah *information overload* [6].

Pada dasarnya, ada tiga jenis sistem rekomendasi ialah “*Collaborative Filtering*, *Content-Based Filtering*, dan *Hybrid Methods*”.

## 2.2. Content-Based Filtering

Salah satu metode yang dipakai dalam sistem rekomendasi dikenal sebagai *content-based filtering* (CBF). Metode ini menyarankan item kepada pengguna tergantung pada seberapa mirip profil mereka dengan pilihan sebelumnya [7][8].

Dalam penelitian ini, CBF digunakan untuk merekomendasikan artikel berdasarkan kemiripan isi abstrak artikel target (*query document*) dengan abstrak artikel lainnya di dalam dataset. Keunggulan utama CBF adalah kemampuannya untuk memberikan rekomendasi kepada pengguna baru tanpa memerlukan riwayat pengguna lain (*cold-start problem*) [6].

Proses utama CBF melibatkan tiga tahapan: *feature extraction* dari item (abstrak), *profile learning* (representasi vektor), dan *similarity calculation* (perhitungan kemiripan).

## 2.3. Term Frequency-Inverse Document Frequency (TF-IDF)

Para profesional di bidang information retrieval serta text mining sering kali menerapkan TF-IDF, sebuah metode pembobotan statistik [9][10], guna merefleksikan kepentingan relatif dari kata-kata (term) di dalam sebuah dokumen (corpus) [11].

Dua bagian yang membentuk pembobotan TF-IDF ialah:

- Term Frequency* (TF): Menemukan frekuensi kemunculan suatu istilah di dalam teks. Kata-kata yang muncul lebih kerap dalam sebuah dokumen dianggap lebih krusial.
- Inverse Document Frequency* (IDF): Menghitung frekuensi keseluruhan kemunculan sebuah kata di semua dokumennya. Keunikan ataupun keistimewaan sebuah kata sebanding dengan kelangkaannya di dalam corpus.

Sebuah alat yang ampuh dalam menghasilkan kata kunci tema dari abstrak artikelnya, TF-IDF mengalikan TF serta IDF untuk memberikan bobot lebih pada istilah yang kerap muncul dalam satu dokumen (relevan) namun jarang muncul di dokumen lain (*distinctive*) [12].

$$TF - IDF_{(t,d)} = TF_{(t,d)} \times IDF_{(t)} \quad (1)$$

Keterangan :

$f_{t,d}$  : frekuensi kata  $t$  dalam dokumen  $d$

$N$  : jumlah seluruh dokumen

$df(t)$  : jumlah dokumen yang mengandung kata  $t$

## 2.4. Cosine Similarity

*Cosine Similarity* ialah ukuran kemiripan antara dua vektor dalam ruang tiga dimensinya [13] dengan menghitung nilai kosinus sudut yang yang tidak bergantung pada besaran vektor serta menghitung kosinus dari sudut yang tercipta di antara keduanya. Ukuran vektor tidak berpengaruh pada teknik ini,

melainkan berfokus pada arah vektor tersebut, sehingga mampu merepresentasikan kedekatan makna antar dokumen secara lebih akurat [14].

Dalam keperluan pemrosesan teksnya, serta khususnya setelah pembobotan TF-IDF, setiap dokumen ditampilkan sebagai sebuah vektor yang mempunyai bobot katanya. Setelah itu, tingkat kemiripan antar teks diukur memakai *Cosine Similarity*, yang memperhitungkan kedekatan vektornya [15]. Dengan nilai 1 yang mengindikasikan tingkat kemiripan yang sangat tinggi (sudut  $0^\circ$ ) serta nilai 0 yang mengindikasikan tidak ada kemiripan (sudut  $90^\circ$ ), nilai kemiripan yang dihasilkannya bisa berupa nilai nol hingga satu.

Keunggulan utama dari *Cosine Similarity* terletak pada kemampuannya dalam mengukur kesamaan berdasarkan distribusi bobot kata tanpa dipengaruhi oleh panjang dokumen [15]. Dengan demikian, metode ini sangat sesuai digunakan pada data teks, karena mampu menekankan kesamaan isi atau konteks dokumen, bukan sekadar jumlah kata yang digunakan. Inilah sebabnya mengapa sistem rekomendasi berbasis teks kerap kali memanfaatkan *Cosine Similarity* agar memperoleh rekomendasi yang lebih baik serta akurat.

## 2.5. Jaccard Similarity

*Jaccard Similarity* adalah metode pengukuran kemiripan berbasis himpunan (*set-based similarity measure*) yang menghitung rasio antara jumlah elemen yang sama (*intersection*) dan jumlah keseluruhan elemen unik (*union*) dari dua dokumen [15].

Berbeda dengan pendekatan berbasis vektor yang mempertimbangkan bobot frekuensi, Jaccard berfokus pada keberadaan kata kunci dalam sebuah dokumen [2][12].

Secara matematis, *Jaccard Similarity* dirumuskan sebagai:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (2)$$

Keterangan :

$A$  dan  $B$  : himpunan kata unik

$\cap$  : irisan

$\cup$  : gabungan

Beberapa penelitian sebelumnya telah menggunakan metode *Content-Based Filtering*, TF-IDF, serta metode pengukuran kemiripan seperti *Cosine Similarity* dan *Jaccard Similarity* dalam sistem rekomendasi berbasis teks.

Penelitian Pradana dan Alamsyah[16], menunjukkan bahwa kombinasi TF-IDF dan *Cosine Similarity* mampu menghasilkan tingkat presisi yang tinggi pada sistem rekomendasi artikel ilmiah berbasis abstrak.

Selain itu, penelitian Rinjeni [17] menunjukkan bahwa *Cosine Similarity* memiliki performa yang lebih baik dibandingkan *Jaccard Similarity* dalam pencocokan dokumen ilmiah.

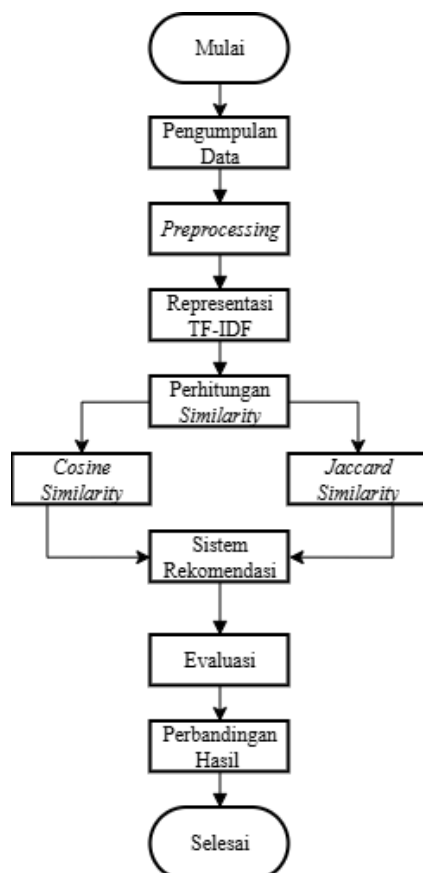
Sementara itu, penelitian Sukestiyarno [18] membuktikan bahwa kedua metode tersebut dapat

digunakan dalam sistem rekomendasi berbasis Content-Based Filtering, meskipun terdapat perbedaan hasil evaluasi.

Berdasarkan penelitian tersebut, masih terdapat celah penelitian berupa minimnya studi yang secara khusus membandingkan Cosine Similarity dan Jaccard Similarity pada sistem rekomendasi artikel ilmiah berbasis abstrak dengan evaluasi precision dan recall.

### 3. METODE PENELITIAN

Kerangka kerja riset dibuat guna memastikan bahwasanya proses studi ini terorganisir, metodis, serta efisien. Seperti yang terlihat pada Gambar 1, diagram alir studi ini terdiri dari beberapa tahapan utamanya.



Gambar 1. Alur Penelitian

#### 3.1. Pengumpulan Data

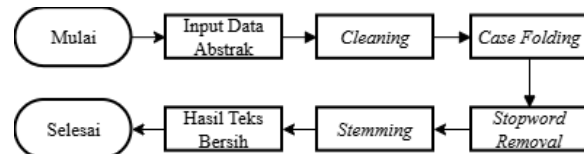
Pada awalnya, penulis menghimpun informasi dari abstrak jurnal ilmiah yang terdapat di database nasional publikasi ilmiah. Terdapat total 110 dokumen dalam format .csv yang dimanfaatkan dalam kumpulan data studi ini. Dalam maksud analisa kemiripan dokumen, dataset ini menyertakan teks abstraknya.

#### 3.2. Pre-Processing

Guna memperoleh data teks yang selaras agar diproses dalam langkah-langkah analisa selanjutnya, tahap *preprocessing* dipergunakan[19].

Mengingat data abstrak yang diperoleh umumnya masih mengandung berbagai karakter yang tidak

terstruktur, diperlukan serangkaian langkah sistematis untuk menstandarisasi data tersebut. Sehingga di perlukan *pre-processing* data dengan alur sebagai berikut, *Cleaning*, *Case Folding*, *Stopword Removal*, *Stemming*. Kerangka langkah *pre-processing* dapat diketahui pada Gambar 2.



Gambar 2. Langkah Pre-Processing

Seluruh rangkaian tahapan ini bertujuan untuk menghasilkan teks bersih (*clean text*) yang memiliki konsistensi tinggi, sehingga dapat meningkatkan akurasi serta efisiensi di seluruh pemrosesan data yang mengikutinya.

#### 3.3. Transformasi Data dengan TF-IDF

Pendekatan TF-IDF dipakai guna mengubah data teks menjadi representasi numerik sesudah tahap preprocessing. Kata-kata diberi bobot selaras dengan signifikansinya dalam dokumen serta corpus lengkap dengan memanfaatkan prosedur ini. Prosedur ini menciptakan representasi matriks vektor dari dokumen agar dipakai dalam perhitungan kemiripan. Berikut ialah formulasi matematisnya dari *TF-IDF*:

$$TF_{(t,d)} = \frac{f_{(t,d)}}{\sum f_{(d)}} \quad (3)$$

$$IDF_{(t)} = \log\left(\frac{N}{df(t)}\right) \quad (4)$$

$$TF-IDF_{(t,d)} = TF_{(t,d)} \times IDF_{(t)} \quad (5)$$

Keterangan :

$f_{t,d}$  : frekuensi kata  $t$  dalam dokumen  $d$

$N$  : jumlah seluruh dokumen

$df(t)$  : jumlah dokumen yang mengandung kata  $t$

```

# =====
# TF-IDF
# =====
vectorizer = TfidfVectorizer(
    max_features=5000,
    ngram_range=(1,2)
)

tfidf_matrix = vectorizer.fit_transform(df['clean'])

print("\nTF-IDF berhasil dibuat")
print("Jumlah fitur:", len(vectorizer.get_feature_names_out()))
print("Shape matrix:", tfidf_matrix.shape)
  
```

Gambar 3. Pembobotan TF-IDF

Dalam implementasinya, pembobotan TF-IDF dilakukan menggunakan library scikit-learn dengan parameter:

`max_features = 5000`

`ngram_range = (1,2)`

Parameter ini digunakan dalam membatasi jumlah fitur serta mempertimbangkan unigram beserta bigram dalam representasi teks.

### 3.4. Cosine Similarity

Dengan menghitung sudut antara dua vektor dalam ruang tiga dimensi, *Cosine Similarity* menentukan seberapa mirip dua kertas satu sama lainnya.

Rumus *Cosine Similarity* adalah:

$$\text{Cosine}(A, B) = \frac{AB}{\|A\| \times \|B\|} \quad (6)$$

Terdapat kisaran nilai bagi *Cosine Similarity* dari 0 hingga 1, di mana:

- Nilai 1 memperlihatkan kemiripan tinggi
- Nilai 0 memperlihatkan tidak ada kemiripan

Metode ini mempertimbangkan bobot kata dalam dokumen sehingga lebih sensitif terhadap distribusi kata.

```
# =====
# COSINE SIMILARITY
# =====
cosine_sim = cosine_similarity(tfidf_matrix)

print("\nCosine Similarity selesai")
```

Gambar 4. *Cosine Similarity*

Studi ini memanfaatkan paket *scikit-learn* di Python dalam menghitung *cosine similarity*, seperti yang terlihat pada Gambar 4 di atas. Matriks kemiripan berbasis ukuran dihasilkan selama metode komputasinya dengan membandingkan vektor TF-IDF di seluruh dokumenya  $N \times N$ , di mana setiap elemen matriks menunjukkan tingkat kemiripan antara dua dokumen.

### 3.5. Jaccard Similarity

```
# =====
# FUNGSI JACCARD
# =====
def jaccard_similarity(text1, text2):
    set1 = set(text1.split())
    set2 = set(text2.split())

    intersection = len(set1 & set2)
    union = len(set1 | set2)

    if union == 0:
        return 0

    return intersection / union
```

Gambar 5. *Jaccard Similarity*

Salah satu cara dalam membandingkan dua dokumen ialah melalui *Jaccard Similarity*, yang memanfaatkan sekelompok kata sebagai proksi dari aspek-aspek yang sama. Teknik ini didasari oleh perbandingan jumlah total kata dari dua dokumenya, yang merupakan gabungan dari semua kata unik (*union*) serta perpotongan semua kata yang sama (*intersection*). Berikut ini ialah formulasi matematis dari *Jaccard Similarity*:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (7)$$

Keterangan :

*A dan B* : himpunan kata unik

$\cap$  : irisan

$\cup$  : gabungan

Skor *Jaccard Similarity* bisa berkisar dari nol hingga satu, di mana:

- Nilai 1 memperlihatkan kedua dokumen identik
- Nilai 0 memperlihatkan tidak ada kesamaan kata

Berdasarkan kode tersebut, setiap dokumen direpresentasikan sebagai kumpulan kata, kemudian dihitung kesamaan kata antar dokumen. Nilai kemiripan diperoleh dari perbandingan total katanya yang sama dengan total seluruh katanya yang unik.

Perhitungan ini dilakukan untuk seluruh dokumen sehingga menghasilkan nilai kemiripan yang digunakan dalam menentukan rekomendasi artikel.

### 3.6. Evaluasi Sistem

Evaluasi sistem dilaksanakan dalam mengukur kinerja metode yang digunakan dalam menghasilkan rekomendasi artikel ilmiah. Pada penelitian ini, evaluasi difokuskan pada perbandingan performa antara metode *Cosine Similarity* dan *Jaccard Similarity* dalam menentukan tingkat relevansi dokumen. *Precision* serta *Recall* ialah dua ukuran krusial yang kerap diterapkan dalam sistem pencarian informasi dalam mengukur kinerja sistemnya.

### 3.7. Precision

*Precision* digunakan untuk mengukur tingkat ketepatan sistem dalam memberikan rekomendasi. Indikator ini memperlihatkan seberapa baik dokumen yang dianjurkan oleh sistem konsisten dengan kebutuhan penggunaanya. Secara matematis, *Precision* dirumuskan sebagai berikut:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (8)$$

Keterangan:

TP (*True Positive*): “jumlah dokumen relevan yang berhasil direkomendasikan

FP (*False Positive*): jumlah dokumen tidak relevan yang direkomendasikan”.

Kemampuan sistem dalam memberikan saran yang akurat diperlihatkan oleh skor *Precision* yang tinggi.

### 3.8. Recall

*Recall* sistem ialah ukuran seberapa baik sistem bisa mengekstrak dokumen yang relevan dari kumpulan datanya yang besar[20].

Rumus *Recall* adalah sebagai berikut:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (9)$$

Keterangan:

FN (*False Negative*): “jumlah dokumen relevan yang tidak berhasil direkomendasikan”.

Kemampuan sistem dalam menemukan sebagian besar dokumen yang relevan diperlihatkan oleh nilai *Recall* yang tinggi.

### 3.9. Prosedur Evaluasi

Berikut ialah langkah-langkah yang dipergunakan dalam melaksanakan evaluasi:

- Menentukan satu dokumen sebagai dokumen acuan (*query*)
- Menghasilkan daftar rekomendasi dokumen menggunakan metode *Cosine Similarity* dan *Jaccard Similarity*
- Mengambil sejumlah dokumen teratas (*top-N*) dari hasil rekomendasi
- Menentukan dokumen yang relevan secara manual berdasarkan kesesuaian topik
- Menghitung nilai *Precision* dan *Recall* untuk masing-masing metode
- Membandingkan hasil evaluasi untuk menentukan metode terbaik

### 3.10. Interpretasi Hasil Evaluasi

Hasil evaluasi digunakan untuk menganalisis performa kedua metode dalam menghasilkan

rekomendasi. Metode dengan nilai *Precision* dan *Recall* yang lebih tinggi dianggap memiliki kinerja yang lebih baik dalam mengidentifikasi dokumen yang relevan. Selain itu, hasil evaluasi juga digunakan untuk memahami karakteristik masing-masing metode, di mana *Cosine Similarity* cenderung lebih mempertimbangkan bobot kata, sedangkan *Jaccard Similarity* lebih sederhana karena hanya berdasarkan kesamaan kata.

## 4. HASIL DAN PEMBAHASAN

### 4.1. Hasil Preprocessing Data

Sebelum pemrosesan selanjutnya, data teks dibersihkan serta dinormalisasinya selama langkah *preprocessing*. Proses yang dilakukan mencakup *cleaning*, *case folding*, *stopword removal*, dan *stemming*.

Berdasarkan hasil *preprocessing*, diperoleh sebanyak 110 data abstrak yang siap digunakan dalam proses analisis. Contoh hasil *preprocessing* diperlihatkan pada Tabel 1.

Tabel 1. Contoh Hasil *Preprocessing*

| No | Teks Asli (Abstract)                                                                                        | Hasil <i>Preprocessing</i> (Clean)                                               |
|----|-------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|
| 1  | SMK Muhammadiyah Majenang merupakan unit pelaksana pendidikan yang bertujuan meningkatkan kualitas siswa... | smk muhamadiah majenang unit pelaksana pendidikan tuju tingkat kualitas siswa... |
| 2  | Sistem yang dapat memudahkan staf administrasi dalam pengolahan data siswa sangat dibutuhkan...             | sistem memudahkan staf administrasi olah data siswa butuh...                     |
| 3  | LPK Shinju telah menyimpan data dalam basis data, namun belum dimanfaatkan secara optimal...                | lpk shinju simpan data basis data namun belum manfaat optimal...                 |
| 4  | Prestasi akademik mahasiswa merupakan indikator penting dalam evaluasi kualitas pendidikan...               | prestasi akademik mahasiswa indikator penting evaluasi kualitas pendidikan...    |
| 5  | Penelitian ini membahas penggunaan metode Naive Bayes untuk menganalisis pola prestasi siswa...             | teliti bahas guna metode naive bayes analisis pola prestasi siswa...             |

Berdasarkan Tabel 4.1, terlihat bahwa proses *preprocessing* berhasil membersihkan teks dengan menghilangkan kata yang tidak penting serta mengubah kata ke bentuk dasar. Hal ini membuat data lebih terstruktur dan siap digunakan pada tahap selanjutnya.

### 4.2. Hasil Transformasi *TF-IDF*

Setelah tahap *preprocessing*, teknik *TF-IDF* dipergunakan dalam mengubah data teksnya ke dalam format numerik. Setiap dokumen diwakili oleh matriks bobot kata yang dihasilkan oleh teknik ini.

Hasil transformasi menunjukkan bahwa:

- Jumlah dokumen: 110
- Jumlah fitur: 5000
- Bentuk matriks:  $(110 \times 5000)$

Dalam sebuah dokumen, *TF-IDF* memberikan bobot lebih pada istilah-istilah yang dianggap sangat relevan. Ini berarti bahwasanya fitur-fitur dari setiap dokumen bisa ditangkap oleh representasi vektor dari dokumen tersebut.

### 4.3. Hasil *Cosine Similarity*

Tingkat kemiripan antara teks diukur dengan menerapkan *Cosine Similarity*, yang didasari oleh vektor *TF-IDF*. Hasil perhitungan menghasilkan matriks kemiripan berukuran  $110 \times 110$ , di mana setiap nilai menunjukkan tingkat kesamaan antara dua dokumen.

Contoh hasil rekomendasi untuk satu dokumen diperlihatkan pada Tabel 2.

Tabel 2. Hasil *Cosine Similarity*

| Index | Nilai <i>Similarity</i> |
|-------|-------------------------|
| 2     | 0.403                   |
| 19    | 0.290                   |
| 106   | 0.221                   |
| 95    | 0.215                   |
| 79    | 0.192                   |

Berdasarkan hasil tersebut, dokumen dengan indeks 2 memiliki tingkat kemiripan tertinggi dengan dokumen acuan. Hal ini menunjukkan bahwa kedua dokumen memiliki kesamaan dalam distribusi kata yang digunakan. Secara umum, *Cosine Similarity*

mampu memberikan hasil yang lebih akurat karena mempertimbangkan bobot kata dalam dokumen.

#### 4.4. Hasil Jaccard Similarity

Selain menggunakan *Cosine Similarity*, penelitian ini juga menerapkan *Jaccard Similarity* untuk mengukur kemiripan berdasarkan kesamaan kata.

Tabel 3 Hasil Jaccard Similarity

| Index | Nilai Similarity |
|-------|------------------|
| 2     | 0.167            |
| 19    | 0.163            |
| 85    | 0.151            |
| 67    | 0.148            |
| 106   | 0.143            |

Nilai kemiripan yang dihasilkan oleh *Jaccard Similarity* cenderung lebih rendah dibandingkan dengan *Cosine Similarity*. Hal ini disebabkan karena metode *Jaccard* hanya mempertimbangkan keberadaan kata tanpa memperhatikan frekuensinya.

#### 4.5. Hasil Evaluasi Sistem

Evaluasi dilakukan untuk mengukur kinerja kedua metode dalam menghasilkan rekomendasi yang relevan. Pengukuran dilakukan menggunakan metrik *Precision* dan *Recall* berdasarkan hasil rekomendasi top-5 Hasil evaluasi diperlihatkan pada Tabel 4.

Tabel 4. Hasil Evaluasi Sistem

| Metode                    | Precision | Recall |
|---------------------------|-----------|--------|
| <i>Cosine Similarity</i>  | 1.00      | 1.00   |
| <i>Jaccard Similarity</i> | 0.60      | 0.60   |

Berdasarkan hasil tersebut, *Cosine Similarity* menunjukkan nilai *Precision* dan *Recall* yang lebih tinggi dibandingkan *Jaccard Similarity*.

#### 4.6. Pembahasan

Ketika membandingkan kedua metode tersebut, temuan memperlihatkan bahwasanya pendekatan *Cosine Similarity* menghasilkan saran yang lebih baik dalam artikel ilmiah daripada *Jaccard Similarity* [21]. Salah satu alasannya ialah dikarenakan *Cosine Similarity* bisa menangkap tingkat relevansinya serta distribusi kata di dalam teks dengan lebih tepat dikarenakan memakai teknik *TF-IDF* untuk mendeskripsikan dokumen sebagai vektor berbobot.

Temuan studi ini konsisten dengan studi yang dilaksanakan oleh Rinjeni dkk. yang membuktikan bahwasanya metode *Cosine Similarity* mampu menghasilkan nilai kemiripan tertinggi dibandingkan metode lainnya, dengan nilai maksimum mencapai 0,8 pada beberapa skenario pengujian[17].

Selain itu, dalam beberapa keadaan, studi ini mendemonstrasikan bahwasanya *Cosine Similarity* mengungguli *Jaccard Similarity* dalam hal efisiensi komputasi.

Riset lain memperlihatkan bahwasanya *Cosine Similarity* lebih efektif daripada *Jaccard Similarity* pada sejumlah kriteria penilaian, termasuk *F1-score*,

*recall*, serta akurasi, sehingga memberikan kepercayaan pada temuan ini [22].

Hal ini disebabkan karena *Cosine Similarity* menggunakan representasi vektor berbobot TF-IDF yang mampu menangkap relevansi kontekstual, sedangkan *Jaccard Similarity* hanya berbasis pada kesamaan himpunan kata tanpa mempertimbangkan frekuensi kemunculan kata.

Sebaliknya, *Jaccard Similarity* memiliki keterbatasan dalam merepresentasikan hubungan semantik antar-dokumen karena hanya mengukur keberadaan kata. Meskipun metode ini memiliki perhitungan yang lebih sederhana, namun dalam beberapa kondisi membutuhkan waktu komputasi yang lebih besar dan menghasilkan tingkat kemiripan yang lebih rendah[17].

Konsisten dengan studi lain, studi ini menemukan bahwasanya sistem rekomendasi berbasis teks mempunyai performa yang lebih baik ketika TF-IDF serta *Cosine Similarity* diterapkan secara bersamaan, khususnya dalam konteks pencarian artikel ilmiah berbasis abstrak, karena mampu menghasilkan rekomendasi yang lebih relevan dan akurat dibandingkan pendekatan berbasis himpunan seperti *Jaccard Similarity*.

## 5. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian yang telah dilaksanakan, dapat disimpulkan bahwa sistem rekomendasi artikel ilmiah berbasis *Content-Based Filtering* berhasil dibangun menggunakan representasi teks dengan metode *Term Frequency-Inverse Document Frequency* (TF-IDF). Sistem ini terbukti mampu mengolah data abstrak artikel menjadi bentuk vektor untuk menghitung tingkat kemiripan antardokumen. Dalam pengukuran kemiripan tersebut, metode *Cosine Similarity* dan *Jaccard Similarity* dapat diterapkan secara efektif, meskipun keduanya memiliki karakteristik yang berbeda. *Cosine Similarity* memperhitungkan bobot kata dalam dokumen, sedangkan *Jaccard Similarity* sebatas mempertimbangkan keberadaan kata tersebut. Hasil evaluasi menunjukkan bahwa pendekatan *Cosine Similarity* jauh lebih unggul dalam menghasilkan rekomendasi artikel yang relevan dibandingkan dengan pendekatan berbasis himpunan seperti *Jaccard Similarity*. Keunggulan kombinasi TF-IDF dan *Cosine Similarity* ini dibuktikan oleh perolehan nilai *precision* dan *recall* yang sempurna, yakni 1,00, berbanding jauh dengan metode *Jaccard Similarity* yang hanya mencapai nilai 0,60.

## DAFTAR PUSTAKA

- [1] F. Dwi and W. Andriyani, "Pengembangan Sistem Rekomendasi Pembimbing Tugas Akhir Menggunakan Teknik Content Based Filtering Development of a Final Project Supervisor Recommendation System Using Content-Based Filtering Technique," *J. Inform. DANKOMPUTER*, no. 2, pp. 474–483, 2025, doi:

- 10.26798/jiko.v9i2.1599.
- [2] A. A. Nuralmi, M. N. Abdurrazaq, and I. Maulana, "Kondisi Information Overload pada Mahasiswa IAI AL-AZIS Akibat Penggunaan Media Sosial yang Tidak Sehat," *ELSEVIER*, vol. 5, no. 12, pp. 5580–5596, 2024.
  - [3] J. Mothe, "Analytics Methods to Understand Information Retrieval Effectiveness—A Survey," *Mathematics*, vol. 10, no. 12, 2022, doi: 10.3390/math10122135.
  - [4] F. Lechtenberg, J. Farreres, A. L. Galvan-Cara, A. Somoza-Tornos, A. Espuña, and M. Graells, "Information retrieval from scientific abstract and citation databases: A query-by-documents approach based on Monte-Carlo sampling," *Expert Syst. Appl.*, vol. 199, no. January, p. 116967, 2022, doi: 10.1016/j.eswa.2022.116967.
  - [5] I. S. Wibowo *et al.*, "Keyword Extraction Judul Berita Online Di Indonesia Menggunakan Metode TF-IDF 1,2," *J. Tek. Inform. dan Sist. Inf.*, vol. 11, no. 1, 2024.
  - [6] I. K. Phan, "Perancangan Skema Evaluasi untuk Sistem Rekomendasi Berita Menggunakan Metrik Precision, Recall, dan F1 - Score TIN : Terapan Informatika Nusantara," *Terap. Inform. Nusan.*, vol. 6, no. 8, pp. 1351–1360, 2026, doi: 10.47065/tin.v6i8.9106.
  - [7] D. Pratiwi, A. Asrianda, and L. Rosnita, "Penerapan Metode Content-Based Filtering dalam Sistem Rekomendasi Objek Wisata di Aceh Tamiang," *J. Ilmu Komput. dan Inform.*, vol. 4, no. 2, pp. 85–96, 2025, doi: 10.54082/jiki.169.
  - [8] A. Y. Timur, "Sistem Rekomendasi Lagu Indonesia Menggunakan Metode Content-Based Filtering Dan Cosine Similarity," *J. Inform. dan Tek. Elektro Terap.*, vol. 13, no. 1, pp. 1415–1423, 2025, doi: 10.23960/jitet.v13i1.5949.
  - [9] M. Liang and T. Niu, "Research on Text Classification Techniques Based on Improved TF-IDF Algorithm and LSTM Inputs," *Procedia Comput. Sci.*, vol. 208, pp. 460–470, 2022, doi: 10.1016/j.procs.2022.10.064.
  - [10] K. N. Singh, S. D. Devi, H. M. Devi, and A. K. Mahanta, "A novel approach for dimension reduction using word embedding: An enhanced text classification approach," *Int. J. Inf. Manag. Data Insights*, vol. 2, no. 1, p. 100061, 2022, doi: 10.1016/j.jjime.2022.100061.
  - [11] D. Ridhwanullah, Y. K. Kumarahadi, and B. D. Raharja, "Content-Based Filtering pada Sistem Rekomendasi Buku Informatika," *J. Ilm. Sinus*, vol. 22, no. 2, pp. 57–66, 2024.
  - [12] D. Sandy, A. Tambunan, and D. A. Dermawan, "Implementasi Content Based Filtering dalam Sistem Rekomendasi Jurnal Scopus Berbasis Web Untuk Menunjang Pelaksanaan Penelitian dan Tugas Mahasiswa Program Studi Bisnis Digital," *J. Manaj. Inform.*, pp. 1–17, 2025.
  - [13] V. Zhelezniak, A. Savkov, A. Shen, and N. Y. Hammerla, "Correlation coefficients and semantic textual similarity," *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, pp. 951–962, 2019, doi: 10.18653/v1/n19-1100.
  - [14] A. A. Huda, R. Fajarudin, and A. Hadinegoro, "Sistem Rekomendasi Content-based Filtering Menggunakan TF-IDF Vector Similarity Untuk Rekomendasi Artikel Berita," *Build. Informatics, Technol. Sci.*, vol. 4, no. 3, pp. 1679–1686, 2022, doi: 10.47065/bits.v4i3.2511.
  - [15] M. Haris, A. Baihaqi, and A. Jananto, "PERBANDINGAN METODE COSINE DAN JACCARD SIMILARITY DALAM KERJA PEMERINTAH DAERAH," *J. Mhs. Tek. Inform.*, vol. 10, no. 2, pp. 2665–2670, 2026.
  - [16] H. Afu Ichsan Pradana, Robi Wariyanto Abdullah, "Deteksi Ketepatan Penggunaan Masker Wajah Dengan Algoritma CNN Dan," *J. Tek. Inform. dan Sist. Inf.*, vol. 9, no. 3, pp. 2305–2316 ISSN, 2022.
  - [17] T. P. Rinjeni, A. Indriawan, and N. A. Rakhmawati, "Matching Scientific Article Titles using Cosine Similarity and Jaccard Similarity Algorithm," *Procedia Comput. Sci.*, vol. 234, no. 2023, pp. 553–560, 2024, doi: 10.1016/j.procs.2024.03.039.
  - [18] Y. L. Sukestiyarno, H. A. Sapolo, and H. Sofyan, "Application of Recommendation Application of Recommendation System on E- Learning Platform Using Content-Based Filtering with Jaccard Similarity and Cosine Similarity Algorithms," *PrePrints*, pp. 1–9, 2023, doi: 10.20944/preprints202306.1672.v1.
  - [19] I. R. Bakti, H. Jati, and Y. P. Bunda, "A comparative analysis of five textual similarity methods for automatic short answer grading," *J. Soft Comput. Explor.*, pp. 43–54, 2026.
  - [20] W. G. S. Parwita, "Pengujian Akurasi Sistem Rekomendasi Berbasis Content-Based Filtering," *Inform. Mulawarman J. Ilm. Ilmu Komput.*, vol. 14, no. 1, p. 27, 2019, doi: 10.30872/jim.v14i1.1272.
  - [21] T. Wahyuningsih, Henderi, and Winarno, "Text Mining an Automatic Short Answer Grading (ASAG), Comparison of Three Methods of Cosine Similarity, Jaccard Similarity and Dice's Coefficient," *J. Appl. Data Sci.*, vol. 2, no. 2, pp. 45–54, 2021, doi: 10.47738/jads.v2i2.31.
  - [22] S. J. Pinastika *et al.*, "ANALISIS KOMPARATIF JACCARD DAN COSINE SIMILARITY," *J. Sist. Inf. dan Teknol. Komputasi*, vol. 2, no. 4, pp. 171–177, 2025.