

# *Extractive Text Summarization Karya Ilmiah Mahasiswa Menggunakan Fuzzy C-Means dan Vector Space Model*

## *Extractive Text Summarization of Student Scientific Using Fuzzy C-Means and Vector Space Model*

Vivin Octavia Cahyani<sup>1</sup>, Muhammad Faisal<sup>2</sup>, Irwan Budi Santoso<sup>3</sup>

<sup>1,2,3</sup>Fakultas Sains dan Teknologi, Program Studi Teknik Informatika, Universitas Islam Negeri Maulana Malik Ibrahim, Kota Malang, Indonesia

E-mail: <sup>1</sup>210605110038@student.uin-malang.ac.id, <sup>2</sup>mfaisal@ti.uin-malang.ac.id, <sup>3</sup>irwan@ti.uin-malang.ac.id

Received 23 April 2025; Revised 15 May 2025; Accepted 19 May 2025

**Abstrak-** Artikel jurnal ilmiah terus meningkat setiap tahunnya, sering kali mempersulit pembaca dalam menyaring informasi inti secara efisien. Informasi yang kurang efisien membuat pembaca harus membaca ulang artikel sehingga memerlukan banyak waktu. Oleh karena itu, dibutuhkan sebuah alat untuk menemukan inti informasi dari artikel jurnal ilmiah secara cepat dan efisien. Untuk mengatasi masalah ini, peringkasan teks secara otomatis diperlukan, khususnya untuk artikel jurnal ilmiah berbahasa Indonesia. Penelitian ini mengembangkan sistem peringkasan teks otomatis menggunakan metode *Fuzzy C-Means* dan *Vector Space Model* menggunakan pembobotan kalimat TF-IDF (*Term Frequency Invers Document Frequency*). Evaluasi sistem menggunakan metrik ROUGE-1 dan ROUGE-2. Hasil pengujian menunjukkan bahwa sistem terbaik, pada tingkat kompresi 30% serta menggunakan *stemming* memberikan hasil terbaik dan seimbang, dengan rata-rata ROUGE-1 *Precision* 0.5331, *Recall* 0.5034, *F1-Score* 0.4975 dan *Accuracy* 0.5183. Hasil penelitian ini menunjukkan bahwa model dengan *stemming* lebih disarankan untuk menghasilkan ringkasan yang lebih relevan dan akurat pada tingkat kompresi yang lebih tinggi.

**Kata kunci:** Fuzzy C-Means, Vector Space Model, Peringkasan Teks

**Abstract-** Scientific journal articles continue to increase every year, often making it difficult for readers to filter core information efficiently. Therefore, a tool is needed to find the core information from scientific journal articles quickly and efficiently. To solve this problem, automatic text summarization is needed, especially for Indonesian scientific journal articles. This research develops an automatic text summarization system using *Fuzzy C-Means* method and *Vector Space Model* using TF-IDF (*Term Frequency Invers Document Frequency*) sentence weighting. The system evaluation uses ROUGE-1 and ROUGE-2 metrics. The test results show that the best system, at 30% compression level and using *stemming* gives the best and balanced results, with an average ROUGE-1 *Precision* 0.5331, *Recall* 0.5034, *F1-Score* 0.4975 and *Accuracy* 0.5183. The results of this study show that the model with *stemming* is more recommended to produce more relevant and accurate summaries at higher compression levels.

**Keywords:** Fuzzy C-Means, Vector Space Model, Text Summarization

## 1. PENDAHULUAN

Artikel jurnal ilmiah adalah sebuah karya tulis ilmiah yang terstruktur menggunakan kaidah-kaidah keilmuan yang berisi tentang masalah nonfiksi atau faktual yang dimuat dalam jurnal, majalah, atau berita dengan tujuan untuk menyampaikan sebuah fakta serta menawarkan solusi dari suatu masalah [1]. Artikel jurnal ilmiah yang semakin meningkat setiap tahunnya,

dapat ditemukan secara online diterbitkan di berbagai situs web, dan dapat dibaca kapan saja serta dimana saja.

Publikasi ilmiah yang dipublikasikan dapat mempermudah mahasiswa untuk mendapatkan referensi yang diperlukan untuk tugas perkuliahan yang mengharuskan mencari rujukan dari sebuah artikel jurnal ilmiah. Banyaknya publikasi artikel jurnal ilmiah membuktikan bahwa ilmu pengetahuan semakin berkembang dengan pesat. Publikasi yang dilakukan oleh dosen, mahasiswa dan peneliti meningkat setiap tahunnya. Tercatat pada tahun 2022, Cina berhasil menduduki peringkat pertama penghasil Artikel Jurnal ilmiah terindeks Scopus paling banyak diantara negara yang lain, dengan total 1.004.745 artikel jurnal ilmiah[2].

Seiring dengan meningkatnya publikasi yang dilakukan setiap tahunnya, pembaca termasuk mahasiswa, peneliti dan akademisi sering menghadapi kesulitan dalam menyaring informasi yang relevan dari banyaknya artikel yang diterbitkan dengan berbagai topik yang disuguhkan. Mencari informasi inti dari setiap artikel membutuhkan waktu yang cukup lama untuk membaca keseluruhan dari teks artikel oleh karena itu, dibutuhkan sebuah alat untuk menemukan inti informasi dari artikel jurnal ilmiah secara cepat dan efisien[3][4]. Ketidakefektifan ini menghambat kemampuan pembaca untuk mengakses dan memanfaatkan literatur ilmiah secara efektif. Kebutuhan akan adanya alat yang dapat meringkas isi dari setiap artikel sangat penting terutama untuk seorang mahasiswa sedang mengerjakan tugas akhir, yang mencari referensi yang sesuai dengan penelitiannya, alat ini akan sangat berguna.

Peringkasan teks otomatis adalah cara peringkasan teks dengan menggunakan bantuan sebuah alat yang sudah di program untuk bisa meringkas sebuah artikel jurnal ilmiah sehingga dapat menghasilkan sebuah informasi inti dari teks yang panjang dengan cepat dan efisien[5]. Tujuan dibuat alat ini untuk memudahkan para pembaca dalam memahami isi dari artikel tanpa perlu membaca keseluruhan teks artikel jurnal ilmiah. Peringkasan teks otomatis memiliki dua jenis berdasarkan jumlah dokumen yang digunakan, yaitu peringkasan teks *single document* dan peringkasan teks *multi document*. Berdasarkan *output* yang dihasilkannya dibagi menjadi dua, yaitu peringkasan teks ekstraktif dan peringkasan teks abstraktif[4][6]. Ringkasan ekstraktif merupakan ringkasan dengan cara memilih kalimat yang dianggap penting dari sebuah paragraf untuk dijadikan satu ringkasan yang utuh dari potongan kalimat-kalimat tersebut, sedangkan ringkasan abstraktif dengan cara membuat kata-kata baru yang dapat direpresentasikan hasil ringkasan dalam kalimat[7].

Dalam beberapa tahun terakhir, teknologi *Natural Language Processing* mengalami perkembangan yang signifikan, dengan diperkenalkannya model *transformer* seperti BERT, GPT, dan T5. Model ini telah merevolusi berbagai tugas NLP, termasuk peringkasan teks, karena kemampuannya dalam memahami konteks secara mendalam dan menangkap makna semantik antar kalimat. *Pre-trained transformer-based models* memungkinkan peringkasan teks secara abstraktif maupun ekstraktif dengan kualitas ringkasan yang mendekati ringkasan manusia. Meskipun pendekatan berbasis *transformer* menawarkan hasil yang sangat baik, implementasinya membutuhkan sumber daya komputasi yang besar dan proses pelatihan kompleks. Oleh karena itu, dalam penelitian ini digunakan pendekatan alternatif yang lebih ringan dan efisien dengan menggabungkan metode *Vector Space Model* dan *Fuzzy C-Means*. Pendekatan ini tetap mampu menghasilkan ringkasan teks yang relevan, khususnya pada artikel jurnal ilmiah berbahasa Indonesia.

Sistem peringkasan dapat dibuat dengan memanfaatkan kecerdasan buatan (*Artificial Intelligent*) yang berkembang pesat saat ini, salah satu model yang dapat digunakan salah satunya adalah *Vector Space Model* dan *Fuzzy C-Means* dimana *Vector Space Model* yang digunakan dalam penelitian [8] sebagai perhitungan untuk pembobotan nilai pada setiap kata dalam artikel, penelitian ini menghasilkan sebuah paragraf ringkasan secara ekstraktif yang mengambil beberapa kalimat yang memiliki nilai kemiripan dengan judul paling tinggi dari 104 kalimat hanya diambil 5 kalimat dengan skor tertinggi, dari 5 kalimat tersebut akan dijadikan satu paragraf ringkasan. *Fuzzy C-Means* adalah sebuah algoritma yang dikembangkan pada tahun 1973 oleh Dunn dan pada tahun 1981 diperbaiki oleh Bezdek yang digunakan untuk pengenalan sebuah

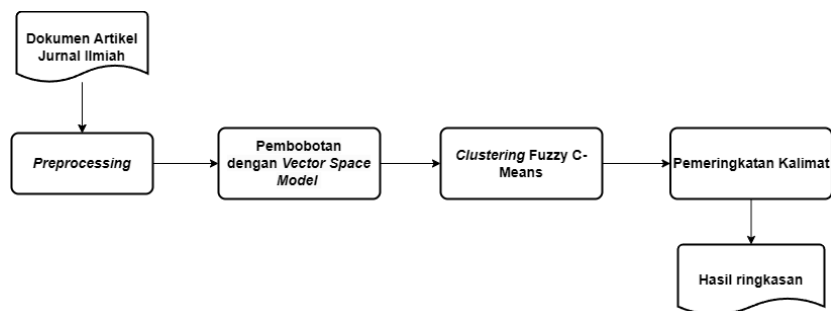
pola[9]. Algoritma ini menjadikan setiap objek dari beberapa *clusters* menjadi pilihan yang tepat, kemudian mengelompokkan sebuah data yang sangat luas, sederhana dan mudah diterapkan. Implementasi metode *Fuzzy C-Means* saat digunakan untuk meringkas dokumen artikel jurnal berbahasa inggris dengan *input* teks berformat .pdf serta menggunakan pembobotan TF-IDF berhasil mendapatkan nilai rata-rata *recall*, *precision*.

Penelitian ini dilakukan untuk mengetahui seberapa baik performa yang dihasilkan oleh sistem peringkasan teks, dengan membandingkan hasil ringkasan sistem dengan ringkasan manual oleh pakar. Pada penelitian ini menggunakan bagian abstrak dari artikel jurnal ilmiah sebagai perbandingan untuk mengevaluasi hasil ringkasan sistem dan hasil ringkasan manualnya, evaluasi menggunakan ROUGE-N. Diharapkan bahwa pembuatan sistem peringkasan ini dapat membantu pembaca dengan memberikan informasi yang relevan dari isi artikel jurnal ilmiah yang diringkas serta tidak menyita banyak waktu untuk membaca keseluruhan isi teks artikel jurnal ilmiah. Beberapa penelitian sebelumnya telah mengembangkan penelitian peringkasan teks yang menggunakan metode yang berbeda telah menghasilkan sistem peringkasan yang relevan dengan hasil ringkasan pakar atau ahli. Seperti penelitian yang dilakukan oleh [10] menunjukkan hasil evaluasi yang kurang optimal dengan menggunakan algoritma *fuzzy c-means* dengan pembobotan TF-ISF tidak cukup representatif untuk menentukan relevansi kalimat dalam dokumen esai, dengan celah penelitian tersebut, penelitian ini berusaha untuk mengisinya dengan metode *fuzzy c-means* dan *vector space model* yang dapat mengukur kemiripan semantik antar kalimat, sehingga mampu mengelompokkan dengan lebih akurat.

Berdasarkan latar belakang tersebut, rumusan masalah yang akan diambil adalah bagaimana mengetahui seberapa baik performa dari hasil sistem peringkasan teks artikel jurnal ilmiah karya mahasiswa menggunakan metode *Fuzzy C-Means* dan *Vector Space Model* dengan menghitung evaluasi menggunakan ROUGE-1 dan ROUGE-2 untuk mendapatkan skor *precision*, *recall*, *f-score* dan *accuracy* yang paling optimal. Sistem peringkasan teks yang dibuat dapat dimanfaatkan untuk pengguna yang membutuhkan informasi inti dari artikel jurnal ilmiah yang cepat dan ringkas, dengan waktu yang sangat efisien.

## 2. METODE PENELITIAN

Tahap awal untuk penelitian ini dengan melakukan persiapan data. *Dataset* yang digunakan pada penelitian ini menggunakan data primer pengumpulan data secara manual dengan mendownload melalui kumpulan artikel pada laman *Google Scholar*, dan *repository UIN Malang*. *Preprocessing* data merupakan tahap selanjutnya yang meliputi proses *split sentence*, *case folding*, *tokenization*, *stopword removal*, dan *stemming*. *Preprocessing* dilakukan untuk mendapatkkan data yang siap digunakan untuk proses pembobotan dan klasterisasi selanjutnya. Tahapan selanjutnya setelah *prerprocessing* pembobotan dengan *Vector Space Model* yang akan menghasilkan matrik vektor TF-IDF yang digunakan untuk *input* pada proses klasterisasi. Setelah itu proses klasterisasi menggunakan *Fuzzy C-Means* hingga menghasilkan pusat *cluster*. Kemudian proses pemeringkatan dari hasil proses klasterisasi yang akan menghasilkan sebuah kalimat ringkasan berdasarkan hasil pemeringkatan. Hasil ringkasan sistem selanjutnya akan dievaluasi menggunakan metrik ROUGE-1 dan ROUGE-2 dengan membandingkan hasil ringkasan sistem dengan ringkasan manusia atau pakar.



Gambar 1 Alur Penelitian

### 2.1 Pengumpulan Data

Penelitian ini menggunakan *dataset* kumpulan artikel jurnal ilmiah karya mahasiswa dengan pengumpulan secara manual melalui *Google Scholar*, *repository* UIN Malang dan *Science and Technology Index* (SINTA). Data yang digunakan sebanyak 100 dokumen artikel jurnal ilmiah karya mahasiswa, dengan atribut Judul Karya Ilmiah, Hasil Ringkasan, dan Isi. Atribut judul berisi judul dari artikel tersebut, hasil ringkasan berisi ringkasan manual yang diambil dari bagian abstrak dari artikel, dan untuk isi merupakan isi dari keseluruhan dari artikel ilmiah, pada bagian isi, mengekstrak isi dari file .pdf diubah menjadi .txt untuk memudahkan dalam pembuatan *dataset*. Jurnal yang di ambil merupakan jurnal berbahasa indonesia dengan berbagai tema. Penelitian ini akan meringkas teks artikel jurnal secara ekstraktif, dengan menggunakan bahasa pemrograman python.

	Judul Karya Ilmiah	Hasil Ringkasan	isi
0	Akademisi Karbitan dan Marwah Perguruan Tinggi	Pemerintah harus lebih mengevaluasi persyaratan...	Jakarta - Pemerintah agaknya perlu mencermati ...
1	Disertasi Bukan Sembarang Karya	Diundang Ketua Prodi Pascasarjana di Universit...	Beberapa hari lalu saya undang Ketua Program...
2	Membangun Tradisi Ilmiah Melalui Penelitian	Universitas pada hakikatnya sebagai institusi ...	Universitas hakikatnya adalah institusi akadem...
3	Analisis Kebutuhan Informasi Program Studi Tad...	Kebutuhan utama bagi manusia yaitu informasi. ...	Salah satu kebutuhan primer bagi manusia yaitu...
4	Membingkai Tasawuf Dengan Tafsir Ilmiah Al-Qur'an	Abdullah Darraz, menjelaskan Ayat-ayat Al-Qur'...	"Ayat-ayat al-Qur'an itu, bagaikan intan yang ...

Gambar 2 Dataset Artikel Jurnal Ilmiah

### 2.2 Preprocessing Data

*Preprocessing* data merupakan tahap awal setelah *dataset* siap digunakan untuk inputan pada proses selanjutnya, ini penting dilakukan untuk pembersihan data dan memperbaiki struktur data yang digunakan. Tahapan pertama dari proses ini adalah *split sentence* ini merupakan tahap dimana membagi teks panjang menjadi potongan kalimat yang akan dipisahkan berdasarkan tanda (.) titik untuk memudahkan model mengenali struktur dan konteks dari data. Tahapan kedua dari *preprocessing* yaitu *case folding* dengan mengubah semua huruf kapital menjadi huruf kecil tahap ini perlu supaya tidak terjadi perbedaan makna dalam kumpulan teks serta mempercepat pemrosesan kalimat untuk tahap selanjutnya. Tahap selanjutnya *tokenization* yang membagi kata dalam kalimat menjadi token-token kata. Setelah dibentuk token dari kata-kata perkalimat, akan di proses untuk *stopword removal* menghapus kata-kata yang tidak penting dan tidak memberikan makna yang signifikan. Tahapan terakhir dari *preprocessing* adalah *stemming*, yakni mengubah kata -kata menjadi bentuk kata dasarnya [11][12].

### 2.3 Pembobotan Vector Space Model

Proses peringkasan pertama setelah *preprocessing* yaitu akan dibentuk vektor kata dengan menggunakan *vektor space model* dengan pembobotan *Term Frequency Invers Document Frequency* (tf-idf), dengan membentuk vektor fitur untuk digunakan *input* pada proses *Clustering*. Proses pembentukan vektor fitur diawali dengan menghitung bobot tf-idf, pertama menghitung *term frequency* menggunakan persamaan 1:

$$tf_{i,j} = \frac{n_{a,b}}{\sum_k n_{a,b}} \quad (1)$$

Keterangan:

$tf_{i,j}$  = Frekuensi kata

$n_{a,b}$  = Jumlah kata yang muncul dalam dokumen

$\sum_k n_{a,b}$  = Total seluruh kata yang ada dalam dokumen

Setelah didapatkan hasil dari *term frequency* selanjutnya akan dihitung *invers document frequency* menggunakan persamaan 2:

$$idf_i = \log \left( \frac{N}{df_i} \right) \quad (2)$$

Keterangan:

N = Jumlah dokumen

idfi = Frekuensi kemunculan kata i pada seluruh dokumen

dfi = Dokumen yang mengandung i

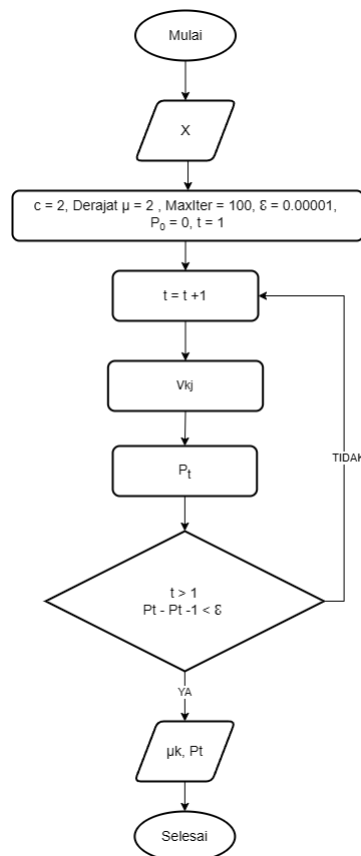
Setelah mendapatkan hasil TF-IDF akan dihitung untuk menentukan bobot dari tf-idf tersebut menggunakan persamaan 3:

$$W_{ij} = tf_{ij} \cdot idf_i \quad (3)$$

Dimana  $W$  adalah bobot dari dokumen yang dihasilkan dari hasil perkalian tf dan idf. Hasil bobot tersebut akan dibentuk vector fitur yang berbentuk matriks berdimensi. Matriks ini yang akan digunakan untuk proses input pada proses *clustering*.

#### 2.4 Clustering Fuzzy C-Means

*Clustering* menggunakan *fuzzy c-means* dengan mengelompokkan kalimat berdasarkan bobot dari tf-idf karena untuk inputan pada algoritma tersebut menggunakan data numerik [13]. Untuk mengatur proses pengumpulan data, pembagian otomatis dari kumpulan data dilakukan. Ini memungkinkan objek yang memiliki persamaan untuk dikelompokkan dalam kelompok berbeda dari kelompok lainnya [14]. Gambar 3 menunjukkan flowchart dari penerapan algoritma *fuzzy c-means* untuk sistem peringkasan otomatis.



Gambar 3 Flowchart Fuzzy C-Means

Pada pembentukan ringkasan teks menggunakan algoritma fuzzy c-means memiliki beberapa langkah seperti berikut:

- 1.) Data input merupakan data hasil bobot perhitungan TF-IDF yang telah terbentuk menjadi matriks berdimensi (jumlah dokumen x jumlah kata unik).
- 2.) Selanjutnya, menginisialisasi parameter apa saja yang akan digunakan:
  - a. Jumlah klaster  $c = 2$  (misalkan)
  - b. Derajat  $\mu$  dari  $(w) = 2$
  - c. Maksimal Iterasi  $MaxIter = 100$
  - d. Threshold error  $\xi = 0.00001$
  - e. Nilai fungsi objektif  $P_0 = 0$
  - f. Iterasi awal  $t = 1$

Inisialisasi diatas sesuai dengan penelitian yang pernah dilakukan oleh [10], untuk pengisian parameter memiliki syarat, parameter jumlah klaster ( $c$ ) nilai lebih dari 1 dan kurang dari jumlah data sampel ( $1 < c < n$ ), parameter derajat dari  $\mu$  dari  $(w)$  nilai lebih dari 1,  $MaxIter$  nilai lebih dari 1 dan error terkecil yang diharapkan sebesar  $\xi$ .

- 3.) Menghitung pusat *cluster*

Data keanggotaan awal diinisialisasi secara acak dalam bentuk matriks  $U$  dengan ukuran jumlah dokumen x jumlah *cluster*. Tahapan ini akan menghasilkan matriks keanggotaan

awal  $U$  yang sudah ternormalisasi. Setiap elemen dalam *cluster* memiliki nilai acak antara 0 dan 1 dan proses normalisasi untuk membantu memastikan bahwa setiap data memiliki total keanggotaan 1. Untuk menghitung pusat kluster dan normalisasi menggunakan persamaan 4 dan 5.

$$V_{kj} = \frac{\sum_{i=1}^n ((\mu_{ik})^w \cdot X_{ij})}{\sum_{i=1}^n ((\mu_{ik})^w)} \quad (4)$$

Keterangan:

$V_{kj}$  = Pusat *cluster* ke-  $k$   
 $n$  = Jumlah total dokumen

*centroid* akan dihitung ulang sesuai dengan jumlah iterasi maksimalnya untuk memastikan pembagian data ke dalam *cluster* semakin akurat. Kemudian dinormalisasi untuk memastikan jumlahnya = 1, menggunakan persamaan berikut.

$$\mu_{ik} = \frac{\mu_{ik}}{Q_i} \quad (5)$$

Keterangan:

$\mu_{ik}$  = Nilai keanggotaan setelah normalisasi  
 $Q_i$  = Total derajat keanggotaan sebelum normalisasi

#### 4.) Perbarui *Mambership* Matrix $U$

Sebelum keanggotaan matriks  $U$  diperbaharui akan dihitung fungsi objektif iterasi ke- $t$  untuk menilai perubahan pada fungsi objektif pada iterasi sebelumnya, menggunakan persamaan berikut.

$$P_t = \sum_{i=1}^n \sum_{k=1}^c \left( \left[ \sum_{j=1}^m (X_{ij} - V_{kj})^2 \right] \mu_{ik}^w \right) \quad (6)$$

Keterangan:

$P_t$  = Nilai fungsi objektif pada iterasi ke - $t$   
 $n$  = Jumlah data  
 $c$  = Jumlah kluster yang dibentuk  
 $m$  = Dimensi atau jumlah fitur dari setiap data  
 $X_{ij}$  = Nilai fitur ke- $j$  dari data ke- $i$   
 $(X_{ij} - V_{kj})^2$  = Jarak antara data  $X_{ij}$  dan pusat kluster  $V_{kj}$  pada dimensi  $j$   
 $\sum_{j=1}^m (X_{ij} - V_{kj})^2$  = Jarak total antara data ke- $i$  dan pusat kluster ke- $k$   
 $\mu_{ik}^w$  = Derajat keanggotaan yang telah dipangkatkan dengan  $w$

Kemudian akan dihitung derajat keanggotaan yang diperbarui menggunakan persamaan berikut.

$$\mu_{ik} = \frac{\left[ \sum_{j=1}^m (X_{ij} - V_{kj})^2 \right]^{\frac{-1}{w-1}}}{\sum_{k=1}^c \left[ \sum_{j=1}^m (X_{ij} - V_{kj})^2 \right]^{\frac{-1}{w-1}}} \quad (7)$$

Keterangan:

$\mu_{ik}$  = Derajat keanggotaan data ke- $i$  dalam kluster ke- $k$  (antara 0 dan 1)  
 $X_{ij}$  = Nilai fitur ke- $j$  dari data ke- $i$   
 $c$  = Jumlah kluster yang dibentuk  
 $m$  = dimensi atau jumlah fitur dari setiap data

## 5.) Periksa Konvergensi

Setiap perubahan nilai  $|P_t - P_{t-1}| < \varepsilon$  pada fungsi objektif harus di cek, jika terdapat perbedaan fungsi objektif kurang dari nilai ambang batas error maka iterasi akan berhenti. Jika tidak, maka iterasi dilanjutkan hingga mencapai maksimal iterasi.

6.) Hasil *cluster*

*Fuzzy C-Means* akan menghasilkan pusat cluster dan matriks keanggotaan dari setiap kalimat terhadap setiap *cluster*.

## 2.5 Pemeringkatan Kalimat

Proses pemeringkatan dengan menghitung nilai kalimat 30 % bobot dengan tf-idf dan 40% kedekatan dengan keanggotaan dalam pusat klaster [15]. Nilai ini digunakan untuk menentukan seberapa penting atau relevan suatu kalimat untuk hasil peringkasan teks. Kalimat dengan skor tertinggi dari setiap *cluster* dipilih sebagai hasil ringkasan.

## 2.6 Evaluasi

Proses evaluasi pada penelitian ini menggunakan metrik ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*). ROUGE merupakan suatu metrik yang sering digunakan untuk proses evaluasi dari sebuah sistem peringkasan teks [16]. Cara kerja dari ROUGE ini dengan menghitung kata-kata yang tumpang tindih antara ringkasan sistem dengan ringkasan manual atau referensi beserta bobot dari masing-masing [17]. Kesesuaian seperti jumlah n-kata (*n-gram*), pasangan kata, dan urutan kata antara ringkasan dari sistem dan ringkasan manual dari pakar [18]. Jumlah *n-gram* memiliki simbol N yang berupa angka 1 atau lebih. Pada penelitian ini menggunakan ROUGE-1 dan ROUGE-2 untuk mengetahui kualitas dan performa dari hasil ringkasan sistem yang dibuat. Selain ROUGE pendekatan evaluasi manual atau human judgement juga sering digunakan [19]. Nilai *precision*, *recall*, dan *f-score* dapat dihitung menggunakan persamaan berikut.

$$Recall = \frac{\sum_{s \in sys} \sum_{gramN \in s} Count_{match}(gram_N)}{\sum_{s \in ref} \sum_{gramN \in s} Count(gram_N)} \quad (8)$$

Nilai *recall* dapat dihitung menggunakan persamaan 8, *s* adalah kalimat atau *sentence* yang ada pada ringkasan, *ref* adalah ringkasan referensi, *Count(gram N)* adalah jumlah *N-gram* yang ada pada ringkasan referensi dan *Count<sub>match</sub>(gramN)* adalah jumlah *N-gram* tertinggi yang muncul dalam ringkasann yang dibuat oleh sistem dan ringkasan referensi.

$$Precision = \frac{\sum_{s \in sys} \sum_{gramN \in s} Count_{match}(gram_N)}{\sum_{s \in sys} \sum_{gramN \in s} Count(gram_N)} \quad (9)$$

Nilai *precision* dihitung menggunakan persamaan 9 dimana *sys* adalah ringkasan sistem, *Count(gramN)* adalah jumlah *N-gram* yang terdapat pada ringkasan sistem. *Count<sub>match</sub>(gramN)* merupakan jumlah maksimum *N-gram* yang muncul pada ringkasan sistem dan ringkasan referensi.

$$Precision = \frac{\sum_{s \in sys} \sum_{gramN \in s} Count_{match}(gram_N)}{\sum_{s \in sys} \sum_{gramN \in s} Count(gram_N)} \quad (10)$$

$\beta$  merupakan parameter untuk menentukan bobot relatif dari *precision* dan *recall*, biasanya diatur ke 1 untuk menyeimbangkan *precision* dan *recall*, sehingga *F-Score* menjadi *F1-Score*. Hasil kombinasi dari nilai *recall* dan *precision* digunakan untuk mengukur kinerja dari suatu sistem.



### 3. HASIL DAN PEMBAHASAN

Uji coba dilakukan pada 100 dokumen artikel jurnal ilmiah yang telah diunduh secara manual dari *repository* UIN Malang, *Google Scholar*, dan *Science Technology Index* (SINTA). Dataset yang telah dibuat dari 100 dokumen artikel jurnal ilmiah memiliki atribut judul karya ilmiah, hasil ringkasan, dan isi. Skenario uji coba dilakukan dari id dokumen-1 sampai dengan dokumen-100 untuk mengukur nilai ROUGE-N yakni dengan ROUGE-1 dan ROUGE-2 pada hasil ringkasan sistem. Sebelum dilakukan proses peringkasan pada setiap dokumen dilakukan *preprocessing*, dan akan dilakukan pengujian *preprocessing* menggunakan *stemming* dan tanpa *stemming*.

Setelah dilakukan prapremprosesan akan dilakukan insialisasi parameter yang akan digunakan pada metode *Fuzzy C-Means* dengan jumlah *cluster* random sesuai hasil yang ditentukan terbaik antara 2-6 *cluster*, parameter *fuzziness*  $f$  2, *max\_iterasi* 100, dan nilai epsilon 0,00001. Proses menghitung kesamaan kosinus antar kalimat dengan vektor dokumen dan menghitung skor akhir untuk setiap kalimat dengan kedekatan *cluster* menggunakan metode *Vector Space Model*. Panjang ringkasan yang dihasilkan sistem sebesar 10%, 20%, 30%, 40% dan 50% dari keseluruhan kalimat pada teks asli. Tabel 1 menunjukkan hasil dari tahapan *preprocessing*.

Tabel 1 Hasil *Preprocessing*

Index Dokumen	Isi Dokumen	Hasil Preprocessing
0	Syntax Jurnal Informatika merupakan sebuah sis.....	syntax jurnal informatika rupa buah sistem inf..
1	Hati atau liver adalah organ tubuh yang terletak....	hati liver organ tubuh letak bagi atas rongga ...
2	Secara geografis, Indonesia adalah salah satu.....	geografis indonesia salah satu negara pulau be....
3	Salah ssatu permasalahan serius yang dihadapi a.....	salah satu masalah serius hadap akibat tanam t...
4	Indonesia merupakan salah satu negara dengan j.....	indonesia rupa salah satu negara jumlah duduk .....
.....	.....	.....
95	Penumpukkan sampah yang tidak sesuai dengan kap.....	tumpu sampah sesuai kapasitas masalah nyata.....
96	Secara umum, Diabetes dikenal sebagai penyakit.....	diabetes kenal sakit epidemi dampak negara kelompok.....
97	Pencemaran udara menjadi permasalahan yang ser.....	cemar udara masalah seringkali jumpa....
98	Kartu Kredit adalah sebuah alat pembayaran yan.....	kartu kredit alat bayar keluar bank.....
99	Serangan Jantung adalah salah satu penyakit ya.....	serang jantung salah satu mati dunia.....

Pada hasil *preprocessing* diatas akan digunakan untuk input kedalam model yaitu pada proses *clustering* dan perhitungan *Term Frequency Invers Document Frequency*.

Tabel 2 Derajat keanggotaan dokumen terhadap *cluster* dengan *Stemming*

Data Point	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
0	0.166571	0.166571	0.166571	0.166571	0.167147	0.166571
1	0.000545	0.000545	0.000545	0.000545	0.997276	0.000545
2	0.166618	0.166618	0.166618	0.166618	0.166912	0.166618
3	0.000545	0.000545	0.000545	0.000545	0.997276	0.000545
4	0.000545	0.000545	0.000545	0.000545	0.997276	0.000545
.....	.....	.....	.....	.....	.....	.....
48633	0.000545	0.000545	0.000545	0.000545	0.997276	0.000545
48634	0.166636	0.166636	0.166636	0.166636	0.166818	0.166636
48636	0.166646	0.166646	0.166646	0.166646	0.166771	0.166646
48636	0.166571	0.166571	0.166571	0.166571	0.167147	0.166571
48637	0.000545	0.000545	0.000545	0.000545	0.997276	0.000545

Pada Tabel 2 menunjukkan nilai derajat keanggotaan semua data terhadap setiap cluster menggunakan *preprocessing stemming*, dibentuk sebanyak 6 cluster. Data point berjumlah 48637 dihasilkan dari *tfidf\_matrix* yang dibentuk sebelumnya, setiap kata unik dalam seluruh korpus

dokumen dianggap sebagai satu fitur, dalam satu dokumen terdiri dari teks panjang dan beragam, dengan begitu kata unik yang dihasilkan ribuan. Pada tabel tersebut terdapat 48637 kata unik di antara dokumen, tfidf\_matrix memiliki dimensi (100, 48637) kemudian di transpose maka matriksnya menjadi berdimensi (48637, 100).

Tabel 3 Derajat keanggotaan dokumen terhadap *cluster* tanpa *Stemming*

Data Point	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
0	0.166604	0.166604	0.166604	0.166604	0.166978	0.166604
1	0.000559	0.000559	0.000559	0.000559	0.997204	0.000559
2	0.166637	0.166637	0.166637	0.166637	0.166813	0.166637
3	0.000559	0.000559	0.000559	0.000559	0.997204	0.000559
4	0.000559	0.000559	0.000559	0.000559	0.997204	0.000559
.....	.....	.....	.....	.....	.....	.....
48633	0.000559	0.000559	0.000559	0.000559	0.997204	0.000559
48634	0.166604	0.166604	0.166604	0.166604	0.166978	0.166604
48636	0.166680	0.166680	0.166680	0.166680	0.166602	0.166680
48636	0.166604	0.166604	0.166604	0.166604	0.166978	0.166604
48637	0.000559	0.000559	0.000559	0.000559	0.997204	0.000559

Pada Tabel 3 diatas menunjukkan nilai derajat keanggotaan semua data terhadap setiap cluster tanpa menggunakan preprocessing stemming, dibentuk sebanyak 6 cluster. Data point berjumlah 48637 dihasilkan dari tfidf\_matrix yang dibentuk sebelumnya, setiap kata unik dalam seluruh korpus dokumen dianggap sebagai satu fitur, dalam satu dokumen terdiri dari teks panjang dan beragam, dengan begitu kata unik yang dihasilkan ribuan. Pada tabel tersebut terdapat 48637 kata unik di antara dokumen, tfidf\_matrix memiliki dimensi (100, 48637) kemudian di transpose, maka matriksnya menjadi berdimensi (48637, 100). Selanjutnya untuk hasil dari perhitungan kedekatan cluster sebesar 30% dan Tf-Idf sebesar 70% yang menggunakan stemming dapat dilihat pada Tabel 4.

Tabel 4 Hasil Perhitungan kedekatan *cluster* dan *TF-IDF* dengan *Stemming*

	Kalimat ke -	skor
Dokumen 1	2	4.1199
	4	4.0949
	5	4.0899
	7	4.0842
	12	4.0783
Dokumen 2	2	3.8720
	4	3.8470
	5	3.8420
	7	3.8363
	12	3.8303
Dokumen 3	2	3.4790
	4	3.4540
	5	3.4490
	7	3.4433
	12	3.4374
Dokumen 4	2	3.4054
	4	3.3804
	5	3.3754
	7	3.3697
	12	3.3637

Tabel 5 Hasil Perhitungan kedekatan *cluster* dan *TF-IDF* tanpa *Stemming*

	Kalimat ke -	skor
Dokumen 1	2	4.4522
	4	4.4272
	5	4.4222
	7	4.4164
	12	4.4105
	2	4.4132
	4	4.3882

Dokumen 2	5	4.3832
	7	4.3775
	12	4.3715
Dokumen 3	2	3.7960
	4	3.7710
	5	3.7660
	7	3.7603
	12	3.7544
Dokumen 4	2	3.5702
	4	3.5452
	5	3.5402
	7	3.5345
	12	3.5286

Dapat dilihat dari kedua tabel diatas urutan kalimat yang dihasilkan sama, karena indeks kalimat diurutkan setelah memilih kalimat dengan skor tertinggi kemudian untuk hasil skor berbeda, karena pengaruh dari proses *stemming* dan tanpa *stemming*, yang menghasilkan dimensi matriks tf-idf yang berbeda. Kemudian dari hasil tersebut nantinya akan diurutkan berdasarkan skor tertinggi dan dimasukkan ke dalam ringkasan pada tingkat kompresi 10%, 20%, 30%, 40%, dan 50%. Hasil keseluruhan dari peringkasan teks dengan tingkat kompresi 10%, 20%, 30%, 40% dan 50% dapat dilihat pada Tabel 6 dengan Stemming dan Tabel 7 tanpa Stemming dibawah ini.

Tabel 6 Rata-rata hasil evaluasi ROUGE-1 dan ROUGE-2 dengan Stemming

Compression Rate	ROUGE-N							
	ROUGE-1				ROUGE-2			
	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F2-Score	Accuracy
10%	0.2321	0.6278	0.3271	0.4300	0.0539	0.1414	0.0751	0.0976
20%	0.4053	0.5643	0.4530	0.4848	0.0984	0.1325	0.1080	0.1154
30%	0.5331	0.5034	0.4975	0.5183	0.1369	0.1243	0.1249	0.1306
40%	0.6325	0.4568	0.5111	0.5447	0.1750	0.1219	0.1382	0.1484
50%	0.7073	0.4121	0.5034	0.5597	0.2198	0.1228	0.1520	0.1713

Tabel 7 Rata-rata hasil evaluasi ROUGE-1 dan ROUGE-2 tanpa Stemming

Compression Rate	ROUGE-N							
	ROUGE-1				ROUGE-2			
	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F2-Score	Accuracy
10%	0.2321	0.6278	0.3271	0.4300	0.0539	0.1414	0.0751	0.0976
20%	0.4049	0.5640	0.4526	0.4844	0.0984	0.1325	0.1080	0.1153
30%	0.5330	0.5033	0.4974	0.5181	0.1368	0.1242	0.1248	0.1305
40%	0.6323	0.4566	0.5109	0.5445	0.1746	0.1216	0.1379	0.1481
50%	0.7069	0.4118	0.5031	0.5594	0.2192	0.1225	0.1516	0.1709

Kedua tabel menunjukkan skor yang seimbang ketika berada pada tingkat kompresi 30%, terlihat bahwa antara nilai precision dan recall seimbang sebesar 0.5331 dan 0.5034, rata-rata nilai precision yang dihasilkan menandakan bahwa lebih dari setengah kata yang muncul dalam ringkasan yang dihasilkan oleh sistem sesuai dengan kata dalam ringkasan referensi. Nilai recall yang didapat dari penelitian ini sedikit lebih rendah, berarti bahwa lebih dari 50% sistem dapat menangkap kata-kata yang relevan dari teks asli dalam ringkasan yang dihasilkan. Hasil penelitian menunjukkan bahwa penggunaan *stemming* berdampak positif terhadap kinerja model dalam proses peringkasan. Dengan mengubah kata-kata ke bentuk dasarnya, *stemming* mengurangi dimensi vektor TF-IDF, sehingga meningkatkan kemiripan semantik antar kalimat yang dianalisis oleh *fuzzy c-means*. Hal ini memungkinkan sistem untuk mengelompokkan dan memilih kalimat penting dengan lebih akurat.

Dibandingkan dengan metode lain seperti *TexRank* yang berbasis *graf*, pendekatan ini menunjukkan kelebihan dalam menangkap kemiripan semantik dengan mempertimbangkan bobot kata dan distribusinya secara numerik. Sementara itu, metode berbasis *transformer* seperti

BART atau T5 cenderung menghasilkan ringkasan yang koheren dan kontekstual, namun memerlukan sumber daya komputasi yang lebih besar seperti hasil penelitian yang pernah dilakukan [20] yang menghasilkan rata-rata ROUGE-1 dan ROUGE-2 terbaik sebesar 0.1197 dan 0.0119 dengan menggunakan BIGRU. Model yang digunakan dalam penelitian ini lebih ringan dan efisien, dengan hasil nilai *f-score* menunjukkan keseimbangan antara nilai precision dan recall, yang berarti bahwa sistem cukup baik dalam mempertahankan informasi yang relevan. Akurasi yang dihasilkan sebesar 0.5183 menandakan bahwa model cukup andal dalam menghasilkan ringkasan yang relevan.

#### 4. KESIMPULAN DAN SARAN

Nilai rata-rata hasil evaluasi ROUGE-1 dan ROUGE-2, yang ditunjukkan pada Tabel 6 dan Tabel 7 menggunakan *stemming* dan tanpa *stemming* menunjukkan bahwa sistem mendapatkan hasil yang relevan dalam menangkap kata-kata serta mengkluster dengan baik, mana kalimat yang cocok dijadikan untuk ringkasan atau kurang cocok berdasarkan hasil perhitungan skor yang dilakukan. Hal ini dibuktikan dengan hasil relevansi rata-rata nilai ROUGE-1 pada tingkat kompresi 30% dengan nilai rata-rata *precision* 0.5331, *recall* 0.5034, *f1-score* 0.4975 dan *accuracy* 0.5183. Meskipun hasil penelitian mampu memberikan hasil ringkasan yang cukup relevan, terdapat beberapa keterbatasan, model kurang optimal dalam menangani dokumen dengan struktur *non-konvensional*, seperti artikel dengan banyak tabel, dan gambar. Berikut beberapa saran untuk peneliti selanjutnya supaya bisa menghasilkan keseimbangan nilai precision dan recall dengan mengembangkan sistem lebih lanjut dengan mengintegrasikan model *Large Language Models* (LLMs) seperti GPT-4 atau T5 yang mampu menghasilkan ringkasan dengan kualitas linguistik yang lebih baik serta memahami konteks secara lebih menyeluruh. Selain itu, peningkatan *preprocessing* pada format dokumen juga dapat membantu memperbaiki hasil ringkasan pada dokumen dengan struktur variatif.

#### DAFTAR PUSTAKA

- [1] Salma, "Artikel Ilmiah: Pengertian, Fungsi, Ciri-ciri dan Sistematika," *deepublish*, 2022. [https://penerbitdeepublish.com/pengertian-artikel-ilmiah/#1\\_Komara\\_2017](https://penerbitdeepublish.com/pengertian-artikel-ilmiah/#1_Komara_2017) (accessed Jul. 11, 2024).
- [2] Wahyono, "Jumlah Publikasi Jurnal Ilmiah Terindeks Scopus asal Indonesia Tahun ke Tahun, Naik Atau Turun?," *SindoNews.com*, 2023. <https://edukasi.sindonews.com/read/1195439/211/jumlah-publikasi-jurnal-ilmiah-terindeks-scopus-asal-indonesia-tahun-ke-tahun-naik-atau-turun-1694063382> (accessed Jul. 18, 2024).
- [3] A. Nenkova and K. McKeown, "Automatic summarization," *Found. Trends Inf. Retr.*, vol. 5, no. 2–3, pp. 103–233, 2011, doi: 10.1561/15000000015.
- [4] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: a survey," *Artif. Intell. Rev.*, vol. 47, no. 1, pp. 1–66, 2017, doi: 10.1007/s10462-016-9475-9.
- [5] M. S. Utomo, J. S. Wibowo, and E. N. Wahyudi, "Text Summarization Pada Artikel Berita Menggunakan Vector Space Model Dan Cosine Similarity," *J. Din. Inform.*, vol. 14, no. 1, pp. 11–24, 2022, doi: 10.35315/informatika.v14i1.9163.
- [6] Halimah, Surya Agustian, and Siti Ramadhani, "Peringkasan teks otomatis (automated text summarization) pada artikel berbahasa indonesia menggunakan algoritma lexrank," *J. CoSciTech (Computer Sci. Inf. Technol.)*, vol. 3, no. 3, pp. 371–381, 2022, doi: 10.37859/coscitech.v3i3.4300.
- [7] M. A. Zamzam, "Sistem Automatic Text Summarization Menggunakan Algoritma Textrank," *Matics*, vol. 12, no. 2, pp. 111–116, 2020, doi: 10.18860/mat.v12i2.8372.
- [8] M. Irfan, Jumadi, W. B. Zulfikar, and Erik, "Implementation of Fuzzy C-Means algorithm

- and TF-IDF on English journal summary,” *Proc. 2nd Int. Conf. Informatics Comput. ICIC 2017*, vol. 2018-Janua, pp. 1–5, 2017, doi: 10.1109/IAC.2017.8280646.
- [9] J. C. Bezdek, R. Ehrlich, and W. Full, “FCM: The fuzzy c-means clustering algorithm,” *Comput. Geosci.*, vol. 10, no. 2, pp. 191–203, 1984, doi: [https://doi.org/10.1016/0098-3004\(84\)90020-7](https://doi.org/10.1016/0098-3004(84)90020-7).
- [10] I. M. Suwija Putra, Y. Adiwinata, D. P. Singgih Putri, and N. P. Sutramiani, “Extractive Text Summarization of Student Essay Assignment Using Sentence Weight Features and Fuzzy C-Means,” *Int. J. Artif. Intell. Res.*, vol. 5, no. 1, pp. 13–24, 2021, doi: 10.29099/ijair.v5i1.187.
- [11] A. tri Jaka, “Belajar Data Science: Text Mining Untuk Pemula,” *J. Inform. UPGRIS*, vol. 1, pp. 1–9, 2015, [Online]. Available: <https://media.neliti.com/media/publications/137435-ID-preprocessing-text-untuk-meminimalisir-k.pdf>
- [12] M. Adriani, J. Asian, B. Nazief, S. M. M. Tahaghoghi, and H. E. Williams, “Stemming Indonesian: A confix-stripping approach,” *ACM Trans. Asian Lang. Inf. Process.*, vol. 6, no. 4, pp. 1–33, Dec. 2008, doi: 10.1145/1316457.1316459.
- [13] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek, “A possibilistic fuzzy c-means clustering algorithm,” *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 4, pp. 517–530, 2005, doi: 10.1109/TFUZZ.2004.840099.
- [14] V. N. Phu, N. D. Dat, V. T. Ngoc Tran, V. T. Ngoc Chau, and T. A. Nguyen, “Fuzzy C-means for english sentiment classification in a distributed system,” *Appl. Intell.*, vol. 46, no. 3, pp. 717–738, 2017, doi: 10.1007/s10489-016-0858-z.
- [15] S. Pawar, H. Manjula Gururaj, and N. N. Chiplunar, “Text Summarization Using Document and Sentence Clustering,” *Procedia Comput. Sci.*, vol. 215, pp. 361–369, 2022, doi: 10.1016/j.procs.2022.12.038.
- [16] C.-Y. Lin, “ROUGE: A Package for Automatic Evaluation of Summaries Chin-Yew,” *Jpn. Circ. J.*, vol. 34, no. 12, pp. 1213–1220, 2004, doi: 10.1253/jcj.34.1213.
- [17] A. Y. Setiawan, I. G. M. Darmawiguna, and G. A. Pradnyana, “Sentiment Summarization Evaluasi Pembelajaran Menggunakan Algoritma LSTM (long short term memory),” *Kumpul. Artik. Mhs. Pendidik. Tek. Inform.*, vol. 11, no. 2, pp. 183–191, 2022.
- [18] P. Verma, S. Pal, and H. Om, “A comparative analysis on Hindi and English extractive text summarization,” *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 18, no. 3, 2019, doi: 10.1145/3308754.
- [19] E. Lloret and M. Palomar, “Text summarisation in progress: A literature review,” *Artif. Intell. Rev.*, vol. 37, no. 1, pp. 1–41, 2012, doi: 10.1007/s10462-011-9216-z.
- [20] R. Adelia, S. Suyanto, and U. N. Wisesty, “Indonesian abstractive text summarization using bidirectional gated recurrent unit,” *Procedia Comput. Sci.*, vol. 157, pp. 581–588, 2019, doi: 10.1016/j.procs.2019.09.017.