

**ANALISIS PERBANDINGAN *TEXTRANK* DAN *LONG SHORT TERM MEMORY* DALAM PERINGKASAN TEKS BERITA
BAHASA INGGRIS**

SKRIPSI

**Oleh :
MUHAMMAD FAZA ABDILLAH
NIM. 200605110068**



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2024**

**ANALISIS PERBANDINGAN *TEXTRANK* DAN *LONG SHORT TERM MEMORY* DALAM PERINGKASAN TEKS BERITA
BAHASA INGGRIS**

SKRIPSI

Oleh :
MUHAMMAD FAZA ABDILLAH
NIM. 200605110068

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2024**

HALAMAN PERSETUJUAN

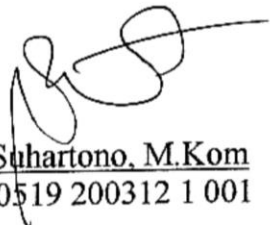
**ANALISIS PERBANDINGAN *TEXTRANK* DAN *LONG SHORT TERM MEMORY* DALAM PERINGKASAN TEKS BERITA
BAHASA INGGRIS**

SKRIPSI

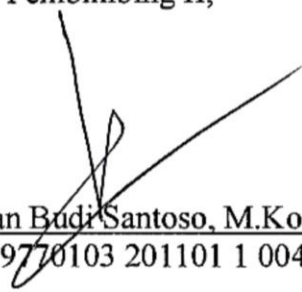
**Oleh :
MUHAMMAD FAZA ABDILLAH
NIM. 200605110068**

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 03 Juni 2024

Pembimbing I,



Prof. Dr. Suhartono, M.Kom
NIP. 19680519 200312 1 001

Pembimbing II,


Dr. Irwan Budi Santoso, M.Kom
NIP. 19770103 201101 1 004

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Chairperson, M.MT, IPM
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

ANALISIS PERBANDINGAN *TEXTRANK* DAN *LONG SHORT TERM MEMORY* DALAM PERINGKASAN TEKS BERITA BAHASA INGGRIS

SKRIPSI

Oleh :
MUHAMMAD FAZA ABDILLAH
NIM. 200605110068

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 03 Juni 2024

Susunan Dewan Penguji

Ketua Penguji : A'la Syauqi, M.Kom
NIP. 19771201 200801 1 077

Anggota Penguji I : Okta Qomaruddin Aziz, M.Kom
NIP. 19911019 201903 1 013


Anggota Penguji II : Prof. Dr. Suhartono, M.Kom
NIP. 19680519 200312 1 001

Anggota Penguji II : Dr. Irwan Budi Santoso, M.Kom
NIP. 19770103 201101 1 004

()
()
()
()

Mengetahui dan Mengesahkan,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Fachrudin Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Muhammad Faza Abdillah
NIM : 200605110068
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika
Judul Skripsi : Analisis Perbandingan *Textrank* dan *Long Short Term Memory* Dalam Peringkasan Teks Berbahasa Inggris.

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 03 Juni 2024
Yang membuat pernyataan,



Muhammad Faza Abdillah
NIM.200605110068

MOTTO

~ Jika kita gagal 1000 kali, maka pastikan kita bangkit 1001 kali ~

*~ Segala sesuatu yang baik, selalu datang di saat terbaiknya. Persis waktunya.
Tidak datang lebih cepat, pun lebih lambat. Itulah kenapa rasa sabar itu harus
disertai keyakinan dan kerja keras ~*

HALAMAN PERSEMBAHAN

Dengan mengucapkan syukur Alhamdulillah dan terima kasih, penulis mempersembahkan skripsi ini untuk orang tua, keluarga, dosen, para sahabat dan semua pihak-pihak yang telah memberikan saran, motivasi, doa, serta dukungan secara aktif sehingga penelitian ini dapat diselesaikan tepat pada waktunya.

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Puji syukur penulis panjatkan kepada Allah SWT yang senantiasa memberikan rahmat serta kesehatan, sehingga penulis mampu menyelesaikan skripsi ini dengan baik. Penulis menyampaikan ucapan terima kasih kepada semua pihak yang pernah terlibat langsung maupun tidak langsung dalam menyelesaikan skripsi ini, bukan hanya karena usaha keras dari penulis sendiri, akan tetapi karena adanya dukungan dari berbagai pihak. Oleh karena itu penulis berterima kasih kepada:

1. Prof. Dr. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Sri Hariani, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Fachrul Kurniawan M.MT., IPM selaku Ketua Prodi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Prof. Dr. Suhartono, M.Kom selaku Dosen Pembimbing 1 yang telah membimbing serta memberikan arahan serta motivasi dalam penulisan skripsi dari awal hingga akhir.
5. Dr. Irwan Budi Santoso, M.Kom selaku Dosen Pembimbing 2 yang telah membimbing serta memberikan arahan serta motivasi dalam penulisan skripsi dari awal hingga akhir.

6. A'la Syauqi, M.Kom selaku penguji I dan Okta Qomaruddin Aziz, M.Kom selaku penguji II yang telah meluangkan waktunya untuk menguji dan dengan sabar memberi arahan dan saran dalam menyelesaikan skripsi ini.
7. Segenap civitas akademik Program Studi Teknik Informatika, dan seluruh dosen yang telah memberikan ilmu serta arahan semasa kuliah.
8. Kedua orang tua penulis, Bapak Sukarnadi dan Ibu Nur Kholis, serta keluarga besar yang telah memberikan banyak dukungan, doa serta selalu menjadi semangat sehingga penulis mampu menyelesaikan masa studi hingga mencapai gelar sarjana.
9. Teman-teman yang telah memberikan bantuan yang sangat besar serta yang menjadi teman diskusi dan teman belajar.

Skripsi yang penulis tulis ini masih sangat jauh dari kata sempurna dan mungkin terdapat kesalahan di dalamnya. Oleh karena itu, penulis sangat mengapresiasi adanya kritik dan saran yang membangun untuk mengembangkan skripsi ini menjadi lebih baik dan lebih bermanfaat untuk diri penulis sendiri dan pembaca pada umumnya.

Malang, 03 Juni 2024

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
ABSTRAK	xiv
ABSTRACT	xv
مستخلص البحث	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Pernyataan Masalah	4
1.3 Tujuan Penelitian	4
1.4 Batasan Masalah	4
1.5 Manfaat Penelitian	4
BAB II STUDI PUSTAKA	5
2.1 Penelitian Terkait	5
2.2 Peringkasan Teks Otomatis	10
2.3 <i>TextRank</i>	11
2.4 <i>Long Short Term Memory</i>	12
BAB III DESAIN DAN IMPLEMENTASI.....	15
3.1 Pengumpulan Data	16
3.2 <i>Preprocessing Data</i>	16
3.2.1 <i>Splitting Sentence</i>	17
3.2.2 <i>Cleaning</i>	18
3.2.3 <i>Case Folding</i>	18
3.2.4 <i>Stopword Removal</i>	19
3.2.5 <i>Tokenizing</i>	19
3.2.6 Hapus <i>Empty List</i>	20
3.3 Implementasi Metode	20
3.3.1 Metode <i>TextRank</i>	20
3.3.1.1 Vektorisasi Menggunakan <i>Global Vector</i>	21
3.3.1.2 <i>Similarity Matrix</i>	22
3.3.1.3 Rumus <i>TextRank</i>	23
3.3.2 Metode <i>Long Short Term Memory</i>	23
3.4 Evaluasi	28
3.5 Skenario Pengujian	30
BAB IV HASIL DAN PEMBAHASAN.....	32

4.1 Tahap Pengujian.....	32
4.2 Hasil Pengujian	33
4.2.1 Pengujian Menggunakan Metode <i>TextRank</i>	33
4.2.2 Pengujian Menggunakan Metode <i>Long Short Term Memory</i>	35
4.3 Pembahasan	43
BAB V KESIMPULAN DAN SARAN.....	57
5.1 Kesimpulan.....	57
5.2 Saran.....	57
DAFTAR PUSTAKA	
LAMPIRAN	

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait	8
Tabel 3.1 Contoh <i>Splitting Sentence</i>	17
Tabel 3.2 Contoh <i>Cleaning</i>	18
Tabel 3.3 Contoh <i>Case Folding</i>	18
Tabel 3.4 Contoh <i>Stopword Removal</i>	19
Tabel 3.5 Contoh <i>Tokenizing</i>	19
Tabel 3.6 Contoh vektorisasi.....	22
Tabel 3.7 Contoh hasil <i>similarity matrix</i>	22
Tabel 3.8 Contoh pembobotan dan vektor probabilitas	25
Tabel 3.9 Perbandingan ringkasan manual dan sistem	29
Tabel 4.1 Hasil pengujian menggunakan <i>TextRank</i>	33
Tabel 4.2 Perbedaan jumlah kata dan kalimat sebelum dan sesudah diringkas	33
Tabel 4.3 Skor evaluasi ROUGE-1 tiap berita	34
Tabel 4.4 Skor evaluasi ROUGE-1 skenario pertama	37
Tabel 4.5 Indeks berita tertinggi pada LSTM skenario pertama	38
Tabel 4.6 Indeks berita terendah pada LSTM skenario pertama	38
Tabel 4.7 Skor evaluasi ROUGE-1 skenario dua.....	39
Tabel 4.8 Indeks berita tertinggi pada LSTM skenario kedua	40
Tabel 4.9 Indeks berita terendah pada LSTM skenario kedua	40
Tabel 4.10 Skor evaluasi ROUGE-1 skenario tiga.....	41
Tabel 4.11 Indeks berita tertinggi pada LSTM skenario ketiga	41
Tabel 4.12 Indeks berita terendah pada LSTM skenario ketiga	42
Tabel 4.13 Perbandingan rata-rata skor pengujian tiga skenario	42
Tabel 4.14 Perbandingan rata-rata skor pengujian <i>TextRank</i> dan LSTM	43
Tabel 4.15 Hasil pengujian Pooja Raundale dan Himanshu Shekhar.....	48
Tabel 4.16 Hasil pengujian <i>Barrios et al.</i>	49
Tabel 4.17 Perbandingan akurasi dengan peneliti terdahulu.....	51

DAFTAR GAMBAR

Gambar 2.1 Struktur jaringan LSTM (Le <i>et al.</i> , 2019).....	13
Gambar 3.1 Diagram metode penelitian	15
Gambar 3.2 Flowchart <i>preprocessing</i>	16
Gambar 3.3 Alur algoritma <i>TextRank</i>	21
Gambar 3.4 Alur algoritma LSTM	24
Gambar 3.5 Arsitektur LSTM	27
Gambar 4.1 Perbandingan Nilai Recall.....	44
Gambar 4.2 Perbandingan Nilai Precision	45
Gambar 4.3 Perbandingan Nilai F1-Score	45
Gambar 4.4 Hasil penelitian Mercan <i>et al.</i>	47
Gambar 4.5 Hasil penelitian Turkey <i>et al.</i>	48
Gambar 4.6 Hasil penelitian Barrios <i>et al.</i>	49

ABSTRAK

Abdillah, Muhammad Faza. 2024. **Analisis Perbandingan *Textrank* Dan *Long Short Term Memory* Dalam Peringkasan Teks Berbahasa Inggris**. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Prof. Dr. Suhartono, M.Kom (II) Dr. Irwan Budi Santoso, M.Kom.

Kata kunci: *Long Short Term Memory, Peringkasan Teks Otomatis, Textrank.*

Penelitian ini bertujuan untuk mengembangkan sistem peringkasan teks otomatis untuk artikel berita, menggunakan dua teknik peringkasan: ekstraktif dengan algoritma *TextRank* dan abstraktif dengan algoritma *Long Short Term Memory* (LSTM). Berita yang mudah ditemukan di berbagai media kini tersebar luas di internet, sehingga pembaca sering kesulitan menemukan artikel yang relevan. Teknologi informasi diharapkan dapat membantu mengatasi tantangan ini dengan mengolah, menganalisis, dan menyajikan informasi yang lebih ringkas dan akurat. Teknik peringkasan dibagi menjadi ekstraktif, yang mempertahankan kalimat asli dari dokumen, dan abstraktif, yang menghasilkan kalimat baru berdasarkan pemahaman konteks dokumen. Hasil penelitian menunjukkan bahwa metode LSTM menghasilkan nilai rata-rata tertinggi untuk *recall*, *precision*, dan *f1-score* dalam skenario dengan 90% data *training* dan 10% data *testing*, masing-masing sebesar 0.462, 0.507, dan 0.480. Semakin banyak data yang digunakan untuk melatih model, semakin akurat hasil ringkasan yang dihasilkan. Namun, terdapat beberapa indeks berita dengan nilai akurasi rendah, menunjukkan bahwa model peringkasan masih dapat disempurnakan. Kesimpulannya, metode LSTM menunjukkan kinerja terbaik dalam peringkasan teks berita berbahasa Inggris, meskipun nilai rata-rata masih di bawah 0.5, sehingga konfigurasi model dan parameter perlu ditingkatkan untuk mencapai akurasi yang lebih tinggi.

ABSTRACT

Abdillah, Muhammad Faza. 2024. **Comparative Analisis of Textrank and Long Short Term Memory in English Text Summarisation**. Essay. Department of Informatics Engineering, Faculty of Science and Technology, Islamic State University of Maulana Malik Ibrahim of Malang. Supervisor: (I) Prof. Dr. Suhartono, M.Kom (II) Dr. Irwan Budi Santoso, M.Kom.

This study aims to develop an automatic text summarization system for news articles using two summarization techniques: extractive summarization with the TextRank algorithm and abstractive summarization with the Long Short Term Memory (LSTM) algorithm. News, easily found in various media, is now widely disseminated on the internet, making it difficult for readers to find relevant articles. Information technology is expected to help overcome this challenge by processing, analyzing, and presenting more concise and accurate information. Summarization techniques are divided into extractive, which retains the original sentences from the document, and abstractive, which generates new sentences based on the document's context. The study results show that the LSTM method achieved the highest average scores for recall, precision, and f1-score in a scenario with 90% training data and 10% testing data, with scores of 0.462, 0.507, and 0.480, respectively. The more data used to train the model, the more accurate the summary produced. However, some news indices had low accuracy scores, indicating that the summarization model can still be improved. In conclusion, the LSTM method demonstrated the best performance in summarizing English news texts, although the average scores were still below 0.5, suggesting that model configuration and parameters need to be enhanced to achieve higher accuracy.

Key words: *Automatic Text Summary, Long Short Term Memory, Textrank.*

مستخلص البحث

عبد الله، محمد فزع. ٢٠٢٤. تحليل مقارن بين *Long Short Term Memory* و *Textrank* في تلخيص النص الإنجليزي. الأطروحة. برنامج دراسة هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف الأول: الأستاذ الدكتور سوهارتونو، املاجستر. المشرف الثاني: الدكتور إروان بودي سانتوسو، املاجستر.

الكلمات الرئيسية: تلخيص النص، تلخيص النص، الذاكرة طويلة المدى

يهدف هذا البحث إلى تطوير نظام لتلخيص النصوص تلقائيًا للمقالات الإخبارية باستخدام تقنيتين للتلخيص: التلخيص الاستخراجي باستخدام خوارزمية *TextRank* والتلخيص التجريدي باستخدام خوارزمية *Long Short Term Memory (LSTM)*. تنتشر الأخبار التي يمكن العثور عليها بسهولة في وسائل الإعلام المختلفة الآن على الإنترنت بشكل واسع، مما يجعل من الصعب على القراء العثور على المقالات ذات الصلة. من المتوقع أن تساعد تكنولوجيا المعلومات في التغلب على هذا التحدي من خلال معالجة وتحليل وتقديم معلومات أكثر دقة واختصارًا. تقنيات التلخيص مقسمة إلى تلخيص استخراجي، والذي يحتفظ بالجمل الأصلية من المستند، وتلخيص تجريدي، والذي يولد جملاً جديدة بناءً على سياق المستند. أظهرت نتائج الدراسة أن طريقة *LSTM* حققت أعلى متوسط للنقاط في الاستدعاء، والدقة، ودرجة *F1* في سيناريو يحتوي على 90% من بيانات التدريب و10% من بيانات الاختبار، حيث كانت الدرجات 0.462، 0.507، و0.480 على التوالي. كلما زادت البيانات المستخدمة لتدريب النموذج، كان الملخص الناتج أكثر دقة. ومع ذلك، كانت هناك بعض المؤشرات الإخبارية التي حصلت على درجات دقة منخفضة، مما يشير إلى أن نموذج التلخيص لا يزال بحاجة إلى تحسين. في الختام، أظهرت طريقة *LSTM* أفضل أداء في تلخيص النصوص الإخبارية الإنجليزية، على الرغم من أن المتوسطات كانت لا تزال أقل من 0.5، مما يشير إلى أن تكوين النموذج والمعايير تحتاج إلى تحسين لتحقيق دقة أعلى.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam Kamus Besar Bahasa Indonesia, berita adalah cerita atau keterangan mengenai kejadian yang sedang hangat. Berita dapat dengan mudah kita temukan di media cetak seperti majalah atau koran, penyiaran radio dan televisi, serta melalui artikel *online* pada internet yang bisa kita akses setiap saat. Seiring dengan perkembangan teknologi informasi yang begitu pesat, banyak artikel berita tersebar luas pada internet dengan jumlah teks yang besar. Karena jumlahnya yang sangat besar, sering kali sulit bagi pembaca dalam memahami artikel yang relevan dengan minat dan kebutuhan mereka (Tuhatussania, 2022).

Pemanfaatan teknologi informasi diharapkan dapat membantu dan memenuhi semua kebutuhan informasi baik untuk pihak luar maupun dalam, sehingga pada akhirnya penggunaan teknologi informasi akan mempermudah dalam mengolah, menganalisa, dan mengkomunikasikan informasi yang relevan guna mengefisiensikan pengelolaan informasi menjadi sebuah berita yang lebih akurat (Robiyanto *et al.*, 2019).

Oleh karena itu, diperlukan peringkasan teks otomatis yang dapat memahami bahasa alami dan memberikan informasi yang relevan dan ringkas. Dua jenis teknik peringkasan yang berbeda diperlukan untuk mewujudkan peringkasan teks otomatis: teknik ekstraktif dan abstraktif. Tanpa mengubah kata sumber, pendekatan ekstraktif berfungsi dengan mengambil setiap kata dalam teks dan mengatur serta

mengkategorikannya. Kalimat akhir biasanya akan sama persis dengan kalimat yang ditemukan dalam materi sumber (Prabowo *et al.*, 2016). Sedangkan untuk pendekatan abstraktif menghasilkan ringkasan seperti manusia yang mengekstrak ide-ide utama dari dokumen yang mereka baca.

Peringkasan teks berdasarkan input memiliki dua jenis, yaitu peringkasan teks *single-document* (dokumen tunggal) dan peringkasan teks *multi-document* (dokumen banyak). Peringkasan teks *single-document* merupakan proses peringkasan atau menghasilkan kutipan singkat dari satu dokumen tunggal, seperti artikel berita, esai, atau laporan, tanpa memerlukan sumber-sumber eksternal. Tujuannya adalah untuk memberikan gambaran cepat dan ringkas tentang isi dokumen tersebut. Sedangkan peringkasan teks *multi-document* merupakan proses peringkasan yang inputnya berasal dari beberapa dokumen teks yang terkait atau relevan dengan topik yang sama. Tujuannya adalah untuk menyajikan informasi inti dari sejumlah dokumen dalam bentuk ringkasan tunggal yang lebih singkat dan lebih mudah dimengerti (Mehr, 2013).

Diharapkan dengan adanya sistem otomatis peringkasan teks ini dapat membantu pembaca dalam efisien waktu saat membaca dan memahami inti informasi yang akurat dari berita yang dibaca. Allah *subhanahu wa ta'ala* berfirman dalam QS. Al-Fussilat ayat 46 yang berbunyi:

مَنْ عَمِلَ صَالِحًا فَلِنَفْسِهِ ۖ وَمَنْ أَسَاءَ فَعَلَيْهَا ۗ وَمَا رَبُّكَ بِظَلَّامٍ لِّلْعَبِيدِ

“Barangsiapa mengerjakan kebajikan maka (pahalanya) untuk dirinya sendiri dan barangsiapa berbuat jahat maka (dosanya) menjadi tanggungan dirinya sendiri.” (QS. Al-Fussilat:46)

Menurut tafsir Ibnu Katsir tentang penjelasan ayat di atas, barangsiapa yang melakukan perbuatan baik akan benar-benar menuai hasil dari perbuatannya, dan barangsiapa yang melakukan perbuatan buruk akan benar-benar menderita akibat dari perbuatannya. Dalam kehidupan sehari-hari manusia akan mendapat kebaikan ketika berbuat baik kepada sesamanya. Oleh karena itu, sistem peringkasan teks ini diharapkan menjadi salah satu kebaikan dalam membantu pembaca memahami inti informasi yang terdapat pada berita.

Peringkasan teks dapat dilakukan secara ekstraktif dan abstraktif. Dalam ekstraktif, kalimat kunci dan objek diekstraksi tanpa memodifikasi objek itu sendiri. Hal ini diperoleh dengan ekstraksi frasa kunci atau kalimat *ad-hoc* yang menjaga agar kalimat tetap utuh. Sementara dalam abstraktif melibatkan parafrase kalimat yang sadar akan konteks setelah memahami bahasanya (Khatri *et al.*, 2018).

Dalam kesempatan ini peneliti hendak membandingkan dua teknik peringkasan tersebut. Untuk teknik ekstraktif peneliti menggunakan algoritma *TextRank*, kemudian untuk teknik abstraktif peneliti menggunakan algoritma *Long Short Term Memory*. *TextRank* adalah sebuah teknik peringkasan berbasis grafik yang merepresentasikan kalimat sebagai *node* serta hubungan antar kalimat sebagai *edge*. Metode ini menggunakan nilai kemiripan (*similarity*) untuk mendapatkan hubungan antar kalimat tersebut (Hernawan *et al.*, 2022). *TextRank* merupakan metode peringkasan teks yang populer karena kinerjanya yang baik. Salah satu kelebihanannya adalah tidak memerlukan data pelatihan, sehingga mudah diterapkan dalam berbagai bahasa. *Long Short Term Memory* (LSTM) merupakan metode peringkasan abstraktif hasil dari modifikasi dari metode *Recurrent Neural Network*

(RNN). Dikarenakan arsitekturnya yang memungkinkannya untuk mengingat dan melupakan data yang akan diproses, LSTM dianggap berkinerja lebih baik daripada RNN (Alfhi Saputra, 2021).

1.2 Pernyataan Masalah

Seberapa baik performa algoritma *TextRank* dan LSTM dalam peringkasan teks berita Bahasa Inggris?

1.3 Tujuan Penelitian

Penelitian ini dibuat untuk membandingkan performa algoritma *TextRank* dan LSTM.

1.4 Batasan Masalah

- a. Penggunaan dataset kumpulan artikel berita berbahasa Inggris dalam penelitian ini.
- b. Performa antara *TextRank* dan LSTM diukur menggunakan ROUGE-1.
- c. Penelitian hanya berfokus pada hasil *precision*, *recall*, dan *f1-score* pada setiap metode yang dipakai.

1.5 Manfaat Penelitian

- a. Manfaat untuk peneliti selanjutnya, menjadi rekomendasi metode peringkasan teks otomatis dengan nilai akurasi tertinggi.
- b. Manfaat untuk pembaca artikel berita, sistem peringkasan teks otomatis dapat membantu para pembaca untuk lebih mudah memahami inti informasi yang ada didalam artikel berita tersebut.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terkait

Penelitian yang diusulkan oleh peneliti yang berjudul “Analisis Perbandingan *TextRank* dan *Long Short Term Memory* Dalam Peringkasan Teks Berita Bahasa Inggris” merupakan penelitian berdasarkan beberapa penelitian terdahulu yang telah dilakukan oleh para peneliti terdahulu dengan topik serupa.

Halimah *et al.* (2022) pernah meneliti penggunaan algoritma *LexRank* untuk meringkas teks otomatis dari artikel-artikel berbahasa Indonesia. Sebuah dataset korpus yang terdiri dari 300 artikel tentang berbagai subjek digunakan dalam penelitian ini. Penulis menyelesaikan sejumlah tugas, termasuk segmentasi, tokenisasi, *case folding*, penghilangan *stopword*, *stemming*, dan penghilangan tanda baca pada tahap prapemrosesan teks. Selanjutnya, konstruksi grafik, peringkat kalimat, bobot TF-IDF, bobot kemiripan, dan pembentukan hasil ringkasan dihitung. Hasil pengujian terbaik diperoleh dari tingkat kompresi 50%, dengan nilai rata-rata *f-measure* pada metrik ROUGE-1, ROUGE-2, dan ROUGE-L berturut-turut sebesar 67,53%, 59,10%, dan 67,05%. Pengujian model menggunakan kombinasi tingkat kompresi 30% dan 50%.

Aisyah *et al.* (2019) meneliti tentang penggunaan *sentence extraction* dan *word sense disambiguation* dalam pengembangan sistem peringkasan teks otomatis multi-dokumen. Menurut temuan pengujian, nilai rata-rata ROUGE-1 adalah 0,712, 0,163, dan 0,216 dalam hal *recall*, *precision*, dan *f-measure*. Sebaliknya, hasil

pengujian yang diperoleh tanpa menggunakan *word sense disambiguation* menunjukkan nilai rata-rata ROUGE-1 untuk *recall*, *precision*, dan *f-measure* masing-masing sebesar 0,685, 0,139, dan 0,216.

Rani & Bidhan (2021) meneliti pengembangan sistem peringkasan teks otomatis menggunakan metode *TextRank*, TF-IDF, dan *Latent Dirichlet Allocation* (LDA). Tiga dataset dari penelitian sebelumnya yang terdiri dari ulasan dokumen, berita, dan informasi hukum digunakan sebagai data perbandingan dengan pengujian menggunakan metode *single-document*. Menurut hasil dari penelitian tersebut, algoritma *TextRank* mengungguli algoritma TF-IDF dan LDA.

Hernawan *et al.* (2022) meneliti tentang sistem peringkasan teks artikel yang ditulis dalam Bahasa Indonesia dengan menggunakan pendekatan *TextRank* dan pembobotan BM25. Sebanyak sepuluh berita mengenai isu lingkungan dan bencana alam dari situs BBC Indonesia digunakan sebagai data penelitian. Tujuan dari pembobotan BM25 adalah untuk menghitung nilai kemiripan berdasarkan bobot setiap kata (*term*) dalam TF-IDF. Panjang ringkasan yang dihasilkan sistem ditunjukkan oleh tingkat kompresi. Tingkat kompresi yang rendah dan tingkat retensi yang tinggi adalah kualitas yang diinginkan untuk sebuah ringkasan. Dengan tingkat kompresi 30%, pengujian memberikan hasil terbaik, dengan rata-rata *f-measure* sebesar 0.552.

Fakhrezi *et al.* (2021) melakukan penelitian tentang peringkasan teks otomatis menggunakan algoritma *TextRank* untuk mengembangkan referensi kosakata Al-Quran. Untuk memberikan hasil rangkuman berdasarkan seluruh arti kata, penelitian ini menggunakan metode untuk mengumpulkan seluruh arti kata,

yang kemudian dirangkum menggunakan *TextRank*. *Query* yang akan dicari dan tafsir Tafsir al-Misbah yang digunakan untuk menentukan arti *query* merupakan input sistem. Setelah itu, algoritma memilih dokumen yang relevan dengan *query* dan menggunakan pendekatan *TextRank* untuk meringkas teks-teks yang telah dipilih. Hasil rangkuman sistem dibandingkan dengan rangkuman yang dibuat secara manual oleh manusia. Nilai rata-rata *f1-score* yang dihasilkan oleh metode *TextRank* pada penelitian tersebut sebesar 0,6173.

Balmuri *et al.* (2021) melakukan penelitian tentang peringkasan teks berbahasa Telugu (India) menggunakan *Long Short Term Memory Deep Learning*. Penelitian ini menggunakan dataset kumpulan artikel online yang dibagi menjadi 6 kategori, yaitu kultur, ekonomi, berita lokal, berita internasional, berita religi, dan berita olahraga. Peneliti juga menggunakan deskriptor fitur yang digunakan untuk memberikan skor informatif dari kalimat. Deskriptor fitur yang dipakai adalah *Latent Semantic Analysis* (LSA) dan TF-IDF. Pada penelitian ini, metode *Long Short Term Memory Deep Learning* mampu menghasilkan akurasi 86% untuk kata kunci tertentu dan mampu meringkas Bahasa Telugu.

Setiawan *et al.* (2022) melakukan penelitian tentang analisis sentiment dan peringkasan teks evaluasi pembelajaran menggunakan algoritma *Long Short Term Memory*. Tujuan dari penelitian ini adalah untuk mengklasifikasikan hasil evaluasi dari pembelajaran dosen kedalam evaluasi yang positif atau negatif, serta meringkas teks evaluasi tersebut untuk memudahkan dosen dalam mencari intisari dalam teks evaluasi. Pengujian dilakukan dengan pengklasifikasikan dan peringkasan menggunakan algoritma *Long Short Term Memory*. Output data klasifikasi

kemudian diukur menggunakan *confusion matrix*, sedangkan hasil peringkasan dievaluasi menggunakan ROUGE. Hasil pengujian mendapatkan nilai akurasi rata-rata dan nilai *f-measure* secara berurutan sebesar 0,902 dan 0,921.

Berikut penelitian-penelitian terkait yang dijadikan sumber acuan untuk melakukan penelitian mengenai Analisis Perbandingan *TextRank* dan *Long Short Term Memory* Dalam Peringkasan Teks Berita Berbahasa Inggris.

Tabel 2.1 Penelitian Terkait

No	Peneliti	Judul	Metode	Hasil
1	(Halimah <i>et al.</i> , 2022)	Peringkasan Teks Otomatis (<i>Automated Text Summarization</i>) Pada Artikel Berbahasa Indonesia Menggunakan Algoritma Lexrank	<i>LexRank</i>	<i>Compression rate</i> 50% menghasilkan nilai <i>ROUGE-1</i> sebesar 67,53%, <i>ROUGE-2</i> sebesar 59,10%, dan <i>ROUGE-L</i> sebesar 67,05%. Sedangkan <i>compression rate</i> 30% menghasilkan rata-rata <i>f-measure</i> pada <i>ROUGE-1</i> sebesar 55,82%, <i>ROUGE-2</i> sebesar 45,51%, dan <i>ROUGE-L</i> sebesar 54,76%.

No	Peneliti	Judul	Metode	Hasil
2	(Aisyah <i>et al.</i> , 2019)	Peringkasan Teks Multi-Dokumen Berdasarkan Metode <i>Sentence Extraction Dan Word Sense Disambiguation</i>	<i>Sentence Extraction dan Word Sense Disambiguation</i>	Nilai rata-rata <i>ROUGE-1</i> adalah 0,712, 0,163, dan 0,216 pada <i>recall</i> , <i>precision</i> , dan <i>f-measure</i> secara berurutan. Sedangkan nilai rata-rata tanpa menggunakan <i>word sense disambiguation</i> sebesar 0,685, 0,139, dan 0,216 pada <i>recall</i> , <i>precision</i> , dan <i>f-measure</i> secara berurutan.
3	(Rani & Bidhan, 2021)	<i>Comparative Assessment of Extractive Summarization: TextRank, TF-IDF and LDA</i>	<i>TextRank, TF-IDF, dan Latent Dirichlet Allocation (LDA)</i>	<i>TextRank</i> menunjukkan nilai terbaik dibandingkan dengan <i>TF-IDF</i> dan <i>LDA</i> dengan nilai <i>recall</i> , <i>precision</i> , dan <i>f-measure</i> tertingginya pada <i>ROUGE-1</i> secara berurutan adalah 79%, 56%, dan 63%.
4	(Hernawan <i>et al.</i> , 2022)	Peringkasan Artikel Berbahasa Indonesia Menggunakan <i>TextRank</i> Dengan Pembobotan <i>BM25</i>	<i>TextRank</i> dan pembobotan <i>BM25</i>	Hasil terbaik menggunakan <i>compression rate</i> 30% dengan nilai rata-rata <i>precision</i> , <i>recall</i> , dan <i>f-measure</i> sebesar 0,552
5	(Fakhrezi <i>et al.</i> , 2021)	<i>Implementation of Automatic Text Summarization with TextRank Method in the Development of Al-Qur'an Vocabulary Encyclopedia</i>	<i>TextRank</i>	Rata-rata nilai <i>f-Score</i> yang dihasilkan pada penelitian ini sebesar 0,6173

No	Peneliti	Judul	Metode	Hasil
6	(Balmuri <i>et al.</i> , 2021)	<i>Telugu Text Summarization Using Lstm Deep Learning</i>	<i>Long Short Term Memory (LSTM)</i>	Metode <i>Long Short Term Memory Deep Learning</i> mampu menghasilkan akurasi 86% untuk kata kunci tertentu dan mampu meringkas Bahasa Telugu.
7	(Setiawan <i>et al.</i> , 2022)	<i>Sentiment Summarization Evaluasi Pembelajaran Menggunakan Algoritma LSTM (Long Short Term Memory)</i>	<i>Long Short Term Memory (LSTM)</i>	Hasil pengujian mendapatkan nilai akurasi sebesar 0,902 dan nilai <i>f-measure</i> sebesar 0,921.

Berdasarkan penelitian terkait diatas, bisa ditarik kesimpulan bahwa algoritma *TextRank* dan LSTM sudah terbukti mampu menghasilkan akurasi tinggi saat digunakan dalam pengujian peringkasan teks dibandingkan dengan algoritma peringkasan teks lainnya. Oleh karena itu, dalam penelitian ini, peneliti ingin membandingkan kinerja *TextRank* dan LSTM untuk peringkasan teks berita berbahasa Inggris.

2.2 Peringkasan Teks Otomatis

Peringkasan teks otomatis adalah proses memadatkan teks secara otomatis menggunakan komputer untuk membuat versi yang lebih pendek yang menyoroti poin-poin penting dan membantu pembaca memahami poin-poin tersebut tanpa harus membaca teks atau dokumen secara keseluruhan (Andriani *et al.*, 2019). Gagasan mengenai sistem peringkasan teks otomatis dicetuskan oleh Luhn (1958) yang pada awalnya mempresentasikan konsep sistem peringkasan teks otomatis

dalam studinya tentang sintesis abstrak otomatis dalam makalah dan artikel majalah. Sebuah program komputer menggunakan peringkasan otomatis untuk memotong bagian-bagian teks dan membuat ringkasan yang menyoroti detail yang paling penting sambil menyajikan konten dokumen lengkap. Materi dalam ringkasan dokumen harus komprehensif, dengan pengulangan yang minimal dan keterbacaan yang sangat baik di seluruh dokumen. (Zong *et al.*, 2021).

Jika dilihat berdasarkan inputnya, sistem peringkasan teks otomatis dapat dilakukan menggunakan *single-document* atau *multi-document*. Sementara itu, ada dua teknik untuk meringkas teks secara otomatis: peringkasan ekstraktif dan peringkasan abstraktif. Peringkasan ekstraktif membuat ringkasan dengan memilih kalimat yang berkaitan dengan konten asli dan kemudian memilih istilah dengan skor relevansi tertinggi (Ammar & Suyanto, 2020). Sedangkan peringkasan abstraktif menghasilkan ringkasan dengan melakukan interpretasi teks asal melalui proses transformasi pada kalimat asli tanpa menghilangkan makna sebenarnya (Alpiani, 2019).

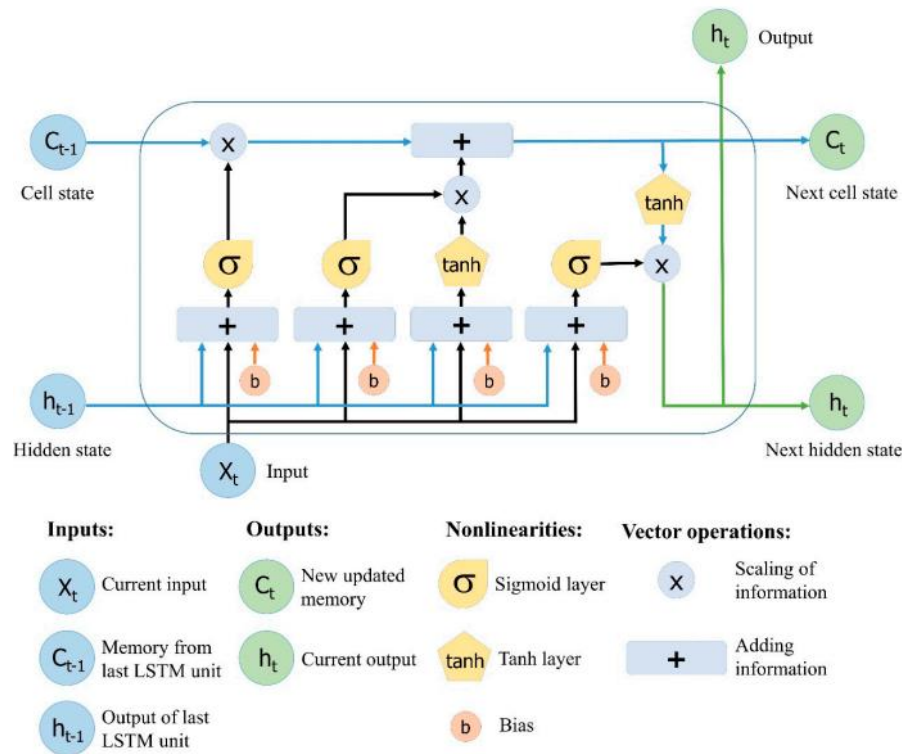
2.3 *TextRank*

TextRank termasuk algoritma peringkasan teks *graph-based* yang merupakan turunan dari algoritma *PageRank* (Zhou *et al.*, 2022). *TextRank* menggunakan jaringan dengan *node* untuk merepresentasikan kalimat dalam dokumen dan *edge* untuk menunjukkan hubungan kemiripan antar kalimat. Memberikan voting atau merekomendasikan sebuah simpul ketika simpul tersebut terhubung dengan simpul lain adalah ide dasar di balik pemeringkatan berbasis graf. Skor sebuah simpul meningkat secara proporsional dengan jumlah *vote* yang diterimanya. Algoritma

TextRank digunakan untuk mengekstrak kalimat dalam gagasan peringkasan teks otomatis, dengan tujuan mendapatkan kalimat-kalimat penting dari dokumen. *TextRank* beroperasi dalam dua tahap yaitu menentukan kepentingan kalimat dan memeriksa seberapa mirip dua frasa satu sama lain dalam dokumen. (Verma *et al.*, 2019).

2.4 Long Short Term Memory

Long Short Term Memory (LSTM) merupakan salah satu algoritma turunan dari *Recurrent Neural Network* (RNN) dengan kelebihan dapat meminimalisir masalah ketergantungan jangka panjang pada RNN. LSTM pertama kali diperkenalkan oleh Hochreiter & Schmidhuber pada tahun 1997. Dibandingkan dengan RNN, LSTM memiliki keunggulan dalam hal data pembelajaran yang nantinya akan disimpan jika perhitungan menghasilkan nilai 1, dan dibuang jika hasilnya 0 (Johan, 2019). Pada umumnya, struktur dalam LSTM terdapat *cell state*, *input gate*, *output gate*, dan *forget gate* (Gambar 2.1).



Gambar 2.1 Struktur jaringan LSTM (Le et al., 2019)

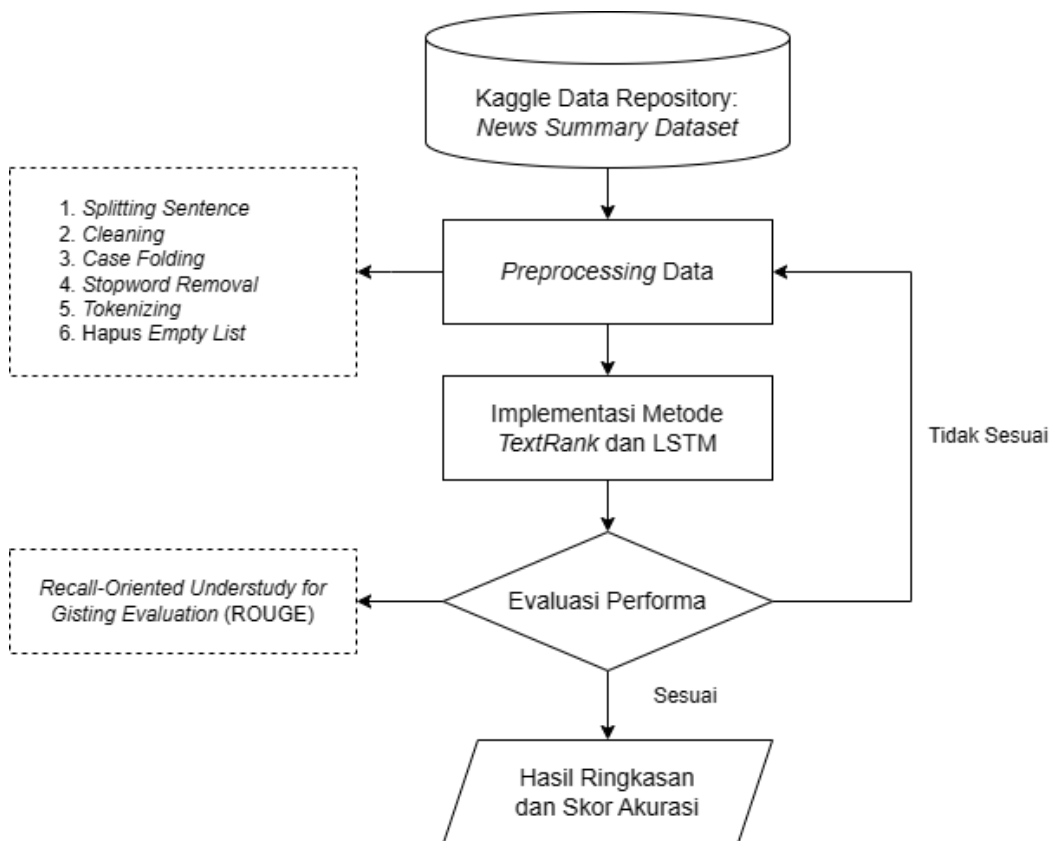
Seperti yang telah divisualisasikan pada Gambar 2.1 dimana *cell* berfungsi sebagai komponen pembangun jaringan LSTM. *Cell state* dan *hidden state* adalah dua *state* yang diteruskan ke *cell* berikutnya. *Cell state* adalah rantai utama aliran data yang memungkinkan data untuk bergerak maju tanpa modifikasi. Tetapi mungkin ada beberapa transformasi linier yang dapat terjadi. *Sigmoid gate* memungkinkan data ditambahkan atau ditarik dari *cell state*. *Gate* dapat dibandingkan dengan sebuah lapisan atau sekumpulan operasi matriks yang masing-masing memiliki sekumpulan bobot yang unik. Karena LSTM menggunakan *gate* untuk mengatur proses penghafalan, *gate* dibuat untuk mencegah masalah ketergantungan jangka panjang (Le et al., 2019).

Algoritma pada lapisan LSTM adalah sebagai berikut, jika *cell* saat ini adalah *cell* pertama dalam lapisan LSTM, *cell state* sebelumnya adalah 0, dan nilai dari *cell state* sebelumnya akan masuk ke dalam *cell* saat ini. Informasi akan masuk dan melewati *forget gate* pada saat yang bersamaan. *Forget gate* menggunakan fungsi sigmoid untuk menghitung hasilnya. Jika hasilnya 0, informasi akan dilupakan, sedangkan jika hasilnya 1 informasi akan diingat. Nilai *cell state* sebelumnya akan dikalikan dengan nilai *forget gate*, menghasilkan nilai *cell state* 0. Fungsi sigmoid digunakan untuk menghitung informasi yang mengalir melalui *input gate*, lalu hasil dari fungsi sigmoid tersebut dikalikan dengan hasil dari fungsi *tanh* yang berguna untuk menghitung informasi. Nilai tersebut juga akan dibagi menjadi dua, yang pertama akan diteruskan ke *cell* berikutnya, dan dua nilai *cell state* yang baru kemudian akan dihitung menggunakan fungsi *tanh* untuk menghasilkan nilai *tanh*. Selain itu, nilai tersebut akan dikalikan dengan nilai hasil dari *input gate*. Hasil dari perkalian tersebut kemudian ditambahkan ke nilai *cell state* untuk menghasilkan nilai *cell state* yang baru. Sebuah fungsi sigmoid akan digunakan untuk menghitung data yang melewati *output gate*, dan hasilnya akan dikalikan dengan nilai *tanh* untuk membuat nilai *hidden state* di *cell* saat ini, yang akan ditransmisikan ke *cell* berikutnya. Prosedur ini terus berjalan hingga mencapai *cell* lapisan LSTM terakhir (Setiawan *et al.*, 2022).

BAB III

DESAIN DAN IMPLEMENTASI

Metode penelitian ini dipilih dengan cermat sebagai alat untuk menjawab pertanyaan penelitian dan mencapai tujuan yang telah ditetapkan. Diharapkan dengan adanya langkah-langkah yang telah terorganisir, penelitian ini dapat dilaksanakan dengan maksimal.



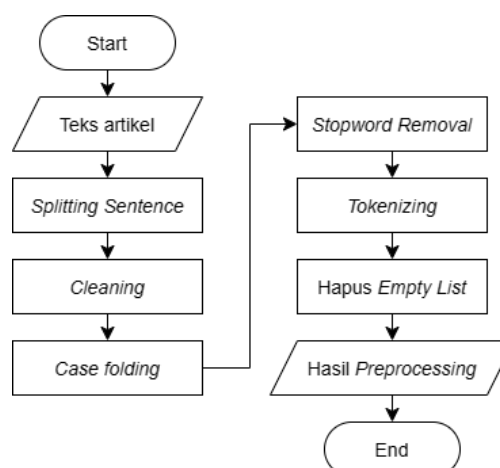
Gambar 3.1 Diagram metode penelitian

3.1 Pengumpulan Data

Sesuai dengan diagram metode penelitian pada Gambar 3.1, dimana langkah awal penelitian ini adalah pengumpulan data. Data yang digunakan pada penelitian ini adalah dataset berita Bahasa Inggris yang berasal dari Kaggle (<https://www.kaggle.com/datasets/sunnysai12345/news-summary>). Dataset tersebut berisi 4515 data yang terbagi dalam beberapa kolom, seperti *Author_name*, *Headline*, Url artikel, teks pendek, dan artikel lengkap. Berita tersebut dirangkum dari Inshorts dan hanya mengambil artikel berita dari *Hindu*, *Indian Times* dan *The Guardian*. Periode berita berkisar dari bulan Februari hingga Agustus 2017.

3.2 Preprocessing Data

Tahap *preprocessing* sangatlah penting guna mengolah dokumen teks mentah menjadi teks yang siap digunakan pada pengembangan sistem peringkasan teks otomatis. Secara umum, urutan tahap *preprocessing* pada NLP adalah *stopword removal*, *tokenizing*, dan *case folding*. Berikut tahapan *preprocessing* data yang digunakan pada penelitian ini.



Gambar 3.2 Flowchart preprocessing

3.2.1 *Splitting Sentence*

Tahap ini membagi artikel menjadi beberapa kalimat yang kemudian dimasukkan ke dalam daftar menggunakan fungsi *split*, *replace*, dan notasi *regular expression (regex)* di Python. Seperti pada contoh Tabel 3.1 dimana teks dibagi menggunakan pembatas yang telah ditentukan melalui fungsi *split*. Tanda seru (!), tanda tanya (?), dan titik (.) digunakan sebagai pembatas untuk memisahkan frasa. Namun, pembatas tidak selalu digunakan sebagai pemisah kalimat. Contohnya adalah titik yang digunakan dalam akronim, situs web, pemisah angka, dan singkatan dari judul dan nama orang. Pemisahan kalimat menggunakan seluruh data teks artikel yang dikumpulkan melalui pengikisan data sebagai data masukannya. Kumpulan kalimat dari teks artikel berfungsi sebagai output sementara.

Tabel 3.1 Contoh *Splitting Sentence*

<p>Sebelum <i>splitting sentence</i></p>	<p><i>The Serious Fraud Office (SFO) in the UK is looking into British American Tobacco (BAT), a tobacco giant, for its participation in bribery and corruption in Africa. Allegations that a whistleblower gave several documents to SFO have been refuted by BAT. But BAT has stated that it is working with UK authorities since activists are now pressuring the US government to take action. The Serious Fraud Office is looking into allegations that British American Tobacco was involved in bribery and corruption in Africa, even though the cigarette firm is trying to minimize the importance of a whistleblower who turned over a dossier of evidence.....</i></p>
<p>Sesudah <i>splitting sentence</i></p>	<p>[<i>'The Serious Fraud Office (SFO) in the UK is looking into British American Tobacco (BAT), a tobacco giant, for its participation in bribery and corruption in Africa. Allegations that a whistleblower gave several documents to SFO have been refuted by BAT.'</i>, <i>'But BAT has stated that it is working with UK authorities since activists are now pressuring the US government to take action.'</i>, <i>'The Serious Fraud Office is looking into allegations that British American Tobacco was involved in bribery and corruption in Africa, even though the cigarette firm is trying to minimize the importance of a whistleblower who turned over a dossier of evidence.'</i>,]</p>

3.2.2 *Cleaning*

Pada tahap ini, kalimat dibersihkan untuk menghilangkan spasi, tanda baca, dan *noise* yang tidak perlu, seperti angka. Karakter yang disebut *noise* tidak diperlukan untuk analisis teks. Notasi *regex* digunakan untuk proses *cleaning*. Pada Tabel 3.2 adalah contoh data sebelum dan sesudah dilakukan proses *cleaning* teks dimana karakter angka dihilangkan karena dianggap sebagai *noise* dan tidak diperlukan untuk analisis teks.

Tabel 3.2 Contoh *Cleaning*

Sebelum <i>cleaning</i>	['UK roadside repair firm AA on Tuesday lost over £200 million nearly 1,690 crore or nearly a fifth of its value.', 'This came after shares fell as much as 18% during the day after it fired its Executive Chairman Bob Mackenzie with immediate effect for 'gross misconduct'. ']
Sesudah <i>cleaning</i>	['UK roadside repair firm AA on Tuesday lost over million nearly crore or nearly a fifth of its value.', 'This came after shares fell as much as during the day after it fired its Executive Chairman Bob Mackenzie with immediate effect for gross misconduct. ']

3.2.3 *Case Folding*

Dengan menggunakan fungsi *lower* di Python, *case folding* digunakan untuk mengubah karakter huruf besar dalam teks menjadi huruf kecil seperti pada contoh Tabel 3.3. Dengan menyeragamkan struktur teks, *case folding* meningkatkan konsistensi dan membantu meminimalkan kesalahan ketik yang disebabkan oleh penggunaan huruf kapital.

Tabel 3.3 Contoh *Case Folding*

Sebelum <i>case folding</i>	['UK roadside repair firm AA on Tuesday lost over million nearly crore or nearly a fifth of its value.', 'This came after shares fell as much as during the day after it fired its Executive Chairman Bob Mackenzie with immediate effect for gross misconduct. ']
Sesudah <i>case folding</i>	['uk roadside repair firm aa on Tuesday lost over million nearly crore or nearly a fifth of its value.', 'this came after shares fell as much as during the day after it fired its executive chairman bob mackenzie with immediate effect for gross misconduct. ']

3.2.4 Stopword Removal

Stopwords adalah kata-kata yang dianggap kurang penting dalam sebuah teks, sehingga kurang berguna dalam analisis teks. Untuk meningkatkan kualitas hasil analisis teks, *stopwords* akan dihilangkan. Kalimat akan diverifikasi untuk setiap kata. Kata tersebut akan dihapus jika muncul dalam daftar kata yang dilarang. Kata tersebut akan muncul dalam daftar hasil penghapusan *stopword* jika tidak ada dalam daftar *stopword*. Pada Tabel 3.4 dapat dilihat bahwa ada beberapa kata yang dianggap tidak penting dihilangkan untuk meningkatkan hasil analisis teks, seperti kata *over*, *or*, *of*, *its*, dan lainnya. Kata yang dihilangkan biasanya berupa kata imbuhan.

Tabel 3.4 Contoh *Stopword Removal*

Sebelum case folding	['uk roadside repair firm aa on Tuesday lost over million nearly crore or nearly a fifth of its value.', 'this came after shares fell as much as during the day after it fired its executive chairman bob mackenzie with immediate effect for gross misconduct. ']
Sesudah case folding	['uk roadside repair firm aa Tuesday lost million nearly crore nearly fifth value.', 'came shares fell much day fired executive chairman bob mackenzie immediate effect gross misconduct . ']

3.2.5 Tokenizing

Tokenisasi adalah proses membagi kalimat menjadi kumpulan token kata. Tahap ini dilakukan untuk membuat pemrosesan teks menjadi lebih sederhana. Tokenisasi adalah proses membagi frasa menjadi daftar kata berdasarkan karakter spasi menggunakan notasi *regex* dan fungsi *split* seperti yang dapat dilihat pada Tabel 3.5 dimana setiap kata dalam paragraf dibagi menjadi token per kata.

Tabel 3.5 Contoh *Tokenizing*

Sebelum case folding	['uk roadside repair firm aa on Tuesday lost over million nearly crore or nearly a fifth of its value.', 'this came after shares fell as much as during the day after it fired its executive chairman bob mackenzie with immediate effect for gross misconduct. ']
-----------------------------	--

Sesudah case folding	[['uk', 'roadside', 'repair', 'firm', 'aa', 'Tuesday', 'lost', 'million', 'nearly', 'crore', 'nearly', 'fifth value'], ['came', 'shares', 'fell', 'much', 'day', 'fired', 'executive', 'chairman', 'bob', 'mackenzie', 'immediate', 'effect', 'gross', 'misconduct']]
---------------------------------	--

3.2.6 Hapus *Empty List*

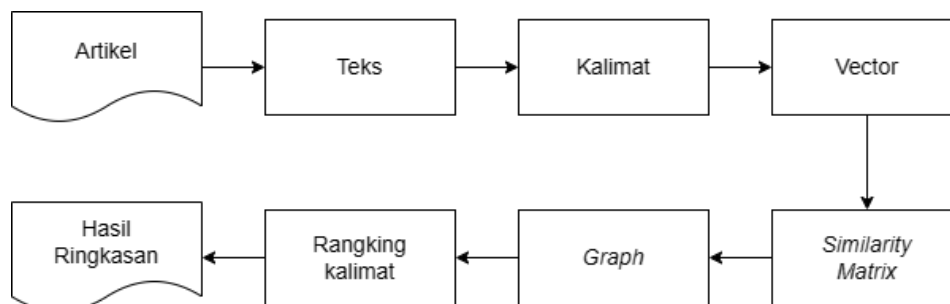
Selama operasi pembersihan dan penghapusan *stopword*, semua token kata dalam item kalimat dihilangkan, sehingga item dalam daftar kalimat menjadi kosong. Ada kalimat yang hanya berisi *stopword*, angka, atau hanya dalam bentuk karakter, sehingga frasa tersebut dihilangkan dan menjadi kosong. Elemen-elemen yang kosong harus dihilangkan karena dapat mengakibatkan kesalahan selama analisis teks.

3.3 Implementasi Metode

Setelah data melalui tahap *preprocessing*, maka tahap selanjutnya adalah implementasi metode yang dipakai untuk membuat sistem peringkasan teks otomatis. Pada penelitian ini, metode yang dipakai adalah *TextRank* dan LSTM yang pada hasil akhirnya nanti akan dibandingkan hasil akurasi.

3.3.1 Metode *TextRank*

TextRank menggunakan grafik teks untuk merepresentasikan hubungan antara istilah untuk teks yang diberikan. Setelah membuat grafik teks, skor *TextRank* dari kata-kata dihitung dengan menggunakan *PageRank*. Awalnya, semua kata yang terdapat dalam grafik teks diperlakukan sebagai kandidat kata kunci, dan tujuan dari *TextRank* adalah untuk menemukan kata terpenting dengan nilai *PageRank* tertinggi dan kemudian menganggapnya sebagai kata kunci (M. Zhang *et al.*, 2020). Berikut adalah alur algoritma *TextRank*:



Gambar 3.3 Alur algoritma *TextRank*

Langkah pertama yang dilakukan oleh *TextRank* adalah menggabungkan beberapa kalimat yang terkandung dalam artikel, kemudian membagi teks tersebut menjadi beberapa kalimat. Kemudian kita akan menemukan *vector representation* (*word embedding*) untuk setiap kalimat, kemiripan (*similarity*) antar vektor kalimat kemudian dihitung dan disimpan dalam sebuah matriks. *Similarity matrix* kemudian dikonversi ke dalam sebuah graf, dengan kalimat sebagai simpul dan skor kemiripan sebagai sisi, untuk perhitungan kalimat. Terakhir, sejumlah kalimat dengan peringkat teratas akan menjadi rangkuman akhir.

3.3.1.1 Vektorisasi Menggunakan *Global Vector*

Kalimat diubah menjadi vektor menggunakan metode *Global Vector* (GloVe). GloVe adalah *unsupervised word embedding* (penyematan kata tanpa pengawasan), yang diperkenalkan oleh Pennington *et al.* (2014) untuk representasi vektor kata. Algoritma ini menggunakan statistik kemunculan bersama dari kata ke kata dalam sebuah korpus dan digunakan untuk kemiripan dan identifikasi entitas (Mohammed *et al.*, 2020). Contoh hasil vektorisasi untuk kalimat “*The Earth is the third planet from the Sun and the only place known to harbor life. It is the fifth-largest planet in the Solar System by diameter, the largest by mass among the four*

terrestrial planets, and the largest in the Solar System relative to its diameter. It is the densest planet in the Solar System.” Adalah sebagai berikut:

Tabel 3.6 Contoh vektorisasi

Kalimat	Vektor
Kalimat pertama	[-0.01855297 0.00417254 -0.00109405 ... -0.00128749 -0.00163248 0.00095206]
Kalimat kedua	[0.00717306 -0.00041978 0.00084715 ... -0.00015643 -0.00059862 0.00045851]
Kalimat ketiga	[0.00506789 -0.00028667 0.00047272 ... -0.00012228 -0.00045653 0.00030242]

3.3.1.2 Similarity Matrix

Kesamaan matriks adalah matriks yang memuat skor kesamaan antaran pasangan elemen dalam suatu kumpulan data. Matriks ini digunakan untuk menggambarkan sejauh mana setiap pasangan elemen saling berhubungan atau serupa dalam suatu konteks tertentu. Pendekatan *cosine similarity* digunakan dalam perhitungan matriks kemiripan dalam penelitian ini. Salah satu matriks kemiripan yang umum digunakan dalam pencarian informasi dan penelitian terkait adalah *cosine similarity*. Dokumen teks dimodelkan sebagai vektor dalam matriks ini. Dengan menggunakan teknik ini, perbedaan kosinus antara vektor *term* dari dua dokumen dapat digunakan untuk menentukan seberapa mirip kedua dokumen tersebut (Rahutomo *et al.*, 2012). Contoh hasil *similarity matrix* seperti yang tertera pada Tabel 3.7.

Tabel 3.7 Contoh hasil *similarity matrix*

	Kalimat pertama	Kalimat kedua	Kalimat ketiga
Kalimat pertama	1.0	0.974074	0.937382
Kalimat kedua	0.974074	1.0	0.931696
Kalimat ketiga	0.937382	0.931696	1.0

3.3.1.3 Rumus TextRank

Rumus *PageRank* dikembangkan menjadi rumus penghitungan skor *TextRank*. Skor *TextRank* dihitung secara rekursif menggunakan algoritma yang didasarkan pada graf. Graf yang terdapat pada *TextRank* merupakan graf yang berbobot namun tidak memiliki arah atau yang biasa disebut *undirected weighted graph* (Mihalcea & Tarau, 2004). Berikut rumus dari *TextRank*:

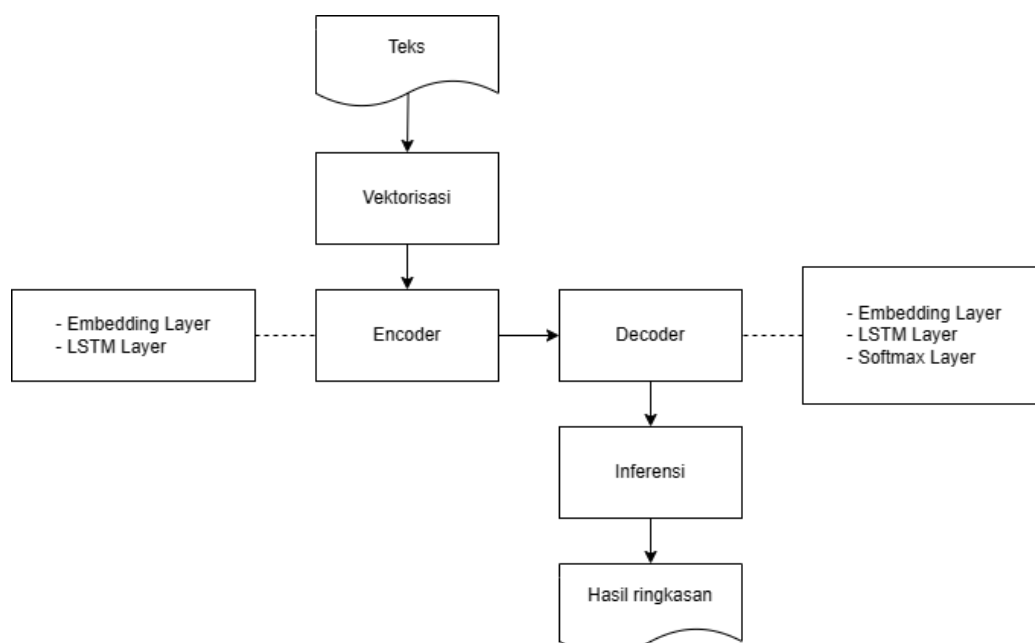
$$WS(V_i) = (1 - d) + d * \sum_{V_j \in In(V_i)} \frac{W_{ji}}{\sum_{V_k \in Out(V_j)} W_{jk}} WS(V_j) \quad (3.1)$$

Skor kalimat dinyatakan sebagai $WS(V_i)$, sementara d adalah *damping factor* yang bernilai 0 sampai 1, dan W_{ji} adalah bobot yang dinyatakan sebagai nilai *similarity* antar kalimat (j dan i). Fungsi $In(V_i)$ menciptakan keluaran yang berisi elemen-elemen simpul yang menunjuk ke simpul V_i (*predecessor*). Di sisi lain, $Out(V_j)$ adalah sekumpulan elemen *vertex* di mana V_j menuju ke simpul tersebut (*successor*). Jika matriks yang dibentuk sebelumnya mengandung kondisi bahwa $V_j \neq V_i$ dan nilai kemiripan antara V_j dan V_i tidak sama dengan 0, maka fungsi $In(V_i)$ menghasilkan *output* berupa kalimat V_j untuk $j = 1, 2, 3$, dan seterusnya.

3.3.2 Metode Long Short Term Memory

Metode kedua yang digunakan pada penelitian ini sebagai pembanding dari metode pertama adalah *Long Short Term Memory* (LSTM). LSTM dipilih karena menjadi metode peringkasan teks abstraktif yang sudah teruji dalam pembuatan sistem peringkasan teks otomatis, serta memiliki performa yang mumpuni. LSTM sendiri merupakan jenis variansi dari metode *Recurrent Neural Network* (RNN)

yang memiliki kemampuan untuk mengolah informasi lebih akurat (H. Zhang *et al.*, 2016). Kelebihan lain dari LSTM daripada RNN adalah adanya arsitektur yang dapat menyimpan dan membuang output yang akan diproses kembali menjadi input serta dapat mempertahankan *error* yang terjadi ketika melakukan *backpropagation* sehingga dapat mengurangi kesalahan (H. Zhang *et al.*, 2016). Berikut alur algoritma LSTM:



Gambar 3.4 Alur algoritma LSTM

Alur pemrosesan LSTM seperti yang telah divisualisasikan pada Gambar 3.4 dimulai dari *Encoder* yang menerima representasi vektor kalimat sebagai input lalu mengubahnya menjadi representasi vektor baru yang nantinya disimpan dan dimasukkan ke dalam *hidden state* dan *cell state* yang terdapat pada *LSTM Layer*. Representasi vektor baru itulah yang nantinya menjadi input untuk *decoder*. Tugas *decoder* selanjutnya adalah mengkonversi representasi vektor baru menjadi kata, proses tersebut terus berulang hingga menghasilkan output kalimat sebagai hasil

ringkasan. Pada *decoder* terdapat *softmax layer* yang memiliki tugas memprediksi probabilitas setiap kata dalam kosakata yang akan dimasukkan ke dalam ringkasan, kata-kata yang memiliki probabilitas tinggi akan lebih mungkin dimasukkan ke dalam hasil ringkasan. Contoh output *softmax layer* terdapat pada Tabel 3.8.

Tabel 3.8 Contoh pembobotan dan vektor probabilitas

Kata	["the earth", "is", "the third", "planet", "from", "the sun", "and", "the only", "place", "known", "to", "harbor", "life"]
Bobot vektor	[0.12, 0.04, 0.06, 0.72, 0.12, 0.04, 0.14, 0.08, 0.26, 0.10, 0.32, 0.44, 0.30]
Vektor probabilitas	[0.024, 0.012, 0.024, 0.072, 0.024, 0.008, 0.014, 0.008, 0.052, 0.030, 0.032, 0.044, 0.030]

Tahapan pemrosesan pada LSTM melewati beberapa jaringan yang terdiri dari *cell state*, *input gate*, *output gate*, dan *forget gate*. Proses awal adalah menggunakan fungsi *sigmoid* untuk menghilangkan data yang tidak relevan. Tujuan dari *sigmoid* adalah untuk mengubah nilai antara -1 hingga 1 menjadi nilai antara 0 hingga 1. Di mana nilai 0 menunjukkan penghapusan elemen ini dan nilai 1 berarti mempertahankan elemen ini. Proses ini berlangsung pada *forget gate* dimana persamaan dari *forget gate* sebagai berikut:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (3.2)$$

Tahap selanjutnya pada persamaan 3.3 adalah mengolah informasi melalui komponen (i_t) untuk menentukan informasi yang diperbarui. Fungsi *tanh* pada persamaan 3.4 digunakan untuk membentuk kandidat vektor baru yang kemudian keduanya digabung untuk menghasilkan keterbaruan pada tahap selanjutnya.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (3.3)$$

$$N_t = \tanh(W_n[h_{t-1}, x_t] + b_n) \quad (3.4)$$

Selanjutnya, pada persamaan 3.5 nilai *memory cell* lama (C_{t-1}) digantikan oleh nilai *memory cell* baru (C_t). Nilai hasil dari *memory cell* baru (C_t) diperoleh dari gabungan antara *input gate* dan *forget gate*. Berikut rumusnya:

$$C_t = C_{t-1} f_t + N_t i_t \quad (3.5)$$

Terakhir, nilai keluaran dari *hidden state* (h_t) adalah versi yang disaring dari *output gate* (o_t), seperti pada persamaan 3.6. Lapisan sigmoid pertama-tama memilih komponen *cell state* yang menghasilkan output. Selanjutnya, pada persamaan 3.7 nilai baru yang dihasilkan oleh lapisan tanh dari *cell gates* (c_t), yang mungkin memiliki nilai antara -1 dan 1, dikalikan dengan output gerbang sigmoid (o_t).

$$O_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (3.6)$$

$$h_t = O_t \tanh(C_t) \quad (3.7)$$

Keterangan:

f_t : *forget gate*

σ : fungsi sigmoid

W : bobot

h_t : *hidden state*

x_t : input pada *time step t*

b : bias

i_t : *input gate*

C_t : *cell aktivasi*

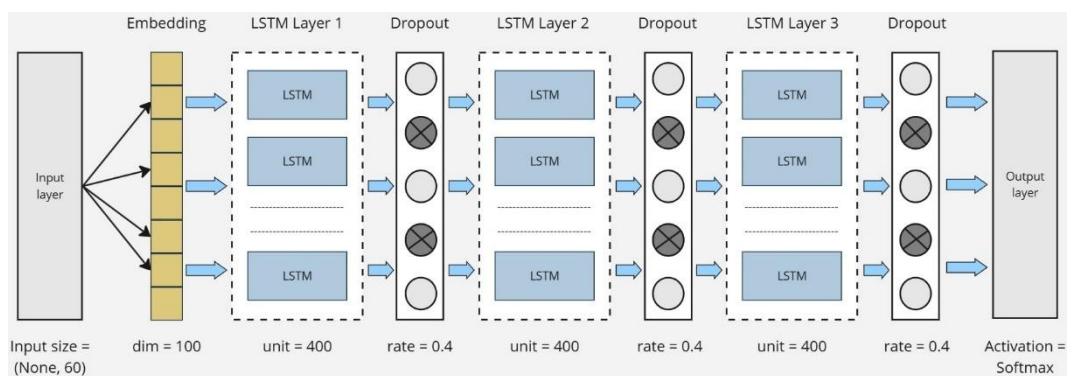
x_t : input ke t

C_t : *cell gates*

f_t : *forget gate*

o_t : *output gate*

Pada penelitian ini, arsitektur LSTM yang dipakai adalah menggunakan 3 *stack layer* dengan jumlah unit setiap *layer*-nya adalah 400 unit. Berikut merupakan arsitektur LSTM yang digunakan pada penelitian ini:



Gambar 3.5 Arsitektur LSTM

Pada Gambar 3.5 dapat dilihat arsitektur LSTM yang digunakan pada penelitian ini memiliki enam *layer* dengan rincian satu *input layer*, satu *embedding layer*, tiga *LSTM layer*, dan satu *output layer*. Kegunaan dari *input layer* adalah menentukan bentuk dan dimensi data yang akan dimasukkan ke dalam model. Sebelum diproses oleh LSTM, data teks yang masuk akan diubah menjadi representasi vektor pada *embedding layer*. Data teks yang telah diubah menjadi vektor kemudian diproses dalam *LSTM layer*, pada *LSTM layer* sendiri terdapat komponen utama yang terdiri dari *cell state*, *hidden state*, dan *gates*. *Cell state* berfungsi untuk mengontrol informasi yang masuk pada *LSTM layer*, apakah informasi tersebut harus disimpan atau dilupakan. *Hidden state* berfungsi untuk menyimpan informasi yang nantinya digunakan untuk menghasilkan output pada langkah waktu saat ini. *Gates* berfungsi mengatur aliran masuk dan keluar informasi dari *cell state*. Pada *LSTM layer* juga terdapat *dropout* yang berfungsi untuk membantu mencegah *overfitting* dengan membuang Sebagian unit secara

acak selama pelatihan. Terakhir, *output layer* memiliki fungsi menghasilkan prediksi akhir berdasarkan informasi yang telah diproses oleh lapisan-lapisan sebelumnya.

3.4 Evaluasi

Pada tahap evaluasi, perhitungan dilakukan untuk membandingkan hasil ringkasan manual pada dataset dengan ringkasan sistem yang dihasilkan oleh masing-masing teknik ini. Metode *Recall-Oriented Understudy for Gisting Evaluation* (ROUGE) digunakan untuk evaluasi pada penelitian ini. ROUGE adalah matriks pengukuran yang digunakan untuk menilai kualitas keluaran peringkasan teks. Kita dapat mengevaluasi keefektifan hasil peringkasan teks dengan membandingkan ringkasan asli dengan ringkasan yang dihasilkan oleh model. ROUGE hadir dalam berbagai bentuk, termasuk *bigram* dan *unigram* (Lin, 2004).

ROUGE-1 dipilih untuk melakukan evaluasi dalam penelitian ini. ROUGE-1 adalah salah satu model ROUGE yang menggunakan *unigram* untuk menilai seberapa dekat ringkasan yang dimodelkan menyerupai ringkasan aslinya. Cara kerja ROUGE yaitu dengan menghitung kata yang sama yang muncul dalam ringkasan sistem dan ringkasan asli (tumpah tindih) secara *unigram* atau per kata (*overlap*) untuk setiap data uji guna mengevaluasi konstruksi sistem. Kemudian menghitung *recall*, *precision*, dan *f1-score*.

$$Recall = \frac{\text{Overlap kata antara ringkasan referensi \& prediksi}}{\text{banyak kata pada ringkasan referensi}} \quad (3.8)$$

$$Precision = \frac{\text{Overlap kata antara ringkasan referensi \& prediksi}}{\text{banyak kata pada ringkasan prediksi}} \quad (3.9)$$

$$F1_{score} = \frac{2 * (Precision * Recall)}{Precision + Recall} \quad (3.10)$$

Nilai recall dihitung dengan membagi jumlah kata yang tumpang tindih dengan jumlah kata dalam ringkasan referensi. Precision dihitung dengan membagi jumlah kata yang tumpang tindih dengan jumlah kata dalam ringkasan prediksi. Terakhir, nilai f1-score dihitung untuk meningkatkan evaluasi. Ketika menghitung nilai recall, precision, dan f1-score, hal yang dilakukan terlebih dahulu adalah menghitung jumlah kata pada ringkasan prediksi yang meng-overlap atau sama dengan kata pada ringkasan manual.

Tabel 3.9 Perbandingan ringkasan manual dan sistem

Ringkasan manual	<i>start gurugram police deploy drones kanwar end</i>
Ringkasan prediksi	<i>start police deploy drones cameras end</i>

Pada Tabel 3.9 merupakan perbandingan hasil ringkasan prediksi dengan ringkasan manual yang terdapat pada dataset. Jika diteliti, jumlah kata yang sama-sama muncul pada kedua ringkasan ada 5 kata, yaitu kata *start*, *police*, *deploy*, *drones*, dan *end*. Setelah mengetahui jumlah kata yang sama antar dua ringkasan, maka jumlah kata tersebut dihitung nilai *recall*, *precision*, dan *f1-score*-nya. Perhitungannya sebagai berikut:

$$Recall = \frac{\text{Jumlah kata yang sama}}{\text{banyak kata pada ringkasan manual}} = \frac{5}{7} = 0.714$$

$$Precision = \frac{\text{Jumlah kata yang sama}}{\text{banyak kata pada ringkasan prediksi}} = \frac{5}{6} = 0.833$$

$$F1_{score} = \frac{2 * (Precision * Recall)}{Precision + Recall} = \frac{2 * (0.833 * 0.714)}{(0.833 + 0.714)} = 0.769$$

Dari rumus diatas dapat dilihat bahwa nilai *recall* didapat dari jumlah kata yang sama antar dua ringkasan dibagi dengan jumlah kata pada ringkasan manual. Sedangkan nilai *precision* didapat dari jumlah kata yang sama antar dua ringkasan dibagi dengan jumlah kata pada ringkasan prediksi. Kemudian nilai *f1-score* didapat dari hasil rata-rata *precision* dan *recall* dibagi dengan hasil dari penjumlahan nilai *precision* dan *recall*. Dilihat dari perhitungan nilai *recall*, *precision*, dan *f1-score* diatas, diperoleh nilai *recall* sebesar 0.714, *precision* sebesar 0.833, dan *f1-score* sebesar 0.769. Nilai tersebut sudah sesuai dengan hasil perhitungan sistem pada berita ketiga LSTM skenario pertama.

3.5 Skenario Pengujian

Data teks berita berbahasa Inggris akan melalui tahap *preprocessing* untuk mengubah data mentah menjadi data yang siap dianalisis, selanjutnya akan diuji dan diolah menggunakan metode *TextRank* dan LSTM. Pada metode *TextRank* hanya dilakukan satu kali percobaan dikarenakan tidak adanya *hyperparameter* pada *library TextRank* yang dipakai. Sedangkan, pada metode LSTM akan dilakukan beberapa percobaan yaitu dengan mengkonfigurasi pembagian dataset sebagai *training* dan *testing set* untuk mendapatkan hasil yang terbaik. Konfigurasi pembagian dataset dilakukan sebanyak tiga kali, yaitu:

1. Membagi dataset menjadi 70% data *training* dan 30% data *testing*.
2. Membagi dataset menjadi 80% data *training* dan 20% data *testing*.
3. Membagi dataset menjadi 90% data *training* dan 10% data *testing*.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Tahap Pengujian

Sebelum mendapatkan hasil akurasi yang diinginkan dari pengujian sistem peringkasan teks ini, ada beberapa tahap yang perlu dilakukan untuk mencapai hasil yang diinginkan, diantaranya:

1. Melakukan *preprocessing* terhadap dataset *news_summary*. Pada dataset tersebut terdapat kolom “text” yang berisi ringkasan manual dan kolom “ctext” yang berisi teks berita asli.
2. Pada pengujian menggunakan metode LSTM, dataset dibagi menjadi data latih dan data validasi. Terdapat tiga skenario pembagian data dengan rasio perbandingan 90% data latih : 10% data validasi, 80% data latih : 20% data validasi, dan 70% data latih : 30% data validasi. Pembagian dataset menggunakan fungsi *train_test_split()*.
3. Menguji performa metode *TextRank* dan LSTM menggunakan metode evaluasi ROUGE-1 yang mencakup nilai *recall*, *precision*, dan *f1-score*. Hasil evaluasi didapatkan melalui perbandingan hasil ringkasan sistem dengan ringkasan manual yang sudah tersedia di dataset.
4. Menganalisis dan membandingkan performa antara metode *TextRank* dan LSTM dengan berbagai skenario untuk mendapatkan hasil yang terbaik.

4.2 Hasil Pengujian

Pengujian dilakukan menggunakan teknik ekstraktif dan abstraktif, pada teknik ekstraktif menggunakan metode *TextRank*, sedangkan pada teknik abstraktif menggunakan metode LSTM. Hasil perhitungan akurasi dua metode tersebut nantinya dibandingkan untuk melihat metode yang terbaik dalam menghasilkan ringkasan teks.

4.2.1 Pengujian Menggunakan Metode *TextRank*

Pada pengujian menggunakan metode *TextRank*, data yang telah melalui tahap *preprocessing* selanjutnya dihitung *similarity*-nya menggunakan metode *cosine similarity*. Hasil dari perhitungan *similarity* tersebut dikonversi menjadi graf kalimat yang disusun menjadi hasil ringkasan sistem. Tabel 4.1 menunjukkan hasil dari perbandingan panjang teks asli dan teks hasil peringkasan sistem.

Tabel 4.1 Hasil pengujian menggunakan *TextRank*

Teks berita asli	<i>Weeks after ex-CBI Director Alok Verma told the Department of Personnel and Training to consider him retired, the Home Ministry asked him to join work on the last day of his fixed tenure as Director on Thursday. The ministry directed him to immediately join as DG, Fire Services, the post he was transferred to after his removal as CBI chief.</i>
Ringkasan sistem	<i>alok verma told department personnel training consider retired home ministry asked join work last day fixed tenure director Thursday. Ministry directed immediately join dg fire services post transferred removal cbi chief.</i>

Pada Tabel 4.2 terlihat perbedaan jumlah kata dan kalimat pada teks berita asli dan hasil ringkasan sistem yang menunjukkan bahwa sistem berhasil membuat ringkasan dari teks berita.

Tabel 4.2 Perbedaan jumlah kata dan kalimat sebelum dan sesudah diringkaskan

	Jumlah kata	Jumlah kalimat
Teks berita asli	60	3

	Jumlah kata	Jumlah kalimat
Ringkasan sistem	34	2

Setelah sistem berhasil membuat ringkasan, selanjutnya adalah mengevaluasi hasil ringkasan sistem dengan membandingkannya dengan ringkasan manual yang telah tersedia di dalam dataset. ROUGE-1 dipilih sebagai metode evaluasi dengan melakukan perhitungan *recall*, *precision*, dan *f1-score*. Evaluasi pada metode *TextRank* dilakukan pada 13.760 data berita yang telah dirangkum oleh sistem. Nilai rata-rata *recall*, *precision*, dan *f1-score* ditunjukkan pada Tabel 4.3.

Tabel 4.3 Skor evaluasi ROUGE-1 tiap berita

Berita	<i>Recall</i>	<i>Precision</i>	<i>F1-Score</i>
Berita 1	0.283	0.416	0.337
Berita 2	0.454	0.641	0.531
Berita 3	0.420	0.677	0.518
Berita 4	0.347	0.571	0.432
Berita 5	0.293	0.566	0.386
...
Berita 13756	0.396	0.65	0.494
Berita 13757	0.346	0.68	0.459
Berita 13758	0.392	0.625	0.481
Berita 13759	0.255	0.4	0.311
Berita 13760	0.304	0.424	0.354
Rata-rata	0.383	0.608	0.469
Nilai terendah	0.044	0.068	0.054
Nilai tertinggi	1.0	1.0	1.0

Pada tabel diatas menunjukkan nilai *recall*, *precision*, dan *f1-score* dari tiap berita. Rata-rata nilai *recall*, *precision*, dan *f1-score* yang dihasilkan dari pengujian

13.760 data berita secara berurutan adalah 0.383, 0.608, dan 0.469. Nilai *recall*, *precision*, dan *f1-score* terendah yang dihasilkan secara berurutan adalah 0.044, 0.068, dan 0.054. Sedangkan untuk nilai tertinggi dari *recall*, *precision*, dan *f1-score* adalah rata di angka 1.0.

4.2.2 Pengujian Menggunakan Metode *Long Short Term Memory*

Pengujian kedua dilakukan menggunakan metode LSTM. Model yang digunakan dalam pengujian ini memiliki 3 *stack layer* LSTM, dimana setiap *layer* memiliki jumlah *node* yang dipakai sebanyak 400 *node*. Pengujian pada metode ini menggunakan 3 skenario pembagian data yaitu 90% data training – 10% data testing, 80% data training – 20% data testing, dan 70% data training – 30% data testing. Proses *preprocessing* yang dilakukan pada pengujian metode ini sama dengan yang telah dilakukan pada metode *TextRank*.

Beberapa parameter yang ditentukan dalam pembentukan model dan dipakai dalam semua skenario diantaranya sebagai berikut:

- a. *Batch size* = 1024

Batch size adalah variabel yang menentukan ukuran data pelatihan yang digunakan dalam sesi pelatihan. Proses pelatihan bergerak lebih cepat dengan semakin rendahnya nilai ukuran *batch* yang digunakan. Sementara itu, karena pelatihan membutuhkan kapasitas penyimpanan yang lebih besar, semakin besar angka ukuran *batch*, semakin lama proses pelatihan akan berlangsung. Hal ini juga berdampak pada akurasi sistem. Karena algoritma akan mempelajari lebih banyak fitur, akurasi

akan meningkat dengan pengaturan ukuran *batch* yang lebih besar (Kusuma *et al.*, 2021).

b. *Epoch* = 100

Istilah “*epoch*” mengacu pada titik di mana semua dataset input untuk model sistem telah menyelesaikan satu putaran *forward propagation* dan *back propagation* melalui *neural network* (Faiz Nashrullah *et al.*, 2020).

c. *Optimizer* = Adam

Adam adalah sebuah metode optimasi berbasis gradien stokastik yang menyimpan rata-rata momentum gradien dan kuadrat momentum gradien untuk setiap parameter. Adam adalah kombinasi dari *optimizer* RMSprop dan *Stochastic Gradient Descent* (SGD) (Arkadia *et al.*, 2022).

d. *Dropout size* = 0.4

Dropout merupakan pendekatan pengaturan jaringan saraf yang digunakan untuk mencegah terjadinya *overfitting* dengan cara memilih neuron tertentu secara acak untuk dibuang dan tidak digunakan selama proses pelatihan. Hal ini berarti bahwa selama *backpropagation*, bobot baru tidak ditambahkan ke neuron dan jaringan akan berhenti menerima kontribusi dari neuron yang dibuang untuk sementara waktu (Khumaidi & Nirmala, 2022). *Dropout* memiliki ukuran 0.4 yang berarti sistem akan memilih 40% dari total neuron secara acak untuk dinonaktifkan selama proses pelatihan.

e. 4 *hidden state* dengan rincian 3 *hidden state* pada *encoder* dan 1 *hidden state* pada *decoder*

Hidden state adalah representasi internal jaringan yang mengangkut informasi dari satu langkah waktu ke langkah berikutnya. Hal ini memungkinkan LSTM untuk menyimpan informasi tentang urutan data sebelumnya dan menggunakan informasi tersebut untuk menganalisis data saat ini.

- f. Ukuran *vector input* = 60

Vector input merujuk pada representasi numerik dari teks yang diberikan sebagai input ke model. Ini adalah urutan vektor yang mewakili kata-kata dalam teks. Dalam penelitian ini, *vector input* yang dimaksud adalah parameter *max_text_len* dengan nilai 60.

Pengujian skenario pertama membagi data menjadi 90% data *training* (49.114 data) dan 10% data *testing* (5.458 data) dari data keseluruhan sebanyak 54.572 data teks. Hasil evaluasi ditunjukkan pada Tabel 4.4.

Tabel 4.4 Skor evaluasi ROUGE-1 skenario pertama

Berita	<i>Recall</i>	<i>Precision</i>	<i>F1-Score</i>
Berita 1	0.625	0.555	0.588
Berita 2	0.571	0.8	0.666
Berita 3	0.571	0.571	0.571
Berita 4	0.375	0.5	0.428
Berita 5	0.25	0.333	0.285
...
Berita 5454	0.222	0.285	0.25
Berita 5455	0.75	0.857	0.8
Berita 5456	0.75	0.6	0.666
Berita 5457	0.285	0.25	0.266
Berita 5458	0.444	0.571	0.5

Berita	Recall	Precision	F1-Score
Rata-rata	0.462	0.507	0.480
Nilai terendah	0.153	0.166	0.173
Nilai tertinggi	1.0	1.0	1.0

Tabel 4.4 menunjukkan hasil perhitungan nilai *recall*, *precision*, dan *f1-score* dari LSTM skenario pertama, dengan jumlah data yang di uji adalah 5.458 data. Nilai *recall*, *precision*, dan *f1-score* yang diperoleh secara berurutan adalah 0.462, 0.507, dan 0.480. Nilai *recall*, *precision*, dan *f1-score* terendah yang dihasilkan secara berurutan adalah 0.153, 0.166, dan 0.173. Sedangkan untuk nilai tertinggi dari *recall*, *precision*, dan *f1-score* adalah rata di angka 1.0.

Tabel 4.5 Indeks berita tertinggi pada LSTM skenario pertama

Matriks	Indeks berita	Aktual	Prediksi
<i>Recall</i>	57	<i>start big b corrects alias spelling end</i>	<i>start big b corrects alias spelling pic end</i>
<i>Precision</i>	69	<i>start sunny leone husband adopt baby girl maharashtra end</i>	<i>start sunny leone husband adopt baby girl end</i>
<i>F1-Score</i>	165	<i>start govt conducting raids aiadm stalin end</i>	<i>start govt conducting raids aiadm stalin end</i>

Pada Tabel 4.5 merupakan indeks berita yang memiliki nilai *recall*, *precision*, dan *f1-score* tertinggi. Dapat dilihat bahwa ringkasan prediksi yang dihasilkan oleh sistem pada indeks berita tersebut sangat mendekati dengan data aktual.

Tabel 4.6 Indeks berita terendah pada LSTM skenario pertama

Matriks	Indeks berita	Aktual	Prediksi
<i>Recall</i>	1605	<i>start courts ko aur nahi hai kya swara pandya rahuls sc hearing end</i>	<i>start sehswag dances sonu nigam end</i>
<i>Precision</i>	4364	<i>start would like son abram play field hockey india srk end</i>	<i>start shah rukh khan meets shah rukh khan win th birthday end</i>

Matriks	Indeks berita	Aktual	Prediksi
<i>F1-Score</i>	4364	<i>start would like son abram play field hockey india srk end</i>	<i>start shah rukh khan meets shah rukh khan win th birthday end</i>

Pada Tabel 4.6 merupakan indeks berita yang memiliki nilai *recall*, *precision*, dan *f1-score* terendah. Dapat dilihat bahwa ringkasan prediksi yang dihasilkan oleh sistem pada indeks berita tersebut berbeda dengan data aktual. Pengujian skenario kedua membagi data menjadi 80% data *training* (43.657 data) dan 20% data *testing* (10.915 data) dari data keseluruhan sebanyak 54.572 data teks. Hasil evaluasi ditunjukkan pada Tabel 4.7.

Tabel 4.7 Skor evaluasi ROUGE-1 skenario dua

Berita	<i>Recall</i>	<i>Precision</i>	<i>F1-Score</i>
Berita 1	0.444	0.5	0.470
Berita 2	0.222	0.222	0.222
Berita 3	0.4	0.444	0.421
Berita 4	0.5	0.444	0.470
Berita 5	0.285	0.25	0.266
...
Berita 10911	0.5	0.6	0.545
Berita 10912	0.375	0.428	0.4
Berita 10913	0.25	0.25	0.25
Berita 10914	0.4	0.444	0.421
Berita 10915	0.5	0.571	0.533
Rata-rata	0.441	0.484	0.458
Nilai terendah	0.153	0.133	0.166
Nilai tertinggi	1.0	1.0	1.0

Pada Tabel 4.7 merupakan hasil perhitungan nilai *recall*, *precision*, dan *f1-score* dari LSTM skenario kedua, dengan jumlah data yang di uji adalah 10.915

data. Nilai *recall*, *precision*, dan *f1-score* yang diperoleh secara berurutan adalah 0.441, 0.484, dan 0.458. Nilai *recall*, *precision*, dan *f1-score* terendah yang dihasilkan secara berurutan adalah 0.153, 0.133, dan 0.166. Sedangkan untuk nilai tertinggi dari *recall*, *precision*, dan *f1-score* adalah rata di angka 1.0.

Tabel 4.8 Indeks berita tertinggi pada LSTM skenario kedua

Matriks	Indeks berita	Aktual	Prediksi
<i>Recall</i>	153	<i>start filmmaker passes away end</i>	<i>start filmmaker passes away end</i>
<i>Precision</i>	153	<i>start filmmaker passes away end</i>	<i>start filmmaker passes away end</i>
<i>F1-Score</i>	153	<i>start filmmaker passes away end</i>	<i>start filmmaker passes away end</i>

Pada Tabel 4.8 merupakan indeks berita yang memiliki nilai *recall*, *precision*, dan *f1-score* tertinggi. Dapat dilihat bahwa ringkasan prediksi yang dihasilkan oleh sistem pada indeks berita tersebut sangat mendekati dengan data aktual.

Tabel 4.9 Indeks berita terendah pada LSTM skenario kedua

Matriks	Indeks berita	Aktual	Prediksi
<i>Recall</i>	5600	<i>start kareena rejected kal ho naa ho later rejected jab met preity end</i>	<i>start amushka said like aamir khan end</i>
<i>Precision</i>	3287	<i>start terrorism state policy one neighbours vp naidu end</i>	<i>start india must need peace sri lanka end</i>
<i>F1-Score</i>	3287	<i>start terrorism state policy one neighbours vp naidu end</i>	<i>start india must need peace sri lanka end</i>

Pada Tabel 4.9 merupakan indeks berita yang memiliki nilai *recall*, *precision*, dan *f1-score* terendah. Dapat dilihat bahwa ringkasan prediksi yang dihasilkan oleh sistem pada indeks berita tersebut sangat berbeda dengan data aktual. Pengujian skenario ketiga membagi data menjadi 70% data *training* (38.200 data) dan 30%

data *testing* (16.372 data) dari data keseluruhan sebanyak 54.572 data teks. Hasil evaluasi ditunjukkan pada Tabel 4.10.

Tabel 4.10 Skor evaluasi ROUGE-1 skenario tiga

Berita	Recall	Precision	F1-Score
Berita 1	0.222	0.333	0.266
Berita 2	0.375	0.375	0.375
Berita 3	0.285	0.4	0.333
Berita 4	0.571	0.5	0.533
Berita 5	0.333	0.428	0.375
...
Berita 16368	0.375	0.375	0.375
Berita 16369	0.6	0.375	0.461
Berita 16370	0.2	0.285	0.235
Berita 16371	0.444	0.571	0.5
Berita 16372	0.3	0.428	0.352
Rata-rata	0.397	0.445	0.415
Nilai terendah	0.166	0.181	0.190
Nilai tertinggi	1.0	1.0	1.0

Pada Tabel 4.10 merupakan hasil perhitungan nilai *recall*, *precision*, dan *f1-score* dari LSTM skenario ketiga, dengan jumlah data yang di uji adalah 16.372 data. Nilai rata-rata *recall*, *precision*, dan *f1-score* yang diperoleh secara berurutan adalah 0.397, 0.445, dan 0.415. Nilai *recall*, *precision*, dan *f1-score* terendah yang dihasilkan secara berurutan adalah 0.166, 0.181, dan 0.190. Sedangkan untuk nilai tertinggi dari *recall*, *precision*, dan *f1-score* adalah rata di angka 1.0.

Tabel 4.11 Indeks berita tertinggi pada LSTM skenario ketiga

Matriks	Indeks berita	Aktual	Prediksi
<i>Recall</i>	281	<i>start indians banned online end</i>	<i>start indians banned online end</i>

Matriks	Indeks berita	Aktual	Prediksi
<i>Precision</i>	281	<i>start indians banned online end</i>	<i>start indians banned online end</i>
<i>F1-Score</i>	281	<i>start indians banned online end</i>	<i>start indians banned online end</i>

Pada Tabel 4.11 merupakan indeks berita yang memiliki nilai *recall*, *precision*, dan *f1-score* tertinggi. Dapat dilihat bahwa ringkasan prediksi yang dihasilkan oleh sistem pada indeks berita tersebut sama dengan data aktual.

Tabel 4.12 Indeks berita terendah pada LSTM skenario ketiga

Matriks	Indeks berita	Aktual	Prediksi
<i>Recall</i>	2267	<i>start aus players team coach end</i>	<i>start england players due world end</i>
<i>Precision</i>	9471	<i>start iran finds new routes smuggle weapons end</i>	<i>start iran declares sanctions venezuelan end</i>
<i>F1-Score</i>	8873	<i>start offers reward recover stolen cheese end</i>	<i>start us man steals fire stealing drugs end</i>

Pada Tabel 4.12 merupakan indeks berita yang memiliki nilai *recall*, *precision*, dan *f1-score* terendah. Dapat dilihat bahwa ringkasan prediksi yang dihasilkan oleh sistem pada indeks berita tersebut sangat berbeda dengan data aktual. Berikut perbandingan rata-rata skor *recall*, *precision*, dan *F1-score* dari pengujian tiga skenario diatas.

Tabel 4.13 Perbandingan rata-rata skor pengujian tiga skenario

Skenario ke-	<i>Recall</i>	<i>Precision</i>	<i>F1-Score</i>
1	0.462	0.507	0.480
2	0.441	0.484	0.458
3	0.397	0.445	0.415

Dari tabel diatas menunjukkan bahwa rata-rata skor *recall*, *precision*, dan *f1-score* tertinggi dari tiga skenario adalah pada skenario pertama yaitu sebesar

0.462, 0.507, dan 0.480 dengan pembagian data 90% data training dan 10% data testing.

4.3 Pembahasan

Penelitian ini menggunakan pendekatan ROUGE dalam melakukan perhitungan evaluasi untuk mendapatkan nilai *recall*, *precision*, dan *f1-score*. *Recall* menghitung tingkat relevansi informasi yang telah diambil oleh sistem dibandingkan dengan keseluruhan informasi yang ada. Informasi yang relevan adalah bagian teks berita yang dianggap penting dari teks asli yang harus ada dalam ringkasan. Nilai *recall* menunjukkan seberapa baik sistem peringkasan teks dalam menangkap semua informasi penting.

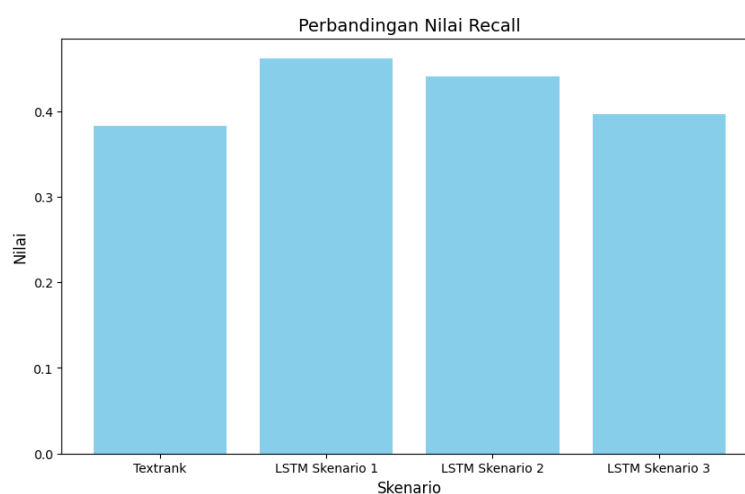
Precision mengukur banyak informasi relevan yang telah ditemukan oleh sistem. Nilai *precision* menunjukkan seberapa banyak teks hasil ringkasan yang relevan dengan teks aslinya. Kemudian *f1-score* adalah rata-rata dari nilai *recall* dan *precision* untuk mengevaluasi performa sistem peringkasan teks otomatis. Nilai *f1-score* menjadi tolak ukur sistem dapat dikatakan berhasil dalam meringkas teks atau tidak.

Tabel 4.14 Perbandingan rata-rata skor pengujian TextRank dan LSTM

Model	<i>Recall</i>	<i>Precision</i>	<i>F1-Score</i>
TextRank	0.383	0.608	0.469
LSTM (Skenario 1)	0.462	0.507	0.480
LSTM (Skenario 2)	0.441	0.484	0.458
LSTM (Skenario 3)	0.397	0.445	0.419

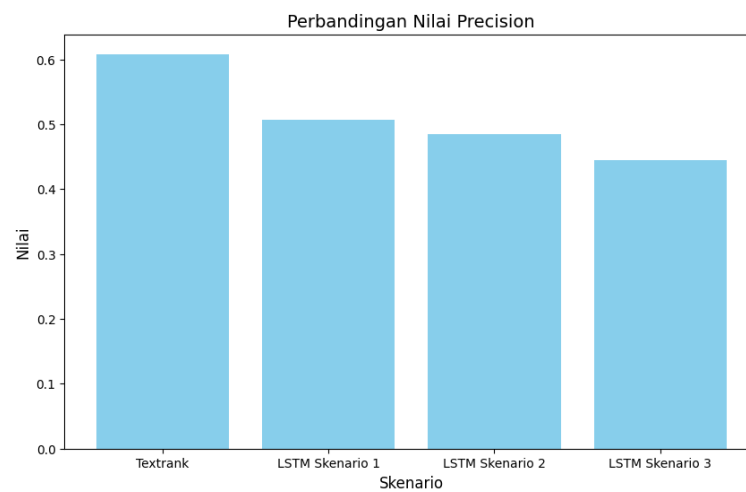
Pada Tabel 4.14 menunjukkan perbandingan nilai rata-rata *recall*, *precision*, dan *f1-score* antara metode *TextRank* dan LSTM. Nilai rata-rata *recall* tertinggi

dihasilkan oleh LSTM skenario pertama yaitu sebesar 0.462 dan nilai terendah dihasilkan oleh *TextRank* dengan nilai sebesar 0.383. Pada Gambar 4.1 merupakan grafik perbandingan nilai *recall* antara *TextRank* dan LSTM skenario 1-3. Grafik menunjukkan bahwa LSTM skenario 1 memiliki nilai tertinggi diantara pengujian yang lain, sedangkan *TextRank* memiliki nilai terendah diantara pengujian yang lain.



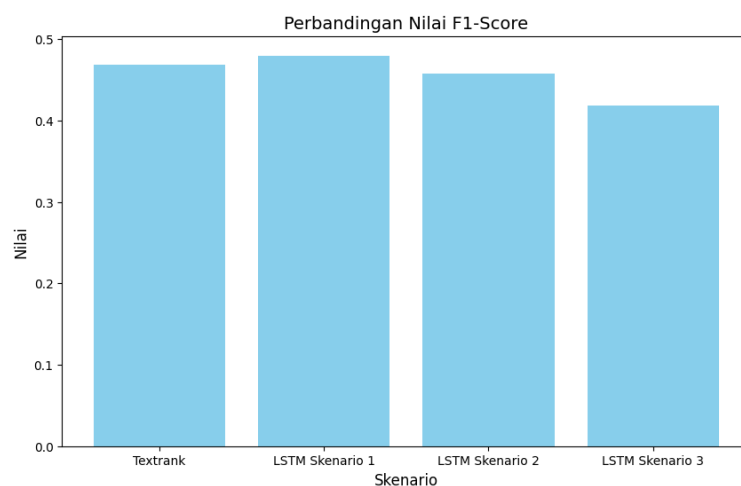
Gambar 4.1 Perbandingan Nilai Recall

Nilai *precision* tertinggi dihasilkan oleh *TextRank* yaitu sebesar 0.608 dan nilai terendah dihasilkan oleh LSTM skenario ketiga dengan nilai sebesar 0.445. Pada Gambar 4.2 menunjukkan grafik perbandingan nilai *precision* antara *TextRank* dan LSTM skenario 1 sampai 3. Grafik menunjukkan bahwa *TextRank* memiliki nilai tertinggi diantara pengujian yang lain, sedangkan LSTM skenario 3 memiliki nilai terendah diantara pengujian yang lain.



Gambar 4.2 Perbandingan Nilai Precision

Nilai *f1-score* tertinggi dihasilkan oleh LSTM skenario pertama yaitu sebesar 0.480 dan nilai terendah dihasilkan oleh LSTM skenario ketiga dengan nilai sebesar 0.419. Pada Gambar 4.3 merupakan grafik perbandingan nilai *f1-score* antara *TextRank* dan LSTM skenario 1-3. Grafik menunjukkan bahwa LSTM skenario 1 memiliki nilai tertinggi diantara pengujian yang lain, sedangkan LSTM skenario 3 memiliki nilai terendah diantara pengujian yang lain.



Gambar 4.3 Perbandingan Nilai F1-Score

Dilihat dari indeks data berita dengan skor akurasi tertinggi pada Tabel 4.5 untuk skenario pertama, Tabel 4.8 untuk skenario kedua, dan Tabel 4.11 untuk skenario ketiga. Jika diteliti, indeks berita yang mendapatkan nilai akurasi tertinggi memiliki jumlah kata yang lebih sedikit daripada indeks berita yang mendapatkan nilai akurasi terendah. Ini membuktikan bahwa model dapat lebih mudah membuat ringkasan dengan output yang lebih sedikit dan sederhana. Konsistensi konten berita juga berpengaruh terhadap kemampuan model dalam menghasilkan output ringkasan dengan nilai akurasi tinggi.

Penelitian yang dilakukan oleh Mercan *et al.*, membahas tentang peringkasan teks abstraktif dengan menggunakan LSTM dan model yang telah dilatih sebelumnya seperti BART, T5, dan PEGASUS. Penelitian ini menggunakan 4 dataset sebagai *benchmark*. Salah satu dari 4 dataset tersebut merupakan dataset berita berbahasa Inggris dengan jumlah data sebanyak 4.515 data yang mencakup informasi seperti nama penulis, judul, url artikel, teks pendek, dan artikel lengkap. Data diperoleh dengan mengumpulkan ringkasan berita dalam bentuk pendek dari artikel berita *The Hindu*, *Indian Times*, dan *The Guardian*. Referensi penelitian ini membahas berbagai topik yang berkaitan dengan peringkasan teks abstraktif dengan model pembelajaran mendalam (*deep learning*). Pada penilaian skor dataset teks berita berbahasa Inggris, skor ROUGE-1 tertinggi diperoleh model BART-Large dengan *Recall* 0.52, *Precision* 0.48, dan *F1-Score* 0.50. Sedangkan untuk model LSTM sendiri mendapatkan nilai terkecil dari model lainnya dengan nilai *Recall* sebesar 0.22, nilai *Precision* sebesar 0.70, dan *F1-Score* sebesar 0.32.

Models	News Summary								
	ROUGE 1			ROUGE 2			ROUGE L		
	P	R	F	P	R	F	P	R	F
LSTM	0,70	0,22	0,32	0,41	0,11	0,16	0,66	0,21	0,30
bart-base	0,42	0,43	0,42	0,20	0,21	0,20	0,30	0,31	0,31
bart-large	0,48	0,52	0,50	0,24	0,27	0,25	0,35	0,39	0,37
t5-base	0,95	0,25	0,37	0,83	0,22	0,32	0,86	0,23	0,34
pegasus-x-base	0,94	0,24	0,36	0,82	0,21	0,32	0,85	0,22	0,32

Gambar 4.4 Hasil penelitian Mercan et al.

Penelitian yang dilakukan oleh Turkey *et al.*, membahas tentang peringkasan teks dokumen COVID-19 menggunakan model *encoder-decoder* LSTM. Dataset yang digunakan adalah CORD-19 (*COVID-19 Open Research Dataset*) yang merupakan sumber untuk makalah tentang COVID-19 atau penyakit terkait. Dataset ini merupakan kumpulan dokumen *sains* yang terkait dengan virus corona. Lebih dari 59.000 makalah penelitian termasuk dalam kumpulan data ini, yang berisi lebih dari 47.000 makalah lengkap tentang COVID-19 atau penyakit terkait. Dalam penelitian ini, data yang digunakan sebanyak 12.000 sampel. Peneliti membandingkan hasil skor evaluasi dengan penelitian terdahulu yang menggunakan model yang berbeda. Pada Gambar 4.2 terlihat model LSTM+Glove yang digunakan oleh peneliti mendapatkan skor ROUGE-1 tertinggi dari model lainnya dengan nilai *Recall* sebesar 36.4%, nilai *Precision* sebesar 56.3%, dan nilai *F1-Score* sebesar 43.07%.

Methods	Rouge-1			Rouge-2			Rouge-L		
	P	R	F-1	P	R	F-1	P	R	F-1
ProphetNet (Zhengzhi Lou, 2020)	29.18	65.29	39.01	18.89	42.98	25.31	23.70	54.06	31.82
T5 (Zhengzhi Lou, 2020)	57.11	14.11	21.75	22.29	5.33	8.22	41.58	10.12	15.61
BART (Zhengzhi Lou, 2020)	31.15	11.04	15.61	8.54	2.79	4.02	26.06	9	12.8
Proposed method LSTM+Glove model	56.3	36.4	43.07	46.9	29.8	35.4	55..6	35.9	42.4

Gambar 4.5 Hasil penelitian Turkey *et al.*

Penelitian yang dilakukan oleh Pooja Raundale dan Himanshu Shekhar, membahas tentang peringkasan teks menggunakan *TextRank* dan *Seq2Seq* model. Pada penelitian ini, penulis membandingkan kinerja peringkasan ekstraktif dengan peringkasan abstraktif serta mempelajari pro dan kontra dari berbagai teknik yang digunakan dengan memanfaatkan penilaian ringkasan serta pemeriksaan manual terhadap ringkasan yang dihasilkan. Dataset yang digunakan pada penelitian ini adalah dataset CNN / Daily Mail *News Summary* yang berisi artikel berita online mencakup 287.226 data pelatihan, 13.368 data pengujian, dan 11.490 artikel validasi. Pada Tabel 4.15 terlihat metode *TextRank* mendapatkan skor rata-rata ROUGE-1 sebesar 39.83%, sedangkan skor rata-rata ROUGE-1 tertinggi didapatkan oleh Teknik *Lead 3 Baseline* sebesar 40.34%.

Tabel 4.15 Hasil pengujian Pooja Raundale dan Himanshu Shekhar

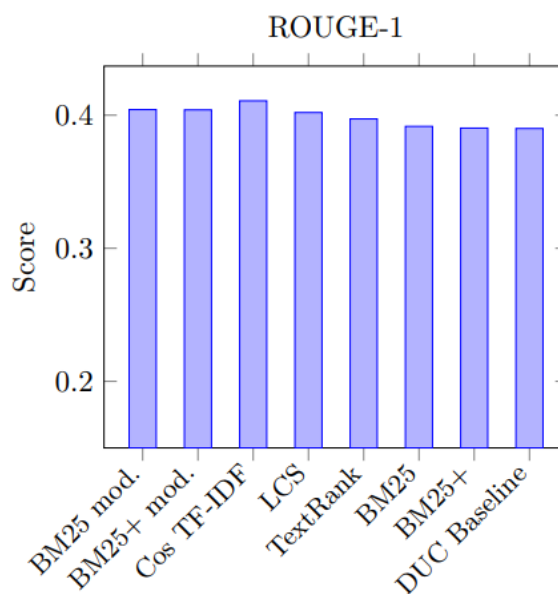
Technique	ROUGE-1 (%)	ROUGE-2 (%)	ROUGE-L (%)
TF-IDF	35.2	17.39	25.23
TextRank	39.83	17.62	19.48
Seq2Seq with Attention	31.33	15.66	33.42
Pointer Generator with Coverage	39.53	17.28	36.38
Lead 3 Baseline	40.34	17.70	36.57

Penelitian yang dilakukan oleh Barrios *et al.*, membahas tentang peringkasan teks menggunakan *TextRank* dengan beberapa variasi guna mendapatkan hasil

terbaik. Variasi yang digunakan diantaranya adalah penggunaan metode BM25, BM25+, LCS, *Cosine Sim*, dan *DUC Baseline*. Dataset yang digunakan penelitian ini adalah dataset dari *Document Understanding Conference (DUC)* tahun 2002. Korpus ini memiliki 567 dokumen yang diringkas menjadi 20% dari ukurannya. Dari Tabel 4.16 terlihat metode *TextRank* mendapatkan nilai rata-rata ROUGE-1 sebesar 0.3983, sedangkan nilai tertinggi dari semua metode yang dipakai diperoleh *Cosine TF-IDF* dengan nilai rata-rata ROUGE-1 sebesar 0.4108.

Tabel 4.16 Hasil pengujian *Barrios et al.*

Method	ROUGE-1
BM25 mod.	0.4042
BM25+ mod.	0.404
Cosine TF-IDF	0.4108
Longest Common Substring (LCS)	0.402
TextRank	0.3983
BM25	0.3916
BM25+	0.3903
DUC Baseline	0.39



Gambar 4.6 Hasil penelitian *Barrios et al.*

Jika dilihat pada Tabel 4.13, hasil terbaik dari penelitian ini adalah pada skenario pertama metode LSTM dengan pembagian dataset menjadi 90% data latih dan 10% data uji. Skor ROUGE-1 yang didapatkan skenario pertama metode LSTM adalah 0.480. Alasan mengapa pada skenario pertama LSTM mampu mendapatkan skor ROUGE tertinggi, dikarenakan dengan menggunakan lebih banyak data untuk melatih model, model memiliki kesempatan yang lebih besar untuk mempelajari pola yang mungkin terdapat dalam data. Selain itu, konfigurasi nilai parameter *epoch* sebesar 100 dan *batch size* sebesar 1024 dapat membantu mengurangi kecenderungan *overfitting*. Sebuah model dikatakan *overfitting* jika model tersebut “sangat cocok” dengan data uji. Akibatnya, model tersebut berkinerja sangat baik pada data pelatihan tetapi buruk pada data pengujian yang baru atau yang sebelumnya tidak terlihat. *Overfitting* terjadi ketika model menafsirkan detail sporadis yang terlihat pada data pengujian sebagai pola yang berarti yang tidak berlaku untuk data baru atau data yang lebih luas. Model LSTM dengan *splitting* data latih sebesar 90% dan data uji sebesar 10%, *epoch* 100, dan *batch size* 1024 dapat menghasilkan ringkasan teks dengan nilai *recall* sebesar 0.462, *precision* sebesar 0.507, dan *f1-score* sebesar 0.480.

Pengaruh *splitting* data juga dibuktikan pada skenario kedua dan ketiga model LSTM, dimana ketika data latih diubah menjadi 80% dan data uji menjadi 20% pada skenario kedua, nilai rata-rata *recall* turun menjadi 0.441, *precision* 0.484, dan nilai *f1-score* menjadi 0.458. Pada skenario ketiga nilai akurasi turun secara signifikan ketika data latih diubah menjadi 70% dan data uji menjadi 30%. Nilai

rata-rata *recall*, *precision*, dan *f1-score* pada skenario ketiga secara berurutan adalah 0.397, 0.445, dan 0.419.

Pembagian data latih dan data uji sangat berguna sebagai parameter pengujian model. Model yang terlatih dengan baik adalah yang dapat men-generalisasi data baru secara akurat ditunjukkan oleh kinerja model yang konsisten. Pembagian jumlah data latih dan data uji menjadi penentu akurasi, sehingga kesalahan dalam menentukan komposisi kedua jenis data tersebut akan berdampak pada nilai akurasi dan presisi yang dihasilkan (Musu *et al.*, 2021). Oleh karena itu, penulis dapat menyimpulkan bahwa hasil akurasi yang dihasilkan pada penelitian ini sangat dipengaruhi oleh pembagian data latih dan data uji. Namun, bukan tidak mungkin parameter yang lain juga dapat mempengaruhi hasil nilai akurasi, oleh karena itu diperlukan adanya penelitian lebih lanjut untuk dapat mengoptimalkan kinerja model dan mendapatkan akurasi yang lebih tinggi lagi.

Berikut perbandingan hasil penelitian yang diperoleh oleh peneliti dibandingkan peneliti terdahulu:

Tabel 4.17 Perbandingan akurasi dengan peneliti terdahulu

Referensi	Model yang dipakai	Data yang dipakai	Konfigurasi Parameter	Hasil Akurasi Terbaik
(Mercan <i>et al.</i> , 2023)	LSTM, BART, T5, PEGASUS	<i>Xsum</i> , <i>CNN/Daily Mail</i> , <i>Amazon Fine Food Review</i> , <i>Inshort News Summary</i>	Konfigurasi <i>splitting</i> data menjadi data <i>training</i> , <i>testing</i> , dan <i>validation</i>	0.50 (BART-Large) 0.32 (LSTM)
(Turky <i>et al.</i> , 2021)	LSTM	CORD-19 (COVID-19 <i>Open Research Dataset</i>)	<i>Batch size</i> = 512, <i>Dropout rate</i> = 0.5, <i>Optimizer</i> = Adam, <i>Learning rate</i> = 0.001	43.07% (LSTM)

Referensi	Model yang dipakai	Data yang dipakai	Konfigurasi Parameter	Hasil Akurasi Terbaik
(Raundale, 2021)	<i>TextRank</i> , <i>Seq2Seq</i>	<i>CNN/Daily Mail News Dataset</i>	Penggunaan <i>Global Vector</i> sebagai <i>word embedding</i>	40.34 (<i>Lead 3 Baseline</i>) 39.83 (<i>TextRank</i>)
(Barrios & Federico, 2016)	<i>TextRank</i>	<i>Document Understanding Conference (DUC)</i>	Menggunakan beberapa variasi model, modifikasi nilai <i>e</i> pada parameter BM25	0.3983 (<i>TextRank</i>)
Peneliti	<i>TextRank</i> , LSTM	<i>Inshort News Summary</i>	<i>TextRank</i> : Penggunaan <i>Global Vector</i> sebagai <i>word embedding</i> LSTM: <i>Batch size</i> = 1024, <i>Dropout rate</i> = 0.4, <i>Optimizer</i> = Adam, <i>Epoch</i> = 100	0.469 (<i>TextRank</i>) 0.480 (LSTM)

Pada Tabel 4.17, dapat dilihat perbandingan akurasi dari berbagai model, konfigurasi parameter, dan dataset yang digunakan oleh peneliti terdahulu. Model yang digunakan diantaranya adalah *TextRank* dan LSTM. Sedangkan konfigurasi yang digunakan oleh peneliti terdahulu diantaranya adalah *splitting* data, konfigurasi parameter seperti *learning rate*, *batch size*, dan *dropout rate*, serta pemilihan *word embedding*. Hasil akurasi terbaik dari perbandingan diatas adalah sebesar 0.480 untuk LSTM, dan 0.469 untuk *TextRank*. Dari tabel diatas, dapat disimpulkan bahwa pemilihan model, konfigurasi parameter, dan penggunaan dataset dapat memberikan pengaruh signifikan terhadap peningkatan hasil akurasi.

Metode peringkasan teks otomatis ini diharapkan dapat membantu orang lain dengan menyederhanakan proses ekstraksi informasi penting dari dokumen teks yang panjang. Ringkasan yang dihasilkan oleh sistem juga dapat membantu orang lain mengatasi kendala ketika mereka membutuhkan informasi penting dengan

cepat. Manfaat dari adanya sistem peringkasan sistem ini sesuai dengan hadist riwayat Muslim sebagai berikut:

عَنْ أَبِي هُرَيْرَةَ رَضِيَ اللَّهُ عَنْهُ عَنِ النَّبِيِّ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ قَالَ مَنْ نَفَسَ عَنْ مُؤْمِنٍ كُرْبَةً مِنْ كُرْبِ الدُّنْيَا ، نَفَسَ اللَّهُ عَنْهُ كُرْبَةً مِنْ كُرْبِ يَوْمِ الْقِيَامَةِ ، وَمَنْ يَسَّرَ عَلَى مُعْسِرٍ ، يَسَّرَ اللَّهُ عَلَيْهِ فِي الدُّنْيَا وَالْآخِرَةِ ، وَمَنْ سَتَرَ مُسْلِمًا ، سَتَرَهُ اللَّهُ فِي الدُّنْيَا وَالْآخِرَةِ ، وَاللَّهُ فِي عَوْنِ الْعَبْدِ مَا كَانَ الْعَبْدُ فِي عَوْنِ أَخِيهِ

“Dari Abu Hurairah radliyallahu anhu, Nabi shallallahu ‘alaihi wa sallam bersabda, “siapa yang membantu menghilangkan satu kesusahan dari sebagian banyak kesusahan orang mukmin ketika didunia maka Allah akan menghilangkan satu kesusahan dari sekian banyak kesusahan dirinya pada hari kiamat kelak. Dan barangsiapa yang memberikan kemudahan (urusan) kepada orang yang kesusahan, niscaya Allah akan membantu memudahkan urusannya didunia dan di akhirat. Dan barangsiapa yang menutup aib orang muslim, niscaya Allah akan menutup aibnya dunia dan akhirat. Sesungguhnya Allah akan selalu menolong seorang hamba selama dia gemar menolong saudaranya...” (HR Muslim).

Berdasarkan hadist diatas, amalan yang dicintai Allah SWT adalah memudahkan urusan dan melapangkan kesusahan orang lain serta saling tolong-menolong sesama mukmin dalam kebaikan. Tolong-menolong sendiri merupakan hal yang sangat dianjurkan oleh Allah SWT dan Rasulullah SAW. Setiap orang yang beriman memiliki kewajiban untuk membantu karena hal ini merupakan tindakan yang mewujudkan prinsip *muamalah ma'a an-nas*. Di mana pengguna dapat memperoleh informasi dengan lebih cepat dengan bantuan teknologi peringkasan teks. Hal ini juga sejalan dengan pengertian *muamalah ma'a Allah*. Dengan adanya sistem ringkasan teks ini, diharapkan dapat mempermudah dan membantu orang lain untuk mendapatkan informasi. Tujuan dari ringkasan adalah untuk memberikan gambaran umum yang ringkas kepada pembaca tentang sebuah teks sehingga mereka dapat dengan cepat memahami poin-poin penting yang ada

di dalam teks tersebut. Perintah membaca sendiri terdapat pada Firman Allah SWT dalam QS. Al-‘Alaq ayat 1-5:

إِقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ ۝١ خَلَقَ الْإِنْسَانَ مِنْ عَلَقٍ ۝٢ اقْرَأْ وَرَبُّكَ الْأَكْرَمُ ۝٣ الَّذِي عَلَّمَ بِالْقَلَمِ ۝٤ عَلَّمَ الْإِنْسَانَ مَا لَمْ يَعْلَمْ ۝٥

“1) Bacalah dengan (menyebut) nama Tuhanmu yang menciptakan, 2) Dia telah menciptakan manusia dari segumpal darah. 3) Bacalah, dan Tuhanmulah Yang Mahamulia, 4) Yang mengajar (manusia) dengan pena. 5) Dia mengajarkan manusia apa yang tidak diketahuinya.” (QS. Al-‘Alaq:1-5)

Berdasarkan Tafsir Jalalain, pada ayat pertama Allah memerintahkan manusia membaca dan memulai dengan menyebut nama Allah SWT. Pada ayat ketiga Allah kembali meminta manusia membaca sebagai pengukuhan makna lafal pertama yang sama, pada ayat kelima, lafal { عَلَّمَ الْإِنْسَانَ } mengandung arti mengajarkan kepada manusia suatu hal yang tidak diketahui oleh manusia. Ayat tersebut menganjurkan manusia untuk selalu membaca, menelaah, dan meneliti setiap apa yang dibacanya. Ayat tersebut juga mendorong manusia untuk belajar sebanyak mungkin. Mencari ilmu itu sendiri merupakan kewajiban yang harus dilakukan oleh semua Muslim. Hal ini juga disebutkan oleh Rasulullah SAW dalam sebuah hadits yang berbunyi, "*Mencari ilmu adalah kewajiban setiap muslim, dan siapa yang menanamkan ilmu kepada yang tidak layak seperti yang meletakkan kalung permata, mutiara, dan emas di sekitar leher hewan.*" (HR Ibnu Majah).

Dari penjelasan tafsir diatas dapat disimpulkan bahwa membaca sangat penting untuk mendapatkan dan mengembangkan ilmu. Saat ini, artikel *online* sangat banyak diminati karena dapat dengan mudah didapatkan. Seseorang dapat memperoleh informasi dalam bentuk pengetahuan praktis dengan membaca artikel

online. Namun, dengan banyaknya artikel *online* yang tersebar sangat rentan dengan informasi *hoax* yang terdapat pada artikel tersebut. Oleh karena itu, dalam Islam sendiri sangat dianjurkan untuk mencari kebenaran dan meneliti informasi yang didapatkan, sebagaimana firman Allah SWT dalam QS. Al-Hujurat ayat 6:

يَا أَيُّهَا الَّذِينَ آمَنُوا إِن جَاءَكُمْ فَاسِقٌ بِنَبَأٍ فَتَبَيَّنُوا أَن تُصِيبُوا قَوْمًا بِجَهَالَةٍ فَتُصْحَبُوا عَلَىٰ مَا فَعَلْتُمْ نَادِمِينَ

”Wahai orang-orang yang beriman, jika seorang fasik datang kepadamu membawa berita penting, maka telitilah kebenarannya agar kamu tidak mencelakakan suatu kaum karena ketidaktahuan(-mu) yang berakibat kamu menyesali perbuatanmu itu.” (QS. Al-Hujurat:6)

Dalam Tafsir Jalalain dijelaskan bahwa Allah memberi peringatan kepada orang mukmin, jika telah datang seorang fasik yang memberi berita maka periksalah terlebih dahulu kebenarannya, apakah berita tersebut benar atau dusta. Karena dikhawatirkan jika kita tidak meneliti terlebih dahulu berita yang datang, maka hal tersebut akan menimpa musibah kepada suatu kaum yang menyebabkan penyesalan atas perbuatan kalian sendiri. Mencari kebenaran juga harus didasari dengan ilmu dan iman yang kuat agar kita tidak mudah untuk terjerumus kedalam informasi *hoax*, dan kita dianjurkan untuk saling membantu dalam mencari ilmu sebagaimana pada QS. Al-Mujadilah ayat 11:

يَا أَيُّهَا الَّذِينَ آمَنُوا إِذَا قِيلَ لَكُمْ تَفَسَّحُوا فِي الْمَجَالِسِ فَافْسَحُوا يَفْسَحِ اللَّهُ لَكُمْ وَإِذَا قِيلَ انشُرُوا فَانشُرُوا يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ

“Wahai orang-orang yang beriman, apabila dikatakan kepadamu “Berilah kelapangan di dalam majelis-majelis,” lapangkanlah, niscaya Allah akan memberi kelapangan untukmu. Apabila dikatakan, “Berdirilah,” (kamu) berdirilah. Allah niscaya akan mengangkat orang-orang yang beriman di antaramu dan orang-orang yang diberi ilmu beberapa derajat. Allah Mahateliti terhadap apa yang kamu kerjakan.” (QS. Al-Mujadilah:11)

Berdasarkan Tafsir Jalalain, dijelaskan bahwa sesungguhnya apabila dikatakan kepada kalian “berlapang-lapanglah” dalam majelis, maka lapangkanlah, niscaya Allah akan memberi kelapangan untuk kalian di surga nanti. Dan apabila dikatakan kepada kalian “berdirilah kalian” untuk melakukan shalat dan amal kebaikan, niscaya Allah akan mengangkat derajat orang-orang yang beriman di antara kalian karena ketaatannya. Dan Dia mengangkat derajat pula orang-orang yang diberi ilmu pengetahuan di surga nanti. Dari ayat ini dapat dipahami bahwa orang-orang yang mempunyai derajat paling tinggi di sisi Allah ialah orang yang beriman dan berilmu. Ilmunya yang diamalkan sesuai dengan perintah Allah dan Rasul-Nya.

Oleh karena itu, sistem peringkasan teks otomatis ini diharapkan dapat membantu masyarakat untuk memahami poin-poin utama berita dengan lebih cepat, sesuai dengan tujuan peringkasan teks, yaitu mempermudah pembaca untuk menemukan informasi (pengetahuan) penting dalam sebuah teks.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dari uji coba metode *TextRank* dan LSTM dalam sistem peringkasan teks berita berbahasa Inggris didapatkan nilai rata-rata tertinggi dihasilkan oleh metode LSTM pada skenario pertama yang memiliki konfigurasi pembagian data 90% data *training* dan 10% data *testing* dengan nilai rata-rata *recall*, *precision*, dan *f1-score* secara berurutan adalah 0.462, 0.507, dan 0.480. Pada ujicoba skenario LSTM dengan konfigurasi pembagian data latih dan data uji, didapatkan bahwa semakin banyak data yang digunakan untuk melatih model, maka semakin akurat teks ringkasan yang dihasilkan dan semakin tinggi skor akurasi yang didapatkan. Namun, jika dilihat dari hasil ringkasan tiap indeks berita, terdapat beberapa indeks yang mendapatkan nilai akurasi sangat kecil yang menandakan bahwa model peringkasan dapat disempurnakan lagi. Jika ditarik kesimpulan berdasarkan uji coba yang telah dilakukan, metode LSTM menjadi yang terbaik dalam melakukan peringkasan teks berita Bahasa Inggris, dengan catatan, nilai rata-rata tersebut masih tergolong rendah dikarenakan nilainya masih dibawah 0.5 yang menandakan bahwa konfigurasi model dan parameter masih bisa ditingkatkan untuk mencapai nilai akurasi yang lebih tinggi.

5.2 Saran

Dari analisa peneliti, sistem peringkasan teks otomatis yang telah dikembangkan ini masih memiliki kekurangan yang dapat diperbaiki. Oleh karena

itu, penulis membuat beberapa rekomendasi yang dapat dipertimbangkan untuk penelitian selanjutnya:

1. Menggunakan ekstraksi fitur yang lebih baik pada model LSTM, seperti penggunaan *word embedding* GloVe atau *sentence embedding* seperti BERT guna menghasilkan representasi kata yang lebih informatif dan sesuai dengan makna aslinya.
2. Penyesuaian *hyperparameter tuning* untuk mendapatkan performa peringkasan yang lebih baik.

DAFTAR PUSTAKA

- Aisyah, K. N., Anggraini, S., & Arifin, A. Z. (2019). Peringkasan Teks Multi-Dokumen Berdasarkan Metode Sentence Extraction Dan Word Sense Disambiguation. *NJCA (Nusantara Journal of Computers and Its Applications)*, 4(1), 1–8. <https://doi.org/10.36564/njca.v4i1.89>
- Alfhi Saputra, M. (2021). Peringkas Teks Otomatis Bahasa Indonesia secara Abstraktif Menggunakan Metode Long Short-Term Memory. *E-Proceeding of Engineering : Vol.8, No.2 April 2021* |, 8(2), 3474–3488.
- Alpiani, A. S. (2019). *Pointer Generator*. 4(August), 169–176. <https://doi.org/10.21108/indojc.2019.4.2.341>
- Ammar, A. N., & Suyanto, S. (2020). Peringkasan Teks Ekstraktif Menggunakan Binary Firefly Algorithm. *Indonesia Journal on Computing (Indo-JC)*, 5(2), 31–42. <https://doi.org/10.21108/indojc.2020.5.2.440>
- Andriani, D., Indriati, & Furqon, M. T. (2019). Peringkasan Teks Otomatis Pada Artikel Berita Hiburan Berbahasa Indonesia Menggunakan Metode BM25. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(3), 2548–2964. <http://j-ptiik.ub.ac.id>
- Arkadia, A., Hananto, B., & Prasvita, D. S. (2022). Optimasi Long Short Term Memory Dengan Adam Menggunakan Data Udara Kota DKI Jakarta. *Seminar Nasional Mahasiswa Ilmu Komputer Dan Aplikasinya (SENAMIKA)*, 92–101.
- Balmuri, R. K., Mangu, V. R., Konda, S., & Gunda, M. (2021). Telugu Text Summerization Using Lstm Deep Learning. *La Pensee, September*, 355–363. <https://www.researchgate.net/publication/354623414>
- Barrios, F., & Federico, L. (2016). Variations of the Similarity Function of TextRank for Automated Summarization. *ArXiv Preprint ArXiv:1602.03606*.
- Faiz Nashrullah, Suryo Adhi Wibowo, & Gelar Budiman. (2020). The Investigation of Epoch Parameters in ResNet-50 Architecture for Pornographic Classification. *Journal of Computer, Electronic, and Telecommunication*, 1(1), 1–8. <https://doi.org/10.52435/complete.v1i1.51>
- Fakhrezi, M. F., Bijaksana, M. A., & Huda, A. F. (2021). Implementation of Automatic Text Summarization with TextRank Method in the Development of Al-Qur'an Vocabulary Encyclopedia. *Procedia Computer Science*, 179(2020), 391–398. <https://doi.org/10.1016/j.procs.2021.01.021>
- Halimah, Surya Agustian, & Siti Ramadhani. (2022). 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>

- Hernawan, Y. F., Adikara, P. P., & Wihandika, R. C. (2022). Peringkasan Artikel Berbahasa Indonesia Menggunakan TextRank dengan Pembobotan BM25. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 9(1), 61–68. <https://doi.org/10.25126/jtiik.2022913765>
- Khatri, C., Singh, G., & Parikh, N. (2018). *Abstractive and Extractive Text Summarization using Document Context Vector and Recurrent Neural Networks*. <http://arxiv.org/abs/1807.08000>
- Khumaidi, A., & Nirmala, I. A. (2022). Algoritma Long Short Term Memory Dengan Hyperparameter Tuning: Prediksi Penjualan Produk. In *Deepublish*.
- Kusuma, T. A. A. H., Usman, K., & Saidah, S. (2021). People Counting for Public Transportations Using You Only Look Once Method. *Jurnal Teknik Informatika (Jutif)*, 2(1), 57–66. <https://doi.org/10.20884/1.jutif.2021.2.2.77>
- Le, X. H., Ho, H. V., Lee, G., & Jung, S. (2019). Application of Long Short-Term Memory (LSTM) neural network for flood forecasting. *Water (Switzerland)*, 11(7). <https://doi.org/10.3390/w11071387>
- Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. *Association for Computational Linguistics*, 8, 74–81.
- Mehr, M. G. (2013). Using AdaBoost Meta-Learning Algorithm for Medical News Multi-Document Summarization. *Intelligent Information Management*, 05(06), 182–190. <https://doi.org/10.4236/iim.2013.56020>
- Mercan, O. B., Cavsak, S. N., Deliahmetoglu, A., & Tanberk, S. (2023). Abstractive Text Summarization for Resumes With Cutting Edge NLP Transformers and LSTM. *2023 Innovations in Intelligent Systems and Applications Conference, ASYU 2023*, 0–3. <https://doi.org/10.1109/ASYU58738.2023.10296563>
- Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into texts. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP 2004 - A Meeting of SIGDAT, a Special Interest Group of the ACL Held in Conjunction with ACL 2004*, 85, 404–411.
- Mohammed, S. M., Jacksi, K., & Zeebaree, S. R. M. (2020). Glove Word Embedding and DBSCAN algorithms for Semantic Document Clustering. *3rd International Conference on Advanced Science and Engineering, ICOASE 2020*, 211–216. <https://doi.org/10.1109/ICOASE51841.2020.9436540>
- Musu, W., Ibrahim, A., Studi, P., Informatika, T., Makassar, U. D., Studi, P., Informatika, M., & Makassar, U. D. (2021). Pengaruh Komposisi Data Training dan Testing terhadap Akurasi Algoritma C4 . 5. *PROSIDING SEMINAR ILMIAH SISTEM INFORMASI DAN TEKNOLOGI INFORMASI*, X(1), 186–195.
- Prabowo, D. A., Fhadli, M., Najib, M. A., Fauzi, H. A., & Cholissodin, I. (2016). TF-IDF-Enhanced Genetic Algorithm Untuk Extractive Automatic Text Summarization. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 3(3), 208.

<https://doi.org/10.25126/jtiik.201633217>

- Rahutomo, F., Kitasuka, T., & Aritsugi, M. (2012). Semantic cosine similarity. *The 7th International Student Conference on Advanced Science and Technology ICAST*, 4(1), 1. <https://www.researchgate.net/publication/262525676>
- Rani, U., & Bidhan, K. (2021). Comparative Assessment of Extractive Summarization: TextRank, TF-IDF and LDA. *Journal of Scientific Research*, 65(01), 304–311. <https://doi.org/10.37398/jsr.2021.650140>
- Raundale, P. (2021). Analytical study of Text Summarization Techniques. *2021 Asian Conference on Innovation in Technology (ASIANCON)*, 1–4. <https://doi.org/10.1109/ASIANCON51346.2021.9544804>
- Robiyanto, R., Nugraha, N., & Apriatna, I. (2019). Peringkasan Teks Otomatis Berita Menggunakan Metode Maximum Marginal Relevance. *JEJARING : Jurnal Teknologi Dan Manajemen Informatika*, 4(1), 23–32. <https://doi.org/10.25134/jejaring.v4i1.6712>
- Setiawan, A. Y., Darmawiguna, I. G. M., & Pradnyana, G. A. (2022). Sentiment Summarization Evaluasi Pembelajaran Menggunakan Algoritma LSTM (Long Short-Term Memory). *Kumpulan Artikel Mahasiswa Pendidikan Teknik Informatika (KARMAPATI)*, 11(2), 183–191.
- Tuhatussania, S. (2022). Automatic Text Summarization Artikel Berita Menggunakan Metode Maximum Marginal Relevance. *EXPLORE*, 12(2).
- Turky, S. N., AL-Jumaili, A. S. A., & Hasoun, R. K. (2021). Abstractive Text Summary of COVID-19 Documents based on LSTM Method and Word Embedding. *Webology*, 18(2), 1011–1022. <https://doi.org/10.14704/web/v18i2/web18370>
- Verma, P., Pal, S., & Om, H. (2019). A comparative analysis on Hindi and English extractive text summarization. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 18(3). <https://doi.org/10.1145/3308754>
- Zhang, H., Yan, Q., Zhang, G., & Jiang, Z. (2016). Theory, Methodology, Tools and Applications for Modeling and Simulation of Complex Systems. In *Communications in Computer and Information Science* (Vol. 643, Issue October). <http://link.springer.com/10.1007/978-981-10-2663-8>
- Zhang, M., Li, X., Yue, S., & Yang, L. (2020). An empirical study of textrank for keyword extraction. *IEEE Access*, 8, 178849–178858. <https://doi.org/10.1109/ACCESS.2020.3027567>
- Zhou, N., Shi, W., Liang, R., & Zhong, N. (2022). TextRank Keyword Extraction Algorithm Using Word Vector Clustering Based on Rough Data-Deduction. *Computational Intelligence and Neuroscience*, 2022. <https://doi.org/10.1155/2022/5649994>

LAMPIRAN

Lampiran I. Data Teks Berita Asli dan Ringkasan Manual

No Berita	Teks Berita	Ringkasan Manual
1	<p><i>Saurav Kant, an alumnus of upGrad and IIT-B's PG Program in Machine learning and Artificial Intelligence, was a Sr Systems Engineer at Infosys with almost 5 years of work experience. The program and upGrad's 360-degree career support helped him transition to a Data Scientist at Tech Mahindra with 90% salary hike. upGrad's Online Power Learning has powered 3 lakh careers.</i></p>	<p><i>upGrad learner switches to career in ML & AI with 90% salary hike</i></p>
2	<p><i>Kunal Shah's credit card bill payment platform, CRED, gave users a chance to win free food from Swiggy for one year. Pranav Kaushik, a Delhi techie, bagged this reward after spending 2000 CRED coins. Users get one CRED coin per rupee of bill paid, which can be used to avail rewards from brands like Ixigo, BookMyShow, UberEats, Cult.Fit and more.</i></p>	<p><i>Delhi techie wins free food from Swiggy for one year on CRED</i></p>
3	<p><i>New Zealand defeated India by 8 wickets in the fourth ODI at Hamilton on Thursday to win their first match of the five-match ODI series. India lost an international match under Rohit Sharma's captaincy after 12 consecutive victories dating back to March 2018. The match witnessed India getting all out for 92, their seventh lowest total in ODI cricket history.</i></p>	<p><i>New Zealand end Rohit Sharma-led India's 12-match winning streak</i></p>
4	<p><i>With Aegon Life iTerm Insurance plan, customers can enjoy tax benefits on your premiums paid and save up to on taxes. The plan provides life cover up to the age of 100 years. Also, customers have options to insure against Critical Illnesses, Disability and Accidental Death Benefit Rider with a life cover up to the age of 80 years.</i></p>	<p><i>Aegon life iTerm insurance plan helps customers save tax</i></p>
5	<p><i>Speaking about the sexual harassment allegations against Rajkumar Hirani, Sonam Kapoor said, "I've known Hirani for many years...What if it's not true, the movement will get derailed." "In the movement, I always believe a woman. But in this case, we need to reserve our judgment," she added. Hirani has been accused by an assistant who worked in 'Sanju'.</i></p>	<p><i>Have known Hirani for yrs, what if MeToo claims are not true: Sonam</i></p>
6	<p><i>Pakistani singer Rahat Fateh Ali Khan has denied receiving any notice from the Enforcement Directorate over allegedly smuggling foreign currency out of India. "It would have been better if the authorities would have served the notice first if any and then publicised this," reads a press release issued on behalf of Rahat. The statement further called the allegation "bizarre".</i></p>	<p><i>Rahat Fateh Ali Khan denies getting notice for smuggling currency</i></p>

No Berita	Teks Berita	Ringkasan Manual
7	<p>India recorded their lowest ODI total in New Zealand after getting all out for 92 runs in 30.5 overs in the fourth ODI at Hamilton on Thursday. Seven of India's batsmen were dismissed for single-digit scores, while their number ten batsman Yuzvendra Chahal top-scored with 18*(37). India's previous lowest ODI total in New Zealand was 108.</p>	<p>India get all out for 92, their lowest ODI total in New Zealand</p>
8	<p>Weeks after ex-CBI Director Alok Verma told the Department of Personnel and Training to consider him retired, the Home Ministry asked him to join work on the last day of his fixed tenure as Director on Thursday. The ministry directed him to immediately join as DG, Fire Services, the post he was transferred to after his removal as CBI chief.</p>	<p>Govt directs Alok Verma to join work 1 day before his retirement</p>
9	<p>Andhra Pradesh CM N Chandrababu Naidu has said, "When I met then US President Bill Clinton, I addressed him as Mr Clinton, not as 'sir'. (PM Narendra) Modi is my junior in politics...I addressed him as sir 10 times." "I did this...to satisfy his ego in the hope that he will do justice to the state," he added.</p>	<p>Called PM Modi 'sir' 10 times to satisfy his ego: Andhra CM</p>
10	<p>Congress candidate Shafia Zubair won the Ramgarh Assembly seat in Rajasthan, by defeating BJP's Sukhwant Singh with a margin of 12,228 votes in the bypoll. With this victory, Congress has taken its total to 100 seats in the 200-member assembly. The election to the Ramgarh seat was delayed due to the death of sitting MLA and BSP candidate Laxman Singh.</p>	<p>Cong wins Ramgarh bypoll in Rajasthan, takes total to 100 seats</p>
...
57	<p>Actress Mishti, who has featured in 'Manikarnika: The Queen of Jhansi', has said the film's co-director Kangana Ranaut made false promises to the cast. "Kangana had said, 'I want dates because all characters look jumbled up...I'm trying my best to give everybody a prominent space on screen.' After watching the movie, there is no scope for other characters," Mishti added.</p>	<p>Kangana made fake promises to cast: 'Manikarnika' actress Mishti</p>
...
69	<p>A 72-year-old man named Terry Sanderson has sued 'Avengers' actress Gwyneth Paltrow for allegedly crashing into him while skiing, and is seeking \$3.1 million (over ₹1.22 crore) in damages. The lawsuit states the actress got up and skied away after knocking the man down, leaving him with broken ribs. The incident reportedly took place in 2016 in Utah, US.</p>	<p>Man sues Avengers actress Gwyneth for crore over ski crash</p>

No Berita	Teks Berita	Ringkasan Manual
...
153	<p><i>Singer and stage performer Shivani Bhatia died in a road accident on Tuesday morning, after her car rammed into a divider on the Yamuna Expressway. The singer was accompanied by her husband Nikhil Bhatia, who survived the accident. "The accident occurred after Nikhil attempted to overtake a high-speed car and lost control of his vehicle," said police.</i></p>	<p><i>Singer Shivani Bhatia dies in car accident, husband survives</i></p>
...
165	<p><i>French sailor Jean-Luc Van Den Heede on Tuesday won the round-the-world yacht race after 212 days alone at sea without modern instruments. The 73-year-old, who completed his sixth circumnavigation of the globe, became the oldest to complete the 30,000-mile Golden Globe race. Of the 19 sailors who started out last July, only five were still in the race on Tuesday.</i></p>	<p><i>73-yr-old wins round-the-world sailing race after 212 days at sea</i></p>
...
281	<p><i>A video of a girl who slipped from a third floor balcony in China's Yunlong County has surfaced online. The video of the incident shows the girl, who was reportedly alone at home, hanging with her neck trapped in between railings. She was rescued by two men who borrowed a ladder and climbed up.</i></p>	<p><i>Girl slips from 3rd floor balcony, dangles by neck in China</i></p>
...
98393	<p><i>Retired US Navy Admiral Bruce Loveless and eight other current and former officers have been arrested for taking bribes in the form of lavish gifts, prostitutes and luxury hotel stays in exchange for classified information. The officers provided the information to a Malaysian defence contractor which helped it gain a competitive advantage in bidding for ship husbanding contracts.</i></p>	<p><i>Retired US Navy Admiral arrested in sex for secrets scandal</i></p>
98394	<p><i>Dinesh Shivnath Upadhyaya from Mumbai is recognised by the Guinness World Records for the record of 'most lit candles in the mouth' at 17. Upadhyaya set the record in July 2016 by putting commercially available candles in his mouth for a minimum of 30 seconds. He previously held the record for the same at 15 candles.</i></p>	<p><i>Indian has world record for most lit candles in mouth</i></p>

No Berita	Teks Berita	Ringkasan Manual
98395	<p>A Los Angeles-based startup has developed 'Paqsule' smart bag priced at A\$23,000 which claims to use UV light and activated oxygen to freshen up dirty clothes. The process can be activated through a button or remotely via a connected smartphone app. Paqsule claims a battery life of 72 hours and can also be used to charge smartphones.</p>	<p>Startup makes smart bag to freshen up dirty clothes</p>
98396	<p>The Enforcement Directorate has summoned Jammu and Kashmir Liberation Front chief Yasin Malik and separatist leader Syed Ali Shah Geelani in different money laundering cases. One of the cases pertains to the Income Tax Department's recovery of \$10,000 at Geelani's residence. Similarly, in 2001 police had seized \$100,000 from a Yasin Malik aide which was to be handed to him.</p>	<p>Kashmiri separatist leaders summoned over money laundering</p>
98397	<p>Pakistan has started building a nearly 2,400-kilometer fence along its border with Afghanistan, according to reports. This comes after Afghan militants reportedly killed six Pakistani soldiers stationed at security checkpoints earlier this month. Pakistan's new fence is aimed at enhancing aerial surveillance of the border to prevent future terror attacks and the transfer of weapons and ammunition.</p>	<p>Pakistan starts building fence along Afghanistan border</p>
98398	<p>A CRPF jawan was on Tuesday axed to death with sharp-edged weapons by Maoists at a local village fair in Chhattisgarh's insurgency-hit Bijapur district. As per preliminary information, Maoists attacked the jawan while he had gone to visit the fair along with his family, a police official said. A combing operation was launched to nab the assailants, he added.</p>	<p>CRPF jawan axed to death by Maoists in Chhattisgarh</p>
98399	<p>'Uff Yeh', the first song from the Sonakshi Sinha starrer upcoming drama film 'Noor' has been released. The song has been composed by Amaal Mallik with vocals by Armaan Malik and lyrics by Manoj Muntashir. Sonakshi will be seen portraying a Pakistani journalist-writer in the film. Directed by Sunhil Sippy, the film is scheduled to release on April 21.</p>	<p>First song from Sonakshi Sinha's 'Noor' titled 'Uff Yeh' out</p>
98400	<p>According to reports, a new version of the 1999 science fiction film 'The Matrix' is in development. Michael B Jordan will reportedly play the lead role in the film. Screenwriter Zak Penn is in talks to write the script of the film, reports added. Actor Keanu Reeves starred in the original film, which was followed by two sequels.</p>	<p>'The Matrix' film to get a reboot: Reports</p>

No Berita	Teks Berita	Ringkasan Manual
98401	<p><i>A new music video shows rapper Snoop Dogg aiming a toy gun at a clown character parodying US President Donald Trump. The video also shows a TV airing a news conference with the headline 'Ronald Klump wants to deport all doggs' airing live from 'The Clown House'. The video is for a remixed version of the song 'Lavender'.</i></p>	<p><i>Snoop Dogg aims gun at clown dressed as Trump in new video</i></p>
98402	<p><i>Madhesi Morcha, an alliance of seven political parties, has withdrawn support to PM Pushpa Kamal Dahal-led Nepal government after it failed to meet a seven-day ultimatum to fulfil their demands including endorsement for the revised Constitution amendment bill. The Morcha has 36 seats in the Parliament, but despite the withdrawal of support, there is no immediate threat to the government.</i></p>	<p><i>Madhesi Morcha withdraws support to Nepalese government</i></p>

Lampiran II. Flowchart Proses Pada LSTM

