

# Peringkasan Teks Multi-Dokumen Berbahasa Indonesia Dengan Sentence Scoring dan SVM

*Multi-Dokumen Text Summarization in Indonesian Using Sentence Scoring and SVM*

Deri Fauzi<sup>1</sup>, Zainal Abidin<sup>2</sup>, Fatchurrochman<sup>3</sup>

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

E-mail: <sup>1</sup>19650074@student-uin-malang.ac.id, <sup>2</sup>zainal@ti.uin-malang.ac.id,

<sup>3</sup>fatchurrochman@ti.uin-malang.ac.id

## Abstrak

Berita online berasal dari berbagai sumber portal berita yang tersedia secara luas di dunia maya. Namun, berita online yang melimpah dapat mengesampingkan detail dan keakuratan berita karena tujuannya untuk memberikan informasi terkini sebanyak mungkin. Ketersediaan berita online di internet dapat menyebabkan penerimaan informasi yang berlebihan. Oleh karena itu, penting untuk menemukan representasi dokumen berita online guna memahami inti dari berita tersebut. Penelitian ini fokus pada menghasilkan ringkasan berita online multi dokumen dari ekstraksi fitur dan proses klasifikasi menggunakan support vector machine. penelitian ini mengklasifikan berita multi dokumen menggunakan ekstraksi fitur Sentence Scoring dan SVM. Hasil pengujian menunjukkan bahwa fold 3 memberikan hasil terbaik, dengan rata-rata Recall 0.946, Presisi 0.487, dan F-measure 0.634. ROUGE-1 juga mencapai nilai tertinggi pada fold 3, yaitu 0.946. Faktor kunci dalam hasil peringkasan adalah proses ekstraksi fitur menggunakan sentence scoring dan pelatihan data dengan SVM. Fitur seperti data numerik dan kemiripan antar kalimat berpengaruh signifikan terhadap hasil akhir dari peringkasan.

**Kata kunci:** Peringkasan Teks, Multi Dokumen, SVM, Berita Online

## Abstract

*Online news comes from a variety of news portal sources that are widely available in the cyber world. However, the abundance of online news can overshadow the detail and accuracy of the news because its purpose is to provide the latest information as much as possible. The availability of online news on the internet can lead to excessive information intake. Therefore, it is important to find a representation of online news documents to understand the essence of the news. This study focuses on generating multi-document online news summaries from feature extraction and classification processes using support vector machines. This study classifies multi-document news to generate summaries using Sentence Scoring feature extraction and SVM. The test results show that fold 3 gives the best results, with an average Recall of 0.946, Precision of 0.487, and F-measure of 0.634. ROUGE-1 also reached the highest value on fold 3, which is 0.946. The key factor in the summary results is the feature extraction process using sentence scoring and data training with SVM. Features such as numerical data and similarity between sentences have a significant effect on the final results of the summary.*

**Keywords:** Text Summarization, Multi-Document, SVM, Online News

## 1. PENDAHULUAN

Perkembangan dunia internet kini memungkinkan akses mudah informasi bagi individu di berbagai waktu dan lokasi. Salah satu akibat dari perkembangan internet ini adalah banyaknya berita online yang tersedia di internet. Akan tetapi, banyaknya berita online yang tersedia di internet dapat membuat informasi terasa berlebihan dan sulit dipahami. Terkadang, pembaca harus meluangkan waktu untuk membaca berita dari berbagai situs guna membandingkan topik

yang sama, yang pada akhirnya dapat menyebabkan tumpang tindih dalam upaya memahami informasi [1]. Oleh karena itu, penting untuk menemukan representasi dokumen berita online guna memahami inti dari berita tersebut.

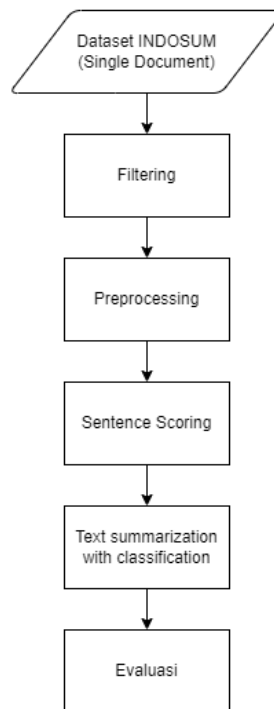
Penelitian tentang peringkasan teks otomatis telah dilakukan dengan beberapa pendekatan. salah satunya dengan pendekatan machine learning[2]. Kemudian terdapat penelitian yang membandingkan metode pendekatan secara semantik untuk menilai pendekatan yang paling efektif untuk menghasilkan ringkasan yang akurat [3]. Penelitian lain menggunakan metode SVM untuk membuat sistem peringkasan dan menghasilkan hasil yang cukup relevan dan akurat [4]. Support Vector Machine (SVM) dapat digunakan untuk melakukan peringkasan yaitu dengan cara mengklasifikasikan kalimat yang penting dan tidak penting. Kalimat yang penting adalah kalimat yang nantinya masuk kedalam ringkasan, sedangkan kalimat yang tidak penting adalah kalimat yang tidak masuk kedalam ringkasan. Untuk mengklasifikasikan hal tersebut, dibutuhkan ekstraksi fitur pada setiap kalimat untuk menentukan kalimat tersebut penting atau tidak. Proses ekstraksi fitur ini sangat penting untuk memberikan bobot pada kalimat dan berpengaruh terhadap hasil akhir dari sebuah ringkasan [5]. Peringkasan teks dengan klasifikasi pernah dilakukan oleh Mandal et al., Namun metode tersebut masih digunakan untuk meringkas dokumen Tunggal [6].

Oleh karena itu makalah ini menjelaskan tentang peringkasan multi dokumen berbahasa Indonesia dengan sentence scoring dan SVM. Dalam pendekatan peringkasan teks ekstraktif, setiap kalimat dalam teks diberi bobot yang unik melalui proses *Sentence Scoring*. Sentence scoring pada penelitian ini meliputi: panjang kalimat, posisi kalimat, jumlah data numerik, TF-ISF, dan kemiripan antar kalimat [7]. Sentence scoring digunakan untuk membuat model SVM. Peringkasan multidokumen dievaluasi menggunakan metric ROUGE. Kami menguji coba menggunakan 50 story. story disusun dari dataset untuk summarization dokumen tunggal. Kami memberikan label baru pada kalimat yang menjadi ringkasan dari story. hasil evaluasi menunjukkan metode ini dapat meringkas multidokumen dengan rata-rata F-measure 0,7.

## 2. METODE PENELITIAN

Tahapan proses sistem yang dibuat pada penelitian ini adalah memasukkan artikel online multidokumen yang dinamakan story. Story ini didapatkan dari dataset INDOSUM yang didalamnya terdapat 14.000 artikel berita online dari berbagai sumber [8]. Kumpulan berita dari dataset ini dibagi menjadi beberapa kategori seperti tajuk utama, teknologi, olahraga, hiburan, dan showbiz. Kemudian waktu terbit berita yang ada di dataset ini berkisar antara tahun 2016-2018. Pada penelitian ini tidak menggunakan keseluruhan atribut dataset tersebut, akan tetapi hanya mengambil berita dan kategori saja. Story ini dibuat berdasarkan query atau kata kunci untuk memilah berita yang sesuai dengan kebutuhan dan masih satu topik pembahasan. Story yang dihasilkan berjumlah 50 yang diambil dari 14.000 berita online pada dataset INDOSUM.

Alur proses untuk sistem peringkasan multidokumen ini dapat dilihat pada Gambar 1. Langkah pertama dilakukan proses filtering untuk mendapatkan story multi dokumen. Selanjutnya proses *preprocessing* yang terdiri dari tahap *lowercase*, *remove punctuation and special character*, *stopword removal*, dan *stemming*. Setelah proses preprocessing, selanjutnya adalah memisahkan story menjadi kalimat-kalimat untuk dilakukan proses sentence scoring. Pada proses ini, setiap kalimat diberikan 5 fitur untuk mendapatkan skor tiap kalimat. Fitur tersebut adalah Panjang kalimat, posisi kalimat, kalimat mengandung data numerik, *TF-ISF*, dan kemiripan antar kalimat jika diukur dengan *cosine similarity*. Proses selanjutnya adalah pengklasifikasian data dengan menggunakan metode SVM. Setelah data berhasil diklasifikasikan, selanjutnya tahapan proses peringkasan yaitu dengan memilih hasil klasifikasi dan dijadikan sebuah ringkasan. Setelah itu terakhir adalah proses evaluasi ringkasan dengan menggunakan metrik ROUGE.



Gambar 1. Alur Peringkasan Teks Multidokumen menggunakan metode klasifikasi

### 2.1 Filtering

Proses filtering bertujuan untuk membuat dataset baru multi dokumen yang dibuat dengan cara melibatkan pengambilan artikel dari dataset INDOSUM yang terdiri dari 14.000 berita online. Proses pertama melibatkan pencarian berita dengan menggunakan query tertentu yang akan membentuk list story. Istilah "story" di sini merujuk pada kumpulan berita online yang berkesinambungan dalam satu topik atau pembahasan tertentu, dengan kata kunci query yang telah disusun sebelumnya. Dengan demikian, langkah-langkah ini membentuk dasar pembentukan dataset baru yang lebih spesifik dan terfokus pada topik atau pembahasan tertentu sesuai dengan karakteristik query yang telah ditentukan sebelumnya.

### 2.2 Preprocessing

Untuk mendapatkan data yang berkualitas dan sesuai dengan model, proses preprocessing perlu dilakukan [9]. Pada umumnya, preprocessing pada NLP melibatkan tahapan tokenizing atau memisahkan kalimat menjadi token / kata, namun pada dataset yang digunakan yaitu dataset INDOSUM, setiap kalimat sudah ditokenisasi sehingga tidak perlu lagi melakukan tahapan tokenisasi. Proses preprocessing untuk setiap kalimat pada story adalah sebagai berikut [10].

- *Case Folding*: semua huruf dalam kalimat menjadi huruf kecil untuk menghindari perbedaan yang tidak perlu dalam teks
- *Remove Punctuation and Special Character*: Proses ini menghapus semua tanda baca pada kalimat. Tanda baca yang dimaksud seperti titik, koma, tanda tanya, dan tanda seru dihapus karena tidak bermakna dan tidak penting untuk analisa teks dan dapat mengganggu hasil analisis. Kemudian special character seperti tanda kurung, simbol matematika, dan karakter khusus lainnya juga dihapus karena tidak memiliki peran penting dalam analisis teks dan dapat mengganggu hasil analisis.
- *Stopword Removal*: proses dimana kata-kata umum atau yang disebut dengan *stopword* akan dihilangkan dari sebuah teks karena kata-kata tersebut tidak memberikan kontribusi penting dalam analisis teks. Contohnya, kata-kata seperti kata depan, kata sambung, dan kata-kata

umum lainnya. Proses stopword removal dilakukan agar proses analisis teks dapat berjalan lebih cepat dan juga dapat mengurangi dimensi data yang tidak perlu, sehingga dapat meningkatkan kualitas dari analisis yang dilakukan.

- *Stemming*: proses transformasi kata dalam teks menjadi bentuk dasar atau kata dasar yang memiliki makna yang sama. Dengan melalui proses ini, kata-kata dalam teks akan memiliki representasi yang lebih konsisten dan mudah diproses pada tahapan berikutnya.

### 2.3 Sentence Scoring

Pada tahap ini teks kalimat yang sudah selesai di preprocessing diberi skor atau nilai yang akan menjadi input dari metode inti yaitu metode SVM. Penelitian ini menggunakan 5 fitur untuk memberikan skor pada setiap kalimatnya. Fitur yang akan digunakan pada penelitian ini akan dijelaskan dibawah ini.

Fitur yang pertama adalah Panjang kalimat. Dalam memilih proses ringkasan kalimat, panjang kalimat menjadi salah satu pertimbangan yang penting. Kandidat kalimat ringkasan dipilih berdasarkan panjang kalimat, di mana kalimat terpanjang dijadikan sebagai referensi [5]. Untuk menghitung panjang kalimat, jumlah kata dalam sebuah kalimat dibagi dengan jumlah kata dalam kalimat terpanjang. Persamaan dari Panjang kalimat adalah sebagai berikut.

$$F1 = \frac{\# \text{ kalimat } j}{\# \text{ kata dalam kalimat terpanjang}} \quad (1)$$

$\# \text{ kalimat } j$  adalah jumlah kata dalam kalimat ke- $j$ , dan  $\# \text{ kata dalam kalimat terpanjang}$  adalah jumlah kata dalam kalimat terpanjang dalam 1 story. Jadi setiap story memiliki kalimat terpanjangnya masing-masing.

Fitur kedua adalah posisi kalimat. Fitur ini digunakan karena biasanya kalimat di awal dokumen memiliki skor lebih tinggi daripada di posisi terakhir [11]. Pembobotan kalimat berdasarkan posisi kalimat dapat dijelaskan pada persamaan berikut ini

$$F2 = \frac{m-i}{m} \quad (2)$$

Disini  $m$  adalah total jumlah kalimat pada story,  $i$  adalah posisi kalimat ke- $i$  pada story tersebut.

Fitur ketiga adalah kalimat yang mengandung data numerik. Untuk meringkas teks perlu mempertimbangkan data numerik dalam dokumen, itu karena biasanya kalimat dengan itu memberikan informasi penting seperti tanggal peristiwa, skor pertandingan, jumlah responden, dan lain lain [4]. Persamaannya adalah sebagai berikut.

$$F3 = \frac{\# \text{ data numerik dalam kalimat}}{\# \text{ kata dalam kalimat}} \quad (3)$$

Persamaan untuk fitur ketiga ini adalah jumlah data numerik dalam kalimat dibagi dengan jumlah kata dalam kalimat dalam satu story.

Fitur keempat adalah *TF-ISF*. TF-ISF ini adalah term frequency inverse sentence frequency. Nilai TF-ISF meningkat seiring dengan bertambahnya jumlah kemunculan term dalam kalimat tetapi diimbangi dengan frekuensi kemunculan term dalam keseluruhan dokumen [6]. Rata-rata nilai TF-ISF yang diperoleh untuk suku-suku kalimat diambil untuk menghitung skor TF-ISF untuk kalimat tersebut.

$$F4 = tf \times \left( \log \frac{n}{sf(t)} + 1 \right) \quad (4)$$

Di mana 'tf' adalah frekuensi istilah, 'n' adalah jumlah kalimat dalam dokumen dan 'sf(t)' adalah jumlah kalimat dalam dokumen yang mengandung istilah 't'.

Fitur kelima adalah kemiripan antar kalimat yang dihitung dengan *cosine similarity*. Fitur ini dihitung antara kalimat dan centroid. Centroid pada fitur ini diperoleh dari skor TF-ISF yang paling besar [12].

$$F5(S_i) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\text{Cos}(\text{sentence}_i \cdot \text{centroid})}{\|\text{sentence}_i\| \cdot \|\text{centroid}\|} \quad (5)$$

Pada fitur ini, yang menjadi pembanding adalah kalimat yang memiliki skor TF-ISF terbesar. Setelah itu dibandingkan dengan kalimat yang lain menggunakan cosine similarity.

#### 2.4 Proses Summarization menggunakan metode klasifikasi SVM

Setiap kalimat diberikan label kelas true atau false, dengan label true menandakan bahwa kalimat tersebut merupakan bagian dari rangkuman, sementara label false menunjukkan bahwa kalimat tidak termasuk dalam rangkuman. Pelabelan dilakukan oleh 5 orang yang memiliki pemahaman yang baik tentang Bahasa Indonesia. Hasil label dari kelima ahli tersebut kemudian diambil suara mayoritas. Jika label terbanyak adalah true, maka kalimat akan dianggap sebagai bagian dari rangkuman, dan sebaliknya jika label terbanyak adalah false.

Metode SVM adalah salah satu metode dalam machine learning untuk proses klasifikasi. Dalam penelitian ini, SVM model yang digunakan adalah binary classifier, karena didalam sistem hanya memisahkan kalimat pada dokumen kedalam kelas positif dan kelas negatif. Model SVM untuk klasifikasi biner memerlukan pelabelan data pelatihan menggunakan variable  $y_i = \pm 1$  (dengan tanda yang sesuai dengan kelas positif atau negatif dari kalimat dokumen pelatihan). Hasil dari klasifikasi berdasarkan posisi nilai inputan pada hyperplane. Linear hyperplane digunakan untuk memisahkan kelas positif dan kelas negatif pada data training [13].

Hasil dari klasifikasi berdasarkan posisi nilai inputan pada hyperplane. Linear hyperplane digunakan untuk memisahkan kelas positif (true) dan kelas negative (false) pada data training. Persamaan 6

$$w_i x_i + b = 0 \quad (6)$$

Dimana  $w$  adalah bobot pada masing-masing fitur yang berbentuk vektor n-dimensi,  $x_i$  adalah nilai parameter masing-masing fitur dan  $b$  adalah bias. Langkah berikutnya adalah menghitung parameter nilai dari setiap fitur yang ada. Fungsi ini akan digunakan sebagai bagian dari pembentukan hyperplane.

$$f(y) = \sum_{i=1}^n w_i x_i + b \quad (7)$$

Maka diperlukan untuk mencari nilai  $w$  dan  $b$  agar fungsi tersebut dapat diselesaikan. Algoritma klasifikasi SVM menghasilkan output berdasarkan posisi nilai input terhadap hyperplane. Hyperplane ini berfungsi sebagai batas yang memisahkan antara kelas negatif dan positif dalam data. Langkah awal yang dilakukan untuk mendapatkan hyperplane optimal adalah dengan menginisialisasi bobot awal. Pada penelitian ini bobot awal dari hasil perhitungan untuk mencari hyperplane adalah sebagai berikut.

$$w_i^0 = \frac{1}{n} \quad (8)$$

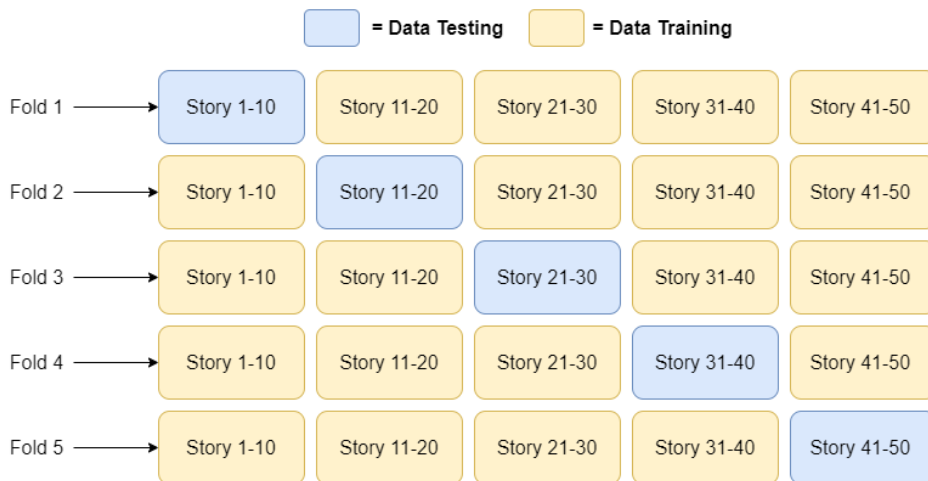
$$w_{baru} = w_{lama} + y_i \cdot x_{ij} \quad (9)$$

Dimana  $w_i^0$  adalah inisialisasi bobot awal dan  $n$  adalah jumlah fitur yang terdapat pada proses sentence scoring. Jika nilai label dari sebuah data, yaitu  $y_i$ , tidak sama dengan hasil klasifikasi dari model pada data tersebut, maka diperlukan perbaruan pada bobot. Bobot baru akan dihitung menggunakan persamaan (9).

Data pelatihan menghasilkan model klasifikasi SVM yang kemudian digunakan untuk menguji model. Pengujian model menggunakan nilai bobot dan bias pada fitur yang sama dengan data latih. Implementasi sistem untuk menentukan kelas dari data pengujian dilakukan dengan memperhitungkan nilai  $f(y)$ . Jika nilai  $f(y) \leq 0$ , maka kalimat tersebut diklasifikasikan sebagai kelas bukan ringkasan, sebaliknya jika nilai  $f(y) \geq 0$ , maka termasuk kelas peringkasan. Ringkasan multidokumen disusun dari kalimat-kalimat dalam kelas ringkasan. Ringkasan akhir disusun dari semua kalimat ringkasan dari semua dokumen.

#### 2.4 Evaluasi

Evaluasi dilakukan dengan memanfaatkan Teknik K-Fold. Dataset dibagi menjadi 5 fold dengan urutan yang sesuai dengan setiap fold memiliki komposisi yang sama jumlahnya, dimana salah satu kelompok tersebut akan menjadi data testing dan data training. Pada setiap fold terdapat 10 story data testing dan 40 story data training yang masing-masing pembagiannya dapat dilihat pada Gambar 2. Fold 1 menunjukkan story 1-10 menjadi data testing dan story 11-50 menjadi data training. Fold 2 berisi story 11-20 yang menjadi data testing dan story 1-10, 21-50 menjadi data training. Pada Fold 3 data testing terdapat pada story 21-30, data training terdapat pada story 1-20, 31-50. Fold 4 data testing terdapat pada story 31-40, data training terdapat pada story 1-30, 41-50. Sedangkan pada fold 5 yang menjadi data testing adalah story 41-50 dan story 1-40 menjadi data training.



Gambar 2. K-Fold Cross Validation

Evaluasi hasil peringkasan menggunakan metrik ROUGE. ROUGE adalah suatu teknik pengukuran secara otomatis untuk mengevaluasi kualitas dari sebuah ringkasan [14]. Teknik ini bekerja dengan cara membandingkan kemiripan antara ringkasan yang dihasilkan oleh mesin dengan ringkasan yang dibuat oleh manusia sebagai acuan yang benar. Dalam hal ini, semakin mirip ringkasan mesin dengan ringkasan manusia, maka semakin tinggi pula nilai ROUGE yang diperoleh dan kualitas ringkasan yang dihasilkan dianggap semakin baik. Perhitungan ROUGE menggunakan persamaan 10.

$$ROUGE - N = \frac{\sum_{S \in Sum_{ref}} \sum_{N-grams \in S} Count_{match}(N-gram)}{\sum_{S \in Sum_{ref}} \sum_{N-grams \in S} Count(N-gram)} \quad (10)$$

Metric ROUGE yang digunakan untuk mengevaluasi hasil peringkasan pada makalah ini adalah ROUGE-1. ROUGE-1, dalam konteks evaluasi sistem ringkasan teks, mengandalkan metrik Recall dan Precision [15]. Recall ROUGE-1 mengukur kemampuan sistem untuk menangkap informasi relevan dari ringkasan mesin (sistem) dan memasukkannya ke dalam ringkasannya, dihitung sebagai jumlah unigram yang sesuai antara ringkasan sistem dan

ringkasan manusia (referensi), dibagi dengan jumlah unigram dalam ringkasan manusia. Persamaan 11 menunjukkan rumus untuk menghitung Recall.

$$Recall = \frac{Overlapping\ Occurrences}{Reference\ Summary} \quad (11)$$

Di sisi lain, Precision ROUGE-1 mengukur sejauh mana informasi yang dimasukkan dalam ringkasan oleh sistem benar-benar relevan, dihitung sebagai jumlah unigram yang sesuai antara ringkasan sistem dan ringkasan manusia, dibagi dengan jumlah unigram dalam ringkasan sistem. Persamaan 12 adalah rumus untuk menghitung Presicion.

$$Presicion = \frac{overlapping\ Occurrences}{System\ Summary} \quad (12)$$

$$F - Measure = 2 \times \frac{(Recall \times Presicion)}{(Recall + Presicion)} \quad (13)$$

Skenario uji coba dilakukan dari fold 1 sampai dengan fold 5 untuk mengukur nilai ROUGE-1 pada masing-masing story dalam dataset. Masing-masing fold berisikan 10 story sebagai data testing dan 40 story lainnya menjadi data training. Uji coba dilakukan sampai fold 5 untuk memperoleh nilai ROUGE-1 dari masing-masing story multi dokumen.

### 3. HASIL DAN PEMBAHASAN

Makalah ini menggunakan dataset INDOSUM. INDOSUM adalah kumpulan berita online berbahasa Indonesia. Kumpulan berita dari dataset ini dibagi menjadi beberapa kategori seperti tajuk utama, teknologi, olahraga, hiburan, dan showbiz. Kemudian waktu terbit berita yang ada di dataset ini berkisar antara tahun 2016-2018 [8].

Data indosum difilter menjadi 50 story. setiap story diperoleh menggunakan query topik cerita. contoh story dapat dilihat dalam Tabel 1. Proses pertama melibatkan pencarian berita dengan menggunakan query tertentu yang membentuk list story. Contoh pada story 1 menggunakan query “laptop”, “gaming”, “asus” untuk mendapatkan story tentang berita yang mengandung kata laptop, gaming, dan asus. Istilah "story" di sini merujuk pada kumpulan berita online yang berkesinambungan dalam satu topik atau pembahasan tertentu, dengan kata kunci query yang telah disusun sebelumnya. Dengan demikian, langkah-langkah ini membentuk dasar pembentukan dataset baru yang lebih spesifik dan terfokus pada topik atau pembahasan tertentu sesuai dengan karakteristik query yang telah ditentukan sebelumnya.

Setelah memperoleh story multi dokumen, langkah selanjutnya adalah melakukan pelabelan terhadap setiap kalimat dalam story tersebut oleh lima orang responden. Pelabelan dilakukan untuk memilih kalimat-kalimat yang akan dimasukkan ke dalam ringkasan dan yang tidak. Setiap responden memberikan label pada setiap kalimat berdasarkan kepentingan dan relevansinya dalam meringkas story. kalimat yang terpilih diberi label true. Sistem melakukan voting terhadap label dari 5 responden. Hasil voting menjadi label kalimat pada story. hasil pelabelan digunakan sebagai ringkasan manusia.

Tabel 1 Contoh query untuk membentuk story sebagai data masukan dari peringkasan multi dokumen

No	Query	Judul
1	['presiden', 'perang', 'melawan']	['Mati Dalam Operasi Antinarkotika Filipina Pro', 'Perang Duterte Lawan Narkoba Dipergencar', 'Filipina Diserang Isu Pelanggaran Ham', 'Rodrigo Duterte Diselidiki Mahkamah Internasional']
2	['laptop', 'gaming', 'asus']	['Asus Xiu Laptop Multimedia Gaming Ready', 'Asus Rilis Laptop Gim Seharga Motor Sport', 'Asus Ramu Notebook Rog Gvik']
3	['gaming', 'acer']	['Asia Pacific Predator League Resmi Dimulai', 'Acer Nitro Spin Laptop Gaming In Terbaru', 'Philips Sedang Garap Monitor K']
4	['musisi', 'synchronize', 'festival']	['Musisi Pengisi Synchronize Festival Siap Beri', 'Musik Dangdut Masuk Dalam Synchronize Fest', 'Jimi The Upstairs Berharap Tampil Di Synchronize F', 'Pujian Jokowi Untuk Musisi Yang Tampil Di Synchronon']
5	['konser', 'jakarta', 'celine', 'dion']	['Celine Dion Akan Gelar Konser Pertama Di Indonesia', 'Netizen Protes Tiket Konser Dp Mobil Celine Dion', 'Celine Dion Ogah Beri Saran Lady Gaga Soal Konser']
6	['film', 'marvel', '2017']	['Lima Hal Yang Harus Diketahui Sebelum Menonton Tho', 'Spider Man Kalahkan Empat Tim Avengers', 'Chris Hemsworth Nyaris Menolak Peran Thor']

Untuk meningkatkan validitas penelitian, proses pelabelan dilakukan kembali dengan melibatkan partisipasi dari 5 responden. Dalam tahap ini, setiap kalimat yang terdapat dalam berita multi-dokumen dinilai sebagai "true" atau "false." Penilaian ini bertujuan untuk menentukan apakah suatu kalimat seharusnya dimasukkan ke dalam ringkasan atau tidak. Label akhir yang dihasilkan adalah label "true," yang diberikan kepada kalimat yang memperoleh lebih dari 50% dari total bobot jawaban yang terkumpul dari responden. Proses ini dibuat dengan cermat untuk memastikan bahwa penentuan keberadaan suatu kalimat dalam ringkasan didasarkan pada penilaian manusia yang akurat dan dapat dipertanggungjawabkan.

Tabel 2. Pelabelan 5 kalimat dari dokumen 1 story 1

Kalimat	R1	R2	R3	R4	R5	Hasil label
Kalimat 1	True	False	True	True	True	True
Kalimat 2	False	True	False	False	False	False
Kalimat 3	True	True	False	False	False	False
Kalimat 4	True	True	True	False	False	True
Kalimat 5	True	False	False	False	False	False

Langkah berikutnya dalam penelitian ini adalah melakukan proses sentence scoring pada setiap berita multi dokumen. Proses ini melibatkan ekstraksi fitur pada setiap kalimat dengan menggunakan lima fitur, yaitu panjang kalimat, posisi kalimat, data numerik, TF-ISF (Term Frequency-Inverse Sentence Frequency), dan kemiripan antar kalimat. Setelah ekstraksi fitur, dilakukan proses klasifikasi menggunakan metode Support Vector Machine (SVM) untuk memisahkan kalimat-kalimat yang dianggap penting dan tidak penting berdasarkan label yang telah diberikan pada setiap kalimat. Hasil uji coba untuk percobaan fold dapat dilihat pada tabel dibawah ini.

Tabel 3. Hasil Perhitungan ROUGE-1 pada Fold 1

List Story	Recall	Presicion	F-Measure
Story 1	0.778	0.710	0.742
Story 2	0.472	0.916	0.623
Story 3	0.690	0.439	0.537
Story 4	0.548	0.326	0.409
Story 5	0.777	0.496	0.606
Story 6	0.897	0.640	0.747
Story 7	1.0	0.477	0.646



Story 8	1.0	0.333	0.5
Story 9	0.921	0.284	0.434
Story 10	0.889	0.412	0.563

Tabel 4. Hasil Perhitungan ROUGE-1 pada Fold 2

List Story	Recall	Presicion	F-Measure
Story 11	1.0	0.581	0.735
Story 12	0.855	0.747	0.797
Story 13	0.747	0.813	0.779
Story 14	0.768	0.393	0.520
Story 15	0.882	0.392	0.543
Story 16	0.916	0.511	0.656
Story 17	0.848	0.435	0.575
Story 18	1.0	0.361	0.530
Story 19	0.880	0.673	0.763
Story 20	0.813	0.674	0.737

Tabel 5. Hasil Perhitungan ROUGE-1 pada Fold 3

List Story	Recall	Presicion	F-Measure
Story 21	0.931	0.481	0.634
Story 22	0.884	0.632	0.737
Story 23	0.998	0.528	0.690
Story 24	0.839	0.618	0.712
Story 25	0.872	0.543	0.669
Story 26	0.997	0.402	0.573
Story 27	1.0	0.5	0.666
Story 28	1.0	0.321	0.486
Story 29	0.994	0.348	0.516
Story 30	0.953	0.506	0.661

Tabel 6. Hasil Perhitungan ROUGE-1 pada Fold 4

List Story	Recall	Presicion	F-Measure
Story 31	0.411	0.466	0.437
Story 32	0.460	0.854	0.598
Story 33	1.0	0.699	0.823
Story 34	1.0	0.8	0.889
Story 35	0.748	0.778	0.763
Story 36	0.756	0.858	0.804
Story 37	0.220	0.382	0.280
Story 38	0.426	1.0	0.597
Story 39	0.523	0.905	0.663
Story 40	0.662	0.813	0.730

Tabel 7. Hasil Perhitungan ROUGE-1 pada Fold 5

Story	Recall	Presicion	F-Measure
Story 41	0.654	0.786	0.714
Story 42	0.613	0.453	0.521
Story 43	0.688	0.836	0.755
Story 44	0.428	0.442	0.435
Story 45	0.714	0.804	0.756
Story 46	0.942	0.576	0.715
Story 47	0.986	0.835	0.904
Story 48	0.876	0.934	0.658
Story 49	0.821	0.549	0.658
Story 50	0.761	0.390	0.516

Pada tabel 3 dapat ditunjukkan bahwa nilai Recall terbesar pada data fold 1 terdapat pada story 7 dan 8 dengan nilai 1. nilai Presicion terbesar pada fold 1 terdapat pada story 2 dengan nilai 0.916 dan nilai F-measure terbesar pada fold 1 terdapat pada story 6 dengan nilai 0,747. Kemudian pada tabel 4 nilai Recall terbesar pada data fold 2 terdapat pada story 11 dan 18, yaitu 1. pada fold 2 nilai Presicion terbesar terdapat pada story 13 dengan nilai 0.813. nilai F-measure terbesar pada fold 2 terdapat pada story 12 dengan nilai 0,797. Kemudian pada tabel 5 nilai Recall terbesar pada fold 3 terdapat pada story 27 dan 28 dengan nilai 1. Pada fold 3 nilai Presicion terbesar terdapat pada story 22 dengan nilai 0,632. Nilai F-measure terbesar pada fold 3 terdapat pada story 22

sebesar 0,737. Kemudian pada tabel 6 nilai Recall terbesar pada fold 4 terdapat pada story 33 dan 34 dengan nilai 1. Pada fold 4 nilai Presicion terbesar terdapat pada story 38 dengan nilai 1. Nilai F-measure terbesar pada fold 4 terdapat pada story 34 sebesar 0,889. Yang terakhir pada tabel 4.15 nilai Recall terbesar terdapat pada story 47 dengan nilai 0.986. Kemudian pada fold 5 nilai Presicion terbesar terdapat pada story 48 dengan nilai 0,934. Kemudian nilai F-measure terbesar pada fold 5 terdapat pada story 47 dengan nilai sebesar 0.904.

Sedangkan nilai terendah dari Recall untuk masing-masing fold adalah fold 1 story 2 dengan nilai 0,472, fold 2 story 13 dengan nilai 0.747, fold 3 story 24 dengan nilai 0.839, fold 4 story 37 dengan nilai 0.220, fold 5 story 44 dengan nilai 0,428. Sedangkan nilai terendah dari Presicion untuk masing-masing fold adalah fold 1 story 9 dengan nilai 0.284, fold 2 story 18 dengan nilai 0.361, fold 3 story 28 dengan nilai 0.321, fold 4 story 37 dengan nilai 0.382, fold 5 story 50 dengan nilai 0.390. Sedangkan nilai terendah dari F-Measure untuk masing-masing fold adalah fold 1 story 4 dengan nilai 0.409, fold 2 story 14 dengan nilai 0.520, fold 3 story 28 dengan nilai 0.486, fold 4 story 37 dengan nilai 0.280, fold 5 story 44 dengan nilai 0.435.

Tabel 8. Nilai Rata-Rata Nilai ROUGE-1 pada 5 Fold

List Fold	Recall	Presicion	F-Measure
Fold 1	0.797	0.503	0.580
Fold 2	0.870	0.558	0.663
Fold 3	0.946	0.487	0.634
Fold 4	0.620	0.755	0.658
Fold 5	0.748	0.660	0.663

Tabel 8 menunjukkan nilai rata-rata hasil akhir dari sistem yang dibangun dengan 5 Fold. Hasil menunjukkan bahwa fold 1 memiliki nilai rata-rata Recall adalah 0.6797, rata-rata Presicion sebesar 0.503, dan nilai rata-rata f-measure mencapai 0.580. Selanjutnya, fold 2 memiliki nilai rata-rata Recall sebesar 0.870, rata-rata Presicion sebesar 0.558, dan nilai rata-rata f-measure adalah 0.663. Fold 3 menunjukkan bahwa nilai rata-rata Recall adalah 0.948, rata-rata Presicion mencapai 0.478, dan nilai rata-rata f-measure adalah 0.634. Pada fold 4, terdapat nilai rata-rata Recall sebesar 0.620, rata-rata Presicion sebesar 0.755, dan nilai rata-rata f-measure mencapai 0.658. Pada fold 5, terdapat nilai rata-rata Recall sebesar 0.748, rata-rata Presicion sebesar 0.660, dan nilai rata-rata f-measure mencapai 0.663.

#### 4. KESIMPULAN DAN SARAN

Hasil uji coba sistem peringkasan teks multi dokumen berbahasa Indonesia dengan metode Sentence Scoring dan Support Vector Machine (SVM) pada 50 data multi dokumen menunjukkan bahwa kinerja sistem mencapai tingkat optimal pada fold 3. Pada fold tersebut, nilai rata-rata Recall mencapai 0.946, Presisi rata-rata sebesar 0.487, dan F-measure rata-rata mencapai 0.634. Fold 1 memiliki nilai rata-rata Recall sebesar 0.6797, sementara fold 2 memiliki nilai rata-rata Recall 0.870. Fold 4 menunjukkan nilai rata-rata Recall sebesar 0.620, dan pada fold 5, nilai rata-rata Recall mencapai 0.748. Hasil ini memberikan gambaran tentang performa sistem dalam mengekstrak informasi yang relevan dari berbagai dokumen pada setiap fold.

Dalam hasil pengujian, terlihat bahwa nilai rata-rata Recall paling tinggi terdapat pada fold 3 dengan skor 0.946. Meskipun nilai terendah untuk rata-rata Recall masih cukup besar, yaitu 0.620, yang terdapat pada fold 4. Dari temuan ini, dapat disimpulkan bahwa sistem peringkasan teks multi dokumen berbahasa Indonesia dengan sentence scoring dan support vector machine (SVM) memiliki kemampuan yang bermanfaat untuk meraih informasi penting dari sebuah berita. Nilai Recall tertinggi tercatat pada fold 3, sementara hasil terendahnya masih menunjukkan kinerja yang relatif baik. Ini menunjukkan potensi sistem dalam menyajikan ringkasan berita dengan baik dan efektif.

## DAFTAR PUSTAKA

- [1] N. Hayatin, K. M. Ghufro, and G. W. Wicaksono, "Summarization of COVID-19 news documents deep learning-based using transformer architecture," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 19, no. 3, pp. 754–761, 2021, doi: 10.12928/TELKOMNIKA.v19i3.18356.
- [2] R. C. Belwal, S. Rai, and A. Gupta, "Extractive text summarization using clustering-based topic modeling," *Soft Comput.*, 2022, doi: 10.1007/s00500-022-07534-6.
- [3] W. Widodo, M. Nugraheni, and I. P. Sari, "A comparative review of extractive text summarization in Indonesian language," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1098, no. 3, p. 032041, 2021, doi: 10.1088/1757-899x/1098/3/032041.
- [4] T. M. P. Aulia, A. Jamaludin, and ..., "Extractive Text Summarization Pada Berita Berbahasa Indonesia Menggunakan Algoritma Support Vector Machine," *J-SAKTI (Jurnal Sains ...*, vol. 5, no. September, pp. 727–735, 2021, [Online]. Available: <http://ejurnal.tunasbangsa.ac.id/index.php/jsakti/article/view/371>.
- [5] A. Qaroush, I. Abu Farha, W. Ghanem, M. Washaha, and E. Maali, "An efficient single document Arabic text summarization using a combination of statistical and semantic features," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 33, no. 6, pp. 677–692, 2021, doi: 10.1016/j.jksuci.2019.03.010.
- [6] S. Mandal, P. Achary, S. Phalke, K. V. K. Poorvaja, and M. Kulkarni, "Extractive Text Summarization Using Supervised Learning and Natural Language Processing," *2021 Int. Conf. Intell. Technol. CONIT 2021*, pp. 1–7, 2021, doi: 10.1109/CONIT51480.2021.9498322.
- [7] 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.
- [8] K. Kurniawan and S. Louvan, "INDO SUM: A New Benchmark Dataset for Indonesian Text Summarization," *2018 Int. Conf. Asian Lang. Process.*, pp. 215–220, 2018.
- [9] A. Naufal Ammar and Suyanto, "Peringkasan Teks Ekstraktif Menggunakan Binary Firefly Algorithm," *Indones. J. Comput.*, vol. 5, no. 2, pp. 31–42, 2020, doi: 10.21108/indojc.2020.5.2.440.
- [10] M. Anandarajan, C. Hill, and T. Nolan, "Text Preprocessing," in *Practical Text Analytics: Maximizing the Value of Text Data*, Cham: Springer International Publishing, 2019, pp. 45–59.
- [11] E. Rainarli and K. E. Dewi, "Relevance Vector Machine for Summarization," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 407, no. 1, 2018, doi: 10.1088/1757-899X/407/1/012075.
- [12] N. S. Shirwandkar and S. Kulkarni, "Extractive Text Summarization Using Deep Learning," *Proc. - 2018 4th Int. Conf. Comput. Commun. Control Autom. ICCUBEA 2018*, pp. 1–5, 2018, doi: 10.1109/ICCUBEA.2018.8697465.
- [13] H. K. H. Vo Duy Thanh, Vo Trung Hung and Tran Quoc Huy, "Text Classification Based on SVM and Text Summarization," *Int. J. Eng. Res. Technol.*, vol. 4, no. 2, pp. 181–186, 2015, [Online]. Available: [www.ijert.org](http://www.ijert.org).
- [14] P. Verma and H. Om, "A novel approach for text summarization using optimal combination of sentence scoring methods," *Sadhana - Acad. Proc. Eng. Sci.*, vol. 44, no. 5, 2019, doi: 10.1007/s12046-019-1082-4.
- [15] N. Khotimah and A. S. Girsang, "Indonesian News Articles Summarization Using Genetic Algorithm," *Eng. Lett.*, vol. 30, no. 1, pp. 152–160, 2022.