

# **PERINGKASAN TEKS MENGGUNAKAN METODE *LEXRANK***

**TUGAS AKHIR**

Diajukan untuk Memenuhi Sebagian Syarat Guna Memperoleh Gelar  
Sarjana Strata Satu (S-1) dalam Teknologi Informasi



Diajukan oleh:

**ARDAN AKHMAD FAJAR SETIADY**

NIM: 2108096085

**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI WALISONGO  
SEMARANG**

**2025**



## **PERNYATAAN KEASLIAN**

Yang bertanda tangan dibawah ini:

Nama : Ardan Akhmad Fajar Setiady

NIM : 2108096085

Jurusan : Teknologi Informasi

Menyatakan bahwa skripsi yang berjudul:

### **PERINGKASAN TEKS MENGGUNAKAN METODE LEXRANK**

Secara keseluruhan adalah hasil penelitian/karya saya sendiri,  
kecuali bagian tertentu yang dirujuk sumbernya.

Semarang, 25 Juni 2025

Pembuat Pernyataan



Handwritten signature of Ardan Akhmad Fajar Setiady.

**Ardan Akhmad Fajar Setiady**

**NIM: 2108096085**





**KEMENTERIAN AGAMA**  
**UNIVERSITAS ISLAM NEGERI WALISONGO**  
**FAKULTAS SAINS DAN TEKNOLOGI**  
Jl. Prof Dr. Hamka kampus III Ngaliyan  
Semarang Telp. 7601295 Fax.7615387

---

**LEMBAR PENGESAHAN**

---

Naskah skripsi berikut ini:

Judul : **PERINGKASAN TEKS MENGGUNAKAN METODE LEXRANK**

Penulis : **Ardan Akhmad Fajar Setiady**

NIM : 2108096085

Jurusan : Teknologi Informasi

Telah diujikan dalam sidang *tugas akhir* oleh Dewan Penguji Fakultas Sains dan Teknologi UIN Walisongo dan dapat diterima sebagai salah satu syarat memperoleh gelar sarjana dalam Ilmu Teknologi Informasi.

Semarang, 25 Juni 2025

**DEWAN PENGUJI**

Penguji I,

**Dr. Masy Ari Ulinuha, M.T**  
NIP. 198108122011011007

Penguji II,

**Hery Mustofa, M.Kom.**  
NIP. 198703172019031007

Penguji III,

**Adzhal Arwani Mahfudh, M.Kom.**  
NIP. 199107032019031006

Penguji IV,

**Mokhammad Ikhlil M., M.Kom.**  
NIP. 198808072019031010

Pembimbing I,

**Hery Mustofa, M.Kom.**  
NIP. 198703172019031007

Pembimbing II,

**Dr. Wenty Dwi Yuniarti,**  
**S.Pd., M.Kom**  
NIP. 197706222006042005





## NOTA PEMBIMBING

Semarang, 16/06/2025

Yth. Ketua Program Studi Teknologi Informasi

Fakultas Sains dan Teknologi

UIN Walisongo Semarang

*Assalamu'alaikum. wr. wb.*

Dengan ini diberitahukan, bahwa saya telah melakukan bimbingan, arahan, dan koreksi naskah skripsi dengan :

Judul : PERINGKASAN TEKS MENGGUNAKAN METODE  
*LEXRANK*

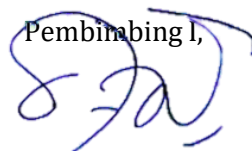
Penulis : Ardan Akhmad Fajar Setiady

NIM : 2108096085

Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqosyah.

*Wassalamu'alaikum. wr. wb.*

Pembimbing I,  


Hery Mustofa, M.Kom

NIP. 19870317 201903 1 007





## NOTA PEMBIMBING

Semarang, 16/06/2025

Yth. Ketua Program Studi Teknologi Informasi

Fakultas Sains dan Teknologi

UIN Walisongo Semarang

*Assalamu'alaikum. wr. wb.*

Dengan ini diberitahukan, bahwa saya telah melakukan bimbingan, arahan, dan koreksi naskah skripsi dengan :

Judul : PERINGKASAN TEKS MENGGUNAKAN METODE  
*LEXRANK*

Penulis : Ardan Akhmad Fajar Setiady

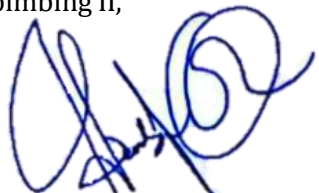
NIM : 2108096085

Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqosyah.

*Wassalamu'alaikum. wr. wb.*

Pembimbing II,



Dr. Wenty Dwi Yuniarti, S.Pd., M.Kom

NIP. 197706222006042005



## **LEMBAR PERSEMBAHAN**

Dengan penuh rasa syukur, penulis mempersembahkan skripsi ini setelah menyelesaikannya kepada:

1. Keluarga besar penulis yang selalu memberikan dukungan dan doa kepada penulis.
2. Seluruh warga akademik UIN Walisongo Semarang, termasuk dosen, pegawai, dan seluruh mahasiswa, diharapkan senantiasa berada dalam kondisi sehat dan terus termotivasi dalam menjalani kegiatan sehari-hari di kampus tercinta UIN Walisongo Semarang.
3. Rekan-rekan penulis yang selalu memberikan dukungan kepada penulis sejak awal hingga akhir.



## **MOTTO**

“Sesungguhnya bersama kesulitan ada kemudahan.”

**(Q.S. Al-Insyirah: 5-6)**



## ABSTRAK

Perkembangan teknologi menyebabkan jumlah dokumen digital meningkat pesat, membuat sulit bagi individu untuk memilih dan memahami isi dokumen dengan cepat. Untuk mengatasi beban ini, merangkum teks menjadi penting. Studi ini menggunakan metode *LexRank* untuk merangkum dokumen dengan memberi nilai pada setiap kalimat. Kualitas ringkasan dinilai dengan skor *ROUGE*, yang mengacu pada standar emas ringkasan oleh ahli Bahasa Indonesia.

Proses penelitian dimulai dengan mengumpulkan dataset menggunakan teknik *scraping*. Setelah itu, data diproses melalui preprocessing. Kemudian, *TF-IDF* digunakan untuk menghitung bobot, dan Cosine Similarity dipakai untuk mengukur kemiripan kalimat. Algoritma *LexRank* kemudian melakukan pemeringkatan kalimat untuk memilih yang paling relevan untuk rangkuman. Evaluasi dilakukan menggunakan *ROUGE* (*ROUGE -1*, *ROUGE -2*, dan *ROUGE -L*) untuk memastikan kualitas dan akurasi dari rangkuman otomatis dibandingkan dengan rangkuman yang dibuat oleh ahli.

Penelitian ini mengumpulkan 50 artikel dari berbagai portal berita Indonesia dengan 7 atribut: penulis, konten, kata kunci, tautan, tanggal terbit, sumber, dan judul. Ringkasan menggunakan *LexRank* berhasil menangkap informasi penting, dengan skor *ROUGE -1* dan *ROUGE -2* antara 21-24%, meskipun struktur kalimat masih perlu diperbaiki. Skor *ROUGE -L* menunjukkan presisi 33% dan *Recall* 36%, menunjukkan susunan kalimat cukup mirip dengan ringkasan manual. Namun, masih ada peluang untuk meningkatkan variasi kata dan kesinambungan bigram agar kualitas ringkasan otomatis semakin mendekati ringkasan manual.

**Kata kunci :** *TF-IDF*, *LexRank*, *Term Frequency*, *Inverse document*, *ROUGE*, *Evaluation*





## KATA PENGANTAR

Puji syukur penulis sampaikan kepada Allah SWT atas segala anugerah, petunjuk, dan karunia-Nya yang telah membantu penulis dalam menyelesaikan laporan penelitian berjudul “Peringkasan Teks Menggunakan Metode *LexRank*” dengan baik dan tanpa kendala yang berarti.

Penelitian ini bertujuan untuk mengembangkan ilmu pengetahuan di bidang teknologi informasi, khususnya pemrosesan bahasa alami. Ini juga merupakan bagian dari kewajiban akademik di Program Studi Teknologi Informasi Universitas Islam Negeri Walisongo Semarang. Penulis berharap temuan dari penelitian ini dapat memberikan manfaat bagi akademik, peneliti, dan pengembang teknologi dalam otomatisasi rangkuman teks.

Dalam pelaksanaan penelitian ini, penulis memperoleh banyak dukungan, bantuan, dan motivasi dari beberapa pihak. Oleh karena itu, penulis ingin mengungkapkan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak Profesor. Dr. H. Nizar, M.Ag., sebagai Rektor Universitas Islam Negeri Walisongo di Semarang.
2. Yang terhormat Bapak Dr. Musahadi, M. Ag, sebagai Dekan Fakultas Sains dan Teknologi di Universitas Islam Negeri Walisongo Semarang.
3. Bapak Dr. Khothibul Umam S.T., M. Kom, yang menjabat sebagai Ketua Program Studi Teknologi Informasi.
4. Bapak Hery Mustofa, M.Kom, dan Ibu Dr. Wenty Dwi Yuniarti, M.Kom sebagai dosen pembimbing yang telah

memberikan petunjuk, saran, serta semangat selama proses penelitian berlangsung.

5. Seluruh pengajar dan tenaga kerja di Universitas Islam Negeri Walisongo Semarang yang telah memberikan bantuan dalam beragam bentuk.
6. Kepada orang tua dan keluarga yang terkasih, saya mengucapkan terima kasih atas semua doa, dukungan emosional, dan semangat yang tidak pernah surut.
7. Rekan-rekan mahasiswa di bidang Teknologi Informasi yang telah memberikan dukungan dan diskusi yang berguna selama berlangsungnya penelitian.
8. Serta semua pihak yang tidak dapat penulis sebutkan secara rinci namun telah memberikan kontribusi dalam penyusunan laporan ini.

Penulis mengakui bahwa laporan ini belum sempurna. Penulis berharap mendapatkan kritik dan saran yang membangun untuk memperbaiki karya ilmiah ini di masa depan. Harapannya adalah laporan ini bermanfaat bagi semua pihak dan menjadi acuan di bidang rangkuman teks otomatis.

Semarang, 24 Juni 2025



Penulis

## DAFTAR ISI

PERINGKASAN TEKS MENGGUNAKAN METODE <i>LEXRANK</i> .....	i
PERNYATAAN KEASLIAN .....	iii
LEMBAR PENGESAHAN.....	v
NOTA PEMBIMBING.....	vii
LEMBAR PERSEMBAHAN .....	xi
MOTTO .....	xiii
ABSTRAK.....	xv
KATA PENGANTAR.....	xvii
DAFTAR ISI .....	xix
DAFTAR TABEL.....	xxiii
DAFTAR GAMBAR.....	xxv
DAFTAR LAMPIRAN.....	xxvii
BAB I.....	1
PENDAHULUAN .....	1
A. Latar Belakang.....	1
B. Rumusan Masalah .....	3
C. Batasan Masalah .....	3
D. Tujuan Penelitian .....	3
E. Manfaat Penelitian.....	4
BAB II .....	5
LANDASAN PUSTAKA .....	5
A. Penelitian Terdahulu .....	5
B. <i>Python</i> .....	18

C.	<i>Supervised learning Dan Unsupervised learning</i> .....	19
D.	Peringkasan <i>Extractive Dan Abstractive</i> .....	21
E.	N-Gram .....	22
F.	Algoritma <i>LexRank</i> .....	23
G.	<i>Preprocessing</i> .....	25
1.	<i>Segmentation</i> .....	26
2.	<i>Case Folding</i> .....	26
3.	<i>Tokenization</i> .....	26
4.	<i>Stemming</i> .....	27
5.	<i>Stopwords Removal</i> .....	27
H.	<i>Term Frequency-Inverse Document Frequency (Tf-Idf)</i> 28	
I.	<i>Cosine Similarity</i> .....	29
J.	<i>ROUGE</i> .....	30
1.	<i>ROUGE -N</i> .....	30
2.	<i>ROUGE -L</i> .....	32
BAB III.....		33
METODOLOGI PENELITIAN.....		33
A.	Alur Penelitian.....	33
B.	Metode Pengumpulan Data.....	34
C.	Perangkat Penelitian.....	35
1.	Perangkat Keras .....	35
2.	Perangkat Lunak.....	36
D.	Uraian Metodologi .....	36
1.	Pengumpulan Dataset .....	36

2.	<i>Preprocessing</i> .....	37
3.	Transformasi .....	43
4.	<i>Algoritma LexRank</i> .....	43
5.	<i>Evaluasi</i> .....	44
BAB IV .....		47
HASIL PENELITIAN DAN PEMBAHASAN .....		47
A.	Pengumpulan Dataset .....	47
1.	<i>Scraping</i> .....	47
B.	Preprocessing .....	52
1.	<i>Segmentation</i> .....	56
2.	<i>Case Folding</i> .....	57
3.	Tokenization .....	58
4.	<i>Stemming</i> .....	60
5.	<i>Stopwords Removal</i> .....	61
C.	Transformasi .....	62
1.	<i>Term Frequency-Inverse Document Frequency</i> .....	62
2.	Cosine Similarity .....	66
D.	<i>Algoritma LexRank</i> .....	69
1.	Peringkatan Kalimat ( <i>LexRank</i> ) .....	69
E.	Evaluasi .....	76
1.	<i>ROUGE</i> .....	76
BAB V .....		89
KESIMPULAN DAN SARAN .....		89
A.	Kesimpulan .....	89

B.   Saran.....	90
DAFTAR PUSTAKA.....	92
DAFTAR LAMPIRAN.....	98

## DAFTAR TABEL

Tabel 3. 1 Perangkat Keras.....	35
Tabel 3. 2 Perangkat Lunak.....	36
Tabel 3. 3 Segmentation.....	38
Tabel 3. 4 Case Folding .....	39
Tabel 3. 5 Tokenization.....	40
Tabel 3. 6 <i>Stemming</i> .....	41
Tabel 3. 7 <i>Stopword</i> Removal .....	42
Tabel 4. 1 Menginstal tools Poetry .....	48
Tabel 4. 2 Menjalankan file main.py .....	48
Tabel 4. 3 Penjelasan Atribut .....	50
Tabel 4. 4 Hasil Proses <i>Scraping</i> .....	50
Tabel 4. 5 Menginstal Library.....	52
Tabel 4. 6 Contoh Hasil <i>Scraping</i> .....	53
Tabel 4. 7 Kode program segmentation .....	57
Tabel 4. 8 Hasil Segmentation .....	57
Tabel 4. 9 Kode program case folding .....	58
Tabel 4. 10 Hasil Case Folding.....	58
Tabel 4. 11 Kode Program tokenization.....	59
Tabel 4. 12 Hasil Tokenization .....	59
Tabel 4. 13 Kode Program <i>Stemming</i> .....	60
Tabel 4. 14 Hasil <i>Stemming</i> .....	61
Tabel 4. 15 Hasil <i>Stopwords</i> Removal.....	62
Tabel 4. 16 Kode program TF-IDF.....	62
Tabel 4. 17 Hasil TF-IDF .....	63
Tabel 4. 18 Frekuensi kata utama .....	64
Tabel 4. 19 Perhitungan IDF.....	64
Tabel 4. 20 Perhitungan TF-IDF.....	65
Tabel 4. 21 Kode program cosine similarity .....	66
Tabel 4. 22 Kode program algoritma LexRank.....	70
Tabel 4. 23 Hasil LexRank .....	72
Tabel 4. 24 Unigram Kedua Ringkasan.....	77
Tabel 4. 25 Kode Program <i>ROUGE</i> -1 .....	79
Tabel 4. 26 Bigram Kedua Ringkasan.....	81
Tabel 4. 27 Kode program <i>ROUGE</i> -2 .....	85
Tabel 4. 28 LCS (Kata Terpanjang yang Cocok) .....	86

Tabel 4. 29 Kode program *ROUGE* -L .....87



## DAFTAR GAMBAR

Gambar 3. 1 Alur Penelitian .....	33
Gambar 4. 1 Pilihan untuk Melanjutkan Proses Pencarian .....	49
Gambar 4. 2 Proses <i>Scrapping</i> .....	49
Gambar 4. 3 Hasil <i>Cosine Similarity</i> .....	66
Gambar 4. 4 Hasil algoritma <i>LexRank</i> .....	71
Gambar 4. 5 Hasil Evaluasi <i>ROUGE</i> .....	76



## **DAFTAR LAMPIRAN**

Lampiran 1 Nilai Rata-rata ROUGE dari Uji Data 50, 100 dan 150 Artikel Berita.....	98
---	----



# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang**

Seiring dengan kemajuan teknologi, jumlah dokumen digital meningkat dengan cepat. Hal ini menyebabkan orang sulit untuk memilih dan memahami inti dokumen dengan cepat karena banyaknya informasi yang tersedia. Oleh karena itu, peringkasan teks adalah salah satu cara penting untuk membantu mengurangi beban tersebut. Peringkasan dokumen teks dapat dilakukan dalam dua cara, peringkasan secara ekstraktif dan peringkasan secara abstrak. Peringkasan secara ekstraktif mengambil beberapa kalimat penting yang mengandung informasi penting dari dokumen asli, dan peringkasan secara abstrak membuat dan menyusun kalimat baru (Fadillah Rihardi et al., n.d.).

Peringkasan ekstraktif bekerja dengan memilih kalimat atau frasa terpenting dari teks asli dan menggunakannya sebagai ringkasan. Metode ini menggunakan algoritma peringkasan untuk menemukan dan mengekstraksi kalimat atau frasa yang paling relevan

dan informatif. Kemudian, algoritma menyusunnya kembali menjadi ringkasan. Sebaliknya, ringkasan abstrak melibatkan proses menulis ulang kalimat baru yang merangkum isi dokumen asli. Metode ini melibatkan pemahaman makna teks sumber dan kemudian menghasilkan ringkasan menggunakan kata-kata dan struktur kalimat yang berbeda sambil menjaga informasi inti (Maulidia et al., n.d.).

Menurut studi sebelumnya (Halimah et al., 2022a), disimpulkan bahwa metode *LexRank* memiliki skor *ROUGE* yang bersaing dalam metode peringkasan dokumen lainnya. Hasil pengujian performa metode yang digunakan sangat dipengaruhi oleh tingkat kompresi yang digunakan. Pada umumnya, hasil yang dicapai akan semakin rendah jika tingkat kompresi yang digunakan lebih rendah, karena peluang pemilihan kalimat oleh metode lebih besar daripada peluang pemilihan kalimat oleh manusia.

Penelitian ini menggunakan metode *LexRank* dalam proses peringkasan suatu dokumen berdasarkan rangking kalimatnya. Untuk menilai kualitas hasil peringkasan, skor *ROUGE* (*Recall -Oriented Understudy for Gisting Evaluation*) digunakan, yang didasarkan pada standar

emas (*gold standard*) ringkasan yang dibuat oleh ahli (Guru Bahasa Indonesia) (Idhafi et al., 2023).

## **B. Rumusan Masalah**

1. Bagaimana penerapan metode *LexRank* untuk peringkasan teks?
2. Bagaimana akurasi metode *LexRank* berdasarkan evaluasi *ROUGE score*?

## **C. Batasan Masalah**

Peneliti menetapkan batasan pada masalah penelitian ini untuk menjaga kejelasan dan objektivitas:

1. Data yang digunakan merupakan hasil dari *scraping* dari beberapa portal berita yang ada di Indonesia seperti CNN Indonesia, Detik, Viva, Mongabay dan Kontan.
2. Algoritma yang digunakan dalam penelitian peringkasan teks ini adalah algoritma *LexRank*.
3. Hasil dari peringkasan akan di evaluasi menggunakan *ROUGE*.

## **D. Tujuan Penelitian**

1. Menerapkan metode *LexRank* pada peringkasan teks agar menghasilkan teks yang singkat dan akurat.

2. Mengukur tingkat keefektifan metode *LexRank* berdasarkan evaluasi *ROUGE score*.

#### **E. Manfaat Penelitian**

1. Memudahkan dalam meringkas teks yang panjang dengan singkat dan akurat.
2. Dapat mempersingkat waktu dalam meringkas teks.
3. Mengimplementasikan metode *LexRank* yang merupakan salah satu algoritma peringkasan teks dan *term* asuk dalam cabang ilmu komputer "*Natural Language Processing (NLP)*"



## **BAB II**

### **LANDASAN PUSTAKA**

#### **A. Penelitian Terdahulu**

Bagian kajian pustaka bertujuan untuk mendukung pelaksanaan penelitian ini, kajian pustaka ini akan membahas penelitian sebelumnya. Dengan mempertimbangkan penelitian sebelumnya tentang artikel dengan judul “Peringkasan teks otomatis (*automated text summarization*) pada artikel berbahasa indonesia menggunakan algoritma *LexRank* ”(Halimah et al., 2022), studi ini menggunakan algoritma *LexRank* untuk membuat peringkasan teks otomatis dari artikel berbahasa Indonesia. Studi ini juga melihat bagaimana pemilihan kalimat berdasarkan ranking yang dihitung *LexRank* mempengaruhi hasil peringkasan. Untuk menilai kualitas hasil peringkasan, skor *ROUGE* digunakan, yang didasarkan pada standar emas ringkasan yang dibuat oleh manusia (penutur asli Bahasa Indonesia).

Menurut penelitian (Nurjannah & Fitri Astuti, 2013) dengan judul “Penerapan Algoritma *Term Frequency-Inverse Document Frequency* (TF-IDF) Untuk *Text Mining*” dari sistem ini melakukan penerapan algoritma *Term*

*Frequency-Inverse Document Frequency* (TF-IDF) untuk text mining. Sistem ini memiliki kemampuan untuk melakukan pencarian berdasarkan *query* untuk sejumlah file secara bersamaan. Metode pembobotan dokumen *TF-IDF* tidak selalu menghasilkan koleksi pengujian yang baik. Hasil ringkasan dari sistem peringkasan teks otomatis yang digunakan dalam penelitian ini adalah kalimat inti yang mirip dengan pertanyaan dan berdasarkan urutan bobot. Oleh karena itu, diharapkan hasil ringkasan memiliki urutan yang sistematis untuk pengembangan penelitian berikutnya.

Pada penelitian "*Indonesian News Extractive Summarization using LexRank and YAKE Algorithm*" (Wijaya & Suganda Girsang, 2024), algoritma *LexRank*, *YAKE* dan *SumBasic* yang di gabungkan mencapai ukuran F1 yang relatif sederhana yaitu 0,388 dalam analisis berbagai metode peringkasan teks. Hal ini menunjukkan bahwa, meskipun dapat ditingkatkan, ada perbedaan presisi-*Recall* yang seimbang. *LSA* dan *TextRank* menunjukkan tingkat *Recall* yang baik, masing-masing 0,637 dan 0,698, tetapi nilai F1 mereka 0,381 untuk *LSA* dan 0.392 untuk *TextRank* menunjukkan bahwa orang lebih suka konten yang lengkap daripada keringkasan, yang ditunjukkan

oleh ketepatannya yang lebih rendah. Dengan ukuran FI terendah 0,344, KLSum menunjukkan pendekatan yang hati-hati, yang mungkin gagal menangkap semua elemen teks utama. Sebaliknya, *LexRank* menunjukkan tingkat presisi dan *Recall* yang lebih baik, dengan ukuran F1 0,433, yang menunjukkan kemampuan mereka untuk menemukan kalimat yang relevan.

Penelitian ini menggunakan algoritma *LexRank* untuk membuat peringkasan teks dari artikel berbahasa Indonesia. Penelitian ini juga melihat bagaimana pemilihan kalimat berdasarkan ranking yang dihitung oleh *LexRank* memengaruhi hasil peringkasan. Untuk menilai kualitas hasil peringkasan, skor *ROUGE* digunakan, yang didasarkan pada standar ringkasan emas yang dibuat oleh penutur asli Bahasa Indonesia.

Tabel 2. 1 Penelitian Terdahulu

No	Paper	Metode	Data (obyek)	Hasil
1	Peringkasan teks otomatis (automated text summarization) pada artikel berbahasa Indonesia	Metode <i>LexRank</i>	Dimulai dengan pengumpulan 300 artikel berita dari internet tentang berbagai topik, <i>term</i> asuk gaya	Dalam pengujian dengan kecepatan kompresi 50%, skor f-measure rata-rata 67,53% pada <i>ROUGE</i> - 1, 59,10% pada <i>ROUGE</i> -

No	Paper	Metode	Data (obyek)	Hasil
	menggunakan algoritma <i>LexRank</i> , 2022 (Halimah et al., 2022a)		hidup, hiburan, kesehatan, politik, teknologi, dan kriminal.	2, dan 67,05% pada <i>ROUGE - L</i> . Dalam pengujian dengan kecepatan kompresi 30%, skor f-measure rata-rata 55,82% pada <i>ROUGE - 1</i> , 45,51% pada <i>ROUGE - 2</i> , dan 54,76% pada <i>ROUGE - L</i> . Hasil ini lebih baik dibandingkan dengan beberapa penelitian metode ekstraktif lainnya. Hasil pengujian performa metode yang digunakan sangat dipengaruhi oleh tingkat kompresi yang digunakan.

No	Paper	Metode	Data (obyek)	Hasil
				Semakin rendah tingkat kompresi yang digunakan, semakin sedikit hasil yang dicapai. Ini disebabkan oleh fakta bahwa peluang untuk metode memilih kalimat yang tidak sesuai dengan pemilihan kalimat manusia menjadi semakin besar.
2	Peringkasan otomatis pada artikel berbahasa Indonesia menggunakan metode maximum	Metode Maximum Marginal Relevance	Dataset ini mengandung 300 artikel dari berbagai website berita online. Selanjutnya, setiap kalimat	Bila dibandingkan dengan metode lain seperti TextRank, <i>LexRank</i> , LDA, dan K-means, metode MMR yang telah

No	Paper	Metode	Data (obyek)	Hasil
	marginal relevance, 2023 (Idhafi et al., 2023)		dalam artikel diberi skor. Ini dilakukan dengan mengambil N kalimat dalam artikel dengan skor tertinggi rata-rata dari dua penilai, yang digunakan sebagai ringkasan manusia (gold standard), di mana N adalah tingkat kompresi artikel. Dari 300 artikel, 150 digunakan sebagai data pelatihan dan 150 lainnya	dioptimasi memberikan nilai <i>ROUGE</i> tertinggi. Hasil ringkasan dengan tingkat kompresi 50% menunjukkan skor F1 rata- rata 73.37% pada <i>ROUGE</i> - 1, 65.52% pada <i>ROUGE</i> - 2, dan 73.05% pada <i>ROUGE</i> - L. Hasil ringkasan dengan tingkat kompresi 30% menunjukkan skor F1 rata- rata 65.17% pada <i>ROUGE</i> - 1, 56.12% pada <i>ROUGE</i> - 2, dan 64.68% pada <i>ROUGE</i> - L.

No	Paper	Metode	Data (obyek)	Hasil
			sebagai data pengujian. Ringkasan manusia akan digunakan untuk mengukur performa ringkasan otomatis dengan MMR.	
3	Peringkasan Teks Otomatis Menggunakan Metode Latent Dirichlet Allocation (LDA), 2023 (Fadillah Rihardi et al., n.d.)	Metode Latent Dirichlet Allocation	Dataset terdiri dari 300 dokumen berita yang dikumpulkan dari berbagai portal berita online.	Proses pencarian model optimal LDA menggunakan data pelatihan dari 150 artikel berita menunjukkan bahwa parameter pemodelan LDA terbaik untuk ringkasan dengan tingkat kompresi 50% adalah tiga set topik dan sepuluh

No	Paper	Metode	Data (obyek)	Hasil
				<p>daftar kata untuk masing-masing topik, dan untuk ringkasan dengan tingkat kompresi 30% adalah dua set topik dan sepuluh daftar kata. Dengan demikian, metode ini menghasilkan skor <i>ROUGE</i> yang kompetitif dibandingkan dengan metode lain. Untuk ringkasan dengan tingkat kompresi dokumen 50%, skor F1 rata-rata adalah 67.81%, 59.96%, 67.44%, dan <i>ROUGE</i> -1,</p>



No	Paper	Metode	Data (obyek)	Hasil
				<i>ROUGE -2</i> dan <i>ROUGE -L</i> . Untuk ringkasan dengan tingkat kompresi dokumen 30%, skor F1 rata-rata adalah 52.37%, 42.11%, dan 51.47%, masing-masing.
4	Penerapan Algoritma Text Mining dan <i>LexRank</i> dalam Meringkas Teks Secara Otomatis, 2022 (Fauzi, 2022)	Text Mining dan <i>LexRank</i>	Data ini dikumpulkan secara real-time dari liputan6.com. Judul berita "Menteri Teten: Kalau UMKM Terganggu, Kita Khawatir Pengangguran Bertambah Banyak" telah	Dengan menggunakan algoritma text mining dan <i>LexRank</i> , ringkasan teks otomatis dapat dibuat tanpa menghilangkan makna teks. Namun, algoritma text mining yang digunakan untuk meringkas teks secara otomatis masih

No	Paper	Metode	Data (obyek)	Hasil
			ditetapkan oleh penulis untuk memudahkan proses analisis.	dianggap lemah karena memiliki nilai yang sama.
5	Literatur Review Terhadap Metode, Aplikasi dan Dataset Peringkasan Dokumen Teks Otomatis untuk Teks Berbahasa Indonesia (Yuliska & Syaliman, 2020)	Maximal Marginal Relevanc e (MMR), Text Rank, Latent Dirichlet Allocation (LDA), Support Vector Machine (SVM), Naive Bayes, Convolutional Neural Network (CNN)	Tersedia dua dataset yang dapat digunakan untuk peringkasan dokumen teks otomatis berbahasa Indonesia: dataset yang berisi kumpulan chat dan IndoSum. Dataset Chat terdiri dari 300 sesi pembicaraan berbahasa Indonesia yang bersumber dari aplikasi online chatting,	Hasil literature review menemukan bahwa peringkasan dokumen teks secara otomatis didominasi oleh teknik ekstraktif. Peringkasan dokumen teks berbahasa Indonesia juga didominasi oleh metode unsupervise, sementara metode yang diawasi seperti machine learning dan deep learning masih sangat

No	Paper	Metode	Data (obyek)	Hasil
			WhatsApp. Dataset Indosum terdiri dari 20 ribu artikel berita	jarang ditemukan.

Studi sebelumnya berkonsentrasi pada pembuatan dan pengujian berbagai metode peringkasan teks otomatis seperti *LexRank*, Maximum Marginal Relevance (MMR), dan Latent Dirichlet Allocation (LDA), serta penggabungannya dengan algoritma text mining, menggunakan dataset yang biasanya terdiri dari artikel berita atau *chat*. Beberapa referensi juga menggunakan teknik kompresi dokumen untuk mengompres atau mereduksi dokumen menjadi sekian persen (Fadillah Rihardi et al., n.d.). Pada referensi Halimah et al., 2022 dilakukan pengujian dengan *Compression Rate* 50% menghasilkan skor *f-measure* 67,53% pada *ROUGE -1*, 59,10% pada *ROUGE -2*, dan 67,05% pada *ROUGE -L*. Sedangkan pada referensi Jain & Rastogi, 2020 dilakukan pengujian tanpa menggunakan teknik kompresi dokumen menunjukkan skor 81% pada *ROUGE -1*, 80% pada *ROUGE -2*, dan 82% pada *ROUGE -L (BBC Dataset)*.

- a. Pengujian Metode *LexRank* dengan dataset khusus, studi ini menguji metode *LexRank* pada dataset yang lebih khusus, yaitu dataset *Excel* dengan kolom seperti identitas, sumber, judul, isi, dan kesimpulan. Seperti *embedding* dan tanggal pembaruan, berbagai metadata dalam dataset ini dapat digunakan untuk eksplorasi tambahan.
- b. Evaluasi dengan ringkasan manual, studi ini membandingkan langsung kinerja ringkasan otomatis *LexRank* dengan ringkasan manual ahli bahasa Indonesia. Ini berbeda dari penelitian sebelumnya yang hanya menilai metode peringkasan yang menggunakan *ROUGE* sebagai standar emas untuk ringkasan manusia.
- c. Metode evaluasi, studi ini menggunakan *ROUGE* sebagai metode utama, tetapi berkonsentrasi pada perbandingan langsung antara ringkasan manual dan otomatis. Ini memberikan pemahaman yang lebih spesifik tentang keakuratan metode *LexRank* dalam konteks linguistik bahasa Indonesia.

Oleh karena itu, penelitian saat ini tidak hanya mengevaluasi kinerja metode peringkasan, tetapi juga memberikan pemahaman lebih lanjut tentang bagaimana

metode *LexRank* dapat mencapai kualitas ringkasan manusia.

Berdasarkan penelitian terdahulu ada beberapa pertimbangan mengapa metode *LexRank* lebih unggul untuk digunakan pada penelitian ini, antara lain:

- a. *LexRank* adalah metode yang sederhana untuk meringkas teks secara ekstraktif tetapi efektif. Metode ini didasarkan pada pemeringkatan berbasis grafik, yang menggunakan kemiripan kosinus untuk mengukur hubungan antar kalimat. Metode ini tidak memerlukan pelatihan atau set data yang besar seperti model berbasis *deep learning*.
- b. *LexRank* bekerja tanpa label data atau anotasi manual (*unsupervised*), yang sangat membantu dalam penelitian ini karena peneliti tidak perlu menghabiskan waktu untuk mengumpulkan data pelatihan tambahan atau melabeli data.
- c. *LexRank* memiliki reputasi yang baik dalam peringkasan otomatis ekstraktif dalam berbagai aplikasi bahasa. Hal ini menunjukkan bahwa teknik ini cukup efektif, dan telah diuji dalam berbagai literatur.

Berdasarkan beberapa pertimbangan diatas, *LexRank* adalah metode yang cocok untuk penelitian

ini karena efisiensi dan kemampuan untuk memberikan hasil yang kompetitif tanpa menggunakan banyak sumber daya.

### **B. Python**

*Python* adalah bahasa pemrograman yang dikembangkan oleh ilmuwan komputer dan matematikawan asal Belanda bernama Guido van Rossum. Van Rossum memulai proyek pengembangan *Python* pada akhir tahun 1989 dengan tujuan membuat bahasa yang sederhana namun kuat. Pada tahun 1991, proyek ini dibuka untuk umum. *Python* telah berkembang pesat sejak saat itu dan digunakan secara luas dalam berbagai industri, *termasuk* pengembangan perangkat lunak, kecerdasan buatan, analisis data, dan komputasi ilmiah.(Fabrizio Romano & Heinrich Kruger, 2021)

*Python* menangani banyak fitur kompleks, seperti manajemen memori, secara otomatis, yang membuatnya lebih mudah digunakan daripada bahasa yang dikompilasi. *Python* memiliki waktu penyelesaian yang cepat, yang sangat bermanfaat untuk pengembangan produk yang membutuhkan iterasi yang cepat. (Unpingco, 2021)

*Python* sangat populer karena beberapa alasan. Misalnya, bersifat *open source*, memiliki komunitas dan sumber belajar *online* yang banyak, mendukung pengelolaan data pada berbagai implementasi dengan paket, dan saat ini dianggap sebagai pemrograman terbaik menurut beberapa komunitas peng-indeks.(Wenty Dwi Yuniarti, 2019)

### **C. *Supervised learning* Dan *Unsupervised learning***

Dalam kecerdasan buatan dan pembelajaran mesin, terdapat dua pendekatan utama, yaitu *supervised learning* dan *unsupervised learning*. Cara paling mudah untuk membedakan keduanya adalah dengan melihat bagaimana model dilatih dan jenis data pelatihan yang digunakan algoritma. Perbedaan utama antara *unsupervised learning* dan *supervised learning* adalah *supervised learning* menggunakan data berlabel untuk menyimpulkan pola dan melatih model untuk memberi label pada data yang tidak terlihat, sementara *unsupervised learning* hanya menggunakan data yang tidak berlabel data, dan melakukannya untuk tujuan menemukan pola, misalnya pengelompokan fitur yang serupa.(Chauhan et al., 2021)

Berikut adalah contoh dari *supervised learning* yang ditunjukkan pada tabel 2.2 (Supriyanto et al., 2023)

Tabel 2. 2 Contoh Supervised learning

No.	Tweet	Hasil	Sentimen
1.	Partai golkar dan demokrat akan bertanding pada saat kampanye 2009	Partai golkar demokrat tanding kampanye 2009	Negatif
2.	Pertandingan pertama antara persema vs persebaya malang	Tanding pertama persema persebaya malang	Positif
3.	Sangat besar hapan wasit saay tanding sepakbola dpt berlaku adil	Besar wasit tanding sepakbola adi	Positif
4.	Partai demokrat menang 2019 krna ada figur sby	Partai demokrat menang 2019 figur sby	Negatif

Penelitian ini menggunakan *unsupervised learning*, dengan kata lain penelitian ini hanya menggunakan data yang tidak berlabel data, dan melakukannya untuk tujuan menemukan pola. Contohnya ada pada tabel 4.5 hasil proses *scraping* .



#### D. Peringkasan *Extractive* Dan *Abstractive*

Peringkasan teks otomatis dibagi menjadi dua kategori, yaitu *extractive* dan *abstractive*. Peringkasan *extractive*, teknik ini menggunakan pemilihan dan ekstraksi kalimat atau frasa penting dari teks sumber untuk membuat ringkasan yang terdiri dari potongan teks asli yang paling relevan. Sedangkan peringkasan *abstractive* ini lebih kompleks, karena membutuhkan pemahaman yang lebih mendalam tentang informasi dan penulisan ulang dengan kata-kata baru untuk menghasilkan kalimat baru yang merangkum isi teks sumber.(Jang & Kim, 2021)

Berikut adalah contoh dari peringkasan *abstractive*

Tabel 2. 3 contoh peringkasan *abstractive (input)*

<i>Input</i>	<i>Output</i>
Istilah Ilmu Data diciptakan pada awal 1960-an untuk menggambarkan profesi baru yang akan mendukung pemahaman dan interpretasi sejumlah besar data yang sedang dikumpulkan pada saat itu. Pada saat itu, tidak ada cara	Ilmu Data mulai dikenalkan pada 1960-an sebagai profesi untuk memahami data dalam jumlah besar. Seiring waktu, bidang ini berkembang dengan menggabungkan ilmu komputer dan statistika guna menghasilkan prediksi dan wawasan di berbagai sektor

<i>Input</i>	<i>Output</i>
untuk memprediksi jumlah data yang benar-benar besar selama lima puluh tahun ke depan. Ilmu Data terus berkembang sebagai disiplin menggunakan ilmu komputer dan metodologi statistik untuk membuat prediksi yang berguna dan mendapatkan wawasan di berbagai bidang.	melalui pemrosesan bahasa alami dan analisis data.

Penelitian ini menggunakan pendekatan ekstraktif, sebab algoritma LexRank memeringkatkan kalimat yang relevan berdasarkan perhitungan bobot *TF-IDF* dan *cosine similarity*. Contohnya ada pada gambar 4.4 hasil algoritma *LexRank*.

### **E. N-Gram**

N-gram adalah mekanisme penting dalam pemrosesan bahasa alam (NLP) untuk mengurai dan memperkirakan data bahasa. N-gram adalah urutan item "n" yang berurutan dari sampel teks atau ucapan tertentu. N-gram dapat berbentuk unigram ( $n=1$ ), bigram ( $n=2$ ),

trigram ( $n=3$ , dll.) dan seterusnya, bergantung pada angka yang ditetapkan untuk " $n$ ". N-gram juga banyak digunakan dalam tugas-tugas yang berkaitan dengan pemrosesan bahasa alami, seperti prediksi teks, pemeriksaan ejaan dan pemodelan bahasa. (Hrytsiv, 2023)

#### **F. Algoritma *LexRank***

*LexRank* adalah metode peringkasan teks otomatis *unsupervised*. Pada dasarnya, algoritma ini mengidentifikasi dan mengekstrak kalimat yang paling informatif untuk ringkasan. Proses dimulai dengan menggambarkan setiap kalimat dalam dokumen sebagai titik di grafik. Untuk mengetahui seberapa mirip kalimat satu sama lain, ukuran kemiripan kosinus dari *TF-IDF* (*Term Frequency-Inverse Document Frequency*) digunakan untuk menghitung jarak antara simpul-simpul ini. Peringkat diberikan berdasarkan seberapa mirip kata-kata tersebut. (Wiratmoko, 2024)

Dalam penelitian (Jain & Rastogi, 2020) menjelaskan bahwa, Algoritma *PageRank Google* digunakan untuk menentukan peringkat halaman web. *LexRank* berasal dari algoritma ini. Metode ini menggunakan gagasan sentralitas leksikal dan berbasis grafik. Matrik kemiripan dibuat dengan menggunakan

skor kemiripan sebagai bobot sisi di antara keduanya dan kalimat sebagai simpul. Meskipun sebagian besar dibuat untuk peringkasan dokumen tunggal, tetapi dapat digunakan untuk berbagai dokumen dengan sedikit modifikasi.

*LexRank* adalah algoritma yang digunakan untuk peringkasan dokumen tunggal atau multi-dokumen, yang berarti ringkasan berasal dari satu atau lebih dokumen. Dalam proses peringkasan, *LexRank* menggunakan pendekatan berbasis *centroid*. Pada sistem yang lebih besar, *LexRank* menggabungkan skor *LexRank* (probabilitas stasioner) dengan fitur lain seperti posisi dan panjang kalimat menggunakan kombinasi bobot yang dapat diatur oleh pengguna atau disesuaikan secara otomatis. (Fauzi, 2022)

Pada dasarnya, *LexRank* menggunakan teknik yang mirip dengan *PageRank*, yang digunakan oleh Google untuk menilai halaman web, memberikan skor kepada setiap poin berdasarkan relevansinya dalam graf. Kalimat dengan skor tertinggi dianggap paling relevan atau penting. (Jain & Rastogi, 2020)

Berikut adalah persamaan untuk perangkian kalimat.

$$p(u) = \frac{d}{N} + (1 - d) \cdot \left( \frac{\text{Sim}(A,B)}{\sum \text{Sim}(B)} \right) \cdot p(B) \quad (2.1)$$

*Rumus Perangkingan Kalimat*

Dengan:

$p$  = skor terakhir untuk simpul (atau sumbu)  $u$  di graf.

$N$  = jumlah titik atau sumbu dalam graf.

$d$  = faktor pendamping.

$u$  = matriks persegi dengan semua elemennya.

$A$  = dokumen A.

$B$  = dokumen B.

Kalimat yang dipilih untuk digunakan sebagai ringkasan dipilih dari kumpulan kalimat yang memiliki peringkat tertinggi. (Halimah et al., 2022)

### ***G. Preprocessing***

*Preprocessing* adalah tahapan mengolah teks agar siap untuk proses berikutnya. Dokumen diproses melalui berbagai prosedur, seperti membuat token, memilih kata-kata yang tidak penting, menghilangkan tanda baca yang tidak perlu, mengubah kata ke dalam bentuk akar kata, dan sebagainya. (Fadillah Rihardi et al., n.d.)

Berikut ini adalah beberapa langkah dan penjelasan *Preprocessing*.

### 1. *Segmentation*

*Segmentation* merupakan tahapan dimana suatu teks dipecah menjadi kalimat. Kemudian setiap kalimat dipecah kembali menjadi kata-kata. Pada tahap ini, kalimat pendek dengan kurang dari empat kata dihilangkan karena memiliki kemungkinan kalimat tersebut tidak terlalu penting.(Jain & Rastogi, 2020)

### 2. *Case Folding*

*Case Folding* adalah proses mengubah semua kata menjadi huruf kecil dan menghilangkan simbol selain huruf, seperti tag (<TEXT>), emotikon (😊), tanda baca (;,:), dan simbol lain. Selain itu, kata biasa dan singkatan diubah menjadi bentuk panjangnya.(Jain & Rastogi, 2020)

### 3. *Tokenization*

*Tokenization* adalah proses membagi rangkaian teks menjadi bagian yang lebih kecil, yang disebut "token". Tujuan utama *tokenization* adalah merepresentasikan teks dengan cara yang bermakna bagi mesin tanpa kehilangan konteksnya. Token-token ini dapat berupa karakter atau sepanjang kata. Algoritma dapat lebih mudah

mengidentifikasi pola dengan mengubah teks menjadi token.(Idhafi et al., 2023)

#### 4. *Stemming*

*Stemming* adalah metode untuk menemukan bentuk dasar dari suatu kata, yang berarti setiap kata yang ada dalam dokumen teks. Teknik *stemming* digunakan untuk mengurangi jumlah indeks yang berbeda dalam dokumen, serta untuk menggabungkan kata-kata lain yang memiliki arti dan kata dasar yang sama tetapi memiliki bentuk atau bentuk yang berbeda.(Yuliska & Syaliman, 2020)

#### 5. *Stopwords Removal*

Salah satu bagian dari proses *Preprocessing* teks adalah penghapusan kata yang tidak relevan dalam suatu kalimat yang didasarkan pada daftar *stopword*. Daftar *stopword* biasanya berasal dari pustaka digital yang sudah tersedia sebelumnya, tetapi tidak semua kata dalam daftar tersebut merupakan kata yang tidak relevan dalam data tertentu. Beberapa contoh *stopword* yang ada di dalam Bahasa Indonesia diantaranya “yang”, “di”, “ke”.(Rinandyaswara et al., 2022)

## ***H. Term Frequency-Inverse Document Frequency (Tf-Idf)***

Algoritma *TF-IDF* adalah skema pembobotan istilah yang paling populer untuk menilai relevansi numerik kata dalam dokumen. *TF-IDF* bekerja dengan melakukan perhitungan atau pembobotan kata dan frekuensi kemunculan kata dalam dokumen tertentu yang menunjukkan tingkat kepentingan kata tersebut dalam dokumen. Skor yang diberikan kepada sebuah kata dengan *TF-IDF* menunjukkan seberapa penting kata tersebut untuk dokumen. (Addiga & Bagui, 2022)

Berdasarkan (Ranu Anugrah et al., 2023), rumus dari *TF-IDF* adalah sebagai berikut:

$$tf = 0,5 + 0,5 \times \frac{tf}{\max(tf)} \quad (2.2)$$

*Rumus Term Frequency (TF)*

$$idf_t = \log \left( \frac{D}{df_t} \right) \quad (2.3)$$

*Rumus Inverse Document Frequency (IDF)*

$$w_{d.t} = tf_{d.t} \times idf_{d.t} \quad (2.4)$$

*Rumus TF-IDF*

Keterangan:

D = dokumen ke-d.

T = *term* ke-t.

W = bobot dokumen ke-d terhadap *term* ke-t.



Tf = banyaknya *term* I pada dokumen.

Idf = Inverse document *frequency*.

Df = banyaknya dokumen yang mengandung *term* I.

### **I. Cosine Similarity**

*Cosine Similarity* merupakan tahapan yang bertujuan untuk mengukur seberapa mirip dari dua dokumen. Ini diukur terhadap dua vektor di dalam ruang dimensi dan dihasilkan dari nilai sudut *cosinus*, yang dihasilkan dari perkalian kedua vektor tersebut. Nilai *cosinus* 0 adalah nilai yang menunjukkan bahwa dokumen yang dibandingkan tidak mirip, sedangkan nilai batas 1 menunjukkan bahwa dokumen yang dibandingkan mirip. (Idhafi et al., 2023)

Berdasarkan penelitian dari (Ranu Anugrah et al., 2023), rumus *cosine similarity* adalah sebagai berikut:

$$\cos \alpha = \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^n A \times B}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}} \quad (2.5)$$

*Rumus Cosine Similarity*

Keterangan:

A = vektor pertama yang dibandingkan, vektor A.

B = vektor kedua yang dibandingkan, vektor B.

$A \cdot B$  = *dot product* dari vektor A dan B.

$|A|$  = panjang vektor A.

$|B|$  = panjang vektor B.

$|A||B|$  adalah produk cross antara vektor A dan B

## **J. ROUGE**

Daftar kriteria untuk menilai teks yang dihasilkan secara otomatis disebut *ROUGE*. *ROUGE* bekerja dengan cara membandingkan ringkasan yang dibuat oleh manusia (dikenal sebagai ringkasan standar atau referensi) dengan ringkasan yang dibuat oleh mesin (dikenal sebagai ringkasan mesin). Untuk memperkirakan metrik ini, evaluasi ini menggunakan ukuran *Precision* dan *Recall*. (Barbella & Tortora, n.d.)

Skor *ROUGE* memiliki rentang antara 0 hingga 1, dengan skor yang mendekati 1 menunjukkan tingkat kecocokan yang lebih tinggi antara teks yang dihasilkan dan teks acuan. *ROUGE* fokus pada penarikan kembali dan memastikan bahwa teks yang dihasilkan mengandung sebanyak mungkin informasi dari teks rujukan. (Ganesan, 2018)

### **1. ROUGE -N**

*ROUGE* -N adalah perhitungan dalam peringkasan teks yang didasarkan pada perbandingan n-gram dalam dokumen referensi

(Halimah et al., 2022a), dan dihitung sebagai berikut.

$$ROUGE_N = \frac{\sum_{S \in \{ReferencesSummaries\}} \sum_{gram_n} Count_{match}(gram_n)}{\sum_{S \in \{ReferencesSummaries\}} \sum_{gram_n} Count(gram_n)} \quad (2.6)$$

*Rumus Evaluasi ROUGE -N*

Di mana  $n$  adalah panjang gram ke- $n$ ,  $Count_{match}(gram_n)$  adalah jumlah  $n$ -gram tertinggi yang muncul bersama dalam set ringkasan kandidat dan referensi. (Lin, 2004)

#### a. ROUGE -1

Merupakan jenis evaluasi ROUGE -N yang didasarkan pada perbandingan unigram atau  $n=1$ .

$$ROUGE - 1 \text{ recall} = \frac{\text{jumlah unigram yang sama}}{\text{total kata di ringkasan manual}} \quad (2.7)$$

*Rumus ROUGE -1 Recall*

$$ROUGE - 1 \text{ precision} = \frac{\text{jumlah unigram yang sama}}{\text{total kata di ringkasan sistem}} \quad (2.8)$$

*Rumus ROUGE -1 Precision*

#### b. ROUGE -2

Merupakan jenis evaluasi ROUGE -N yang didasarkan pada perbandingan unigram atau  $n=2$ .

$$ROUGE - 2 \text{ recall} = \frac{\text{jumlah bigram yang sama}}{\text{total kata di ringkasan manual}} \quad (2.9)$$

*Rumus ROUGE -2 Recall*

$$ROUGE - 2 \text{ precision} = \frac{\text{jumlah bigram yang sama}}{\text{total kata di ringkasan sistem}} \quad (2.10)$$

## 2. *ROUGE -L*

*ROUGE -L* mencocokkan ringkasan sistem dan ringkasan manual dengan seluruh susunan kata terpanjang yang sama yang dikenal sebagai subseri terpanjang yang umum (*Longest Common Subsequence*).

$$ROUGE - L recall = \frac{LCS(sistem, manual)}{total\ kata\ di\ ringkasan\ manual} \quad (2.11)$$

*ROUGE -L Recall*

$$ROUGE - L precision = \frac{LCS(sistem, manual)}{total\ kata\ di\ ringkasan\ sistem} \quad (2.12)$$

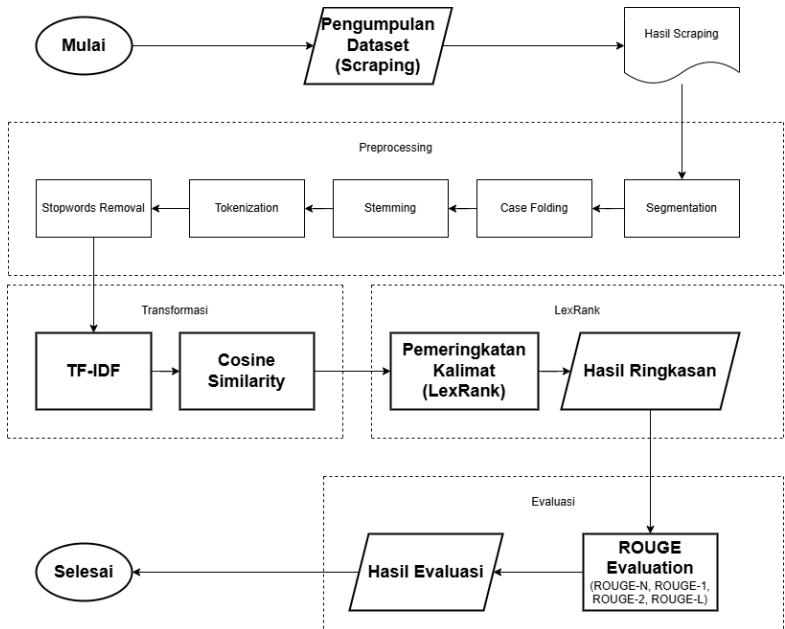
*Rumus ROUGE -L Precision*

### BAB III

## METODOLOGI PENELITIAN

### A. Alur Penelitian

Dalam penelitian ini ada beberapa proses yang harus dilakukan, proses-proses tersebut digambarkan dibawah ini.



Gambar 3. 1 Alur Penelitian

Gambar 1 menunjukkan bahwa proses penelitian meliputi pengumpulan dataset (*scraping* ), kemudian

*Preprocessing*, penggunaan *TF-IDF* untuk perhitungan bobot, dan penggunaan *Cosine Similarity* untuk mengukur kemiripan kalimat. Algoritma *LexRank* melakukan pemeringkatan kalimat untuk memilih kalimat yang paling relevan untuk dimasukkan ke dalam rangkuman. Evaluasi menggunakan *ROUGE* (*ROUGE -1*, *ROUGE -2*, dan *ROUGE -L*) untuk memastikan kualitas dan keakuratan output rangkuman otomatis dibandingkan dengan rangkuman yang dibuat oleh ahli (Guru Bahasa Indonesia).

## **B. Metode Pengumpulan Data**

### **1. Studi Pustaka**

Dataset yang digunakan berasal dari *Scraping* artikel berita. Setiap artikel memiliki atribut seperti author, content, keyword, link, publish date, source (seperti CNN Indonesia, Detik, Viva, Mongabay dan Kontan), dan title.

### **2. Wawancara**

Pada metode ini melibatkan wawancara langsung dengan seorang guru bahasa Indonesia, seorang ahli dalam bidang bahasa. Tujuan wawancara ini adalah untuk mendapatkan pemahaman yang lebih mendalam tentang metode atau teknik yang

digunakan untuk melakukan peringkasan teks secara manual. Salah satu fokus utama diskusi adalah bagaimana menggunakan metode peringkasan manual yang efektif untuk menghasilkan ringkasan yang tidak hanya penuh dengan informasi, tetapi juga akurat dan representatif dari isi teks aslinya.

### **C. Perangkat Penelitian**

Secara khusus, untuk memastikan bahwa proses sistem dapat berjalan dengan baik, perlu dipenuhi sejumlah perangkat keras dan perangkat lunak yang diperlukan. Berikut adalah spesifikasi persyaratan penelitian:

#### **1. Perangkat Keras**

Tabel 3. 1 Perangkat Keras

No.	Perangkat Keras	Spesifikasi
1.	Device	Lenovo LOQ-15AHP9-37iD
2.	Processor	AMD Ryzen 5 8645HS
3.	Memori (RAM)	8,00 GB
4.	Monitor	15,6 inch
5.	Mouse	LOQ M100

## 2. Perangkat Lunak

Tabel 3. 2 Perangkat Lunak

No.	Perangkat Lunak	Spesifikasi
1.	Sistem Operasi	Windows 11 64-bit
2.	Bahasa Pemrograman	Python
3.	Microsoft Office	Word & Excel 2021
4.	Browser	Google Chrome
5.	Kode Editor	Visual Code & Google Colab

### D. Uraian Metodologi

Pada titik ini, alur penelitian di atas diuraikan untuk memberikan penjelasan lebih lanjut. Berikut adalah beberapa contohnya.

#### 1. Pengumpulan Dataset

Data yang digunakan untuk penelitian ini berasal dari berbagai portal berita online terkemuka di Indonesia, seperti Mongabay, VIVA, Kontan, dan CNN Indonesia. Data yang dikumpulkan dalam penelitian ini berbentuk teks dan diperoleh melalui teknik *scraping* web, yang dilakukan secara otomatis dengan menggunakan program atau skrip tertentu untuk mengekstrak berbagai jenis informasi dari halaman web, seperti teks, gambar, dan data lainnya. Jenis informasi ini kemudian disimpan dalam format yang mudah diolah, seperti *spreadsheet* atau basis data.



Dalam penelitian ini, kata kunci yang digunakan adalah “indonesia”. Dengan menggunakan kata kunci tersebut, program secara otomatis akan memfilter dan mengakses artikel berita dari berbagai portal yang berada dalam cakupan wilayah negara Indonesia.

## 2. *Preprocessing*

Sebelum data diolah lebih lanjut, data harus melewati tahapan *Preprocessing*. Proses ini merupakan tahap dimana dokumen dapat dibagi menjadi paragraf, kalimat, bab, sub bab, dan akhirnya potongan kata. Proses ini juga menghilangkan digit angka, huruf kapital, atau karakter lainnya dirubah (Fauzi, 2022).

Proses penjelasan tahapan *Preprocessing* teks diberikan sebagai berikut:

### a. *Segmentation*

Merupakan tahap awal proses *Preprocessing* teks. Teks berita yang terdiri dari paragraph dibagi menjadi beberapa kalimat selama proses ini. Setiap kalimat harus dipisahkan dengan tanda baca seperti titik (.), seru (!), dan tanya (?). (Dimas et al., n.d.)

Tabel 3. 3 *Segmentation*

<i>Input</i>	<i>Output</i>
<p>Grup Astra mengonfirmasi investasi perdana ke sektor digital sebesar US\$ 150 juta (2 triliun Rupiah) untuk Go-Jek. Diharapkan dengan investasi ini kedua perusahaan bisa melakukan banyak kolaborasi bisnis sebagai upaya mengembangkan UMKM di Indonesia. Penandatanganan nota kesepahaman tersebut diselenggarakan hari ini (12/2) dihadiri Presiden Direktur Astra International Prijono Sugiarto CEO dan Founder Go-Jek Nadiem Makarim dan disaksikan Menkominfo Rudiantara.</p>	<p>'Grup Astra mengonfirmasi investasi perdana ke sektor digital sebesar US\$ 150 juta (2 triliun Rupiah) untuk Go-Jek.', 'Diharapkan dengan investasi ini kedua perusahaan bisa melakukan banyak kolaborasi bisnis sebagai upaya mengembangkan UMKM di Indonesia.', 'Penandatanganan nota kesepahaman tersebut diselenggarakan hari ini (12/2) dihadiri Presiden Direktur Astra International Prijono Sugiarto CEO dan Founder Go-Jek Nadiem Makarim dan disaksikan Menkominfo Rudiantara.'</p>

b. *Case Folding*

Merupakan langkah mengubah huruf besar menjadi huruf kecil. (Umam Syaliman & Caltex Riau yuliska, 2022)

Tabel 3. 4 *Case Folding*

Input	Output
Grup Astra mengonfirmasi investasi perdana ke sektor digital sebesar US\$ 150 juta (2 triliun Rupiah) untuk Go-Jek. Diharapkan dengan investasi ini kedua perusahaan bisa melakukan banyak kolaborasi bisnis sebagai upaya mengembangkan UMKM di Indonesia. Penandatanganan nota kesepahaman tersebut diselenggarakan hari ini (12/2) dihadiri Presiden Direktur Astra International Prijono Sugiarto CEO dan Founder Go-Jek Nadiem Makarim dan disaksikan Menkominfo Rudiantara.	'grup astra mengonfirmasi investasi perdana ke sektor digital sebesar us\$ 150 juta (2 triliun rupiah) untuk go-jek.', 'diharapkan dengan investasi ini kedua perusahaan bisa melakukan banyak kolaborasi bisnis sebagai upaya mengembangkan umkm di indonesia.', 'penandatanganan nota kesepahaman tersebut diselenggarakan hari ini (12/2) dihadiri presiden direktur astra international prijono sugiarto ceo dan founder go-jek nadiem makarim dan disaksikan menkominfo rudiantara.'

c. *Tokenization*

Langkah ketika teks dibagi menjadi kata, frasa, simbol, atau komponen lain yang memiliki arti yang dikenal sebagai token. (BAITA & CAHYONO, 2021)

Tabel 3. 5 *Tokenization*

<i>Input</i>	<i>Output</i>
<p><i>Grup Astra</i>  menganfirmasi  investasi perdana ke  sektor digital sebesar  US\$ 150 juta (2 triliun  Rupiah) untuk Go-Jek.  Diharapkan dengan  investasi ini kedua  perusahaan bisa  melakukan banyak  kolaborasi bisnis  sebagai upaya  mengembangkan  UMKM di Indonesia.  Penandatanganan nota  kesepahaman tersebut  diselenggarakan hari  ini (12/2) dihadiri  Presiden Direktur Astra  International Prijono  Sugiarto CEO dan  Founder Go-Jek  Nadiem Makarim dan  disaksikan Menkominfo  Rudiantara.</p>	<p>'grup', 'astra',  'menganfirmasi',  'investasi', 'perdana',  'ke', 'sektor', 'digital',  'sebesar', 'us', '\$', '150',  'juta', '[', '2', 'triliun',  'rupiah', ']', 'untuk',  'go-jek', '.',  ['diharapkan',  'dengan', 'investasi',  'ini', 'kedua',  'perusahaan', 'bisa',  'melakukan', 'banyak',  'kolaborasi', 'bisnis',  'sebagai', 'upaya',  'mengembangkan',  'umkm', 'di',  'indonesia', '.'],  ['penandatanganan',  'nota', 'kesepahaman',  'tersebut',  'diselenggarakan',  'hari', 'ini', '[', '12/2',  ']', 'dihadiri',  'presiden', 'direktur',  'astra', 'international',  'prijono', 'sugiarto',  'ceo', 'dan', 'founder',  'go-jek', 'nadiem',  'makarim', 'dan',  'disaksikan',  'menkominfo',  'rudiantara', '.']</p>

d. *Stemming*

Proses mengubah kata menjadi kata dasar.

Tujuan utama dari proses ini adalah

mengurangi kata menjadi bentuk dasar atau akarnya. (Diki Hendriyanto et al., 2022)

Tabel 3. 6 *Stemming*

<i>Input</i>	<i>Output</i>
Grup Astra mengonfirmasi investasi perdana ke sektor digital sebesar US\$ 150 juta (2 triliun Rupiah) untuk Go-Jek. Diharapkan dengan investasi ini kedua perusahaan bisa melakukan banyak kolaborasi bisnis sebagai upaya mengembangkan UMKM di Indonesia. Penandatanganan nota kesepahaman tersebut diselenggarakan hari ini (12/2) dihadiri Presiden Direktur Astra International Prijono Sugiarto CEO dan Founder Go-Jek Nadiem Makarim dan disaksikan Menkominfo Rudiantara.	'grup astra konfirmasi investasi perdana ke sektor digital besar us 150 juta 2 triliun rupiah untuk go-jek', 'harap dengan investasi ini dua usaha bisa laku banyak kolaborasi bisnis sebagai upaya kembang umkm di indonesia', 'penandatanganan nota paham sebut selenggara hari ini 12 2 hadir presiden direktur astra international prijono sugiarto ceo dan founder go-jek nadiem makarim dan saksi menkominfo rudiantara'

e. *Stopwords Removal*

Prosedur untuk menghapus kata-kata yang tidak penting atau tidak perlu dalam dokumen (Zakawaly et al., 2023).

Tabel 3. 7 *Stopword Removal*

<i>Input</i>	<i>Output</i>
Grup Astra mengonfirmasi investasi perdana ke sektor digital sebesar US\$ 150 juta (2 triliun Rupiah) untuk Go- Jek. Diharapkan dengan investasi ini kedua perusahaan bisa melakukan banyak kolaborasi bisnis sebagai upaya mengembangkan UMKM di Indonesia. Penandatanganan nota kesepahaman tersebut diselenggarakan hari ini (12/2) dihadiri Presiden Direktur Astra International Priyono Sugiarto CEO dan Founder Go-Jek Nadiem Makarim dan disaksikan Menkominfo Rudiantara.	'grup', 'astra', 'mengonfirmasi', 'investasi', 'perdana', 'sektor', 'digital', 'us', '\$', '150', 'juta', '(', '2', 'triliun', 'rupiah', ')', 'go-jek', '.', ['diharapkan', 'investasi', 'perusahaan', 'kolaborasi', 'bisnis', 'upaya', 'mengembangkan', 'umkm', 'indonesia', ''], ['penandatanganan', 'nota', 'kesepahaman', 'diselenggarakan', '(', '12/2', ')', 'dihadiri', 'presiden', 'direktur', 'astra', 'international', 'priyono', 'sugiarto', 'ceo', 'founder', 'go- jek', 'nadiem', 'makarim', 'disaksikan', 'menkominfo', 'rudiantara', '.']

### 3. Transformasi

Setelah semua dataset telah melalui tahap Preprocessing, maka tahap berikutnya adalah transformasi. Pada fase transformasi termasuk penentuan bobot. Kata diolah menggunakan metode TF-IDF dan cosine similarity. Dalam proses TF-IDF memberikan nilai bagi setiap kata pada Setiap kalimat atau dokumen. Di sisi lain, cosine similarity merupakan metode yang sering digunakan dalam ringkasan teks otomatis untuk menilai tingkat kesamaan antara dua teks atau kalimat. Ini juga digunakan untuk menilai seberapa berkaitan suatu teks dengan teks lainnya serta untuk mengidentifikasi kalimat-kalimat yang penting untuk disertakan dalam ringkasan, sehingga dataset telah siap untuk digunakan untuk proses berikutnya.

### 4. *Algoritma LexRank*

Dataset yang telah melalui tahap preprocessing dan ekstraksi fitur, kemudian akan diteruskan dengan proses pengurutan kalimat dengan menggunakan algoritma LexRank. Algoritma ini mengenali dan

mengambil kalimat yang paling berisi informasi untuk disusun menjadi ringkasan. Proses ini dimulai dengan merepresentasikan setiap kalimat dalam dokumen sebagai sebuah titik di dalam grafik. Untuk menentukan tingkat kesamaan antara kalimat, ukuran kemiripan kosinus berdasarkan *TF-IDF* digunakan untuk menghitung jarak antara node-node tersebut.(Wiratmoko, 2024) Kalimat yang dipilih untuk dijadikan ringkasan diambil dari sekumpulan kalimat yang memiliki peringkat paling tinggi.(Halimah et al., 2022a)

## 5. *Evaluasi*

Daftar kriteria untuk menilai teks yang dihasilkan secara otomatis disebut *ROUGE* . *ROUGE* bekerja dengan cara membandingkan ringkasan yang dibuat oleh manusia (dikenal sebagai ringkasan standar atau referensi) dengan ringkasan yang dibuat oleh mesin (dikenal sebagai ringkasan mesin). Untuk memperkirakan metrik ini, evaluasi ini



menggunakan ukuran *Precision* dan *Recall* .  
(Barbella & Tortora, n.d.)

a. *ROUGE -N*

*ROUGE -N* adalah perhitungan dalam peringkasan teks yang didasarkan pada perbandingan n-gram dalam dokumen referensi (Halimah et al., 2022a), dan dihitung sebagai berikut.

$$ROUGE_N = \frac{\sum_{S \in \{ReferencesSummaries\}} \sum_{gram_n} Count_{match}(gram_n)}{\sum_{S \in \{ReferencesSummaries\}} \sum_{gram_n} Count(gram_n)} \quad (3.6)$$

*Rumus Evaluasi ROUGE -N*

1) *ROUGE -1*

Merupakan jenis evaluasi *ROUGE -N* yang didasarkan pada perbandingan unigram atau n=1.

$$ROUGE - 1 recall = \frac{jumlah\ unigram\ yang\ sama}{total\ kata\ di\ ringkasan\ manual} \quad (3.7)$$

*Rumus ROUGE -1 Recall*

$$ROUGE - 1 precision = \frac{jumlah\ unigram\ yang\ sama}{total\ kata\ di\ ringkasan\ sistem} \quad (3.8)$$

*Rumus ROUGE -1 Precision*

2) *ROUGE -2*

Merupakan jenis evaluasi *ROUGE -N* yang didasarkan pada perbandingan bigram atau n=2.

$$ROUGE - 2 recall = \frac{\text{jumlah bigram yang sama}}{\text{total kata di ringkasan manual}} \quad (3.9)$$

*Rumus ROUGE -2 Recall*

$$ROUGE - 2 precision = \frac{\text{jumlah bigram yang sama}}{\text{total kata di ringkasan sistem}} \quad (3.10)$$

*Rumus ROUGE -2 Precision*

**b. ROUGE -L**

*ROUGE -L* mencocokkan ringkasan sistem dan ringkasan manual dengan seluruh susunan kata terpanjang yang sama yang dikenal sebagai subseri terpanjang yang umum (Longest Common Subsequence).

$$ROUGE - L recall = \frac{LCS(sistem, manual)}{\text{total kata di ringkasan manual}} \quad (3.11)$$

*Rumus ROUGE -L Recall*

$$ROUGE - L precision = \frac{LCS(sistem, manual)}{\text{total kata di ringkasan sistem}} \quad (3.12)$$

*Rumus ROUGE -L Precision*

## BAB IV

### HASIL PENELITIAN DAN PEMBAHASAN

#### A. Pengumpulan Dataset

##### 1. *Scraping*

Data yang dikumpulkan terdiri dari kumpulan teks dari artikel berita yang diperoleh melalui proses otomatisasi menggunakan alat bantu yang disebut *Scrapy*. *Scrapy* adalah *framework* open-source yang dibuat untuk melakukan *scraping web* menggunakan bahasa pemrograman Python. Arsitekturnya yang fleksibel dan modular memudahkan pengembang dan peneliti mengotomatisasi proses ekstraksi data dari berbagai situs web. Dengan kemampuan untuk mengumpulkan data secara sistematis dan terstruktur, *framework* ini sangat cocok untuk pengumpulan data skala besar untuk keperluan analisis tambahan.

Pada penelitian ini penulis menggunakan kode *repository* dari *github* *News-Scraper* (<https://github.com/binsarjr/news-scraper>) sebagai alat bantu untuk mendapatkan artikel berita. Langkah pertama yang harus dilakukan adalah menginstal *Poetry* pada system dengan menggunakan perintah *Poetry install* di *terminal*. Berikut adalah kode program untuk menginstal *tools Poetry* yang terdapat pada table 4.1

Tabel 4. 1 Menginstal *tools Poetry*

```
News-scraper>Poetry install
```

Setelah menginstal *Poetry*, Langkah selanjutnya adalah menjalankan *file main.py* yang ada pada *repository* dengan menggunakan perintah “*poetry run python main.py -q indonesia --since 2025-01-01 --until 2025-03-31 --output data.csv*” dengan keterangan “*-q/query*” adalah kueri pencarian, “*-since*” adalah tanggal dimulainya pencarian, “*-until*” adalah tanggal berakhirnya pencarian, “*-output*” *file output* (mendukung format *.json*, *.csv* dan *.xml*). Adapun kode program untuk menjalankan *file main.py* pada tabel 4.2

Tabel 4. 2 Menjalankan *file main.py*

```
News-scraper>poetry run python main.py -q indonesia  
--since 2025-01-01 --until 2025-01-31 --output  
data.csv
```

Lalu akan muncul pertanyaan pilihan *yes* atau *no*, ini dikarenakan argumen “*-q*”, “*-since*” dan “*-until*” merupakan opsional. Maka dari itu tanpa argumen tersebut kode akan tetap berjalan menggunakan hari pada saat kode tersebut dijalankan. Pertanyaan tersebut ditunjukkan pada gambar 4.1



```
2025-05-27 23:27:23 [scrapy.core.scraper] WARNING: Dropped: Item created after until
{'author': 'Lucas Aditya - Sepakbola',
 'content': '\n'
            '\n'
            '\n'
            '\n'
            'Jakarta - Stefano Lilipaly kembali dipanggil Timnas Indonesia. '
            'Pelatih Garuda, Patrick Kluivert, menyebut pesepakbola 35 tahun '
            'itu bisa menjadi teladan tim asuhannya.Terakhir kali Lilipaly '
            'dipanggil Indonesia pada Maret 2023. Waktu itu, dia menggantikan '
            'Egy Maulana Vikri yang cedera saat Indonesia menghadapi '
            'Burundi.Setelah dua tahun, Lilipaly akhirnya kembali ke Timnas. '
            'Kluivert menyertakannya dalam daftar 32 nama pemain di skuad '}
```

Gambar 4. 2 Proses *Scraping*

Hasil dari proses ini adalah terkumpulnya sejumlah 50 artikel berita yang berhasil diambil dari berbagai sumber. Seluruh artikel yang diperoleh tersebut memiliki atribut 7 variabel, yaitu *author*, *content*, *keyword*, *link*, *publish date*, *source*, *title*. Berikut adalah penjelasan dari atribut artikel berita pada tabel 4.3.

Tabel 4. 3 Penjelasan Atribut

No.	Atribut	Keterangan
1.	<i>Author</i>	Pencipta atau pemilik karya tulis.
2.	<i>Content</i>	Isi dari artikel berita.
3.	<i>Keyword</i>	Kata kunci dari artikel berita.
4.	<i>Link</i>	Tautan dari artikel berita.
5.	<i>Publish date</i>	Tanggal publikasi artikel berita.
6.	<i>Source</i>	Sumber dari artikel berita.
7.	<i>Title</i>	Judul dari artikel berita.

Berikut adalah hasil dari proses *scraping* ditunjukkan pada tabel 4.4

Tabel 4. 4 Hasil Proses *Scraping*

Author	CNN INDONESIA
Content	<p>Jurus andalan Pratama Arhan lewat lemparan jauh seketika hilang dari Timnas Indonesia dalam dua laga Kualifikasi Piala Dunia 2026. Nama Arhan tak masuk dalam skuad Indonesia lawan Australia (20 Maret) dan Bahrain (25 Maret). Perannya sebagai bek kiri digeser oleh Dean James, Calvin Verdonk, dan Shayne Pattynama. Hal ini turut membuat lemparan jauh khas Pratama Arhan tak menghiasi penampilan skuad Garuda. Eksekutor lemparan jauh pun diambil oleh Kevin Diks yang tampil sebagai bek sayap kanan kontra Australia dan Bahrain.</p> <p>Pilihan Redaksi Apa Timnas Indonesia Masih Bisa Lolos Piala Dunia Usai Gebuk Bahrain? Ole Romeny: Atmosfer di GBK Aneh, Saya Belum</p>

	<p>Pernah Mengalaminya Cara Timnas Indonesia Finis Posisi 2 dan Lolos Langsung ke Piala Dunia Saat melawan Bahrain, Kevin Diks tercatat dua kali melakukan lemparan jauh. Namun, lemparan jauh pemain FC Copenhagen itu tak setajam lesatan bola dari kedua tangan Arhan. Bola yang dilempar Kevin Diks cenderung melambung tinggi dan mudah diatasi pemain Bahrain dalam duel udara. Kesempatan membuka peluang pun hilang. Usai gagal melancarkan upaya lemparan jauh, Kevin Diks yang masih jadi algojo pun mengenyahkan siasat tersebut dan memilih jalur aman dengan lemparan jarak dekat. Sebelumnya, salah satu kekuatan Timnas Indonesia berasal dari lemparan Arhan. Ada tujuh gol tim Merah Putih yang berawal dari skema ini medio 2022 hingga 2024. Korban-korban lemparan Arhan di antaranya Timor Leste, Curacao, Filipina, Jepang, Vietnam, China, dan Myanmar. Tak hanya di Timnas Indonesia, lemparan Arhan juga jadi senjata mematikan di level klub. Kemenangan Bangkok United, klub Arhan, pada Februari lalu di Liga Thailand berawal dari lemparan jauh yang berbuah gol tunggal. Kendati demikian, lemparan jauh tak mesti jadi satu-satunya kekuatan Timnas Indonesia. Variasi skema dalam usaha mencetak gol dapat lebih efektif dan terhindar dari taktik yang terbaca oleh lawan.</p>
Keyword	indonesia
Link	<a href="https://www.cnnindonesia.com/olahraga/20250326144446-142-1213337/lemparan-maut-pratama-arhan-yang-hilang-di-timnas-indonesia">https://www.cnnindonesia.com/olahraga/20250326144446-142-1213337/lemparan-maut-pratama-arhan-yang-hilang-di-timnas-indonesia</a>
Publish Date	2025-03-27 03:50:00
Source	cnnindonesia.com
Title	Lemparan-Maut-Pratama-Arhan-Yang-Hilang-Di-Timnas-Indonesia

## B. Preprocessing

Sebelum data diolah, tahapan preprocessing harus dilakukan. Pada titik ini, dokumen dibagi menjadi bagian yang lebih kecil, yang terdiri dari kalimat, bab, sub-bab, paragraf, dan akhirnya kata. Angka dihapus, huruf kapital diubah menjadi huruf kecil, dan karakter khusus lainnya dibersihkan selama proses ini.

Tahap ini memerlukan *library* Python NLTK (Natural Language Toolkit) Salah satu *library* dalam bidang pemrosesan bahasa alami (NLP) yang menawarkan berbagai fitur untuk memproses teks, seperti *segmentation* , *Case Folding* , tokenisasi, *stemming*, dan penghapusan *stopword*. Berikut kode program dari tahap menginstal *library* yang ditunjukkan pada tabel 4.5

Tabel 4. 5 Menginstal *Library*

```
!pip install nltk
!pip install scikit-learn
!pip install ROUGE -score
!pip install sastrawi
!pip install sumy
!pip install LexRank
```

Berikut adalah salah satu contoh artikel berita hasil dari proses *scraping* yang ditunjukkan pada tabel 4.6



Tabel 4. 6 Contoh Hasil *Scraping*

Respons dingin ditunjukkan nelayan skala kecil terhadap kebijakan penangkapan ikan terukur (PIT) berbasis kuota, yang resmi diberlakukan Januari 2025 di seluruh Indonesia. Nelayan skala kecil asal Kelurahan Terboyo Wetan, Kecamatan Ginuk, Kota Semarang, Jawa Tengah, mengaku tidak mendapatkan informasi jelas kebijakan. Kebijakan ini jauh dari kehidupan nelayan kecil, yang hanya menggunakan perahu sederhana. Kajian Yayasan Econusa yang bekerja sama dengan Universitas Patimura Ambon. Dalam analisis bertajuk Persepsi Masyarakat Perikanan Kecil Terhadap Penangkapan Ikan Terukur di Kepulauan Aru (WPP 718) dijelaskan bahwa pengetahuan masyarakat, khususnya nelayan skala kecil, mengenai PIT sangat rendah. Ini disebabkan, minimnya akses informasi yang menjangkau masyarakat perikanan kecil di berbagai wilayah. Meskipun, KKP telah menyebarkan informasi tersebut hingga akar rumput. Meskipun, KKP menyebarkan informasi tersebut hingga akar rumput. Respons dingin ditunjukkan nelayan skala kecil terhadap kebijakan penangkapan ikan terukur (PIT) berbasis kuota, yang resmi diberlakukan 1 Januari 2025 di seluruh Indonesia. Mengapa? Agus Isnaini, nelayan skala kecil asal Kelurahan Terboyo Wetan, Kecamatan Ginuk, Kota Semarang, Jawa Tengah, mengaku tidak mendapatkan informasi jelas kebijakan tersebut. Ia tidak begitu paham, aturan yang ditandatangani Menteri Kelautan dan Perikanan (KP), Sakti Wahyu Trenggono, itu. Saya hanya dengar, ada pembatasan hasil tangkapan. Itu saja. Kebijakan ini jauh dari kehidupan nelayan kecil seperti saya, yang hanya menggunakan perahu sederhana sepanjang empat meter,â€ ujarnya, akhir Desember 2024. Nelayan kecil melintas diantara perahu nelayan skala besar di area Pelabuhan Muara Angke, Jakarta Utara, Jakarta. Walaupun KKP telah menyebarkan informasi mengenai kebijakan PIT, upaya penyampaian informasi dinilai belum optimal di tingkat akar rumput. Pengakuan Agus, sejalan dengan kajian Yayasan Econusa yang bekerja sama dengan Universitas Patimura Ambon. Dalam analisis bertajuk â€œPersepsi Masyarakat Perikanan Kecil Terhadap Penangkapan Ikan Terukur di Kepulauan Aru (WPP 718)â€ dijelaskan bahwa pengetahuan masyarakat, khususnya nelayan skala kecil, mengenai PIT sangat rendah. Ini disebabkan,

minimnya akses informasi yang menjangkau masyarakat perikanan kecil di berbagai wilayah. Meskipun, KKP menyebarkan informasi tersebut hingga akar rumput. Survei menunjukkan, 70 persen nelayan tidak setuju dengan kebijakan PIT. Nelayan berpendapat, kebijakan tersebut lebih menguntungkan korporasi perikanan skala besar, sedangkan nelayan kecil kian terpinggirkan. Pelibatan nelayan skala kecil dalam penyusunan kebijakan, penting untuk membangun pemahaman dan perubahan persepsi, serta memastikan implementasi kebijakan berjalan lebih efektif,â€ tulis laporan yang dilakukan di Desa Samang dan Desa Benjina, Aru Tengah, Kepulauan Aru, Provinsi Maluku, tahun 2023. Pendekatan ekosistem perikanan berbasis wilayah, menjadi solusi yang direkomendasikan. Alasannya, pendekatan ini menekankan pentingnya co-management atau pengelolaan bersama antara pemerintah dan pemangku kepentingan lokal. Pemerintah perlu memperluas akses informasi, dengan menysasar nelayan kecil melalui pendekatan lebih efektif, seperti sosialisasi langsung, pendampingan, dan penyediaan materi edukasi yang mudah dipahami,â€ jelas laporan tersebut. Pelibatan nelayan skala kecil dalam penyusunan kebijakan dianggap penting untuk membangun pemahaman dan perubahan persepsi. Foto: Falahi Mubarak/Mongabay Indonesia

#### Perlindungan Nelayan Kecil

Masyithah Aulia Adhiem, dari Pusat Penelitian Badan Keahlian DPR RI, meyoroti pentingnya peraturan pelaksana yang belum disiapkan memadai. Peraturan Pemerintah (PP) No.11/2023 tentang Penangkapan Ikan Terukur, memerlukan waktu untuk diterapkan maksimal.

Perlindungan nelayan lokal dan kecil harus dilakukan, sekaligus mendukung peningkatan Penerimaan Negara Bukan Pajak (PNBP) dari sektor perikanan,â€ jelas dia, dikutip dari kajiannya tentang â€œTantangan Penerapan Kebijakan Penangkapan Ikan Terukur,â€ tahun 2023.

Kebijakan PIT dengan sistem kuota memang bertujuan menjaga keberlanjutan sumber daya laut. Namun, ada kekhawatiran nelayan lokal terpinggirkan akibat persaingan dengan pemodal asing yang diizinkan dalam sistem ini.

â€œPastikan, tidak menekan posisi nelayan lokal, yang merupakan tulang punggung perikanan Indonesia,â€ tegasnya. DPR melalui Komisi IV, diharapkan berperan aktif mendesak pemerintah, segera menyusun peraturan pelaksana sebagai pedoman implementasi PIT.

â€œPengawasan yang ketat, akan memastikan nelayan lokal terlindungi dan ekosistem laut terjaga,â€ jelasnya.

Kolonel Laut (S) Taufik Wijoyoko, Kabid Strahan Kemenko Polhukam, melalui kertas karya ilmiah perorangan bertajuk â€œOptimalisasi Implementasi Kebijakan Penangkapan Ikan Terukur Guna Mewujudkan Laut Sehat Indonesia Sejahteraâ€ menjelaskan bahwa untuk mencapai tujuan laut yang sehat dan Indonesia sejahtera melalui kebijakan PIT, masih diperlukan berbagai evaluasi.

â€œEvaluasi bertujuan agar implementasi kebijakan tepat guna. Juga, mengakomodasi semua kepentingan serta permasalahan yang muncul, serta memastikan proses pelaksanaan berjalan efektif dan optimal,â€ paparnya. Nelayan kecil melakukan pengecatan perahunya yang rusak. Sebagian besar nelayan skala kecil mengaku minim informasi tentang kebijakan PIT yang berbasis kuota. Foto: Falahi Mubarak/Mongabay Indonesia

Menteri KP Sakti Wahyu Trenggono, seperti diberitakan Mongabay sebelumnya, menyatakan bahwa PIT merupakan satu dari lima program prioritas ekonomi biru yang diusung kementeriannya. Kebijakan itu dibuat untuk menjaga keberlanjutan ekosistem laut, meningkatkan mutu dan daya saing produk perikanan, meningkatkan kesejahteraan masyarakat, hingga mendorong pertumbuhan ekonomi nasional yang berkualitas.

Tentang perizinan, Pelaksana Tugas (Plt) Direktur Jenderal Perikanan Tangkap KKP Agus Suherman meminta para pelaku usaha untuk melaporkan data sesuai kondisi sebenarnya. Setiap data yang dilaporkan secara mandiri oleh pelaku usaha, akan digunakan sebagai bahan pertimbangan dalam menetapkan keputusan penting.

â€œDi antaranya mengenai pengurangan atau pencabutan alokasi usaha dalam SIUP tanpa permohonan,â€ terangnya belum lama ini di Jakarta.

Pada 3 Januari 2025, Doni Ismanto Darwin, Staf Khusus Menteri Kelautan dan Perikanan Bidang Komunikasi Publik memberikan respon atas berita ini.

Dia menjelaskan, Å status Kebijakan penangkapan ikan terukur. Pertama, kebijakan PIT saat ini masih tahap transisi, belum sepenuhnya berlaku secara operasional di lapangan. Kedua, Surat Edaran Menteri Kelautan dan Perikanan Nomor B.2403/MEN-KP/XII/2024 mengatur persiapan dan panduan pelaksanaan kebijakan ini untuk semua pihak terkait. *Term* asuk pemerintah daerah, pelaku usaha, dan otoritas pengawasan.

Kementerian Kelautan dan Perikanan (KKP), katanya, Å berkomitmen menjaga keseimbangan antara kelestarian sumber daya laut dan kesejahteraan nelayan kecil. Dalam kebijakan PIT, kuota penangkapan ikan terancang untuk memastikan alokasi yang adil, dengan memperhatikan kebutuhan nelayan lokal dan tradisional.

â€œKami terus mengupayakan dialog dan sosialisasi dengan semua stakeholders perikanan tangkap untuk memastikan pemahaman yang jelas mengenai manfaat dan mekanisme kebijakan ini jika nanti berlaku penuh,â€ katanya.

Menteri, kata Doni, sudah menginstruksikan kepada seluruh jajaran, *term* asuk mitra di pemerintah daerah, untuk pendampingan intensif kepada nelayan agar siap kalau kebijakan berlaku penuh.

KKP, katanya, Å terus memantau dan mengakomodasi masukan dari seluruh pihak untuk memastikan kebijakan ini berjalan efektif, adil, dan inklusif saat berlaku penuh.

Proses selanjutnya adalah preprocessing data, dalam penelitian ini dilakukan dalam lima langkah berurutan:

1. *Segmentation*

Pada tahap ini kalimat dipisahkan dengan tanda baca seperti titik (.), seru (!), dan tanya (?). Kode program dari *segmentation* ditunjukkan pada tabel dibawah 4.7

Tabel 4. 7 Kode program *segmentation*

```
# Segmentation
def segment_text(text_list):
    all_sentences = []
    for text in text_list:
        all_sentences.extend(sent_tokenize(text))
    return all_sentences

sentences = segment_text(cleaned_text)
print("Segmented Sentences:", sentences)
```

Berikut adalah hasil dari proses *segmentation* yang ditunjukkan pada tabel 4.8:

Tabel 4. 8 Hasil *Segmentation*

```
Segmented Sentences: ['Respons dingin ditunjukkan nelayan skala kecil terhadap kebijakan penangkapan ikan terukur (PIT) berbasis kuota, yang resmi diberlakukan Januari 2025 di seluruh Indonesia.', 'Nelayan skala kecil asal Kelurahan Terboyo Wetan, Kecamatan Ginuk, Kota Semarang, Jawa Tengah, mengaku tidak mendapatkan informasi jelas kebijakan Kebijakan ini jauh dari kehidupan nelayan kecil, yang hanya menggunakan perahu sederhana.'...]
```

## 2. *Case Folding*

Tahap *Case Folding* merupakan proses mengubah teks menjadi teks kecil dengan mengganti semua huruf kapital menjadi huruf kecil. Berikut adalah kode program dari tahap *Case Folding* yang ditunjukkan pada tabel 4.9

Tabel 4. 9 Kode program case folding

```
# Case Folding
def casefold_text(sentences):
    return [sentence.lower() for sentence in
sentences]

casefolded_sentences = casefold_text(sentences)
print("Casefolded Sentences:",
casefolded_sentences)
```

Setelah kode program dijalankan, didapatkan hasil seperti yang ada pada tabel 4.10

Tabel 4. 10 Hasil *Case Folding*

```
Casefolded Sentences: ['respons dingin ditunjukkan nelayan
skala kecil terhadap kebijakan penangkapan ikan terukur (pit)
berbasis kuota, yang resmi diberlakukan januari 2025 di
seluruh indonesia.', 'nelayan skala kecil asal kelurahan terboyo
wetan, kecamatan ginuk, kota semarang, jawa tengah, mengaku
tidak mendapatkan informasi jelas kebijakan kebijakan ini jauh
dari kehidupan nelayan kecil, yang hanya menggunakan
perahu sederhana.'...]
```

### 3. Tokenization

Merupakan fase di mana teks dibagi menjadi bagian yang lebih kecil, seperti kata, frasa, simbol, dan elemen penting lainnya. Berikut adalah kode program dari tahapan *tokenization* yang ada pada tabel 4.11

Tabel 4. 11 Kode Program *tokenization*

```
# Tokenization
def tokenize_sentences(sentences):
    return [word_tokenize(sentence) for
sentence in sentences]

tokenized_sentences =
tokenize_sentences(casefolded_sentences)
print("Tokenized Sentences:",
tokenized_sentences)
```

Setelah kode program *tokenization* dijalankan, didapatkan hasil seperti yang ada pada tabel 4.12

Tabel 4. 12 Hasil Tokenization

```
Tokenized Sentences: [['respons', 'dingin', 'ditunjukkan',
'nelayan', 'skala', 'kecil', 'terhadap', 'kebijakan',
'penangkapan', 'ikan', 'terukur', '(', 'pit', ')', 'berbasis', 'kuota',
'', 'yang', 'resmi', 'diberlakukan', 'januari', '2025', 'di',
'seluruh', 'indonesia', '.'], ['nelayan', 'skala', 'kecil', 'asal',
'kelurahan', 'terboyo', 'wetan', '', 'kecamatan', 'ginuk', '',
'kota', 'semarang', '', 'jawa', 'tengah', '', 'mengaku', 'tidak',
'mendapatkan', 'informasi', 'jelas', 'kebijakan', 'kebijakan',
'ini', 'jauh', 'dari', 'kehidupan', 'nelayan', 'kecil', '', 'yang',
'hanya', 'menggunakan', 'perahu', 'sederhana', '.']...]
```

#### 4. *Stemming*

Merupakan metode untuk mengubah kata menjadi bentuk akarnya. Tujuannya adalah untuk menyederhanakan kata menjadi bentuk akarnya atau bentuk dasarnya. Berikut adalah kode program untuk menjalankan proses *stemming* yang ada pada tabel 4.13

Tabel 4. 13 Kode Program *Stemming*

```
factory = StemmerFactory()
stemmer = factory.create_stemmer()

def stem_sentences(sentences):
    return [stemmer.stem(sentence) for
sentence in sentences]

stemmed_sentences =
stem_sentences(casefolded_sentences)
print("Stemmed Sentences:",
stemmed_sentences)
```

Setelah kode program *stemming* dijalankan, didapatkan hasil seperti yang ada pada tabel 4.14



Tabel 4. 14 Hasil *Stemming*

Stemmed Sentences: ['respons dingin tunjuk nelayan skala kecil hadap bijak tangkap ikan ukur pit bas kuota yang resmi laku januari 2025 di seluruh indonesia', 'nelayan skala kecil asal lurah terboyo wetan camat ginuk kota semarang jawa tengah aku tidak dapat informasi jelas bijak bijak ini jauh dari hidup nelayan kecil yang hanya guna perahu sederhana'...]

## 5. *Stopwords* Removal

Metode untuk menyingkirkan kata-kata dalam teks yang dianggap tidak penting atau tidak relevan.

```
stop_words = set(stopwords.words('indonesian'))

def remove_stopwords(sentences):
    return [[word for word in sentence if word not in
stop_words] for sentence in sentences]

filtered_sentences = remove_stopwords(tokenized_sentences)
print("Filtered Sentences:", filtered_sentences)
```

Tabel 4.15 menunjukkan hasil dari proses *stopwords* removal

Tabel 4. 15 Hasil *Stopwords* Removal

Filtered Sentences: [['respons', 'dingin', 'nelayan', 'skala', 'kebijakan', 'penangkapan', 'ikan', 'terukur', '(', 'pit', ')', 'berbasis', 'kuota', ',', 'resmi', 'diberlakukan', 'januari', '2025', 'indonesia', '.'], ['nelayan', 'skala', 'kelurahan', 'terboyo', 'wetan', ',', 'kecamatan', 'ginuk', ',', 'kota', 'semarang', ',', 'jawa', ',', 'mengaku', 'informasi', 'kebijakan', 'kebijakan', 'kehidupan', 'nelayan', ',', 'perahu', 'sederhana', '!']...]

## C. Transformasi

### 1. *Term Frequency-Inverse Document Frequency*

Pada tahapan ini dilakukan pembobotan kata dengan menggunakan *library sklearn* yaitu *TfidfVectorizer*. Berikut adalah kode program proses *TF-IDF* yang ditunjukkan pada tabel 4.16

Tabel 4. 16 Kode program *TF-IDF*

```
processed_texts = [' '.join(sentence) for sentence in
filtered_sentences]

vectorizer = TfidfVectorizer()
tfidf_matrix = vectorizer.fit_transform(processed_texts)

tfidf_df = pd.DataFrame(tfidf_matrix.toarray(),
columns=vectorizer.get_feature_names_out())

print("TF-IDF Matrix (preview):")
print(tfidf_df.head())
```

Berikut adalah hasil dari proses *TF-IDF* yang ditunjukkan pada tabel 4.17

Tabel 4. 17 Hasil TF-IDF

Dokumen 0, Kata '2025': $TF-IDF = 0.1777$
Dokumen 0, Kata 'berbasis': $TF-IDF = 0.2444$
Dokumen 0, Kata 'diberlakukan': $TF-IDF = 0.3148$
Dokumen 0, Kata 'dingin': $TF-IDF = 0.3282$
...
Dokumen 1233, Kata 'varian': $TF-IDF = 0.122$
Dokumen 1233, Kata 'warna': $TF-IDF = 0.1220$
Dokumen 1233, Kata 'warnamitsubishi': $TF-IDF = 0.0696$
Dokumen 1233, Kata 'white': $TF-IDF = 0.0696$
Dokumen 1233, Kata 'xpander': $TF-IDF = 0.4882$

Contoh penelitian ini menggunakan tiga dokumen pendek dari artikel berita. Tujuannya adalah untuk melakukan perhitungan manualisasi pada pembobotan TFIDF.

1. Nelayan kecil tidak mendapatkan informasi jelas kebijakan PIT. (D1)
2. Kebijakan PIT belum disosialisasikan optimal di akar rumput. (D2)
3. Nelayan kecil merasa kebijakan PIT hanya menguntungkan korporasi besar. (D3)

Kemudian hitung frekuensi pada 5 kata utama yang muncul.

Tabel 4. 18 Frekuensi kata utama

Kata	D1	D2	D3
Nelayan	1	0	1
Kecil	1	0	1
Kebijakan	1	1	1
PIT	1	1	1
Korporasi	0	0	1

Selanjutnya perhitungan  $TF$ , untuk D1 semua kata muncul 1 kali ["nelayan", "kecil", "tidak", "mendapatkan", "informasi", "jelas", "kebijakan", "PIT"]. Sehingga untuk nilai  $max(f)=1$ .

$$tf = 0,5 + 0,5 \times \frac{1}{1} = 1$$

Jadi nilai TF dari D1 adalah 1. Sedangkan untuk D2 dan D3 hasilnya juga sama, karena setiap kata dalam dokumen tersebut hanya muncul sekali. Tahap berikutnya adalah perhitungan  $IDF$ . Berikut adalah tabel yang menjelaskan perhitungan  $IDF$  dengan D3.

Tabel 4. 19 Perhitungan IDF

Kata	df <sub>t</sub> (dokumen yg mengandung kata)	IDF
Nelayan	2	$\log(3/2)=0.176$

Kecil	2	$\log(3/2)=0.176$
Kebijakan	3	$\log(3/3)=0$
PIT	3	$\log(3/3)=0$
Korporasi	1	$\log(3/1)=0.477$

Langkah terakhir adalah menghitung TF-IDF. Berikut adalah tabel perhitungan TF-IDF.

Tabel 4. 20 Perhitungan TF-IDF

Kata	TF			IDF	TF-IDF		
	D1	D2	D3		D1	D2	D3
Nelayan	1	0	1	0.176	0.176	0	0.176
Kecil	1	0	1	0.176	0.176	0	0.176
Kebijakan	1	1	1	0	0	0	0
PIT	1	1	1	0	0	0	0
Korporasi	0	0	1	0.477	0	0	0.477

Sehingga jika di tuliskan dalam bentuk array, maka akan terlihat seperti dibawah.

```
array([
    [0.176, 0.000, 0.000],
    [0.176, 0.000, 0.176],
    [0.000, 0.000, 0.000],
    [0.000, 0.000, 0.000],
    [0.000, 0.000, 0.477],
])
```

## 2. Cosine Similarity

Cosine similarity adalah teknik yang umum digunakan dalam peringkasan teks otomatis untuk mengukur seberapa mirip dua teks atau kalimat satu sama lain. Ini juga digunakan untuk menentukan seberapa relevan sebuah teks dengan teks lain dan juga untuk menentukan kalimat mana yang penting untuk dimasukkan ke dalam ringkasan. Berikut adalah source code dari tahapan cosine similarity yang ditunjukkan pada tabel 4.18

Tabel 4. 21 Kode program cosine similarity

```
cosine_sim = cosine_similarity(tfidf_matrix)
print("Cosine Similarity Matrix:")
print(cosine_sim)
```

Metode ini mengubah teks menjadi representasi vektor. Setelah setiap kata atau frasa dalam teks diberi bobot yang sesuai, vektor ini digunakan untuk melihat seberapa mirip kedua teks. Sehingga diperoleh hasil matriks seperti pada gambar dibawah.

```
Cosine Similarity Matrix:
[[1.         0.18281239 0.         ... 0.         0.00308879 0.01198091]
 [0.18281239 1.         0.         ... 0.         0.         0.         ]
 [0.         0.         1.         ... 0.         0.         0.         ]
 ...
 [0.         0.         0.         ... 1.         0.05867038 0.09482572]
 [0.00308879 0.         0.         ... 0.05867038 1.         0.17281419]
 [0.01198091 0.         0.         ... 0.09482572 0.17281419 1.         ]]
```

Gambar 4. 3 Hasil Cosine Similarity

Sebagai contoh 2 teks berbeda untuk perhitungan manual dari tahap ini sebagai berikut:

T1 = “Respons dingin ditunjukkan nelayan skala kecil terhadap kebijakan penangkapan ikan terukur (PIT) berbasis kuota”

T2 = “Perlindungan nelayan lokal dan kecil harus dilakukan, sekaligus mendukung peningkatan PNBP dari sektor perikanan”

Langkah selanjutnya yaitu bentuk vektor frekuensi T1 dan T2 yang dijelaskan pada tabel.

Kata	Frekuensi T1	Frekuensi T2
respons	1	0
dingin	1	0
ditunjukkan	1	0
nelayan	1	1
skala	1	0
kecil	1	1
terhadap	1	0
kebijakan	1	0
penangkapan	1	0
ikan	1	0
terukur	1	0
berbasis	1	0

kuota	1	0
perlindungan	0	1
lokal	0	1
harus	0	1
dilakukan	0	1
sekaligus	0	1
mendukung	0	1
peningkatan	0	1
pnpb	0	1
dari	0	1
sektor	0	1
perikanan	0	1

Kemudian hitung *dot product* dari T1 dan T2. Pada tabel, hanya kata “nelayan” dan “kecil” yang muncul pada kedua teks.

- Nelayan :  $1 \times 1 = 1$
- Kecil :  $1 \times 1 = 1$
- Sehingga  $A.B = 1 + 1 = 2$

Langkah berikutnya adalah menghitung panjang vektor T1 dan T2.

- $|A| = \sqrt{1^2 + 1^2 + 1^2 + 1^2 + \dots + 1^2}$  (13 kali)
- Sehingga  $|A| = \sqrt{13} = 3.606$



Dari sebelumnya, diketahui bahwa vektor T2 terdiri dari 11 kata yang muncul masing-masing satu kali. Maka perhitungannya akan menjadi seperti dibawah:

- $|B| = \sqrt{1^2 + 1^2 + 1^2 + 1^2 + \dots + 1^2}$  (11 kali)
- Sehingga  $|B| = \sqrt{1^2 + 1^2 + 1^2 + \dots + 1^2} = 3.317$

Langkah terakhir adalah memasukkan semua nilai yang sudah dihitung sebelumnya kedalam rumus.

$$\cos \alpha = \frac{A \cdot B}{|A||B|} = \frac{2}{\sqrt{13} \times \sqrt{11}} = \frac{2}{3.606 \times 3.317} = \frac{2}{11.96} = 0.167$$

Jadi hasil dari perhitungan cosine similarity adalah 0.167.

#### D. Algoritma *LexRank*

##### 1. Peringkat Kalimat (*LexRank*)

Algoritma *LexRank* kemudian memberikan peringkat pada setiap kalimat berdasarkan dua faktor:

1. Kesamaan dengan kalimat yang lain, maksudnya peringkat suatu kalimat lebih tinggi jika ada lebih banyak kalimat yang memiliki kesamaan dengan kalimat lain dalam teks.
2. Peringkat kalimat yang berhubungan, dengan kata lain peringkat suatu kalimat juga dipengaruhi oleh peringkat kalimat lain yang berhubungan dengannya.

Berikut adalah source code dari penggunaan algoritma *LexRank* yang ada pada tabel 4.19

Tabel 4. 22 Kode program algoritma *LexRank*

```
def rank_sentences(text, num_sentences=4):
    parser = PlaintextParser.from_string(text,
    Tokenizer("english"))
    summarizer = LexRank Summarizer()
    summary = summarizer(parser.document, num_sentences)
    return ' '.join(str(sentence) for sentence in summary)

summarized_data = []
for i, article in enumerate(input_text):
    try:
        summary = rank_sentences(article)
    except Exception as e:
        summary = f"[Error saat merangkum: {e}]"
    summarized_data.append({'Artikel_Ke': i + 1,
    'Ringkasan': summary})

print("Contoh hasil ringkasan (5 artikel pertama):")
print(df_summary.head())a
```

Setelah algoritma *LexRank* melakukan tahapan pemeringkatan kalimat, hasilnya seperti yang ditunjukkan pada gambar 4.4

Contoh hasil ringkasan (5 artikel pertama):

Artikel_Ke	Ringkasan
0	1 Kebijakan PIT dengan sistem kuota memang bertu...
1	2 Masyarakat Adat Malalo Tigo Jurai merasa dilan...
2	3 Baca juga: Masyarakat di Rempang, Ada Sebelum ...
3	4 Sebarannya ada di empat desa Sebagai kawasan r...
4	5 Karena itu, politikus Nasdem tersebut memastik...

Gambar 4. 4 Hasil algoritma *LexRank*

Berikut ini adalah perhitungan manual dari algoritma *LexRank*.

1. Langkah awalnya adalah menyiapkan data *cosine similarity* yang sudah dihitung sebelumnya, seperti pada tabel.

A	B
1	0.307
0.307	1

2. Kemudian hitung jumlah *similarity* untuk masing-masing baris.

$$\text{Sim A} = \text{Sim}(A,A) + \text{Sim}(A,B) = 1 + 0.307 = 1.307$$

$$\text{Sim B} = \text{Sim}(B,A) + \text{Sim}(B,B) = 0.307 + 1 = 1.307$$

3. Langkah selanjutnya inisialisasi  $p(A)$  dan  $p(B)$ .

$$p(A) = p(B) = \frac{1}{N} = \frac{1}{2} = 0.5$$

4. Hitung  $p(A)$

$$p(A) = \frac{0.85}{2} + (1 - 0.85) \cdot \left( \frac{\text{Sim}(A,B)}{\sum \text{Sim}(B)} \right) \cdot p(B)$$

$$p(A) = \frac{0.85}{2} + (1 - 0.85) \cdot \left(\frac{0.307}{1.307}\right) \cdot 0.5$$

$$p(A) = 0.425 + 0.15 \cdot (0.1175)$$

$$p(A) = 0.425 + 0.0176 = 0.4426$$

5. Hitung  $p(B)$

$$p(B) = \frac{0.85}{2} + (1 - 0.85) \cdot \left(\frac{0.307}{1.307}\right) \cdot 0.5$$

$$p(B) = 0.425 + 0.15 \cdot (0.1175)$$

$$p(B) = 0.425 + 0.0176 = 0.4426$$

6. Hasil *LexRank* kalimat A = 0.4426 dan kalimat B = 0.4426

Hasil ini adalah hanya iterasi awal saja, untuk selanjutnya membutuhkan beberapa iterasi agar hasilnya konvergen. Berikut adalah contoh hasil dari artikel berita yang sudah melewati semua proses peringkasan

Tabel 4. 23 Hasil *LexRank*

<p>Respons dingin ditunjukkan nelayan skala kecil terhadap kebijakan penangkapan ikan terukur (PIT) berbasis kuota, yang resmi diberlakukan Januari 2025 di seluruh Indonesia. Nelayan skala kecil asal Kelurahan Terboyo Wetan, Kecamatan Ginuk, Kota Semarang, Jawa Tengah, mengaku tidak mendapatkan informasi jelas kebijakan. Kebijakan ini jauh dari</p>
--

kehidupan nelayan kecil, yang hanya menggunakan perahu sederhana. Kajian Yayasan Econusa yang bekerja sama dengan Universitas Patimura Ambon. Dalam analisis bertajuk Persepsi Masyarakat Perikanan Kecil Terhadap Penangkapan Ikan Terukur di Kepulauan Aru (WPP 718) dijelaskan bahwa pengetahuan masyarakat, khususnya nelayan skala kecil, mengenai PIT sangat rendah. Ini disebabkan, minimnya akses informasi yang menjangkau masyarakat perikanan kecil di berbagai wilayah. Meskipun, KKP telah menyebarkan informasi tersebut hingga akar rumput. Meskipun, KKP menyebarkan informasi tersebut hingga akar rumput. Respons dingin ditunjukkan nelayan skala kecil terhadap kebijakan penangkapan ikan terukur (PIT) berbasis kuota, yang resmi diberlakukan 1 Januari 2025 di seluruh Indonesia. Mengapa? Agus Isnaini, nelayan skala kecil asal Kelurahan Terboyo Wetan, Kecamatan Ginuk, Kota Semarang, Jawa Tengah, mengaku tidak mendapatkan informasi jelas kebijakan tersebut. Ia tidak begitu paham, aturan yang ditandatangani Menteri Kelautan dan Perikanan (KP), Sakti Wahyu Trenggono, itu. Saya hanya dengar, ada pembatasan

hasil tangkapan. Itu saja. Kebijakan ini jauh dari kehidupan nelayan kecil seperti saya, yang hanya menggunakan perahu sederhana sepanjang empat meter, ujarinya, akhir Desember 2024. Nelayan kecil melintas diantara perahu nelayan skala besar di area Pelabuhan Muara Angke, Jakarta Utara, Jakarta. Walaupun KKP telah menyebarkan informasi mengenai kebijakan PIT, upaya penyampaian informasi dinilai belum optimal di tingkat akar rumput. Pengakuan Agus, sejalan dengan kajian Yayasan Econusa yang bekerja sama dengan Universitas Patimura Ambon. Dalam analisis bertajuk “Persepsi Masyarakat Perikanan Kecil Terhadap Penangkapan Ikan Terukur di Kepulauan Aru (WPP 718) dijelaskan bahwa pengetahuan masyarakat, khususnya nelayan skala kecil, mengenai PIT sangat rendah. Ini disebabkan, minimnya akses informasi yang menjangkau masyarakat perikanan kecil di berbagai wilayah. Meskipun, KKP menyebarkan informasi tersebut hingga akar rumput. Survei menunjukkan, 70 persen nelayan tidak setuju dengan kebijakan PIT. Nelayan berpendapat, kebijakan tersebut lebih menguntungkan korporasi perikanan

skala besar, sedangkan nelayan kecil kian terpinggirkan. Pelibatan nelayan skala kecil dalam penyusunan kebijakan, penting untuk membangun pemahaman dan perubahan persepsi, serta memastikan implementasi kebijakan berjalan lebih efektif, tulis laporan yang dilakukan di Desa Samang dan Desa Benjina, Aru Tengah, Kepulauan Aru, Provinsi Maluku, tahun 2023. Pendekatan ekosistem perikanan berbasis wilayah, menjadi solusi yang direkomendasikan. Alasannya, pendekatan ini menekankan pentingnya co-management atau pengelolaan bersama antara pemerintah dan pemangku kepentingan lokal. Pemerintah perlu memperluas akses informasi, dengan menyasar nelayan kecil melalui pendekatan lebih efektif, seperti sosialisasi langsung, pendampingan, dan penyediaan materi edukasi yang mudah dipahami,â€ jelas laporan tersebut. Pelibatan nelayan skala kecil dalam penyusunan kebijakan dianggap penting untuk membangun pemahaman dan perubahan persepsi. Perlindungan Nelayan Kecil. Masyithah Aulia Adhiem, dari Pusat Penelitian Badan Keahlian DPR RI, menyoroti pentingnya peraturan pelaksana yang

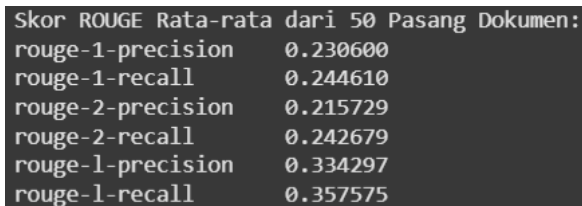
belum disiapkan memadai. Peraturan Pemerintah (PP) No.11/2023 tentang Penangkapan Ikan Terukur, memerlukan waktu untuk diterapkan maksimal.

## E. Evaluasi

### 1. *ROUGE*

Pada tahap ini, hasil pemeringkatan kalimat yang dilakukan oleh proses algoritma *LexRank* akan dievaluasi. Ini akan menunjukkan seberapa baik algoritma *LexRank* dalam menangkap, meringkas, dan menerjemahkan informasi. Untuk mencapai tujuan ini, *ROUGE* melakukan pengukuran tumpang tindih antara konten referensi yang dibuat manusia dan teks yang dihasilkan dengan *LexRank*.

Setelah dilakukan evaluasi, didapatkan hasil seperti pada gambar 4.5



Skor ROUGE Rata-rata dari 50 Pasang Dokumen:	
rouge-1-precision	0.230600
rouge-1-recall	0.244610
rouge-2-precision	0.215729
rouge-2-recall	0.242679
rouge-l-precision	0.334297
rouge-l-recall	0.357575

Gambar 4. 5 Hasil Evaluasi *ROUGE*

Hasil evaluasi *ROUGE* dari gambar 4.5 merupakan nilai rata-rata dari 50 ringkasan *LexRank* dan ringkasan



manual. Tabel diatas menunjukkan nilai *ROUGE -1 Precision* dan *Recall* berada di angka 0.23 dan 0.24, *ROUGE -2 Precision* dan *Recall* di angka 0.21 dan 0.24, *ROUGE -L Precision* dan *Recall* di angka 0.33 dan 0.35.

a. *ROUGE -1*

Perhitungan *ROUGE -1* didasarkan pada perbandingan unigram atau  $n=1$ . Berikut adalah contoh dan rumus perhitungan *ROUGE -1*.

$$ROUGE - 1 recall = \frac{\text{jumlah unigram yang sama}}{\text{total kata di ringkasan manual}}$$

*Rumus ROUGE -1 Recall*

$$ROUGE - 1 precision = \frac{\text{jumlah unigram yang sama}}{\text{total kata di ringkasan sistem}}$$

*Rumus ROUGE -1 Precision*

Langkah pertama adalah identifikasi unigram antara kedua ringkasan. Berikut adalah kata-kata yang muncul pada kedua ringkasan pada tabel 4.20

Tabel 4. 24 Unigram Kedua Ringkasan

No.	Kata	Total Unigram <i>LexRank</i>	Total Unigram Manual
1.	Kebijakan	4	11
2.	Pit	4	13
3.	Sistem	1	1
4.	Kuota	3	1
5.	Bertujuan	1	0
6.	Menjaga	1	1
7.	Keberlanjutan	1	1
8.	Sumber	1	0

No.	Kata	Total Unigram <i>LexRank</i>	Total Unigram Manual
9.	Daya	1	0
10.	Laut	1	1
11.	Nelayan	2	7
12.	Skala	1	2
13	Kecil	2	4
14	Mengaku	1	1
15	Minim	1	0
16	Informasi	1	3
17	Tentang	1	2
18	Yang	3	10
19	Berbasis	1	2
20	Dalam	1	0
21	Penangkapan	1	2
22	Ikan	1	1
23	Terancang	1	0
24	Untuk	2	5
25	Memastikan	1	3
26	Alokasi	1	1
27	Adil	1	1
28	Dengan	1	1
29	Memperhatikan	1	0
30	Kebutuhan	1	0
31	Lokal	1	2
32	Dan	2	11
33	Tradisional	1	0
34	Kkp	1	3
35	Katanya	1	0
36	Terus	1	1
37	Memantau	1	1
38	Mengakomodasi	1	1
39	Masukan	1	0
40	Dari	1	1
41	Seluruh	1	0
41	Pihak	1	2



```
ref_lgrams = ngrams(ref_tokens, 1)
cand_lgrams = ngrams(cand_tokens, 1)

ref_lgram_count =
pd.Series(ref_lgrams).value_counts()
cand_lgram_count =
pd.Series(cand_lgrams).value_counts()

overlap_1 = sum((ref_lgram_count &
cand_lgram_count).values)
Precision_1 = overlap_1 / len(cand_tokens) if
len(cand_tokens) > 0 else 0
Recall_1 = overlap_1 / len(ref_tokens) if
len(ref_tokens) > 0 else 0
```

Skor *ROUGE -1* menunjukkan seberapa banyak unigram (kata tunggal) yang tumpang tindih antara ringkasan otomatis dan ringkasan manual. Nilai ketepatan 0,2306 dan *Recall* 0,2446 menunjukkan bahwa sekitar 23% kata dalam ringkasan otomatis cocok dengan ringkasan manual, dan sekitar 24% kata dari ringkasan manual berhasil ditangkap oleh ringkasan otomatis. Ini menunjukkan bahwa, meskipun tidak sangat akurat, ringkasan otomatis sudah dapat mengumpulkan beberapa informasi penting.

b. *ROUGE -2*

Perhitungan *ROUGE -2* didasarkan pada perbandingan bigram atau  $n=2$ . Berikut adalah contoh dan rumus perhitungan *ROUGE -2*.

$$ROUGE - 2 \text{ recall} = \frac{\text{jumlah bigram yang sama}}{\text{total kata di ringkasan manual}}$$

*Rumus ROUGE -2 Recall*

$$ROUGE - 2 \text{ precision} = \frac{\text{jumlah bigram yang sama}}{\text{total kata di ringkasan sistem}}$$

*ROUGE -2 Precision*

Langkah pertama adalah identifikasi bigram antara kedua ringkasan. Berikut adalah kata-kata yang muncul pada kedua ringkasan yang ditunjukkan pada tabel 4.22

Tabel 4. 26 Bigram Kedua Ringkasan

No.	Bigram yang sama	Keterangan
1.	kebijakan PIT	<b>tidak ada di manual</b>
2.	sistem kuota	ada di manual
3.	menjaga keberlanjutan	ada di manual
4.	sumber daya	ada di manual
5.	nelayan skala	ada di manual
6.	skala kecil	ada di manual
7.	minim informasi	<b>tidak ada di manual</b>

No.	Bigram yang sama	Keterangan
8.	tentang kebijakan	ada di manual
9.	yang berbasis	ada di manual
10.	berbasis kuota	ada di manual
11.	penangkapan ikan	ada di manual
12.	memastikan alokasi	ada di manual
13	alokasi yang	ada di manual
14	yang adil	ada di manual
15	memperhatikan kebutuhan	tidak ada di manual
16	nelayan lokal	ada di manual
17	lokal dan	ada di manual
18	dan tradisional	<b>tidak ada di manual</b>
19	KKP katanya	tidak ada di manual
20	terus memantau	ada di manual
21	memantau dan	ada di manual
22	dan mengakomodasi	ada di manual
23	masukan dari	<b>tidak ada di manual</b>
24	dari seluruh	<b>tidak ada di manual</b>
25	untuk memastikan	ada di manual

No.	Bigram yang sama	Keterangan
26	kebijakan ini	ada di manual
27	ini berjalan	ada di manual
28	berjalan efektif	ada di manual
29	efektif adil	<b>tidak ada di manual</b>
30	adil dan	ada di manual
31	dan inklusif	ada di manual
32	inklusif saat	ada di manual
33	saat berlaku	ada di manual
34	berlaku penuh	<b>tidak ada di manual</b>
35	Penangkapan Ikan	ada di manual
36	Ikan Terukur	ada di manual
37	Terukur berbasis	ada di manual
38	Mengakomodasi	ada di manual
39	basis kuota	ada di manual
40	akses informasi	ada di manual
41	terbatasnya akses	ada di manual
41	kecil tentang	ada di manual

Langkah kedua adalah menghitung jumlah bigram yang sama.

- Jumlah bigram yang sama = 30

Langkah selanjutnya adalah menghitung total kata dan bigram pada ringkasan manual.

- Jumlah kata pada ringkasan manual = 219
- Jumlah bigram pada ringkasan manual = 218

Selanjutnya, hitung jumlah total kata dan bigram pada ringkasan *LexRank*

- Jumlah kata pada ringkasan *LexRank* = 83
- Jumlah bigram pada ringkasan *LexRank* = 82

Langkah terakhir, hitung menggunakan rumus *ROUGE - 2 Recall* dan *Precision*.

- $ROUGE - 2 Recall = \frac{30}{218} = 0,1376$
- $ROUGE - 2 Precision = \frac{30}{82} = 0,3658$



Berikut adalah kode program untuk menghitung *ROUGE -2 Recall* dan *Precision* pada tabel 4.23

Tabel 4. 27 Kode program *ROUGE -2*

```
ref_2grams = ngrams(ref_tokens, 2)
cand_2grams = ngrams(cand_tokens, 2)

ref_2gram_count =
pd.Series(ref_2grams).value_counts()
cand_2gram_count =
pd.Series(cand_2grams).value_counts()

overlap_2 = sum((ref_2gram_count &
cand_2gram_count).values)
Precision_2 = overlap_2 / len(cand_tokens)
if len(cand_tokens) > 0 else 0
Recall_2 = overlap_2 / len(ref_tokens) if
len(ref_tokens) > 0 else 0
```

Skor *ROUGE -2* mengukur tingkat kesamaan antara dua ringkasan menggunakan bigram, atau dua kata berurutan. Ringkasan otomatis menunjukkan struktur kalimat atau hubungan antar kata sedikit lebih rendah daripada *ROUGE -1*, dengan *Precision* 0.2157 dan *Recall* 0.2427. Ini adalah hasil yang wajar karena bigram lebih peka terhadap urutan kata daripada *ROUGE -1*. Ini menunjukkan bahwa susunan kalimat dalam ringkasan

otomatis belum terlalu mirip dengan ringkasan manual, meskipun ada beberapa kata yang sama.

c. *ROUGE -L*

Perhitungan *ROUGE -1* didasarkan pada seberapa panjang urutan kata berurutan yang cocok antara dua ringkasan. Berikut adalah contoh perhitungan dan rumus *ROUGE -L*.

$$ROUGE - L recall = \frac{LCS(sistem, manual)}{total\ kata\ di\ ringkasan\ manual}$$

*ROUGE -L Recall*

$$ROUGE - L precision = \frac{LCS(sistem, manual)}{total\ kata\ di\ ringkasan\ sistem}$$

*Rumus ROUGE -L Precision*

Langkah pertama adalah identifikasi kata terpanjang yang cocok antara kedua ringkasan. Berikut adalah kata-kata yang cocok pada kedua ringkasan.

Tabel 4. 28 LCS (Kata Terpanjang yang Cocok)

No.	Longest Common Subsequence
1.	kebijakan PIT yang berbasis kuota
2.	informasi tentang kebijakan PIT
3.	untuk memastikan alokasi yang adil
4.	KKP terus memantau dan mengakomodasi masukan dari seluruh pihak untuk memastikan kebijakan ini berjalan efektif adil dan inklusif saat berlaku

Langkah selanjutnya hitung menggunakan rumus *ROUGE -L Recall* dan *Precision* .

- $ROUGE -L Recall = \frac{4}{219} = 0,0182$
- $ROUGE -L Precision = \frac{4}{83} = 0,0481$

Dibawah ini adalah kode program dari *ROUGE -L Recall* dan *Precision* .

Tabel 4. 29 Kode program *ROUGE -L*

```
def lcs(X, Y):
    m = len(X)
    n = len(Y)
    L = [[0] * (n + 1) for _ in range(m + 1)]
    for i in range(m + 1):
        for j in range(n + 1):
            if i == 0 or j == 0:
                L[i][j] = 0
            elif X[i-1] == Y[j-1]:
                L[i][j] = L[i-1][j-1] + 1
            else:
                L[i][j] = max(L[i-1][j], L[i][j-1])
    return L[m][n]

lcs_length = lcs(ref_tokens, cand_tokens)
Precision_l = lcs_length / len(cand_tokens) if
len(cand_tokens) > 0 else 0
Recall_l = lcs_length / len(ref_tokens) if
len(ref_tokens) > 0 else 0
```

Skor *ROUGE -L*, yang memperhitungkan *Longest Common Subsequence (LCS)*, menunjukkan seberapa panjang urutan kata berurutan yang cocok antara dua

ringkasan. Skor *Precision ROUGE -L* sebesar 0.3343 dan skor *Recall* sebesar 0.3576 adalah yang tertinggi dari ketiga metrik, menunjukkan bahwa *LexRank* cukup baik dalam membuat kalimat atau urutan informasi yang linear dan relevan terhadap ringkasan manual.

## BAB V

### KESIMPULAN DAN SARAN

#### A. Kesimpulan

Berdasarkan hasil dari penelitian yang dilakukan, dapat disimpulkan bahwa:

1. Penelitian ini berhasil mengumpulkan 50 data artikel dari berbagai portal berita di Indonesia. Seluruh artikel yang diperoleh tersebut memiliki atribut 7 variabel, yaitu *author*, *content*, *keyword*, *link*, *publish date*, *source*, *title*. Dataset kemudian melalui tahapan *preprocessing*, pembobotan kata dengan *TF-IDF*, *cosine similarity*, pemeringkatan kalimat menggunakan algoritma *LexRank*, dan yang terakhir adalah tahap evaluasi menggunakan *ROUGE -1*, *ROUGE -2* dan *ROUGE -L*
2. Secara keseluruhan, hasil evaluasi menunjukkan bahwa ringkasan otomatis berbasis *LexRank* mampu mengumpulkan beberapa informasi penting. Tingkat kesesuaian unigram (*ROUGE -1*) adalah 23–24% dan bigram (*ROUGE -2*) adalah 21–24%, menunjukkan bahwa struktur detail kalimat masih perlu diperbaiki meskipun beberapa kata dan frasa sudah sesuai. Di sisi lain, skor *ROUGE -L*

lebih tinggi (*Precision* ~33% dan *Recall* ~36%), yang menunjukkan bahwa urutan informasi dan konstruksi kalimat *LexRank* menyerupai alur logis ringkasan manual. Namun, masih ada ruang untuk peningkatan pada nilai-nilai ini, terutama dalam hal meningkatkan variasi kata dan mempertahankan kesinambungan bigram. Dengan demikian, ringkasan otomatis dapat semakin mendekati kualitas ringkasan manual.

#### B. Saran

Penulis menyarankan beberapa hal berikut berdasarkan penelitian yang dilakukan:

1. Mengumpulkan lebih banyak data artikel dari berbagai sumber dan topik untuk meningkatkan kinerja algoritma peringkasan teks. Dataset yang lebih besar dan beragam akan membantu algoritma mengenali pola informasi yang lebih luas, meningkatkan akurasi, dan meningkatkan kemampuan model untuk generalisasi terhadap berbagai jenis teks.
2. Sangat disarankan untuk menggabungkan teknik ekstraktif seperti *LexRank* dengan teknik abstraksi berbasis deep learning (seperti T5, BART, atau

mT5). Dibandingkan dengan metode tunggal, pendekatan hybrid ini dapat menghasilkan ringkasan yang lebih informatif dan alami karena mampu menyusun kembali data dengan kalimat yang lebih ringkas dan koheren.

## DAFTAR PUSTAKA

- Addiga, A., & Bagui, S. (2022). Sentiment Analysis on Twitter Data Using Term Frequency-Inverse Document Frequency. *Journal of Computer and Communications*, 10(08), 117–128.  
<https://doi.org/10.4236/jcc.2022.108008>
- Baita, A., & Cahyono, N. (2021). Analisis Sentimen Mengenai Vaksin Sinovac Menggunakan Algoritma Support Vector Machine (Svm) Dan K-Nearest Neighbor (Knn). *Infos*, 4(2), 42–42.
- Barbella, M., & Tortora, G. (n.d.). *ROUGE metric evaluation for Text Summarization techniques*.  
<https://ssrn.com/abstract=4120317>
- Chauhan, T., Rawat, S., Malik, S., & Singh, P. (2021). Supervised and Unsupervised Machine Learning based Review on Diabetes Care. *2021 7th International Conference on Advanced Computing and Communication Systems, ICACCS 2021*, 581–585.  
<https://doi.org/10.1109/ICACCS51430.2021.9442021>
- Diki Hendriyanto, M., Ridha, A. A., & Enri, U. (2022). Analisis Sentimen Ulasan Aplikasi Mola Pada Google Play Store Menggunakan Algoritma Support Vector Machine Sentiment Analysis of Mola Application Reviews on Google Play Store Using Support Vector Machine Algorithm. *Journal of Information Technology and Computer Science (INTECOMS)*, 5(1), 1–7.
- Dimas, F., Al-Hafiidh, F., Rozi, I., & Kusumaning, P. (n.d.). *JIP (Jurnal Informatika Polinema) PERINGKASAN TEKS*



*OTOMATIS PADA PORTAL BERITA OLAHRAGA  
MENGUNAKAN METODE MAXIMUM MARGINAL  
RELEVANCE.*

- Fabrizio Romano, & Heinrich Kruger. (2021). *Learn\_Python\_Programming*.
- Fadillah Rihardi, A., Agustian, S., Pandu Cynthia, E., Informatika, T., Sains dan Teknologi, F., & Sultan Syarif Kasim, U. (n.d.). *Peringkasan Teks Otomatis Menggunakan Metode Latent Dirichlet Allocation (LDA) (Rihardi, AF) Peringkasan Teks Otomatis Menggunakan Metode Latent Dirichlet Allocation (LDA) Automatic Text Summarization using Latent Dirichlet Allocation (LDA)*.
- Fauzi, A. (2022). Bulletin of Data Science Penerapan Algoritma Text Mining dan Lexrank dalam Meringkas Teks Secara Otomatis. *Media Online*, 1(2), 65–72. <https://ejurnal.seminar-id.com/index.php/bulletinds>
- Ganesan, K. (2018). *ROUGE 2.0: Updated and Improved Measures for Evaluation of Summarization Tasks*. <http://arxiv.org/abs/1803.01937>
- Halimah, Surya Agustian, & Siti Ramadhani. (2022a). Peringkasan teks otomatis (automated text summarization) pada artikel berbahasa indonesia menggunakan algoritma lexrank. *Jurnal CoSciTech (Computer Science and Information Technology)*, 3(3), 371–381. <https://doi.org/10.37859/coscitech.v3i3.4300>
- Halimah, Surya Agustian, & Siti Ramadhani. (2022b). Peringkasan teks otomatis (automated text summarization) pada artikel berbahasa indonesia menggunakan algoritma lexrank. *Jurnal CoSciTech*

- (*Computer Science and Information Technology*), 3(3), 371–381. <https://doi.org/10.37859/coscitech.v3i3.4300>
- Halimah, Surya Agustian, & Siti Ramadhani. (2022c). Peringkasan teks otomatis (automated text summarization) pada artikel berbahasa indonesia menggunakan algoritma lexrang. *Jurnal CoSciTech (Computer Science and Information Technology)*, 3(3), 371–381. <https://doi.org/10.37859/coscitech.v3i3.4300>
- Hrytsiv, N. M. (2023). N-GRAMS IN MEANING TRANSLATION MODEL OF FICTION: COGNITIVE AND SEMANTIC INTERSECTION. *Transcarpathian Philological Studies*, 1(27), 170–178. <https://doi.org/10.32782/tps2663-4880/2022.27.1.29>
- Idhafi, Z., Agustian, S., Yanto, F., & Safaat H, N. (2023). Peringkasan teks otomatis pada artikel berbahasa indonesia menggunakan metode maximum marginal relevance. *Jurnal CoSciTech (Computer Science and Information Technology)*, 4(3), 609–618. <https://doi.org/10.37859/coscitech.v4i3.6311>
- Jain, M., & Rastogi, H. (2020). Automatic Text Summarization using Soft-Cosine Similarity and Centrality Measures. *Proceedings of the 4th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2020*, 1021–1028. <https://doi.org/10.1109/ICECA49313.2020.9297583>
- Jang, H., & Kim, W. (2021). Reinforced Abstractive Text Summarization with Semantic Added Reward. *IEEE Access*, 9, 103804–103810. <https://doi.org/10.1109/ACCESS.2021.3097087>

Lin, C.-Y. (2004). *ROUGE : A Package for Automatic Evaluation of Summaries*.

Maulidia, Y., #1, S., Siti, N., & #2, F. (n.d.). *JEPIN (Jurnal Edukasi dan Penelitian Informatika) Peringkasan Teks Otomatis pada Modul Pembelajaran Berbahasa Indonesia Menggunakan Metode Cross Latent Semantic Analysis (CLSA)*. [www.kompas.com](http://www.kompas.com).

Nurjannah, M., & Fitri Astuti, I. (2013). PENERAPAN ALGORITMA TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF) UNTUK TEXT MINING Mahasiswa S1 Program Studi Ilmu Komputer FMIPA Universitas Mulawarman Dosen Program Studi Ilmu Komputer FMIPA Universitas Mulawarman. *Jurnal Informatika Mulawarman*, 8(3), 110–113.

Ranu Anugrah, R., Rekayasa Sistem Komputer, J., & MIPA Universitas Tanjungpura Jl Hadari Nawawi, F. H. (2023). *Coding : Jurnal Komputer dan Aplikasi PENERAPAN COSINE SIMILARITY DAN PEMBOBOTAN TF-IDF UNTUK KLASIFIKASI PENGADUAN MASYARAKAT BERBASIS WEB (Studi Kasus : BAGWASSIDIK DITRESKRIMUM POLDA KALBAR)* (Vol. 11, Issue 01).

Rinandyaswara, R., Sari, Y. A., & Furqon, M. T. (2022). *PEMBENTUKAN DAFTAR STOPWORD MENGGUNAKAN TERM BASED RANDOM SAMPLING PADA ANALISIS SENTIMEN DENGAN METODE NAÏVE BAYES (STUDI KASUS: KULIAH DARING DI MASA PANDEMI)*. 9(4). <https://doi.org/10.25126/jtiik.202294707>

Supriyanto, J., Alita, D., & Isnain, A. R. (2023). Penerapan Algoritma K-Nearest Neighbor (K-NN) Untuk Analisis Sentimen Publik Terhadap Pembelajaran Daring. *Jurnal*

- Informatika Dan Rekayasa Perangkat Lunak*, 4(1), 74–80.  
<https://doi.org/10.33365/jatika.v4i1.2468>
- Umam Syaliman, K., & Caltex Riau Yuliska, P. (2022). PERINGKASAN DOKUMEN TEKS OTOMATIS BERDASARKAN SEBUAH KUERI MENGGUNAKAN BIDIRECTIONAL LONG SHORT TERM MEMORY NETWORK AUTOMATIC TEXT DOCUMENT SUMMARY BASED ON A QUERY USING BIDIRECTIONAL LONG SHORT TERM MEMORY NETWORK. *Journal of Information Technology and Computer Science (INTECOMS)*, 5(2).
- Unpingco, J. (2021). Python programming for data analysis. In *Python Programming for Data Analysis*. Springer International Publishing. <https://doi.org/10.1007/978-3-030-68952-0>
- Wenty Dwi Yuniarti. (2019). *Dasar-dasar pemrograman dengan python*. Deepublish.
- Wijaya, J., & Suganda Girsang, A. (2024). Indonesian News Extractive Summarization using Lexrank and YAKE Algorithm. *Optim. Inf. Comput.*, 0–10.  
<https://doi.org/10.19139/soic-2310-5070-1976>
- Wiratmoko, G. (2024). Evaluating the Effectiveness of the LexRank and LSA Algorithm in Automatic Text Summarization for Indonesian Language. In *Syntax Admiration* (Vol. 5, Issue 9).
- Yuliska, Y., & Syaliman, K. U. (2020). Literatur Review Terhadap Metode, Aplikasi dan Dataset Peringkasan Dokumen Teks Otomatis untuk Teks Berbahasa Indonesia. *IT Journal*

*Research and Development*, 5(1), 19–31.  
[https://doi.org/10.25299/itjrd.2020.vol5\(1\).4688](https://doi.org/10.25299/itjrd.2020.vol5(1).4688)

Zakawaly, G., Hayatin, N., Rahmayanti, V., & Nastiti, S. (2023).  
Improvisasi Algoritma Dijkstra Pada Peringkasan Teks  
Otomatis Untuk Artikel Politik. *REPOSITOR*, 5(2), 709–  
716.

## DAFTAR LAMPIRAN

### Lampiran 1 Nilai Rata-rata *ROUGE* dari Uji Data 50, 100 dan 150 Artikel Berita

Uji Data 50 Artikel Berita						
No.	<i>ROUGE -1</i>		<i>ROUGE -2</i>		<i>ROUGE -L</i>	
	<i>Precision</i>	<i>Recall</i>	<i>Precision</i>	<i>Recall</i>	<i>Precision</i>	<i>Recall</i>
1.	0.23	0.24	0.23	0.33	0.33	0.35

Uji Data 100 Artikel Berita						
No.	<i>ROUGE -1</i>		<i>ROUGE -2</i>		<i>ROUGE -L</i>	
	<i>Precision</i>	<i>Recall</i>	<i>Precision</i>	<i>Recall</i>	<i>Precision</i>	<i>Recall</i>
1.	0.23	0.31	0.23	0.33	0.33	0.44

Uji Data 150 Artikel Berita						
No.	<i>ROUGE -1</i>		<i>ROUGE -2</i>		<i>ROUGE -L</i>	
	<i>Precision</i>	<i>Recall</i>	<i>Precision</i>	<i>Recall</i>	<i>Precision</i>	<i>Recall</i>
1.	0.30	0.37	0.37	0.45	0.45	0.55