

***MULTI DOCUMENT SUMMARIZATION MENGGUNAKAN ALGORITMA  
RECURRENT NEURAL NETWORK***

**SKRIPSI**

**Oleh:  
MOH. ALFIN  
NIM. 19650024**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2023**

***MULTI DOCUMENT SUMMARIZATION MENGGUNAKAN  
ALGORITMA RECURRENT NEURAL NETWORK***

**SKRIPSI**

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

**Oleh:  
MOH. ALFIN  
NIM. 19650024**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2023**

**HALAMAN PERSETUJUAN**

***MULTI DOCUMENT SUMMARIZATION MENGGUNAKAN  
ALGORITMA RECURRENT NEURAL NETWORK***

**SKRIPSI**

**Oleh:  
MOH ALFIN  
NIM. 19650024**

Telah Diperiksa dan Disetujui untuk Diuji  
Tanggal : 27 November 2023

Pembimbing I



Dr. Zainal Abidin, M.Kom  
NIP. 19760613 200501 1 004

Pembimbing II



Puspa Miladin N.S.A.B., M.Kom  
NIP. 19930828 201903 2 018

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



  
Dr. Fachrul Kurniawan, M.MT., IPM  
NIP. 19771020 200912 1 001

## HALAMAN PENGESAHAN

### **MULTI DOCUMENT SUMMARIZATION MENGGUNAKAN ALGORITMA RECURRENT NEURAL NETWORK**

#### SKRIPSI

Oleh:  
**MOH ALFIN**  
**NIM. 19650024**

Telah Dipertahankan di Depan Dewan Penguji  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
untuk Memperoleh Gelar Sarjana Komputer (S.Kom)  
Pada Tanggal: 11 Desember 2023

#### Susunan Dewan Penguji

- Ketua Penguji : Dr. M. Ainul Yaqin, M.Kom  
NIP. 19761013 200604 1 004
- Anggota Penguji I : A'la Syauqi, M.Kom  
NIP. 19771201 200801 1 007
- Anggota Penguji II : Dr. Zainal Abidin, M.Kom  
NIP. 19760613 200501 1 004
- Anggota Penguji III : Puspa Miladin N.S.A.B., M.Kom  
NIP. 19930828 201903 2 018



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



Dr. Fachrul Kurniawan, M.MT., IPM  
NIP. 19771020 200912 1 001

## PERNYATAAN KEASLIAN TULIAN

Saya yang bertanda tangan di bawah ini:

Nama : MOH. ALFIN

NIM : 19650024

Fakultas / Prodi : Sains dan Teknologi / Teknik Informatika

Judul Skripsi : *Multi Document Summarization Menggunakan Algoritma Recurrent Neural Network.*

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, 27 November 2023

Yang membuat pernyataan,



MOH. ALFIN  
NIM.19650024

**MOTTO**

*... KEEP GOING ...*

## **KATA PENGANTAR**

Dengan menyebut nama Allah SWT. yang Maha Pengasih lagi Maha Penyayang. Puji syukur senantiasa penulis panjatkan kehadiran Allah SWT. atas rahmat, nikmat, sehat, serta hidayah-Nya sehingga dapat menyelesaikan penulisan skripsi ini dengan baik dalam rangka memenuhi syarat kelulusan untuk memperoleh gelar sarjana komputer (S.Kom) pada Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Penulisan skripsi ini tidak akan berjalan dengan baik tanpa adanya dukungan dan bantuan dari berbagai pihak baik secara langsung maupun tidak langsung sehingga penulis dengan segala hormat berterimakasih sebesar-besarnya kepada:

1. Prof. Dr. H. M. Zainuddin, MA selaku Rektor Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
2. Prof. Dr. Sri Harini, M.Si selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
3. Dr. Fachrul Kurniawan, M.MT selaku Ketua Prodi Teknik Informatika Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
4. Dr. Zainal Abidin, M.Kom selaku Dosen Pembimbing I atas segala bimbingan, saran, kritik, waktu, kesabaran yang diberikan selama penulisan skripsi ini dari awal hingga selesai.
5. Puspa Miladin Nuraida Safitri A Basid, M.Kom selaku Dosen Pembimbing II atas bimbingannya selama penulisan skripsi ini hingga selesai dengan baik.
6. Dosen dan Staff Program Studi Teknik Informatika UIN Malang atas ilmu dan pengalaman yang telah diberikan selama menempuh perkuliahan.
7. Ibu Siti Nur Hasanah selaku orangtua penulis, serta Nurul Fitriah dan Moh. Alfian selaku kakak penulis yang senantiasa memberikan dukungan dalam segala hal serta kepercayaan sehingga penulis mampu menyelesaikan skripsi dengan baik.

8. Seluruh keluarga besar Saudara Teknik Informatika UIN Maulana Malik Ibrahim Malang atas segala rangkaian cerita, dukungan bantuan dalam bentuk apapun selama perkuliahan hingga terselesaikannya penyusunan skripsi ini.
9. Seluruh pihak yang terlibat secara langsung maupun tidak langsung selama penulisan skripsi yang tidak dapat penulis sebutkan satu persatu atas segala jenis bantuan yang telah diberikan sehingga penulis bisa sampai pada tahap ini, semoga dipermudah segala urusan kalian kedepannya.

Akhir kata, penulisan skripsi ini masih jauh dari kata sempurna sehingga diperlukan banyak kritik dan saran yang membangun. Penulis berharap skripsi ini dapat memberikan manfaat bagi berbagai macam pihak.

Malang, 27 November 2023

Penulis

## DAFTAR ISI

<b>HALAMAN JUDUL .....</b>	<b>ii</b>
<b>HALAMAN PERSETUJUAN.....</b>	<b>iii</b>
<b>HALAMAN PENGESAHAN.....</b>	<b>iv</b>
<b>PERNYATAAN KEASLIAN TULIAN .....</b>	<b>v</b>
<b>MOTTO .....</b>	<b>vi</b>
<b>KATA PENGANTAR.....</b>	<b>vii</b>
<b>DAFTAR ISI.....</b>	<b>ix</b>
<b>DAFTAR GAMBAR.....</b>	<b>xi</b>
<b>DAFTAR TABEL.....</b>	<b>xii</b>
<b>ABSTRAK .....</b>	<b>xiii</b>
<b>ABSTRACT .....</b>	<b>xiv</b>
<b>المُلخَص.....</b>	<b>xv</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Pernyataan Masalah .....	6
1.3 Tujuan Penelitian .....	6
1.4 Batasan Masalah .....	7
1.5 Manfaat Penelitian .....	7
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>9</b>
2.1 Penelitian Terkait .....	9
2.2 <i>Text Summarization</i> .....	12
2.2.1 <i>Abstractive Summarization</i> .....	12
2.2.2 <i>Extractive Summarization</i> .....	13
2.3 <i>Natural Language Processing</i> .....	13
2.4 <i>Word2Vec</i> .....	14
2.4.1 <i>Skip-gram</i> .....	15
2.4.2 <i>Continuous Bag of Word (CBOW)</i> .....	16
2.5 <i>Artificial Neural Network</i> .....	17
2.6 <i>Recurrent Neural Network</i> .....	18
2.7 <i>Long Short-Term Memory</i> .....	20
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>25</b>
3.1 Pengumpulan Data .....	25
3.2 Desain Sistem.....	25
3.3 <i>Preprocessing</i> .....	26
3.3.1 <i>Text Cleaning</i> .....	26

3.3.2 Tokenization.....	27
3.3.3 Stopword removal .....	28
3.3.4 Stemming .....	28
3.4 Ekstraksi Fitur .....	29
3.5 LSTM.....	30
3.6 Peringkasan.....	35
3.7 Evaluasi.....	35
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>	<b>37</b>
4.1 Pengelolaan Dataset.....	37
4.2 Skenario Pengujian .....	41
4.2.1 Ekstraksi Fitur Menggunakan Word2vec.....	41
4.2.2 <i>Splitting Data</i> .....	42
4.2.3 Pelatihan Model LSTM.....	43
4.3 Hasil Uji Coba.....	48
4.3.1 Hasil Uji Coba Menggunakan Model CBOW .....	48
4.3.2 Hasil Uji Coba Menggunakan Model <i>Skip-gram</i> .....	52
4.4 Pembahasan.....	53
4.5 Integrasi Penelitian.....	55
<b>BAB V PENUTUP.....</b>	<b>58</b>
5.1 Kesimpulan .....	58
5.2 Saran .....	58
<b>DAFTAR PUSTAKA</b>	
<b>LAMPIRAN</b>	

## DAFTAR GAMBAR

Gambar 2. 1 Arsitektur <i>Skip-gram</i> .....	16
Gambar 2. 2 Arsitektur CBOW .....	17
Gambar 2. 3 Arsitektur <i>Neural Network</i> .....	17
Gambar 2. 4 Arsitektur RNN .....	19
Gambar 2. 5 Ilustrasi LSTM .....	21
Gambar 3. 1 Desain Sistem .....	26
Gambar 3. 2 Ilustrasi proses <i>Continous Bag of Word</i> .....	29
Gambar 3. 3 Flowchart perhitungan LSTM .....	31
Gambar 4. 2 Dokumen gabungan .....	38
Gambar 4. 3 <i>K-Fold Cross Validation</i> .....	43
Gambar 4. 4 Grafik loss dan akurasi model 50 neuron .....	44
Gambar 4. 5 Grafik loss dan akurasi model 100 neuron .....	45
Gambar 4. 6 Grafik loss dan akurasi model 150 neuron .....	46
Gambar 4. 7 Grafik loss dan akurasi model 200 neuron .....	47
Gambar 4. 8 Ringkasa Sytem .....	49
Gambar 4. 9 Ringkasan manual .....	49

## DAFTAR TABEL

Tabel 2. 1 Tabel <i>Novelty</i> .....	11
Tabel 3. 1 Proses <i>text cleaning</i> .....	27
Tabel 3. 2 Proses <i>Tokenization</i> .....	27
Tabel 3. 3 Proses <i>Stopword</i> .....	28
Tabel 3. 4 Proses <i>Stemming</i> .....	28
Tabel 3. 5 Vektor <i>One-hot</i> .....	29
Tabel 4. 1 Dataset <i>multi-document</i> .....	39
Tabel 4. 2 Tabel berita perkalimat .....	40
Tabel 4. 3 Label tiap responden .....	41
Tabel 4. 4 Tabel Skenario pengujian.....	41
Tabel 4. 5 Vector tiap kalimat.....	42
Tabel 4. 6 Parameter model LSTM.....	43
Tabel 4. 7 Hasil pelatihan model 50 neuron .....	44
Tabel 4. 8 Hasil pelatihan model 100 neuron .....	45
Tabel 4. 9 Hasil pelatihan model 150 neuron .....	46
Tabel 4. 10 Hasil pelatihan model 200 neuron .....	47
Tabel 4. 11 Statistika Ringkasan.....	50
Tabel 4. 12 Statistik Jumlah Kata overlapping Dalam Ringkasan.....	50
Tabel 4. 13 Hasil ROUGE-1 fold-1 pada model CBOW vektor 100 window 5... 51	
Tabel 4. 14 Hasil rata-rata rouge-1 model CBOW .....	51
Tabel 4. 15 Hasil rata-rata rouge-1 menggunakan model <i>Skip-gram</i> .....	52
Tabel 4. 16 Hasil Rata-rata Rouge-1 tiap fold .....	54

## ABSTRAK

Alfin, Moh. 2023. *Multi Document Summarization Menggunakan Algoritma Recurrent Neural Network*. Skripsi. Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Zainal Abidin, M.Kom. (II) Puspa Miladin Nuraida Safitri A Basid., M.Kom.

**Kata Kunci:** Peringkasan, *Recurrent Neural Network*, *Long Short-Term Memory*, *Word2vec*.

Peringkasan dokumen berita adalah sebuah aspek penting dalam pemrosesan bahasa alami dan jurnal ini bertujuan untuk menggambarkan perkembangan terbaru dalam bidang ini. Dengan ledakan informasi dan jumlah berita yang terus meningkat, peringkasan dokumen berita menjadi kunci dalam menghadapi tantangan untuk mengakses informasi yang relevan dan berharga. Pada paper ini dilakukan peringkasan multi dokumen berbahasa Indonesia menggunakan algoritma *Recurrent Neural Network* (RNN), model yang digunakan yaitu *Long-Short Term Memory* (LSTM), kemudian ekstraksi fitur yang adalah *Word2Vec*, model *Word2vec* yang digunakan ada dua yaitu *Continuous Bag of Word* (CBOW) dan *Skip-gram*. Hasilnya menunjukkan nilai presisi, *recall*, dan *f-measure* lebih tinggi menggunakan model CBOW. Untuk model CBOW mendapatkan nilai *recall* sebesar 0.487, presisi sebesar 0.704, dan *F-measure* sebesar 0.550. Sementara itu, untuk model *Skip-gram*, hasil pengujian menunjukkan nilai *recall* 0.414, presisi dengan nilai 0.687, dan *F-measure* sebesar 0.504.

## ABSTRACT

Alfin, Moh. 2023. **Multi-Document Summarization Using Recurrent Neural Network Algorithm**. Thesis. Informatics Engineering Department, Faculty of Science and Technology. *State Islamic University of Maulana Malik Ibrahim Malang*. Supervisors: (I) Zainal Abidin, M.Kom. (II) Puspa Miladin Nuraida Safitri A Basid., M.Kom.

News document summarization is an important aspect of natural language processing and this journal aims to describe the latest developments in this field. With the explosion of information and the ever-increasing number of news, news document summarization is key to facing the challenge of accessing relevant and valuable information. In this paper, multi-document summarization in Indonesian is performed using the RNN (Recurrent Neural Network) method, the variation used is Long Short-Term Memory (LSTM), with feature extraction using two different Word2Vec models, namely CBOW (Continuous Bag of Words) and *Skip-gram*. The results show significant recall, precision, and F-measure values. For the CBOW model, the recall, precision, and F-measure values found are 0.487, 0.704, and 0.550. Meanwhile, for the *Skip-gram* model, the test results show a recall value of 0.414, a precision value of 0.687, and an F-measure value of 0.504

**Keyword:** *Summarization, Recurrent Neural Network, Long Short-Term Memory, Word2vec.*

## الملخص

الفين، محمد. ٦٠٦٣. " تلخيص المستندات المتعددة باستخدام خوارزمية الشبكة العصبية المتكررة ". رسالة نخرج البكالوريوس. قسم الهندسة الحاسوبية، كلية العلوم والتكنولوجيا، الجامعة الإسلامية الحكومية مولانا مالك إبراهيم مالانج. المشرفون: (١) زَيْنُ العابدين، ماجستير. (٦) فوسفا ميلادين نورايده سافيتري ابدول باسيد ، ماجستير.

**الكلمات الرئيسية:** تلخيص ، الشبكة العصبية المتكررة ، الذاكرة الطويلة وقصيرة المدى.

يعد تلخيص المستندات الإخبارية جانبًا مهمًا من معالجة اللغة الطبيعية وتهدف هذه المجلة إلى وصف أحدث التطورات في هذا المجال. مع انفجار المعلومات والعدد المتزايد باستمرار من الأخبار، يعد تلخيص المستندات الإخبارية أمرًا أساسيًا لمواجهة التحدي المتمثل في الوصول إلى المعلومات ذات الصلة والقيمة. في هذا البحث، يتم تلخيص المستندات المتعددة باللغة الإندونيسية باستخدام طريقة (الشبكة العصبية المتكررة)، والتنوع المستخدم هو الذاكرة طويلة المدى ، مع استخراج الميزات باستخدام نموذجين مختلفين من Word2Vec ، وهما CBOW (حقيقية مستمرة من الكلمات) وتخطي غرام. تظهر النتائج قيمًا كبيرة للاستدعاء والدقة والقياس. بالنسبة لنموذج CBOW ، فإن قيم الاستدعاء والدقة والقياس الموجودة هي ٠.٤٨٧ و ٠.٧٤٦ و ٠.٥٥٠ وفي الوقت نفسه، بالنسبة لنموذج Skip-gram ، تظهر نتائج الاختبار قيمة استدعاء تبلغ ٠.٦٨٧ ، وقيمة دقة تبلغ ٠.٦٨٧ ، وقيمة قياس تبلغ ٠.٦٨٧ .

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Kemajuan teknologi pada zaman sekarang menyebabkan jumlah dokumen dan artikel yang dapat diakses melalui internet meningkat secara signifikan. Dalam era digital, menulis dan mempublikasikan informasi menjadi semakin mudah dilakukan oleh hampir semua orang, sehingga jumlah informasi atau berita yang terdapat pada internet sangat banyak. Hal tersebut menjadi salah satu sumber utama informasi dari berbagai banyak orang, seperti situs berita, blog, sosial media, dan sejenisnya yang menyediakan berita dan informasi secara real-time dan kontinu. Hal ini menyebabkan jumlah teks berita yang tersedia di internet semakin melonjak dan menjadi sulit untuk dikelola. (Holton & Chyi, 2012). Banyaknya teks berita di internet bisa mempengaruhi minat baca berita orang. Dengan banyaknya berita dan informasi yang tersebar menyebabkan pembaca memerlukan banyak waktu untuk membaca semua informasi tersebut. Untuk memudahkan manusia dalam membaca artikel berita, maka diperlukan sebuah sistem peringkasan teks secara otomatis atau *Automatic text summarization system*.

Peringkasan teks otomatis adalah mengambil kalimat-kalimat yang dianggap penting dalam suatu teks yang panjang dan menyajikannya kembali dalam bentuk yang lebih singkat atau ringkas. Sehingga teks tersebut berisi esensi dari teks yang asli. (Saputra, 2017). Peringkasan teks otomatis dapat digunakan dalam berbagai bidang, seperti jurnalisme, riset, dan bisnis. Metode yang

digunakan dalam peringkasan teks otomatis meliputi teknik statistik, pembelajaran mesin, dan analisis semantik. Bidang peringkasan teks otomatis merupakan salah satu penelitian yang cukup banyak dilakukan dalam bidang *Natural Language Processing*. (Gupta & Gupta, 2019). Automatic Text Summarization dapat diaplikasikan pada berbagai jenis teks seperti pada hasil pencarian di *search engine*, ulasan produk pada *e-commerce*, dan artikel berita di media massa. Pada aplikasi tersebut, peringkasan teks otomatis digunakan untuk menghasilkan ringkasan informasi yang relevan dan mudah dipahami oleh pengguna.

Peringkasan teks dapat meningkatkan minat baca berita seseorang. Peringkasan teks otomatis dapat membuat informasi yang tersedia lebih mudah dipahami dan disajikan dalam bentuk yang lebih ringkas. Hal ini dapat membuat proses pemahaman informasi lebih cepat dan mengurangi kejenuhan pada teks yang panjang. Selain itu, peringkasan teks juga dapat membuat informasi yang tersedia lebih terfokus pada informasi yang penting. Peringkasan teks juga dapat digunakan untuk menyajikan berita dalam bentuk yang lebih sederhana dan mudah diingat. Dengan demikian, peringkasan teks dapat menjadi alat yang efektif untuk meningkatkan minat baca seseorang terhadap berita. (Rossana Ramirez-Avila *et al.*, 2021)

Dalam melakukan peringkasan teks sendiri terdapat dua pendekatan yaitu pendekatan abstraktif dan pendekatan ekstraktif. Pendekatan ekstraktif hanya memilih kalimat-kalimat penting dari dokumen asli untuk dijadikan ringkasan, sehingga informasi penting dalam dokumen tetap terjaga dan tercakup dalam

ringkasan. Hasil ringkasan pada pendekatan berupa kalimat-kalimat yang memiliki frekuensi kemunculan tertinggi pada teks (Ramanujam & Kaliappan, 2016). Sedangkan pendekatan abstraktif adalah melakukan peringkasan secara abstrak sebagaimana peringkasan yang dilakukan oleh manusia dalam melakukan peringkasan (Khan & Salim, 2014). Karena pendekatan ekstraktif hanya memilih kalimat-kalimat yang penting dari dokumen asli, maka algoritma dapat bekerja lebih cepat dalam menghasilkan ringkasan. Sementara itu, pendekatan abstraktif memerlukan lebih banyak waktu untuk menghasilkan ringkasan, karena model harus menghasilkan kalimat-kalimat baru yang dapat merepresentasikan informasi penting dalam dokumen. Pendekatan ekstraktif pada peringkasan teks dapat menghasilkan hasil yang objektif dan dapat diukur. Hal ini karena metode ekstraktif hanya mengekstrak informasi penting yang telah didefinisikan, sedangkan pendekatan abstraktif dapat dipengaruhi oleh faktor subjektif peringkasan.

Dalam agama Islam, terdapat perintah izin untuk mengqashar shalat (menyingkat durasi shalat) ketika sedang melakukan perjalanan dan merasa takut diserang oleh orang-orang kafir. Tujuan dari shalat qashar adalah untuk memudahkan seorang muslim yang sedang melakukan perjalanan untuk melakukan ibadah shalat. Dalam hal ini sama halnya dengan peringkasan teks yang dilakukan untuk mengurangi durasi membaca atau mendengarkan, mengurangi informasi yang tidak relevan yang bertujuan untuk memudahkan pemahaman. Allah SWT menyebutkannya dalam Qur'an surat An-nisa' ayat 101 tentang melakukan shalat qashar.

وَإِذَا ضَرَبْتُمْ فِي الْأَرْضِ فَلَيْسَ عَلَيْكُمْ جُنَاحٌ أَنْ تَقْصُرُوا مِنَ الصَّلَاةِ إِنْ خِفْتُمْ أَنْ يُفْتِنَكُمْ الَّذِينَ كَفَرُوا إِنَّ الْكُفْرَيْنَ كَانَوا  
لَكُمْ عَدُوًّا مُبِينًا ﴿١٠١﴾

”Apabila kamu bepergian di bumi, maka tidak dosa bagimu untuk mengqasar salat jika kamu takut diserang orang-orang yang kafur. Sesungguhnya orang-orang kafir itu adalah musuh yang nyata bagimu.” (QS. An-Nisa’ 4: ayat 101)

Tafsir jalalain memberikan penjelasan mengenai ayat ini yaitu (Dan jika kamu melakukan perjalanan) atau bepergian (di muka bumi, maka tidak ada salahnya kamu) (apabila mengqashar shalat) dengan membuat yang empat rakaat menjadi dua (jika kamu khawatir akan diperangi) atau mendapatkan luka dari orang-orang kafir (menyatakan peristiwa yang terjadi kala itu maka mafhumnya tidak berlaku). Menurut keterangan dari sunnah, yang dimaksud dengan suatu perjalanan panjang ialah empat pos atau dua marahlah. Dan dari firman-Nya, “Maka tidak salahnya kamu.” Ditarik kesimpulan bahwa mengqashar shalat itu merupakan keringanan dan bukan kewajiban. Dan ini pendapat Imam Syafi’i. (Sesungguhnya orang-orang kafir itu bagi kamu adalah musuh yang nyata) maksud jelas dan terang permusuhannya terhadap kamu”. (Al-Mahalli *et al.*, 1469)

Ada banyak metode dan Algoritma yang tersedia untuk peringkasan teks, termasuk metode berbasis *machine learning* dan *deep learning*. Pilihan metode tergantung pada kasus penggunaan tertentu dan tingkat akurasi yang diinginkan. RNN merupakan algoritma mesin pembelajaran yang biasa digunakan sebagai metode dalam kasus peringkasan teks secara abstraktif. Contoh penelitian tentang kasus ini dilakukan oleh Adelia R (2019). Dalam penelitiannya, ia menggunakan metode RNN BiGRU dan menganalisis 500 dokumen jurnal berbahasa Indonesia

yang dibagi menjadi 400 data *training* dan 100 data *testing*. Dari dua pengujian yang dilakukan, ia menggunakan model bigram dan unigram. Hasil yang diperoleh menunjukkan bahwa pada bigram mendapatkan hasil Rouge-1 mencapai 0.119751 kemudian untuk Rouge-2 mencapai 0.0119, sementara pada unigram, Rouge-1 mencapai 0.06745 dan Rouge-2 mencapai 0.0055 (Adelia *et al.*, 2019).

Penelitian lainnya dilakukan oleh Saputra M. A (2021) dengan pendekatan abstrak menggunakan metode *Long Short-Term Memory* (LSTM). LSTM dinilai cocok untuk peringkasan teks karena dapat mengingat dan melupakan informasi yang di prosesnya serta memiliki mekanisme perhatian. Model LSTM dibangun dengan model encoder/decoder LSTM tiga lapis. Data yang digunakan dalam artikel ini adalah IndoSum, kumpulan dokumen berita Indonesia. Studi ini juga mendefinisikan tiga skenario berbeda dari perspektif pra pemrosesan satu dengan penghapusan stemming dan stopword, satu dengan hanya stemming, dan satu lagi tanpa keduanya. Hasil pengujian menunjukkan hasil yang buruk untuk skenario pertama (stemming dan stopword removal). Hal ini karena menghapus kata-kata berhenti secara signifikan mengurangi jumlah kata. Hasil penelitian menunjukkan rata-rata skor F1 sekitar 0,08612 dan skor maksimum F1 sekitar 0,4597. Penelitian menggunakan metode LSTM juga dilakukan oleh Ivanedra & Mustikasari, (2019).

Penelitian ini menggunakan metode LSTM untuk meringkas berita dari tiga sumber portal berita, yaitu Indian Times, Guardian, dan Hindu yang totalnya mencapai 4.515 artikel berita dan juga ringkasannya. Penelitian ini melakukan dua pengujian, yaitu evaluasi intrinsik pada artikel non-stemming dan artikel

stemming. Hasil evaluasi menunjukkan bahwa artikel yang telah dilakukan stemming menghasilkan ringkasan yang lebih bagus dengan rata-rata *recall* 0.44, *precision* 0.88, dan *f-measure* 0.5820. Sedangkan untuk artikel yang *non-stemming* didapatkan nilai *recall* 0.41, *precision* 0.81, dan *f-measure* 0.5427. (Ivanedra & Mustikasari, 2019)

Peringkasan otomatis pada penelitian sebelumnya masih menggunakan dataset single dokumen, oleh karena itu pada penelitian ini akan dilakukan peringkasan pada dataset multi dokumen menggunakan metode RNN. Dataset multi dokumen sendiri adalah gabungan dari beberapa dataset single dokumen yang dijadikan satu menjadi dokumen utuh. Tujuannya adalah untuk merangkum informasi dari berbagai sumber yang berbeda sehingga pembaca dapat memahami gambaran keseluruhan dari beberapa berita sekaligus.

## **1.2 Pernyataan Masalah**

Dengan latar belakang yang dipaparkan maka pernyataan masalah pada penelitian ini yaitu, bagaimana menghemat waktu dalam mendapatkan inti informasi dari berbagai dokumen yang banyak tanpa membacanya secara menyeluruh?

## **1.3 Tujuan Penelitian**

Untuk membangun sebuah sistem peringkasan teks berita otomatis berbahasa Indonesia dengan jumlah yang besar menggunakan metode *Recurrent Neural Network*.

#### **1.4 Batasan Masalah**

Batasan masalah pada penelitian ini adalah sebagai berikut:

1. Model *Recurrent Neural Network* (RNN) yang akan digunakan adalah *Long-Short Term Memory* (LSTM).
2. Penelitian ini berfokus pada peringkasan ekstraktif.
3. Dataset yang digunakan 190 artikel berita berbahasa Indonesia IndoSum yang digabung menjadi 50 multi dokumen berdasarkan kategori.

#### **1.5 Manfaat Penelitian**

Dalam penerapan metode RNN dalam peringkasan teks otomatis berita bahasa Indonesia diharapkan mampu memberikan manfaat yang berguna, antara lain:

1. Bagi pelaku bisnis
  - a. Meningkatkan produktivitas, peningkatan produktivitas dapat dicapai dengan penggunaan Peringkasan dokumen otomatis untuk membantu karyawan dalam melakukan tugas-tugas mereka. Misalnya, summarization dokumen dapat digunakan untuk membantu karyawan dalam membuat laporan, melakukan riset, atau membuat keputusan.
  - b. Meningkatkan kepuasan pelanggan, peningkatan kepuasan pelanggan dapat dicapai dengan penggunaan summarization dokumen untuk memberikan informasi yang lebih ringkas dan mudah dipahami kepada pelanggan. Hal ini dapat membantu pelanggan dalam memahami informasi yang mereka butuhkan dengan lebih cepat dan mudah.

c. Menghemat biaya tenaga kerja. Peringkasan dokumen secara otomatis dapat menggantikan pekerjaan manusia dalam melakukan peringkasan dokumen secara manual. Hal ini dapat menghemat biaya tenaga kerja, terutama untuk perusahaan yang memiliki banyak dokumen yang perlu diringkaskan.

## 2. Bagi masyarakat

Peringkasan teks otomatis dapat membantu untuk mempercepat dan mempermudah proses pembacaan dan pemahaman serta menghemat waktu dan tenaga dalam membaca dan memahami teks yang panjang.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Terkait**

Penelitian terkait adalah kumpulan penelitian yang membahas topik atau permasalahan yang sama atau serupa dengan topik penelitian yang sedang dibahas. Kemudian penelitian tersebut dijadikan pedoman dan acuan dalam penelitian. berikut adalah penelitian terkait.

Penelitian mengenai peringkasan teks yang dilakukan oleh Adelia *et al* (2019). Dalam penelitiannya, ia menggunakan metode RNN BiGRU dan menganalisis 500 dokumen jurnal berbahasa Indonesia yang dibagi menjadi 400 data *training* dan 100 data *testing*. Dari dua pengujian yang dilakukan, ia menggunakan model bigram dan unigram. Hasil yang diperoleh menunjukkan bahwa pada bigram mendapatkan hasil Rouge-1 mencapai 0.119751 kemudian untuk Rouge-2 mencapai 0.0119, sementara pada unigram, Rouge-1 mencapai 0.06745 dan Rouge-2 mencapai 0.0055. (Adelia *et al.*, 2019)

Dalam penelitian M. A. Saputra (2021) melakukan peringkasan teks otomatis dengan pendekatan abstraktif menggunakan metode LSTM, LSTM sendiri merupakan salah satu model dari metode RNN. LSTM dianggap baik untuk peringkasan teks karena dapat mengingat dan melupakan informasi yang akan diproses, serta memiliki mekanisme attention. Model LSTM dirancang menggunakan model *encoder-decoder* dan 3 *layer* LSTM. Data dalam penelitian ini adalah dataset kumpulan berita berbahasa Indonesia yaitu IndoSum. Penelitian ini juga mendefinisikan tiga skenario yang berbeda dalam hal *preprocessing*, yaitu

skenario dengan *stemming* dan *stopword removal*, hanya *stemming*, dan tanpa keduanya. Hasil pengujian menunjukkan hasil yang rendah untuk skenario pertama (*stemming* dan *stopword removal*) karena pengurangan kata yang signifikan akibat *stopword removal*. Hasilnya memiliki F1-score rata-rata sekitar 0.08612 dan F1-score maksimum sekitar 0.45977. (M. A. Saputra, 2021)

Penelitian menggunakan metode LSTM juga dilakukan oleh Ivanedra & Mustikasari (2019). Penelitian ini menggunakan metode LSTM untuk meringkas berita dari tiga sumber portal berita, yaitu Indian Times, Guardian, dan Hindu yang totalnya mencapai 4.515 artikel berita dan juga ringkasannya. Penelitian ini melakukan dua pengujian, yaitu evaluasi intrinsik pada artikel non-stemming dan artikel stemming. Hasil evaluasi menunjukkan bahwa artikel yang telah dilakukan stemming menghasilkan ringkasan yang lebih bagus dengan rata-rata *recall* 0.44, *precision* 0.88, dan *f-measure* 0.5820. Sedangkan untuk artikel yang *non-stemming* didapatkan nilai *recall* 0.41, *precision* 0.81, dan *f-measure* 0.5427. (Shini & Kumar, 2021)

Penelitian selanjutnya dilakukan oleh Shini & Kumar (2021). Pada penelitian dilakukan pendekatan abstraktif dalam melakukan peringkasan. Data yang digunakan adalah single-dokumen berita berbahasa Inggris. Penelitian ini menggunakan Dual Encoding Model-Dem untuk ekstraksi fitur dan digunakan metode Recurrent Neural Network untuk meringkas. Hasilnya adalah Rouge-1 sebesar 0.4099, Rouge-2 sebesar 0.1897, dan Rouge-L dengan nilai 0.3665. (Shini & Kumar, 2021)

Tabel 2. 1 Tabel *Novelty*

Sumber	Pendekatan	Dokumen	Ekstraksi fitur	Metode	Hasil
(Shini & Kumar, 2021)	Abstraktif	Single-Dokumen berita berbahasa Inggris	Dual Encoding Model-Dem	<i>Recurrent Neural Network</i>	ROUGE-1 sebesar 0.4099, ROUGE-2 sebesar 0.1897, dan ROUGE-L sebesar 0.3665
(Ivanedra & Mustikasari, 2019)	Abstraktif	Artikel berita Single-Dokumen bahasa Indonesia	<i>Word embedding</i>	<i>Long-Short Term Memory</i>	Recall 41%, precision 81%, dan F measure sebesar 54,27%
(Rahman & Siddiqui, 2019)	Ekstraktif	Artikel berita Single-Dokumen berbahasa Inggris	Multi-layered Attentional Peephole Convolutional LSTM	<i>Long-Short Term Memory</i>	ROUGE-1 39.61%, ROUGE-2 20.87%, ROUGE-L 39.42%
(Kalangi <i>et al.</i> , 2020)	Abstraktif	Teks artikel Single-Dokumen bahasa Inggris	TF-IDF, <i>Sentence length, sentence position, similarity to tittle</i>	<i>Long-Short Term Memory</i>	<i>Precision 0.54, recall, 0.66, f-measure 0.58</i>
(Hanunggul, 2019)	Abstraktif	Amazon Fine Food Reviews	Word2vac	<i>Long-Short Term Memory</i>	ROUGE-1 0.05689, ROUGE-2 0.00131
Fokus penelitian	Ekstraktif	Artikel Multi-Dokumen berita berbahasa Indonesia	Word2vec	<i>Long-Short Term Memory</i>	-

Pada penelitian terdahulu, peringkasan dokumen dilakukan untuk satu dokumen saja. Sedangkan pada penelitian ini, peringkasan dokumen dilakukan untuk beberapa dokumen sekaligus (multi dokumen). multi dokumen sendiri adalah gabungan dari beberapa dokumen yang dijadikan satu menjadi dokumen utuh. Tujuannya adalah untuk merangkum informasi dari berbagai sumber yang berbeda sehingga pembaca dapat memahami gambaran keseluruhan dari beberapa berita sekaligus.

## **2.2 Text Summarization**

*Text Summarization* atau dalam bahasa Indonesia adalah peringkasan teks merupakan proses mengurangi jumlah kata dalam sebuah teks tanpa mengubah makna dasar dari teks tersebut. Peringkasan teks dapat dilakukan dengan berbagai cara, seperti menghilangkan kata-kata yang (Holton & Chyi, 2012). Banyaknya teks berita di internet bisa mempengaruhi minat baca berita orang. Karena jumlah informasi yang tersedia sangat besar, orang mungkin merasa kewalahan dan sulit untuk memilih informasi yang penting dan akurat. Beberapa orang mungkin juga merasa bosan atau tidak terstimulasi dengan jumlah informasi yang terlalu banyak.

Peringkasan teks otomatis adalah proses mengambil informasi penting dari sebuah teks yang panjang dan menyederhanakannya menjadi versi ringkas yang menyampaikan esensi dari teks asli (Saputra, 2017). Peringkasan teks otomatis dapat digunakan dalam berbagai bidang, seperti jurnalisme, riset, dan bisnis. Metode yang digunakan dalam peringkasan teks otomatis meliputi teknik statistik, pembelajaran mesin, dan analisis semantik. *Extractive summarization* sering digunakan dalam aplikasi seperti pencarian berita, pemantauan media, dan analisis dokumen untuk menyediakan ringkasan teks yang cepat dan mudah dibaca. (Anjar Prabowo *et al.*, 2016)

### **2.2.1 Abstractive Summarization**

*Abstractive summarization* adalah metode peringkasan teks yang menciptakan ringkasan baru dengan menggabungkan atau menyimpulkan poin-poin kunci dari teks asli (Adelia *et al.*, 2019). Ini berbeda dari metode ekstraktif

yang hanya mengambil kalimat atau bagian dari teks yang dianggap paling penting untuk ditampilkan dalam ringkasan. *Abstractive summarization* menggunakan Algoritma yang dapat memahami konteks dan makna dari teks asli, sehingga dapat menciptakan ringkasan yang lebih sintetis dan menyederhanakan teks asli. *Abstractive summarization* sering digunakan dalam aplikasi seperti chatbot, asisten virtual, dan sistem pemahaman mesin untuk menyajikan informasi yang lebih singkat dan mudah dipahami.

### **2.2.2 *Extractive Summarization***

Peringkasan ekstraktif adalah metode peringkasan teks yang dilakukan dengan cara mengambil bagian-bagian penting dari teks asli, seperti kalimat atau paragraf, dan menyusunnya kembali menjadi teks yang lebih singkat dan padat. Bagian-bagian yang diambil biasanya adalah bagian-bagian yang mengandung informasi penting, seperti topik utama, fakta-fakta penting, dan kesimpulan. Pendekatan ini menghasilkan ringkasan dengan cara mengambil kalimat-kalimat penting dari dokumen asli dan menyusunnya menjadi dokumen yang lebih pendek.

## **2.3 *Natural Language Processing***

*Natural Language Processing (NLP)* merupakan salah satu ilmu AI yang mempelajari cara memproses bahasa alami yang digunakan oleh manusia. Bahasa manusia tidak langsung dapat dipahami oleh mesin, sehingga perlu diproses terlebih dahulu agar komputer dapat memahami maksud dari manusia. NLP

merupakan bidang penting dalam ilmu komputer yang mencakup linguistik komputasional dan pembelajaran mesin. (A. Jain *et al.*, 2018).

Langkah awal dalam metode NLP adalah *Text Processing*. *Text Preprocessing* bertujuan untuk menyiapkan teks *raw* menjadi teks yang siap diproses oleh mesin. Ada berbagai proses yang dapat dilakukan pada tahap *Text Preprocessing* dan tidak ada aturan baku mengenai urutan dan jenis proses yang digunakan. Semua tergantung pada *output* yang diinginkan dari data tersebut. Beberapa proses yang umumnya digunakan dalam *Text Processing* antara lain *text cleaning*, *tokenization*, *stopword removal*, dan *stemming*.

## 2.4 Word2Vec

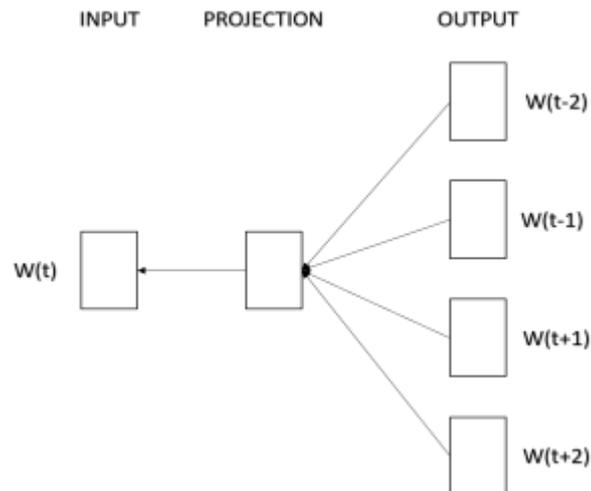
Word2Vec adalah teknik pemodelan bahasa alami berbasis jaringan saraf yang digunakan untuk menghasilkan representasi vektor untuk kata-kata dalam korpus teks. Representasi vektor ini dapat digunakan untuk mewakili makna kata dan mengidentifikasi hubungan antara kata-kata dalam teks. (Mikolov *et al.*, 2013) Word2vec melakukan pelatihan seperti sistem jaringan saraf untuk mendapatkan bobot yang digunakan sebagai vector kata. Arsitektur word2vec terdiri dari tiga *layer* yaitu *input*, *hidden*, dan *output*. Untuk *inputan* pada word2vec adalah vector *one-hot encoded* pada setiap kata unik yang terdapat pada data yang akan dilatih.

Dalam Word2Vec, kata-kata yang sering muncul bersama dalam konteks yang sama akan diberikan vektor yang mendekati satu sama lain dalam ruang vektor. Sebagai contoh, kata-kata seperti "raja" dan "ratu" atau "pria" dan "wanita" akan memiliki vektor yang mendekati satu sama lain dalam ruang vektor, karena mereka cenderung muncul bersama-sama dalam konteks yang

sama. Dengan menggunakan representasi vektor ini, Word2Vec memungkinkan mesin untuk memahami dan memprediksi relasi antara kata-kata yang berbeda, seperti sinonim, antonim, dan hubungan semantik lainnya. Sehingga, Word2Vec dapat digunakan untuk berbagai tugas NLP seperti analisis sentimen, klasifikasi teks, peringkasan teks, mesin terjemah, dan lainnya. Dalam *word2vec* sendiri terdapat dua model yang digunakan untuk proses pelatihan yaitu *Skip-gram* dan *Continuous Bag of Words* (CBOW).

#### **2.4.1 *Skip-gram***

Model ini mengambil sebuah kata target dan mencoba memprediksi kata-kata yang mungkin muncul di sekitarnya. Model *Skip-gram* menggunakan informasi dari kata target untuk memprediksi kata-kata di sekitarnya, sehingga model ini juga dikenal sebagai "context-based" model (Mikolov *et al.*, 2013). Input dari model ini adalah *current word* yang akan digunakan untuk memprediksi kata atau konteks di sekitarnya sebagai *output* nya, kemudian model akan mempelajari distribusi probabilitas kata dalam konteks dengan jumlah *windows* yang telah ditentukan.

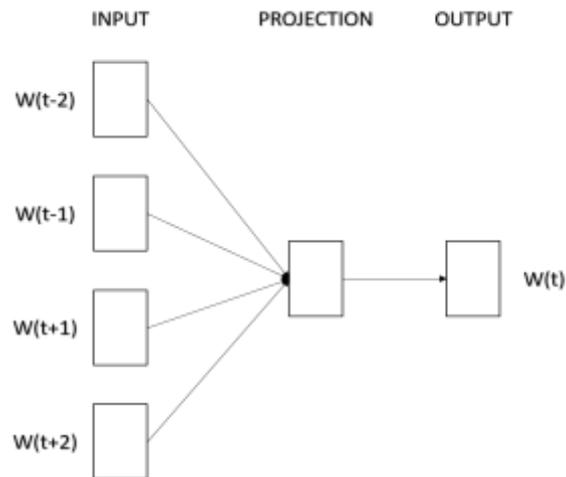


Gambar 2. 1 Arsitektur *Skip-gram*

#### 2.4.2 *Continuous Bag of Word (CBOW)*

Model CBOW bekerja dengan cara sebaliknya. Model CBOW bekerja dengan mengambil beberapa kata di sekitar kata target (yang ingin direpresentasikan sebagai vektor) dan mencoba memprediksi kata target tersebut. Model CBOW menggunakan informasi dari kata-kata di sekitar kata target untuk memprediksi kata target, sehingga model ini juga dikenal sebagai "*bag-of-words*" model. Model CBOW adalah model yang digunakan untuk memprediksi kata tertentu (*current word*) dari konteks yang ada di sekitarnya (*input*).

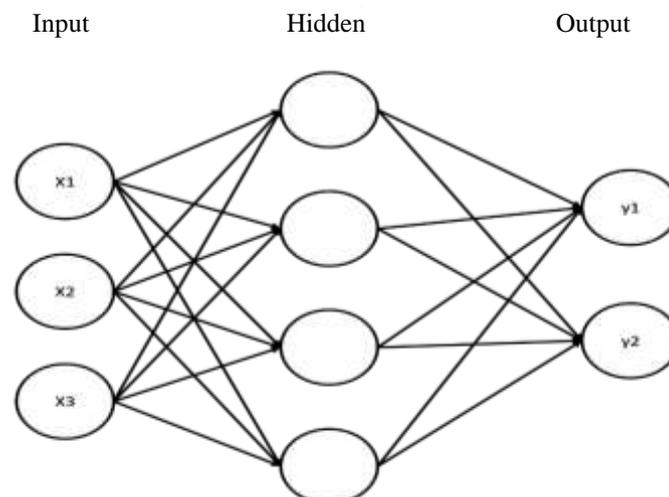
Dalam model ini, distribusi probabilitas konteks akan dipelajari dengan menggunakan jendela yang telah ditentukan. Dengan demikian, dapat dikatakan bahwa CBOW adalah kebalikan dari model *Skip-gram*. Seperti pada ilustrasi implementasi model *Skip-gram*, konteks yang digunakan dalam model CBOW juga menggunakan vektor one-hot encoded yang serupa. Arsitektur CBOW dapat ditunjukkan pada gambar 2.2



Gambar 2. 2 Arsitektur CBOW

## 2.5 Artificial Neural Network

*Artificial Neural Network* (ANN) atau jaringan saraf tiruan adalah salah satu jenis teknologi AI yang terinspirasi dari sistem saraf biologis manusia. ANN terdiri dari kumpulan unit pengolah informasi atau "neuron" yang saling terhubung dalam struktur yang rumit dan terorganisir untuk melakukan tugas tertentu seperti klasifikasi, prediksi, dan pengenalan pola. Arsitektur ANN terdiri dari tiga layer, yaitu *input*, *hidden*, dan *output*.

Gambar 2. 3 Arsitektur *Neural Network*

Lapisan pertama dalam ANN adalah *input layer*, yang menerima data input dari sumber eksternal. Setiap neuron di dalam *input layer* mewakili satu fitur dari data input. Fungsi dari *input layer* adalah untuk meneruskan informasi input ke lapisan selanjutnya. Lapisan selanjutnya *hidden layer*, yang bertindak sebagai penghubung antara *input layer* dan *output layer*. Fungsi dari *hidden layer* adalah untuk memproses informasi input dan menghasilkan fitur-fitur kompleks, kemudian akan menghasilkan representasi yang lebih baik dari data input. Lapisan terakhir *output layer*, yang menghasilkan *output* dari input yang telah diproses melalui *hidden layer*. *Output* dari lapisan ini dapat berupa kelas prediksi, nilai kontinu, atau kategori. Fungsi dari *output layer* adalah untuk menghasilkan *output* yang akurat dan dapat diinterpretasikan dari data input.

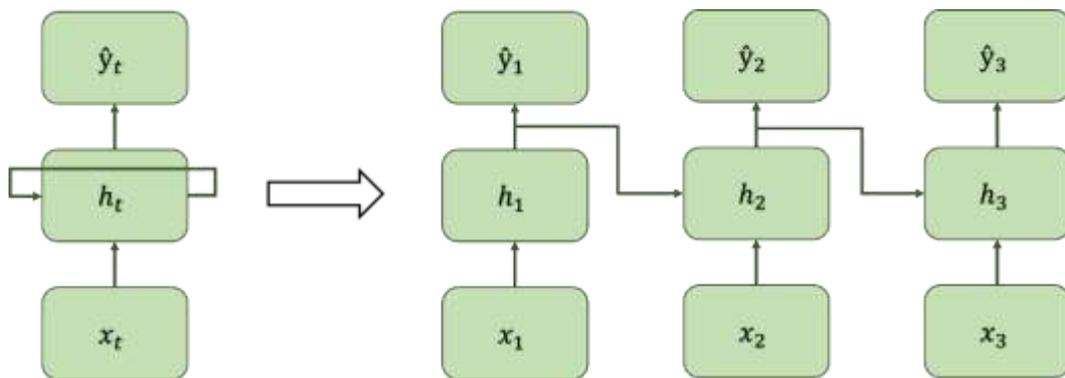
## 2.6 Recurrent Neural Network

*Recurrent Neural Network* (RNN) adalah jenis jaringan saraf tiruan yang memiliki kemampuan untuk memproses data sekuensial, seperti teks, waktu, atau serangkaian data lainnya. RNN memiliki mekanisme loop yang memungkinkan informasi dalam satu tahap pemrosesan diteruskan ke tahap berikutnya, membuat RNN dapat memproses dan mengingat informasi dalam data secara bersamaan. Ini membuat RNN cocok untuk tugas-tugas seperti analisis sentiment, penerjemahan bahasa alami, dan prediksi Series waktu. (Liu *et al.*, 2016)

Secara umum, arsitektur RNN terdiri dari tiga bagian utama: *input layer*, *hidden layer*, dan *output layer*. *Input layer* menerima masukan sekuensial, seperti serangkaian data waktu atau teks. *Hidden layer* adalah tahap pemrosesan utama

yang berulang dan memiliki memori internal. *Output layer* menghasilkan prediksi atau label untuk setiap tahap pemrosesan.

Untuk memodelkan hubungan antara tahap pemrosesan yang berbeda, RNN menggunakan parameter dalam bentuk matriks yang dikenal sebagai *weight matrices*. Ini menentukan bagaimana informasi dalam tahap pemrosesan saat ini mempengaruhi tahap pemrosesan berikutnya dan seberapa besar informasi dari tahap sebelumnya mempengaruhi tahap saat ini.



Gambar 2. 4 Arsitektur RNN

Berdasarkan gambar tersebut. Persamaan tersebut terdiri dari tiga bagian yaitu input, hidden, dan output. Fungsi input berfungsi untuk mengubah input  $x_t$  menjadi bentuk vektor. Fungsi *hidden* digunakan untuk menghitung *hidden state*  $h_t$  berdasarkan input  $x_t$  dan *hidden state* pada timestamp sebelumnya, yaitu  $h_{t-1}$ . Sedangkan fungsi *output* digunakan untuk menghitung nilai prediksi  $\hat{y}_t$  berdasarkan *hidden state* pada timestamp tersebut, yaitu  $h_t$ . Proses perhitungan dilakukan untuk setiap timestamp  $t$  pada kalimat yang sedang diproses. Berikut merupakan formula perhitungan RNN.

$$h_t = \sigma(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (2.1)$$

Untuk menghitung *hidden state* yaitu dengan cara mengalikan *Weight matrix* dari masukan ( $x$ ) ke *hidden (h)* dengan masukan ke-t ( $x_t$ ) lalu ditambahkan perkalian *Wight matrix* dari *hidden* ke *hidden* dengan *hidden* ke t-1 kemudian ditambah dengan *bias term* kemudian hasilnya dimasukan kedalam fungsi aktivasi ( $\sigma$ ). Setelah itu kita dapat menghitung keluaran ( $\hat{y}$ ) dengan persamaan berikut

$$\hat{y} = \sigma(W_{hy}h_t + b_y) \quad (2.2)$$

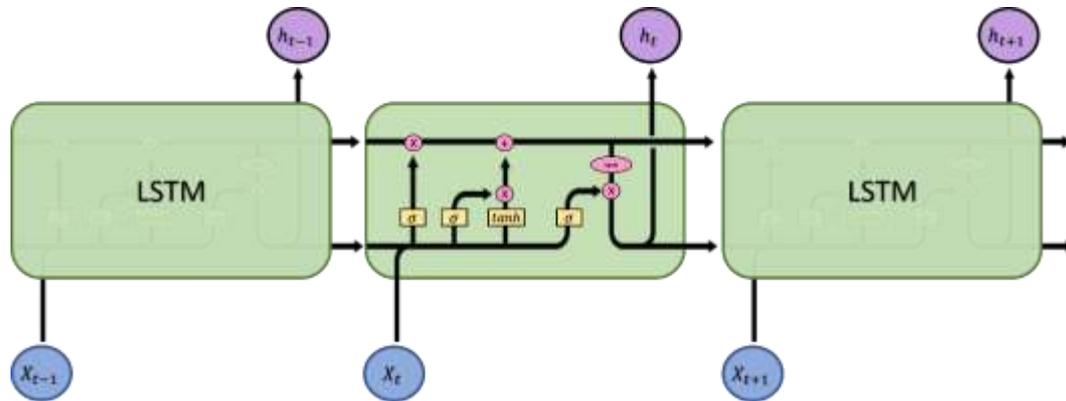
setelah mendapatkan nilai *hidden layer* kita bisa menghitung keluaran dengan cara mengalikan *Weight matrix* dengan nilai *hidden* ke t ditambahkan dengan bias term kemudian hasilnya dimasukan kedalam fungsi aktivasi.

## 2.7 Long Short-Term Memory

LSTM (*Long Short-Term Memory*) adalah varian dari jenis jaringan saraf tiruan RNN. LSTM pertama kali diusulkan oleh (Hochreiter & Schmidhuber, 1997). LSTM dikembangkan untuk mengatasi masalah yang dihadapi oleh RNN dalam mengingat informasi yang lama dalam sekuensial panjang. LSTM memiliki kemampuan yang lebih baik untuk menangani permasalahan yang lebih kompleks, seperti pemrosesan bahasa alami, teks generasi, dan pengenalan suara (Liu *et al.*, 2016)

Arsitektur LSTM lebih rumit dan kompleks dibandingkan dengan RNN, yang terdiri dari beberapa "unit LSTM" yang dapat mengingat informasi yang lama dan melupakan informasi yang tidak relevan. Setiap unit LSTM memiliki tiga jenis operasi yang disebut *input gate*, *forget gate* dan *output gate*. *Input gate* akan digunakan untuk menentukan informasi yang akan disimpan dalam *hidden state*, *forget gate* digunakan untuk menentukan informasi yang akan dihapus dari

*hidden state*, dan *output gate* digunakan untuk menentukan informasi yang akan digunakan untuk membuat prediksi.



Gambar 2. 5 Ilustrasi LSTM

Pada gambar 2.4 Dalam sebuah sel LSTM, dua keadaan ditransfer ke sel berikutnya, yaitu *cell state* dan *hidden state*. *Cell state* adalah aliran data utama, dimana data akan dikirimkan ke depan tanpa terjadi banyak perubahan, dan juga mungkin akan terjadi beberapa transformasi atau perubahan linear. Kemudian saat perhitungan aktivasi sigmoid dapat menambah atau menghapus data dari *cell state*. *Gate* adalah operasi matematika yang serupa dengan serangkaian atau lapisan operasi matriks. *Gate* memiliki bobot individu yang berbeda, yang menentukan bagaimana data ditambahkan atau dihapus.

a. *Forget Gates*

*Forget gate* bertanggung jawab untuk memutuskan apakah informasi lama yang ada dalam sel memori akan dihapus atau tidak. Jika *forget gate* menentukan bahwa informasi lama harus dihapus, maka informasi tersebut akan dihapus dari sel memori dan tidak akan mempengaruhi prediksi yang dibuat oleh jaringan (G. Jain *et al.*, 2019).

*Forget gate* memainkan peran penting dalam mengendalikan informasi yang diteruskan melalui jaringan LSTM dan memastikan bahwa hanya informasi yang relevan yang tersimpan dalam sel memori. Ini membantu memastikan bahwa jaringan LSTM memelihara informasi jangka panjang yang akurat dan membuat prediksi yang baik. Berikut rumus yang digunakan:

$$f_t = \sigma(W_f x_t + W_f h_{t-1} + b_f) \quad (2.3)$$

di mana  $f_t$  adalah vektor yang berisi nilai antara 0 dan 1 yang menentukan seberapa banyak informasi pada sel memori yang akan dihapus pada timestep  $t$ .  $W_f$  dan  $b_f$  adalah parameter untuk *forget gate*,  $h_{t-1}$  adalah *output* dari sel LSTM pada timestep sebelumnya, dan  $x_t$  adalah input pada timestep saat ini.  $\sigma$  adalah fungsi aktivasi sigmoid yang menghasilkan vektor  $f_t$  dengan elemen yang bernilai antara 0 dan 1.

#### b. *Input Gates*

*Input gate* bertanggung jawab untuk memutuskan apakah informasi baru yang masuk ke jaringan LSTM harus dimasukkan ke dalam sel memori atau tidak. Jika *input gate* menentukan bahwa informasi baru harus dimasukkan ke dalam sel memori, informasi tersebut akan disimpan dan digunakan untuk membuat prediksi. Pada *input gate* dta akan diproses melalui komponen  $i_t$ . Proses ini bertujuan untuk mengenali informasi yang perlu diperbarui, kemudian informasi tersebut akan disimpan di *cell state* menggunakan fungsi aktivasi *sigmoid*. Pada langkah ini, sebuah vektor *candidate cell* juga dibuat dengan menggunakan aktivasi *tanh*. Kemudian ditambahkan ke *cell state*  $c_t$ . Rumus untuk *input gate* dapat dijelaskan sebagai berikut:

$$i_t = \sigma(W_i x_t + W_i h_{t-1} + b_i) \quad (2.4)$$

Tahap berikutnya adalah menentukan apakah informasi baru ( $X_t$ ) akan disimpan dalam *cell state* dan mengupdate *cell state*. Tahap ini terdiri dari dua bagian, yang pertama adalah fungsi aktivasi sigmoid dengan kondisi jika outputnya 0 maka informasi akan diabaikan jika outputnya 1 maka informasi akan diperbarui. Kemudian dilanjutkan menggunakan fungsi aktivasi tanh yang digunakan untuk membuat vector baru yang kemudian disimpan pada *cell memory* dengan nilai kepentingan -1 hingga 1. Keduanya akan dilakukan perhitungan yang akan digunakan untuk memperbarui *cell state*, perhitungan tersebut dirumuskan sebagai berikut:

$$\check{c}_t = \tanh(W_c x_t + W_c h_{t-1} + b_c) \quad (2.5)$$

c. *Cell state*

Langkah berikutnya adalah menambahkan nilai vektor memori yang baru dihasilkan ke memori yang ada sebelumnya ( $c_{t-1}$ ). Proses ini menggantikan memori lama dengan memori baru dan menghasilkan rumus untuk *Cell gates* ( $C_t$ ) yang didapatkan dengan forgate gate dikalikan dengan *cell state* sebelumnya ditambah input kemudian dikalikan *cell state* pada current timestep.

$$c_t = f_t x_t * c_{t-1} + i_t * \check{c}_t \quad (2.6)$$

d. *Output Gates*

Tahap akhir terjadi pada komponen *output gate*, untuk mendapatkan output dari hidden dilakukan perhitungan fungsi aktivasi sigmoid, untuk *cell state* digunakan fungsi aktivasi tanh. Setelah mendapatkan kedua nilai tersebut

dilakukan perhitungan untuk mendapatkan nilai *output gate*. Berikut adalah rumus untuk *output gate*:

$$o_t = \sigma(W_{x_o}x_t + W_{h_o}h_{t-1} + b_o) \quad (2.7)$$

$$h_t = o_c * \tanh(c_t) \quad (2.8)$$

## **BAB III**

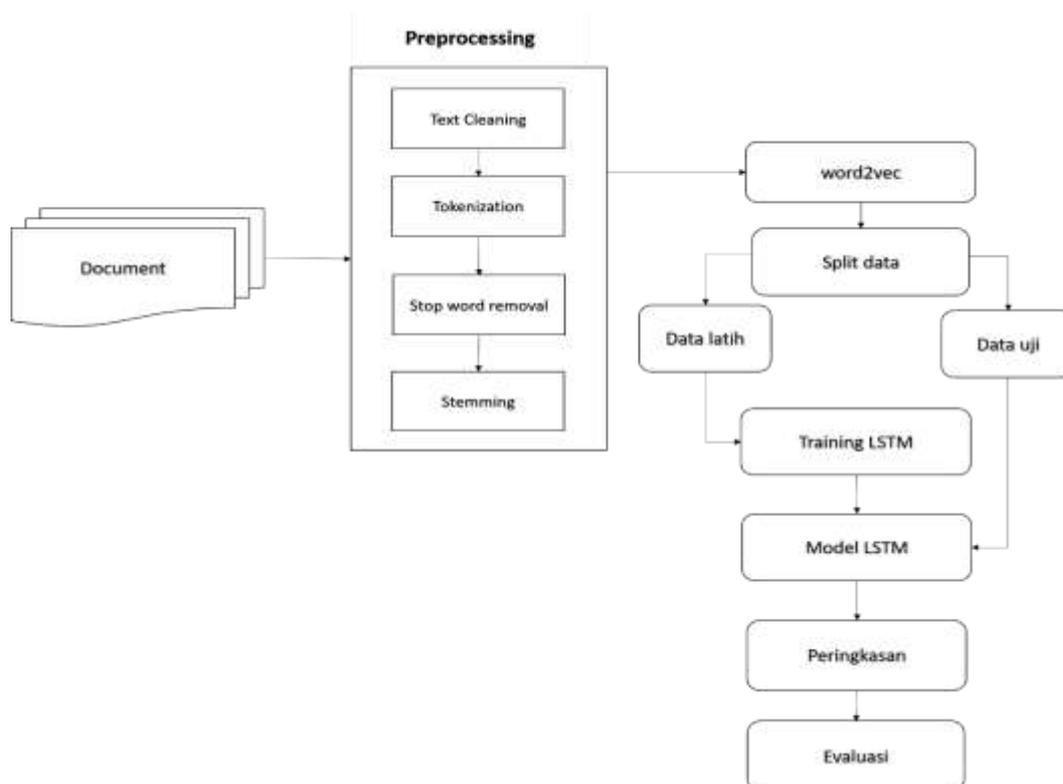
### **METODOLOGI PENELITIAN**

#### **3.1 Pengumpulan Data**

Dataset yang dipakai merupakan sekumpulan artikel berita Bahasa Indonesia bernama IndoSum yang dianggap sebagai benchmark untuk kasus peringkasan teks otomatis. IndoSum terdiri dari 19 ribu pasangan artikel berita dan ringkasannya yang dibuat oleh dua ahli bahasa Indonesia. Artikel-artikel pada dataset tersebut didapatkan dari beberapa portal berita online seperti CNN Indonesia, Kompas, dan Kumparan. Terdapat enam kategori berita pada dataset ini, yaitu tajuk utama, hiburan, olahraga, politik, headline, dan teknologi (Kurniawan & Louvan, 2019). Dalam setiap dataset juga terdapat label pada setiap kalimat yang digunakan sebagai acuan ringkasan.

#### **3.2 Desain Sistem**

Tahap pertama dalam membuat sistem yaitu *preprocessing* meliputi *text cleaning*, *tokenization*, *stopword removal*, dan *stemming*. Selanjutnya adalah melakukan ekstraksi fitur menggunakan *word2vec*. *word2vec* akan mengubah setiap kalimat menjadi vector. Vector yang didapat akan dilatih menggunakan metode LSTM. Model LSTM akan digunakan untuk memprediksi ringkasan dan akan dilakukan evaluasi menggunakan rouge untuk mendapatkan nilai akurasi dari model yang telah dibuat. Alur perancangan sistem dapat dilihat pada gambar 3.1 .



Gambar 3. 1 Desain Sistem

### 3.3 *Preprocessing*

*Preprocessing* adalah tahap awal dari proses peringkasan teks otomatis meliputi proses pembersihan dan pengolahan data sebelum digunakan untuk melatih model machine learning. Tujuannya adalah untuk membuat data yang akan digunakan menjadi berkualitas dan sesuai dengan kebutuhan model.

#### 3.3.1 **Text Cleaning**

*Text cleaning* adalah proses membersihkan teks dari karakter-karakter pada teks yang tidak digunakan, hal tersebut dapat memengaruhi performa pelatihan, karakter tersebut berupa, simbol, angka, tanda baca, dan karakter yang tidak memiliki makna, seperti (~!) @ & # \$% ^ \* \_ + <>.,? []: { } |.). Text cleaning

juga dapat mencakup proses menghilangkan kata-kata yang tidak relevan atau tidak penting, seperti kata-kata yang digunakan dalam bahasa sehari-hari, seperti "ah" atau "ya" dan kata-kata yang digunakan dalam bahasa yang tidak digunakan dalam teks yang dianalisis. *Text cleaning* juga dapat mencakup proses untuk mengonversi teks ke dalam bentuk yang lebih sederhana, seperti mengubah huruf besar menjadi huruf kecil.

Tabel 3. 1 Proses *text cleaning*

<b>Sebelum Cleaning</b>	<i>Dilansir AFP seorang warga Mesir yang dipercaya sebagai wanita terberat di dunia masuk sebuah rumah sakit di Abu Dhabi Kamis (5/5) dimana ia akan melanjutkan perawatan setelah menjalankan operasi menurunkan badan di India.</i>
<b>Setelah Cleaning</b>	<i>dilansir afp seorang warga mesir yang dipercaya sebagai wanita terberat di dunia masuk sebuah rumah sakit di abu dhabi kamis dimana ia akan melanjutkan perawatan setelah menjalankan operasi menurunkan badan di india</i>

### 3.3.2 Tokenization

Proses *Tokenization* dilakukan untuk mempermudah analisis. Dengan memecah teks menjadi token, kita dapat menganalisis kata-kata secara terpisah dan melakukan operasi seperti menghitung frekuensi kata, identifikasi frase, dan melakukan analisis lainnya. Tokenisasi juga dapat membantu meningkatkan akurasi model dengan memperjelas struktur teks dan memudahkan pemrosesan.

Tabel 3.2 merupakan contoh proses tokenisasi.

Tabel 3. 2 Proses *Tokenization*

<b>Sebelum Token</b>	<i>dilansir afp seorang warga mesir yang dipercaya sebagai wanita terberat di dunia masuk sebuah rumah sakit di abu dhabi kamis dimana ia akan melanjutkan perawatan setelah menjalankan operasi menurunkan badan di india</i>
<b>Setelah Token</b>	<i>dilansir, afp, seorang, warga, mesir, yang, dipercaya, sebagai, wanita, terberat, di, dunia, masuk, sebuah, rumah, sakit, di, abu, dhabi, kamis, dimana, ia, akan, melanjutkan, perawatan, setelah, menjalankan, operasi, menurunkan, badan, di, india</i>

### 3.3.3 Stopword removal

Menghilangkan *Stopword* dapat membantu meningkatkan kinerja dari algoritma pemrosesan teks dengan mengurangi jumlah data yang tidak relevan yang perlu dianalisis. Hal ini juga dapat mengurangi dimensi dari data dan membuatnya lebih mudah untuk mengidentifikasi kata atau frase yang penting.

Tabel 3. 3 Proses *Stopword*

<b>Sebelum <i>stopword</i></b>	<i>dilansir, afp, seorang, warga, mesir, yang, dipercaya, sebagai, wanita, terberat, di, dunia, masuk, sebuah, rumah, sakit, di, abu, dhabi, Kamis, dimana, ia, akan, melanjutkan, perawatan, setelah, menjalankan, operasi, menurunkan, badan, di, india</i>
<b>Setelah <i>stopword</i></b>	<i>dilansir, afp, warga, mesir, dipercaya, wanita, terberat, dunia, masuk, rumah, sakit, abu, dhabi, Kamis, dimana, melanjutkan, perawatan, setelah, menjalankan, operasi, menurunkan, badan, india</i>

### 3.3.4 Stemming

Dalam analisis teks, jumlah kata yang berbeda-beda dapat sangat besar, sehingga dapat memperbesar dimensi data dan mempersulit analisis. Dengan melakukan stemming, jumlah kata yang berbeda-beda dapat dikurangi, kemudian kata-kata yang memiliki akar kata yang sama akan dianggap sebagai satu kata, sehingga memudahkan analisis data dan meningkatkan akurasinya. Tabel 3.4 merupakan contoh proses *stemming*.

Tabel 3. 4 Proses *Stemming*

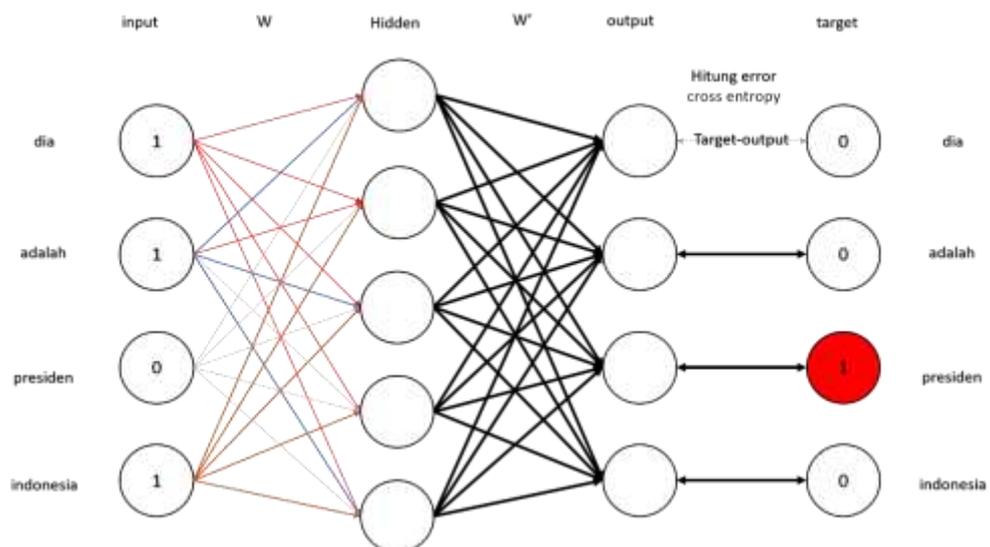
<b>Sebelum <i>Stemming</i></b>	<i>dilansir, afp, warga, mesir, dipercaya, wanita, terberat, dunia, masuk, rumah, sakit, abu, dhabi, Kamis, dimana, melanjutkan, perawatan, setelah, menjalankan, operasi, menurunkan, badan, india</i>
<b>Setelah <i>Stemming</i></b>	<i>dilansir, afp, warga, mesir, percaya, wanita, berat, dunia, masuk, rumah, sakit, abu, dhabi, Kamis, mana, lanjut, rawat, setelah, jalan, operasi, turun, badan, india</i>

### 3.4 Ekstraksi Fitur

Ekstraksi fitur pada peringkasan teks berita ini digunakan teknik *Word2Vec* untuk melakukan Ekstraksi fitur. *Word2Vec* digunakan sebagai salah satu metode untuk mengubah teks menjadi representasi vektor. Vector tersebut kemudian dijadikan input pada model LSTM. Langkah pertama adalah dalam perhitungan *word2vec* adalah mengubah setiap kata unik yang digunakan sebagai input menjadi *one-hot encoded vectors*. Setelah itu akan dilakukan Inisialisasi bobot secara acak pada  $W$  dan  $W'$

Tabel 3. 5 Vektor *One-hot*

Kata	vector
Dia	[1,0,0,0]
adalah	[0,1,0,0]
presiden	[0,0,1,0]
indonesia	[0,0,0,1]

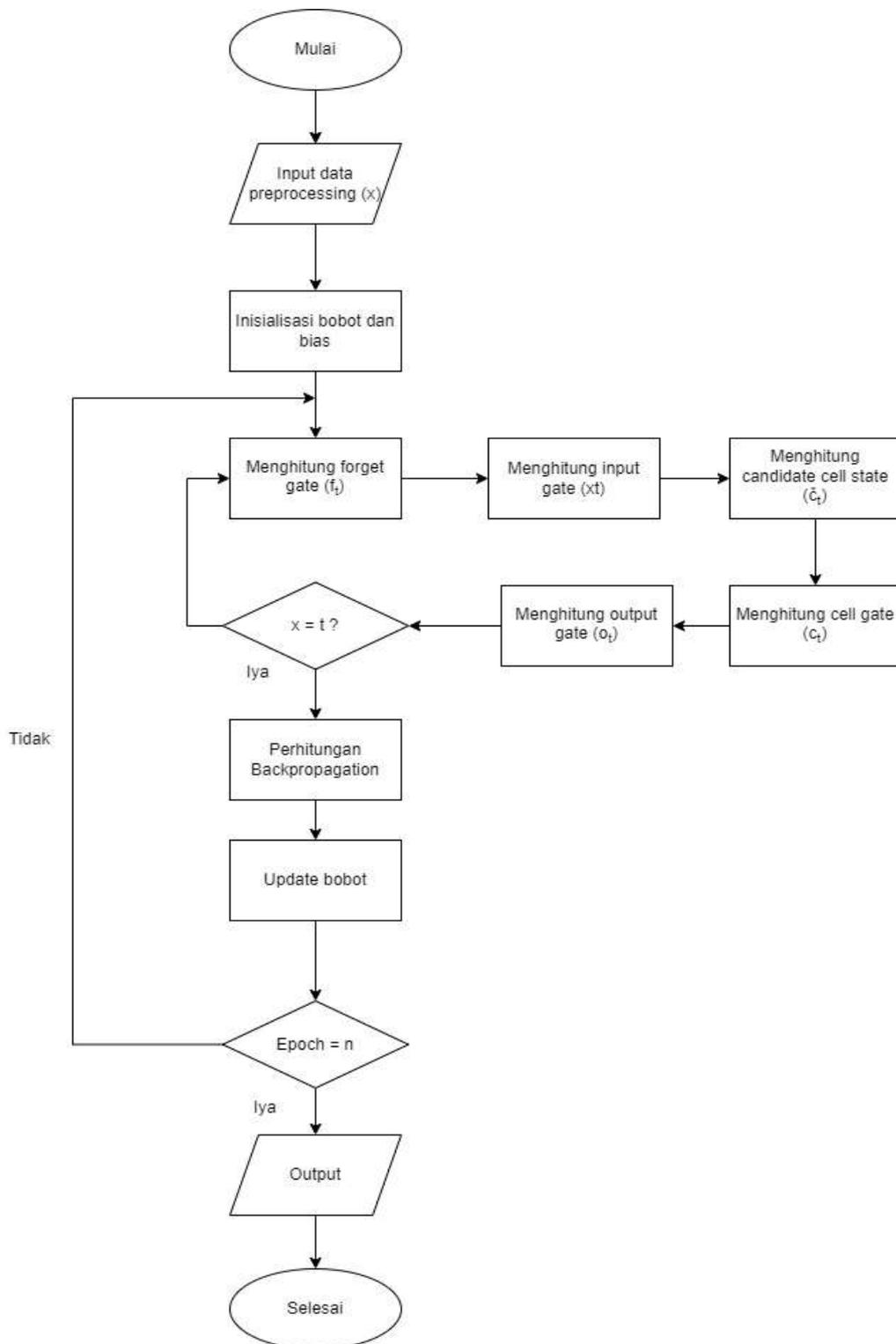


Gambar 3. 2 Ilustrasi proses *Continuous Bag of Word*

Proses Inisialisasi bobot pada  $W$  dan  $W'$  dilakukan secara acak. Kemudian akan dilakukan perhitungan *feedforward* dengan formula  $W = V \times N$  dan  $W' = N \times V$  dimana  $V$  adalah *vector one-hot* dan  $N$  adalah bobot, kemudian perhitungan formula tersebut akan menghasilkan nilai *hidden layer* yang akan digunakan untuk menghitung *output layer* dengan formula  $o = W' \times \text{hidden}$ . Kemudian akan dilakukan perhitungan nilai error menggunakan metode *cross-entropy*. Tahap tersebut diulangi sesuai dengan *epoch* yang di. Bobot  $W$  akhir dari pelatihan akan diambil dan dikalikan dengan vektor *one-hot encoded* dan menjadi vektor representasi kata

### 3.5 LSTM

Tahap selanjutnya yaitu melakukan proses pelatihan menggunakan metode LSTM. Sebelum dilakukan pelatihan data akan dibagi menjadi dua yaitu data latih dan data uji. Data latih digunakan sebagai dasar dalam melatih model peringkasan, sementara data uji digunakan untuk menguji sejauh mana model yang sudah dilatih dapat menghasilkan ringkasan yang akurat. Pada tahap pelatihan, setiap kalimat sudah diberikan label relevansi atau true/false, yang akan digunakan sebagai acuan dalam memilih kalimat-kalimat yang paling relevan dan signifikan dari dokumen-dokumen tersebut untuk dijadikan ringkasan. Proses perhitungan LSTM ditunjukkan pada Gambar 3.3



Gambar 3. 3 Flowchart perhitungan LSTM

Pelatihan model LSTM diawali dengan menghitung *forget gate*, *input gate*, *cell state*, *output gate*, dan *hidden*, perhitungan tersebut diulang sebanyak panjang inputan. Selanjutnya dilakukan proses backpropagation, dan update bobot. Proses tersebut diulang sebanyak epoch yang ditentukan atau diulang hingga model mencapai konvergensi dengan nilai loss dan akurasi mencapai nilai yang ditentukan.

Berikut adalah contoh sederhana langkah-langkah perhitungan LSTM. Sebelum melakukan perhitungan dilakukan pembobotan dan memberi nilai bias *term* terlebih dahulu pada setiap kata. Setelah pembobotan kata telah dilakukan Selanjutnya, akan dilakukan perhitungan menggunakan metode Long Short-Term Memory. Dalam perhitungan manual ini, bobot (W, U) dan bias (b) telah ditetapkan dengan nilai yang di asumsikan dari hasil penelitian sebelumnya. Nilai-nilai tersebut didapatkan dengan membatasi bobot dan bias yang digunakan dengan nilai yang telah ditentukan seperti yang tertera di bawah ini.:

$$\begin{bmatrix} W_i & U_i & b_i \\ W_c & U_c & b_c \\ W_f & U_f & b_f \\ W_o & U_o & b_o \end{bmatrix} = \begin{bmatrix} 0.5 & 0.25 & 0.01 \\ 0.3 & 0.4 & 0.05 \\ 0.03 & 0.06 & 0.002 \\ 0.02 & 0.04 & 0.025 \end{bmatrix}$$

Setelah itu kita telah memiliki nilai vector dari proses ekstraksi fitur, vector tersebut digunakan pada awal input-an. Berikut matriks dari kalimat yang akan dilakukan proses klasifikasi. Berikut adalah nilai vektor dari kalimat “dia adalah presiden Indonesia”:

[[−0.03451, 0.03468901, −0.21107519, −0.613516], [1.3572439, 0.1357186, 1.3350066, 0.5047232], [0.8071401, 0.0050894, −1.7244711, −0.4231063], [−2.5198526, −0.43120527, −0.6436673, 0.8652567]]

Vector diatas yang akan digunakan sebagai inputan  $x_1$  hingga  $x_4$ . Yang mana nilai  $x_1$  sampai  $x_4$ . Setelah itu akan dilakukan perhitungan LSTM sebagai berikut.

Input:  $x_1 = -0.03451$

Setelah mendapatkan nilai bobot,  $x$  dan bias, langkah pertama adalah melakukan perhitungan *forget gate*. Diketahui  $w_f$  adalah 0.03,  $U_f$  bernilai 0.06,  $h_{t-1}$  bernilai 0, dan  $b_f$  bernilai 0.002, maka didapatkan *forget gate* dengan nilai 0,51973092. Dari perhitungan manual seperti berikut.

Forget Gate:

$$f_1 = \sigma(W_f x_1 + W_f h_0 + b_f)$$

$$f_1 = \sigma(0.03 * -0.03451 + 0.06 * 0 + 0.002)$$

$$f_1 = \sigma(0.0789647)$$

$$f_1 = 0,51973092$$

Setelah mendapatkan nilai *forget gate* adalah menentukan input gate  $W_i$  sebesar 0,5  $U_i$  sebesar 0,25 dan  $b_i$  0.01 didapatkan nilai input 0.560390008 dengan formula sebagai berikut:

$$i_1 = \sigma(W_i x_1 + W_i h_{t-1} + b_i)$$

$$i_1 = \sigma(0.5 * -0.03451 + 0.25 * 0 + 0.01)$$

$$i_1 = \sigma(0.242745)$$

$$i_1 = 0.560390008$$

Setelah menghitung nilai forget gate dan input gate, tahap selanjutnya adalah menghitung candidate *cell state*. Perhitungan candidate *cell state* menggunakan persamaan yang sama dengan forget gate digunakan nilai bobot

$W_c = 0,3$  dan bias  $b_c = 0,05$ . Dengan menggunakan nilai tersebut, didapatkan hasil perhitungan candidate *cell state* sebagai berikut:

$$\check{c}_1 = \tanh(W_c x_t + W_c h_{t-1} + b_c)$$

$$\check{c}_1 = \tanh(0.3 * -0.03451 + 0.4 * 0 + 0.05)$$

$$\check{c}_1 = \tanh(0.34647)$$

$$\check{c}_1 = 0.5857613$$

Setelah melakukan ketiga operasi perhitungan yang dibutuhkan oleh *Cell Gate*, dilakukan perhitungan *Cell State*. Hal ini karena nilai *Cell Gate* membutuhkan nilai dari *Forget Gate*, *Input Gate*, dan *Candidate Cell State*. Dalam proses ini, nilai *Cell State* akan dihasilkan sebagai berikut:

$$C1 = (f_1 + C_t + i_1 * C'_1)$$

$$C1 = (0,51973092 * 0 + 0.560390008 * 0.5857613)$$

$$C1 = 0,3279868750327284$$

Tahap terakhir adalah menentukan nilai *output gate*. Setelah melakukan perhitungan *output gate* kita bisa mendapatkan nilai *hidden* pada *output*. Nilai *output gate* didapatkan 0.5057556071, dan *hidden* dengan nilai 0.003017. berikut perhitungannya:

$$o_1 = \sigma(W_{xo} x_t + W_{ho} h_{t-1} + b_o)$$

$$o_1 = \sigma(0.02 * -0.03451 + 0.04 * 0 + 0.025)$$

$$o_1 = \sigma(0.02431)$$

$$o_1 = 0.5057556071$$

$$h_1 = o_1 * \tanh(C1)$$

$$h_1 = 0.5057556071 * \tanh(0,3279868750327284)$$

$$h_1 = 0.003017$$

Dari perhitungan diatas didapatkan nilai  $h_1$  adalah 0.003017, kemudian nilai tersebut akan digunakan sebagai inputan pada tahap selanjutnya ( $x_{t+1}$ ). Perhitungan tersebut diulang sampai pada inputan terakhir yaitu  $x_n$  yang digunakan hidden sebelum *sequence* terakhir ( $h_{n-1}$ ) sebagai inputan.

### 3.6 Peringkasan

Setelah model LSTM dilatih, Model tersebut dapat digunakan untuk melakukan peringkasan teks ekstraktif. Caranya adalah dengan memberikan teks berita sebagai input, dan model akan memberikan label true atau false pada setiap kalimat dalam teks. Kalimat dengan label true akan dipertahankan dan dijadikan sebagai ringkasan teks, sedangkan kalimat dengan label false akan dihapus.

### 3.7 Evaluasi

Evaluasi pada penelitian ini akan menggunakan metode Rouge. Rouge adalah metode yang digunakan untuk evaluasi peringkasan. Terdapat 3 model metode Rouge yaitu Rouge-1, Rouge-2 dan Rouge-L. Pada penelitian ini akan digunakan Rouge-1. Rouge-1 digunakan untuk mengukur sejauh mana sebuah sistem ringkasan teks mampu menghasilkan ringkasan yang sesuai dengan dokumen aslinya. ROUGE-1 umumnya digunakan dalam penelitian dan pengembangan sistem ringkasan otomatis. Model ROUGE bekerja dengan cara menghitung kesamaan kata-kata yang terdapat dalam ringkasan yang dihasilkan oleh sistem dengan gold summaries pada setiap data uji secara unigram atau satu kata demi satu kata. Dari hasil perhitungan tersebut, akan dihitung nilai recall,

precision, dan f1-score untuk mengevaluasi kinerja sistem. (Naufal Ammar, 2020). Rumus ROUGE-1 adalah sebagai berikut:

$$precision = \frac{(jumlah\ kata\ overlapping)}{(jumlah\ kata\ di\ ringkasan\ otomatis)} \quad (3.1)$$

$$recall = \frac{(jumlah\ kata\ overlapping)}{(jumlah\ kata\ di\ ringkasan\ manusia)} \quad (3.2)$$

$$f - measure = \frac{(2 * precision * recall)}{(precision + recall)} \quad (3.3)$$

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Pengelolaan Dataset

Data yang digunakan adalah 50 artikel multi-dokumen dari IndoSum. Artikel multi dokumen tersebut merupakan kumpulan dari beberapa artikel berita IndoSum digabung menjadi satu *story (multi-document)* berdasarkan kata kunci dan kategori yang ditentukan secara manual dengan masing-masing multi dokumen berisi 3 sampai 5 dokumen berita IndoSum. Kemudian 50 multi dokumen tersebut akan dilabeli oleh 5 responden, label dengan pilihan terbanyak akan digunakan untuk menentukan label pada setiap kalimat dalam rangkuman. Dataset multi dokumen ditunjukkan pada tabel 4.1

Berikut adalah contoh artikel dari beberapa berita yang digabung berdasarkan kata kunci “Filipina, Perang, Ham” dengan kategori “tajuk utama”. Dari kata kunci dan kategori tersebut didapatkan tiga berita dengan judul “*Mati Dalam Operasi Anti narkotika Filipina Pro*”, “*Perang Duterte Lawan Narkoba Dipergencar*”, “*Filipina Diserang Isu Pelanggaran Ham*”, “*Rodrigo Duterte Diselidiki Mahkamah Internasional*”. Ketiga berita tersebut akan digabung menjadi 1 berita seperti pada gambar 4.1.

Jakarta ( ANTARA News ) - Sekretaris Komunikasi Presiden Filipina Martin Andanar menyatakan di Istana Malacanang bahwa pemerintah telah memenangkan fase pertama perang melawan narkoba . Andanar menegaskan bahwa operasi antinarkoba tanpa lelah pemerintah yang sudah merenggut 2.000 nyawa dalam 60 hari pemerintahan Presiden Rodrigo Duterte itu akan memasuki fase kedua . Menurut Andanar , fase kedua itu adalah termasuk rehabilitasi sekitar 700.000 pengguna narkoba yang menyerahkan diri kepada polisi karena takut dibunuh . " Presiden ingin mengakhiri masalah narkoba ini sesegera mungkin . Presiden sudah berjanji bahwa dalam jangka tiga sampai enam bulan , masalah itu terselesaikan 70 sampai 80 persen , " kata Andanar dalam laman koran terkemuka Filipina , The Inquirer . Andanar mengatakan yang dimaksud Duterte dengan pengguna narkoba tak hanya mereka yang menghisap shabu setiap hari , namun juga mereka yang kadang-kadang menggunakan substansi non sintesis dan lainnya .

FILIPINA- Pemberantasan kejahatan narkoba di Filipina tidak menunjukkan tanda-tanda mereda , meski sejumlah kalangan mengecam tindakan yang dilakukan pemerintah pimpinan Presiden Rodrigo Duterte . Dalam operasi yang digelar sepanjang Rabu ( 16 / 08 ) malam di ibu kota Manila , polisi mengatakan mereka menewaskan setidaknya 26 tersangka pelaku kejahatan narkoba . Senin ( 14 / 08 ) lalu polisi menggelar operasi serupa di Provinsi Bulacan , provinsi tetangga Manila , yang menewaskan 32 orang , menjadikan total korban keseluruhan dalam empat hari terakhir mencapai setidaknya 58 orang . Presiden Duterte memuji langkah polisi dan mengatakan Filipina memang perlu mempercepat perang melawan kejahatan dan narkoba . Ia mengatakan jika banyak tersangka penjahat yang ditembak mati setiap hari , negaranya akan bisa mengatasi masalah narkoba . " Tiga puluh dua orang tewas dalam operasi besar di Bulacan . Ini perkembangan yang bagus . Jika kita bisa menewaskan 32 tersangka setiap hari , kita akan bisa mengatasi masalah narkoba di negara ini , " kata Presiden Duterte . Juru bicara kepolisian Manila , Kolonel Erwin Margarejo , kepada kantor berita Reuters , mengatakan bahwa operasi di Manila pada Rabu malam digolongkan sebagai ' operasi besar ' yang mengacu pada operasi terkoordinasi yang mengerahkan personel dalam jumlah besar . Lokasi operasi adalah distrik - distrik rawan kejahatan , biasanya dihuni oleh masyarakat kalangan bawah . Dikecuan Komisi HAM Ia mengatakan tindakan tegas harus diambil karena para tersangka melawan aparat . Tidak ada penjelasan mengapa polisi Filipina mempercepat operasi menindak penjahat narkoba pekan ini . Laporan polisi menyebutkan 223 orang ditahan dalam operasi di Manila dan Bulacan . " Presiden tidak memerintahkan ( polisi ) untuk membunuh .

Jakarta , CNN Indonesia - - Sejumlah negara memperingatkan Filipina atas dugaan pembunuhan massal di luar hukum dalam perang melawan narkoba yang telah berlangsung sejak Presiden Rodrigo Duterte menjabat . Dalam sidang siklus ke - 3 Universal Periodic Review ( UPR ) Dewan HAM PBB , di Jenewa , diplomat dari seluruh benua dilaporkan mengecam lonjakan kematian yang terjadi selama operasi anti - narkoba Duterte berjalan . " Kanada menyerukan Manila untuk mengakhiri pembunuhan di luar hukum , penghilangan secara paksa , penangkapan hingga penyiksaan secara ilegal , " bunyi pernyataan delegasi pemerintahan Perdana Menteri Justin Trudeau , Senin ( 8 / 5 ) . Selain Ottawa , Australia , Brasil , Perancis , Jerman , dan Ghana merupakan salah satu negara yang turut mengecam tindakan Filipina itu . Sejak pertengahan 2016 lalu , 2.692 orang dilaporkan tewas di tangan polisi Filipina dalam operasi anti - narkoba . Sekitar 1.847 orang dibunuh oleh pihak yang tidak bisa ditentukan . Sementara itu , sedikitnya 5.500 terduga kriminal narkoba tewas tanpa melalui proses hukum jelas , di saat penyelidikan berlangsung . Senator Filipina Alan Cayetano , sekutu Duterte , mencela kecaman terhadap negaranya itu , menganggap kampanye advokat HAM dan media selama ini dilakukan untuk mengubah persepsi upaya anti - narkoba pemerintahnya . " Tidak ada gelombang pembunuhan di Filipina , " ucap Cayetano dalam sidang itu . Dia mengatakan , beberapa musuh menggunakan " taktik politik " .

FILIPINA - Mahkamah Pidana Internasional akan menyelidiki dugaan pelanggaran hukum dalam perang melawan narkoba yang dilancarkan Presiden Filipina Rodrigo Duterte . Jaksa Mahkamah Pidana Internasional , Fatou Bensouda , mengatakan pihaknya secara khusus akan menyelidiki laporan pembunuhan yang diduga dilakukan polisi saat menggelar operasi memburu pedagang dan pengedar narkoba . " Saya memantau kasus ini sejak 2016 ... setelah melalui pertimbangan yang mendalam , saya memutuskan untuk melakukan penyelidikan awal , " kata Bensouda melalui pernyataan tertulis hari Kamis ( 08 / 02 / 2018 ) . Bensouda menyebut ribuan orang tewas sejak 1 Juli 2016 dalam perang melawan narkoba yang dilancarkan Presiden Rodrigo Duterte . Duterte dituding memerintahkan pembunuhan ribuan tersangka pengedar narkoba . Beberapa pihak mengatakan mereka dibunuh secara sewenang-wenang dan pembunuhan tersebut tidak melalui proses hukum yang semestinya . " Menurut laporan , mereka tewas setelah terlibat bentrok dengan kelompok pengedar lain , ( namun ) diduga ada pula yang tewas akibat tindakan polisi yang melakukan pembunuhan tanpa didahului proses hukum , " jelas Bensouda . Pemerintah di Manila sudah mendapatkan pemberitahuan tentang penyelidikan awal ini . Juru bicara Presiden Duterte , Harry Roque , mengatakan tidak ada hukum yang dilanggar oleh Presiden Duterte dalam perang melawan narkoba .

Gambar 4. 1 Dokumen gabungan

Tabel 4. 1 Dataset *multi-document*

id_story	id_document	query	kategori	judul	content	responden1	responden2	responden3	responden4	responden5	label
1	[3097, 5405, 7571, 10518]	[presiden, perang, melawan]	tajuk utama	[ Mati Dalam Operasi Antinarkotika Filipina Pr...	[[[jakarta, news, sekretaris, komunikasi, pres...	[[True, False, False, True], [Tr...	[[True, True, False, True, True, False], [True...	[[False, True, False, True], [Tru...	[[True, False, True, False, True], [Tru...	[[True, True, True, True, False], [Fals...	[[True, True, False, True, True], [True...
2	[6, 221, 4474]	[printer, portable]	teknologi	[Hp Luncurkan Kamera Instan Sprocket In , Jaj...	[[[hp, luncur, buah, printer, portable, cetak,...	[[True, True, False, False, False, Fals...	[[True, True, False, False, True, False, True,...	[[True, False, True, False, True, False, True,...	[[True, False, True, True, False, True, ...	[[True, True, False, True, True, True, F...	[[True, True, False, True, True, False, True,...
3	[6126, 10164, 14239]	[startup, bukalapak, founder]	teknologi	[Arkavidia Itb Akan Fokus Bahas Perkembangan ...	[[[arkavidia, kali, himpun, mahasiswa, informa...	[[False, True, True, False, False, False, Fals...	[[True, True, True, False, False, False, True,...	[[True, False, True, False, False, True, False...	[[True, True, True, False, False, True, False,...	[[True, False, True, True, True, False, ...	[[True, True, True, False, False, True, False,...
4	[206, 779, 7159]	[laptop, gaming, asus]	teknologi	[Asus Xiu Laptop Multimedia Gaming Ready, Asus...	[[[laptop, gaming, komoditas, panas, jujur, ha...	[[True, True, False, False, False, True, False...	[[True, True, False, False, False, True, False...	[[True, True, False, False, False, True, False...	[[True, True, True, False, True, False, False,...	[[True, True, True, True, True, False, T...	[[True, True, False, False, True, False, False...

Berita yang telah digabung akan dipisah per kalimat. Kalimat yang telah dipisah akan dilabeli satu-persatu oleh kelima responden. Label tiap berita berupa angka 1 dan 0 dimana angka 1 menunjukkan kalimat yang akan digunakan sebagai ringkasan, sementara kalimat yang labelnya 0 akan dibuang. Label yang paling banyak dipilih akan dijadikan acuan sebagai ringkasan. Proses pelabelan ditunjukkan pada tabel 4.3, jika label 1 dipilih 3 orang atau lebih maka labelnya 1 jika sebaliknya maka labelnya 0.

Tabel 4. 2 Tabel berita perkalimat

No. Kalimat	Kalimat
1	Jakarta (ANTARA News) - Sekretaris Komunikasi Presiden Filipina Martin Andanar menyatakan di Istana Malacanang bahwa pemerintah telah memenangkan fase pertama perang melawan narkotik
2	Andanar menegaskan bahwa operasi antinarkotika tanpa lelah pemerintah yang sudah merenggut 2.000 nyawa dalam 60 hari pemerintahan Presiden Rodrigu Duterte itu akan memasuki fase kedua.
3	Menurut Andanar , fase kedua itu adalah termasuk rehabilitasi sekitar 700.000 pengguna narkotika yang menyerahkan diri kepada polisi karena takut dibunuh.
4	" Presiden ingin mengakhiri masalah narkotika ini sesegera mungkin . Presiden sudah berjanji bahwa dalam jangka tiga sampai enam bulan , masalah itu terselesaikan 70 sampai 80 persen , " kata Andar dalam laman koran terkemuka Filipina , The Inquirer
5	Andanar mengatakan yang dimaksud Duterte dengan pengguna narkotika tak hanya mereka yang menghisap shabu setiap hari, namun juga mereka yang kadang-kadang menggunakan substansi non sintetis dan lainnya.
.....	.....
60	Filipina adalah negara pertama di Asia Tenggara yang diselidiki oleh Mahkamah Pidana Internasional.

Kalimat yang telah dipisah akan dilabeli satu-persatu oleh kelima responden. Label yang paling banyak dipilih akan dijadikan acuan sebagai ringkasan. Jika label 1 dipilih 3 orang atau lebih maka labelnya 1 jika sebaliknya maka labelnya 0. Proses pelabelan ditunjukkan pada tabel 4.3

Tabel 4. 3 Label tiap responden

No. Kalimat	Responden dan label pilihan responded					Label
	1	2	3	4	5	
1	1	1	0	1	0	1
2	0	1	1	0	1	1
3	0	0	0	0	0	0
4	0	1	1	1	1	1
5	0	1	0	0	0	0
.....	.....	.....	.....	.....	.....	.....
60	0	0	1	0	1	0

## 4.2 Skenario Pengujian

### 4.2.1 Ekstraksi Fitur Menggunakan Word2vec

Dalam skenario pengujian, dilakukan penggunaan dua jenis Model *Word2Vec* yaitu *Skip Gram* dan *CBOW*. Selanjutnya, digunakan beberapa window pada *Word2Vec* yaitu *window* 5 dan 10 dan akan digunakan dimensi vektor 100 dan 200 pada setiap window pada model untuk menguji kinerja model dalam melakukan ekstraksi fitur pada teks yang akan diproses:

Tabel 4. 4 Tabel Skenario pengujian

Algoritma Pelatihan	Ukuran Jendela (Window Size)	Dimensi Vektor
<i>Continuous Bag-of-Words (CBOW)</i>	5	100
		200
	10	100
		200
<i>Skip-gram</i>	5	100
		200
	10	100
		200

Setelah menentukan parameter-parameter yang digunakan, langkah selanjutnya adalah melanjutkan dengan proses pelatihan. Dari proses pelatihan tersebut akan menghasilkan vector yang merepresentasikan kata, setelah itu akan

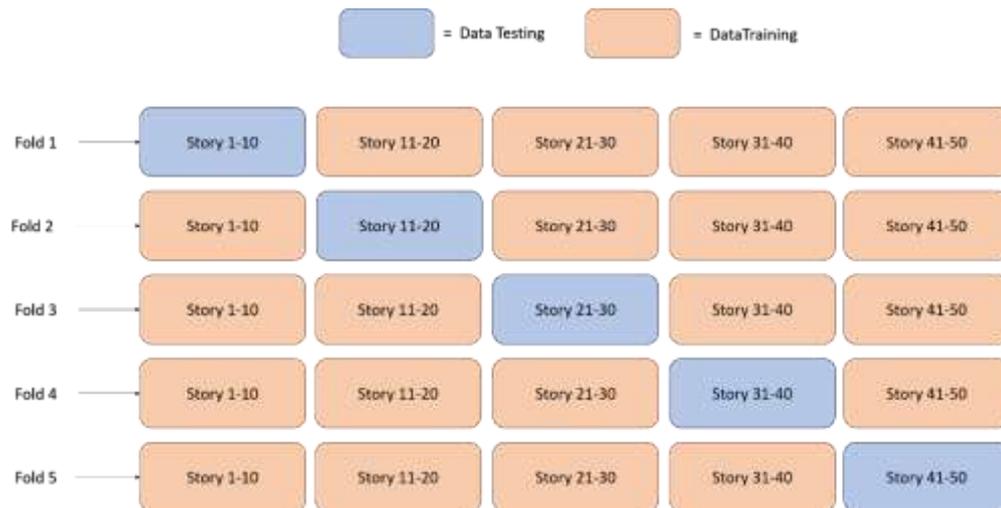
dilakukan perhitungan untuk mendapatkan vector kalimat dengan cara menghitung rata-rata semua vector kata dalam satu kalimat.

Tabel 4. 5 Vector tiap kalimat

<b>Id Berita</b>	<b>Kalimat dari kumpulan kata</b>	<b>Vector 100 dimensi</b>
1501893029-lula-kamal-dokter-ryan-thamrin-sakit-sejak-setahun	['jakarta', 'cnn', 'indonesia', 'dokter', 'ryan', 'thamrin', 'kenal', 'acara', 'dokter', 'oz', 'indonesia', 'tinggal', 'dunia', 'jumat'],	[ 2.1233628 , -0.42484412, -3.1331725 , ..., -0.1663143 , 2.6908634 , -0.64754945 ]
Presiden Filipina Melawan narkoba	['dokter', 'lula', 'kamal', 'rupa', 'selebriiti', 'rekan', 'kerja', 'ryan', 'kawan', 'sakit']	[ 0.43217626, -1.2343202 , -3.38685 , ..., 2.3532481 , 0.6580384 , 1.6339358 ]
	['lula', 'sakit', 'ryan', 'mesti', 'vakum', 'giat', 'masuk', 'bawa', 'acara', 'dokter', 'oz', 'indonesia']	[-2.2838519 , -0.47990146, 0.96118516, ..., 2.7701907 , -2.4300852 , -2.0495987 ]
	['kondisi', 'ryan', 'kampung', 'halaman', 'pekanbaru', 'riau', 'jalan', 'istirahat']	[ 0.129537 , -1.1804565 , -5.2839155 , ..., 2.2329247 , -1.8160264 , 1.4195149 ]
	['ryan', 'thamrin', 'kenal', 'dokter', 'rutin', 'tips', 'informasi', 'sehat', 'tayang', 'dokter', 'oz', 'indonesia']	[-1.3686111 , -0.5360907 , 0.28794196, ..., -0.3411763 , 0.9752968 , 4.1221366 ]
	['ryan', 'tempuh', 'didik', 'dokter', 'fakultas', 'dokter', 'universitas', 'gadjah', 'mada']	[-1.5214654 , -1.4726365 , -5.4147825 , ..., 1.4491113 , 1.5164956 , 2.5264192 ]
	['didik', 'klinis', 'sehat', 'reproduksi', 'sakit', 'tular', 'seksual', 'mahachulalongkornrajavidyalaya', 'university', 'bangkok', 'thailand']	[ 3.706257 , -1.6682606 , -0.45880204, ..., 1.9889855 , -1.6105212 , 0.2123917 ]

#### 4.2.2 *Splitting Data*

Data akan dibagi menjadi data training dan testing menggunakan metode *K-fold Cross Validation* dengan 5-fold. Dilakukan dari fold 1 sampai dengan fold 5 untuk mengukur nilai Rouge-1 pada masing-masing story dalam dataset. Pada fold 1, story 1-10 akan menjadi data testing dan story 11-50 akan menjadi data training. Percobaan tersebut dilakukan sampai fold 5 untuk memperoleh nilai ROUGE-1 dari masing-masing story multi dokumen.



Gambar 4. 2 K-Fold Cross Validation

### 4.2.3 Pelatihan Model LSTM

Parameter yang diujikan pada model LSTM adalah banyaknya unit/neuron pada model dan jumlah learning rate. Jumlah neuron yang digunakan pada pengujian adalah 50, 100, 150, dan 200 dimana pada setiap neuron akan digunakan *learning rate* 0,001-0,009. Tujuan pengujian adalah untuk mengetahui pengaruh jumlah *neuron* dan *learning rate* terhadap performa kinerja model

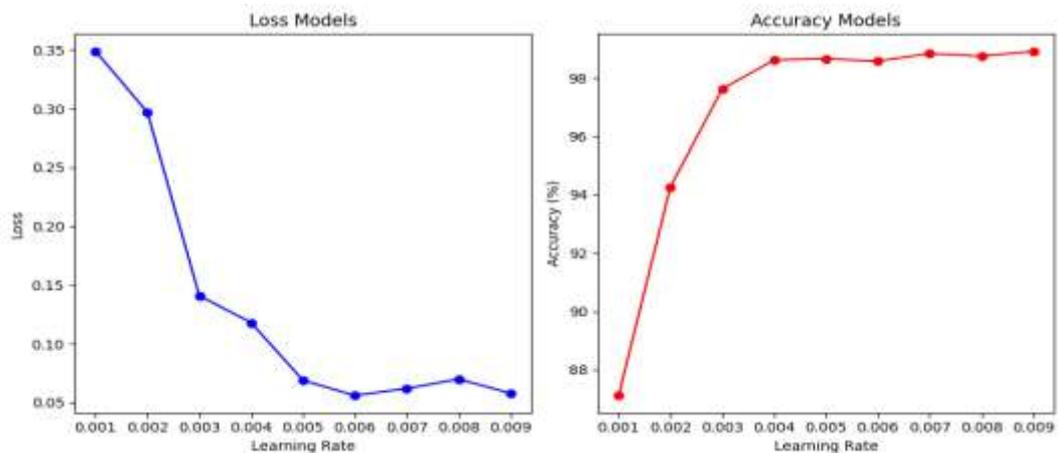
Tabel 4. 6 Parameter model LSTM

Unit / neuron	Learning rate
50	0,001
	0.002
	0.003
	0.004
	0,005
	0.006
	0.007
	0.008
	0.009
100	0,001
	0.002
	0.003
	0.004
	0,005
	0.006
	0.007
	0.008
	0.009

150	0,001
	0.002
	0.003
	0.004
	0.005
	0.006
	0.007
	0.008
	0.009
200	0,001
	0.002
	0.003
	0.004
	0.005
	0.006
	0.007
	0.008
	0.009

Tabel 4. 7 Hasil pelatihan model 50 neuron

Model	Learning Rate	Loss	Akurasi	Waktu Pelatihan (detik)
1	0.001	0.349	83.20%	41.641
2	0.002	0.297	86.69%	46.027
3	0.003	0.141	94.36%	42.582
4	0.004	0.118	96.43%	47.232
5	0.005	0.069	98.02%	47.447
6	0.006	0.056	98.06%	43.005
7	0.007	0.062	98.15%	46.851
8	0.008	0.07	97.98%	45.64
9	0.009	0.058	97.67%	42.561

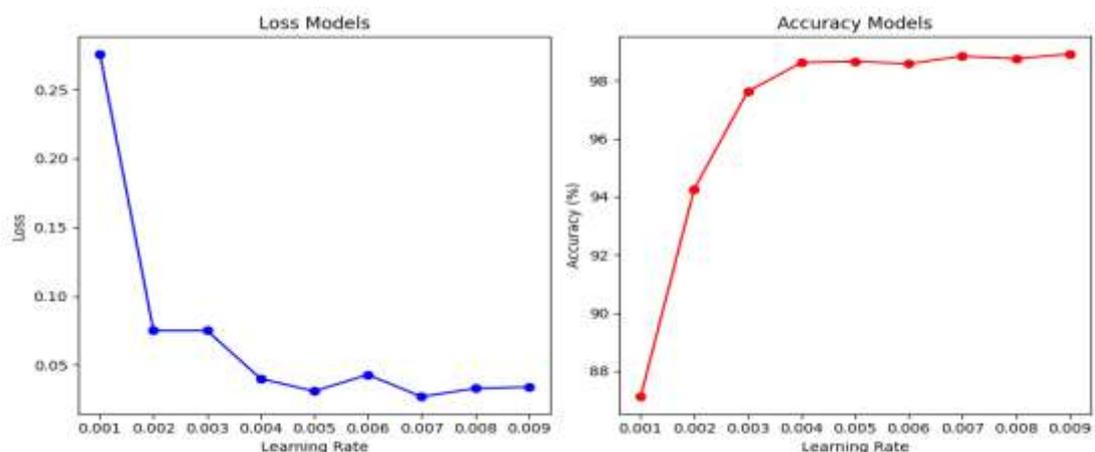


Gambar 4. 3 Grafik loss dan akurasi model 50 neuron

Tabel 4.7 merupakan hasil dari pelatihan dari model menggunakan 50 neuron. Model 1 memiliki nilai loss tertinggi yaitu 0.349 dan akurasi terendah yaitu 83.20%. Model 6 memiliki nilai loss terendah dan akurasi tertinggi yaitu 0,056 dan 98,06%.

Tabel 4. 8 Hasil pelatihan model 100 neuron

Model	Learning Rate	Loss	Akurasi	Waktu Pelatihan (detik)
10	0.001	0.276	87.12%	52.656
11	0.002	0.075	94.27%	52.938
12	0.003	0.075	97.63%	89.285
13	0.004	0.04	98.62%	86.449
14	0.005	0.027	98.84%	55.002
15	0.006	0.043	98.58%	54.014
16	0.007	0.031	98.66%	54.425
17	0.008	0.033	98.75%	86.49
18	0.009	0.034	98.92%	86.613

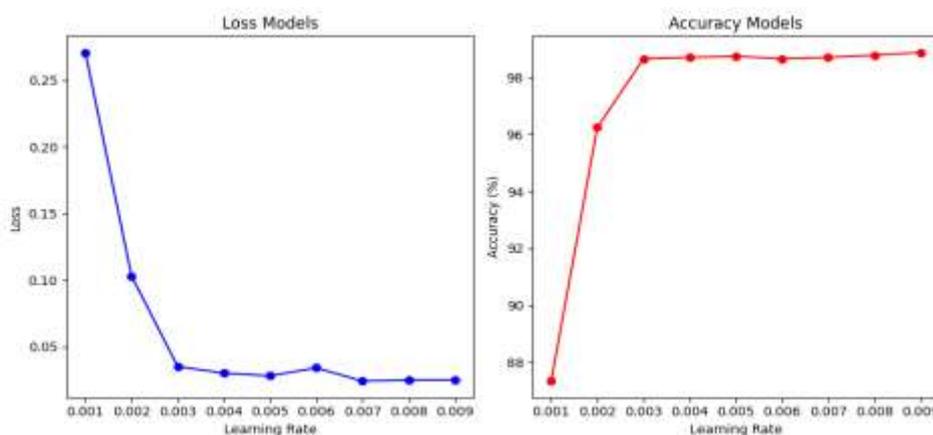


Gambar 4. 4 Grafik loss dan akurasi model 100 neuron

Tabel 4.8 Menunjukkan nilai loss dan akurasi pada model 100 neuron. Nilai loss terendah adalah 0.027 terdapat pada model 15 dengan learning rate 0.005, untuk akurasi tertinggi adalah 98.92% terdapat pada model 18. Sedangkan untuk loss tertinggi dan akurasi terendah terdapat pada model 10 dengan learning rate 0.001 yaitu 0.276 dan 87.12%.

Tabel 4. 9 Hasil pelatihan model 150 neuron

Model	Learning Rate	Loss	Akurasi	Waktu Pelatihan (detik)
19	0.001	0.271	87.34%	146.356
20	0.002	0.103	96.25%	146.649
21	0.003	0.035	98.66%	146.484
22	0.004	0.03	98.71%	94.721
23	0.005	0.028	98.75%	92.631
24	0.006	0.034	98.66%	90.64
25	0.007	0.024	98.71%	146.878
26	0.008	0.025	98.79%	147.928
27	0.009	0.025	98.88%	147.555



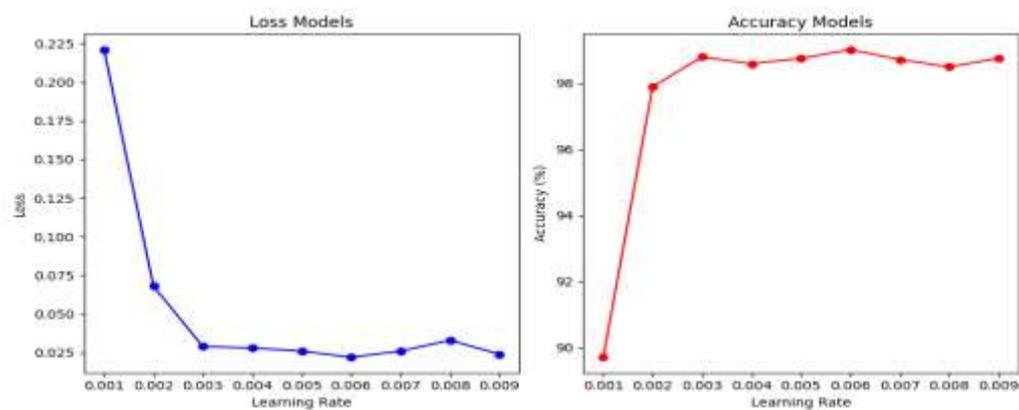
Gambar 4. 5 Grafik loss dan akurasi model 150 neuron

Tabel 4.9 Menunjukkan nilai loss dan akurasi pada model 150 neuron. Nilai loss terendah adalah 0.024 terdapat pada model 25 dengan learning rate 0.007, untuk akurasi tertinggi adalah 98.88% terdapat pada model 27 dengan learning rate 0.009. Sedangkan untuk loss tertinggi dan akurasi terendah terdapat pada model 19 yaitu 0.271 dan 87.34%.

Tabel 4.10 Menunjukkan nilai loss dan akurasi pada model 200 neuron. Nilai loss terendah adalah 0.022 akurasi tertinggi adalah 99.01% yang terdapat pada model 33. Sedangkan untuk loss tertinggi dan akurasi terendah terdapat pada model 28 yaitu 0.221 dan 89.71%.

Tabel 4. 10 Hasil pelatihan model 200 neuron

Model	Learning Rate	Loss	Akurasi	Waktu Pelatihan (detik)
28	0.001	0.221	89.71%	147.99
29	0.002	0.068	97.89%	118.049
30	0.003	0.029	98.79%	120.654
31	0.004	0.028	98.58%	146.902
32	0.005	0.026	98.75%	120.898
33	0.006	0.022	99.01%	146.471
34	0.007	0.026	98.71%	151.276
35	0.008	0.033	98.49%	146.839
36	0.009	0.024	98.75%	148.68



Gambar 4. 6 Grafik loss dan akurasi model 200 neuron

Dari Hasil pengujian menunjukkan bahwa jumlah neuron yang lebih besar cenderung menunjukkan performa yang lebih baik, terutama dalam hal akurasi. Kemudian, terdapat pola bahwa learning rate yang optimal dapat berbeda-beda tergantung pada jumlah neuron. Misalnya, pada model dengan 50 neuron, learning rate 0.005 memberikan hasil terbaik, sementara pada model dengan 100 dan 150 neuron, learning rate 0.005 dan 0.007 memberikan hasil terbaik. Kemudian, terdapat variasi waktu pelatihan yang signifikan, terutama pada model dengan jumlah neuron yang lebih tinggi. Model dengan jumlah neuron 200 membutuhkan waktu pelatihan yang lebih lama dibandingkan dengan model-model lainnya. Hal

ini mungkin disebabkan oleh kompleksitas yang lebih tinggi dari model dengan jumlah neuron yang lebih besar.

Berdasarkan hasil pengujian, pemilihan model yang optimal adalah model dengan 100 neuron dan learning rate 0.005. Model ini memberikan akurasi yang tinggi sekitar 98.84% dan juga loss yang rendah yaitu 0.027 dengan waktu pelatihan yang relatif efisien sekitar 55.002 detik. Untuk model dengan akurasi tertinggi dan loss terendah sebenarnya terdapat pada model dengan 200 neuron dan learning rate 0.006 dengan nilai akurasi mencapai 99.01%, dan loss mencapai 0.022 namun model tersebut membutuhkan waktu pelatihan yang cukup lama yaitu sekitar 146.471 detik.

### **4.3 Hasil Uji Coba**

Hasil uji coba menggunakan metode *Rouge-1* untuk mengevaluasi kinerja model dalam melakukan peringkasan dokumen berita. *Rouge-1* merupakan metrik evaluasi yang fokus pada kesamaan unigram antara ringkasan yang dihasilkan oleh model dengan ringkasan referensi yang sebenarnya.

#### **4.3.1 Hasil Uji Coba Menggunakan Model CBOW**

Untuk mendapatkan nilai *rouge-1* akan dilakukan perhitungan dengan membandingkan hasil ringkasan manusia dengan hasil ringkasan system. Hasil uji coba akan menampilkan nilai *recall*, *presisi*, *f-measure*. Dimana *recall* merupakan perbandingan jumlah kata ringkasan system dan manusia dengan ringkasan manusia sedangkan *presisi* merupakan perbandingan jumlah kata pada ringkasan system dan manusia dengan ringkasan system.

Game - game MOBA di perangkat bergerak belakangan menjadi sorotan berkat dilangsungkannya berbagai turnamen yang menjanjikan jumlah hadiah melimpah . Tapi meski MOBA semakin mainstream , PC sebagai tempat lahirnya genre tersebut masih jadi platform eSport utama . Acer yakin dengan hal itu , dan memutuskan buat mengadakan event terbesarnya di kawasan Asia Pasifik . Asia Pacific Predator League 2018 mempertandingkan Dota 2 sebagai permainan MOBA terpopuler di PC , dan di sana , peserta akan memperebutkan potongan terbanyak dari total prize pool sebesar US\$ 150 – atau sekitar Rp 2 miliar . Di tanggal 20 Oktober kemarin , Acer juga melakukan pembukaan Asia Pacific Predator League 2018 di Singapura , Malaysia , India , Thailand , Filipina dan Hong Kong . Khalayak eSport diperkirakan akan mencapai 385 juta jiwa di tahun 2017 , terdiri dari 192 juta peserta dan 194 juta penonton – mayoritas menikmatinya secara online . Untuk populasi penggemar eSport sendiri , Asia Pasifik diprediksi menyumbang porsi paling banyak , yaitu 51 persen . Bagian ini yang menjadi kelemahan Dell Inspiron 15 7000 , karena dianggap terlalu mini untuk sebuah notebook gaming . Jika sesuai rencana , Acer Nitro 5 Spin akan dipasarkan mulai bulan Oktober di Amerika Utara dan Eropa dengan banderol di kisaran \$ 999-\$1,199 . Sumber berita Ubergizmo , Techradar . Namun perhatian sejumlah perusahaan , terutama produsen monitor , telah mulai melirik tingkatan yang lebih tinggi lagi . Menyusul pengungkapan monitor gaming 49 - inci dari Samsung dan panel curved 35 - inci 200 Hz buatan Acer , kali ini giliran TPV Technology memamerkan Philips 328P8 K di IFA 2017 . Jika sesuai rencana , Acer Nitro 5 Spin akan dipasarkan mulai bulan Oktober di Amerika Utara dan Eropa dengan banderol di kisaran \$ 999-\$1,199 . Kamu bisa menjadi member komunitas startup dan inovasi DailySocial.id ,.

Gambar 4. 7 Ringkasa Sytem

Game - game MOBA di perangkat bergerak belakangan menjadi sorotan berkat dilangsungkannya berbagai turnamen yang menjanjikan jumlah hadiah melimpah . Acer yakin dengan hal itu , dan memutuskan buat mengadakan event terbesarnya di kawasan Asia Pasifik . Asia Pacific Predator League 2018 mempertandingkan Dota 2 sebagai permainan MOBA terpopuler di PC , dan di sana , peserta akan memperebutkan potongan terbanyak dari total prize pool sebesar US\$ 150 – atau sekitar Rp 2 miliar . “ [ Acara ] ini ialah tonggak sejarah penting bagi Acer di ranah gaming sekaligus bentuk komitmen kami untuk terus berinovasi dan menghantarkan keunggulan teknologi kepada para gamer melalui lini perangkat Predator . ” Dari bincang - bicang singkat bersama perwakilan Acer , pendaftaran Asia Pacific Predator League 2018 telah dibuka pada tanggal 9 Oktober 2017 lalu , dan penyisihan dimulai pada hari Sabtu dan Minggu kemarin di Jakarta , Blitar serta Balikpapan . 12 kota yang disambangi oleh Predator League 2018 meliputi Jakarta , Blitar , Balikpapan , Malang , Medan , Samarinda , Surabaya , Tangerang , Palembang , Semarang , Karawang , dan Makassar . Di tanggal 20 Oktober kemarin , Acer juga melakukan pembukaan Asia Pacific Predator League 2018 di Singapura , Malaysia , India , Thailand , Filipina dan Hong Kong . Presiden IeSPA Eddy Lim turut mengapresiasi perhatian Acer pada industri ini , dan menyampaikan harapannya agar Predator League 2018 berjalan lancar hingga momen penutupannya nanti . Sebelum melangsungkan Asia Pacific Predator League 2018 , Acer sudah terlibat dalam berbagai macam event eSport , misalnya berpartisipasi di Intel Extreme Masters ( IEM ) Season 11 , juga menjadi sponsor resmi sekaligus penyedia perangkat display di League of Legends eSports 2017 . Herbert Ang hanya bilang ada uang tunai senilai US\$ 15 ribu ditambah notebook gaming Predator menanti buat masing-masing anggota tim yang jadi juara pertama Predator League 2018 . Acer kembali bermain di ranah gaming kasual dengan merilis satu lagi laptop convertible gaming , Nitro 5 Spin . Sebagai laptop gaming , Acer Nitro 5 Spin memberi jawaban dengan menyematkan prosesor generasi ke delapan Intel Core i7 bersama dengan grafis Nvidia GeForce GTX 1050GPU dan VRAM GDDR5 dedicated . Penyimpanan internal Nitro 5 Spin juga istimewa dengan basis SSD PCI Express dengan pilihan kapasitas hingga 512 GB . Berbekal paket ini , di atas kertas spesifikasi Nitro 5 Spin mampu mengalahkan HP Spectre x360 dan Samsung Note 9 Spin . Apalagi , Acer juga mengadopsi teknologi antena nirkabel MU - MIMO yang menjanjikan kecepatan unduhan tiga kali lebih baik . Jika sesuai rencana , Acer Nitro 5 Spin akan dipasarkan mulai bulan Oktober di Amerika Utara dan Eropa dengan banderol di kisaran \$ 999-\$1,199 . Kamu bisa menjadi member komunitas startup dan inovasi DailySocial.id ,.

Gambar 4. 8 Ringkasan manual

Tabel 4. 11 Statistika Ringkasan

<b>Statistika</b>	<b>Jumlah Kalimat</b>	<b>Jumlah kata</b>	<b>Jumlah kata unik</b>
Ringkasan Sistem	13	308	203
Ringkasan Manusia	14	348	216

Tabel 4. 12 Statistik Jumlah Kata overlapping Dalam Ringkasan

<b>Kata</b>	<b>Jumlah</b>
Kata unik sistem	203
Kata unik manusia	216
Kata yang sama pada sistem dan manusia ( <i>overlapping</i> )	151

Ringkasan *system* dan manual masing-masing akan dihitung jumlah kalimat, kata unik dan kata yang overlapping, kemudian akan digunakan untuk menghitung nilai Rouge-1. Tabel pada table 4.10 dan 4.11 menunjukkan statistika jumlah kalimat dan kata pada ringkasan. Nilai recall didapatkan dari perbandingan kata yang sama pada ringkasan manusia dan sistem dengan jumlah kata pada ringkasan manusia Perhitungan nilai recall dilakukan menggunakan formula 3.1 kemudian berdasarkan statistika pada tabel 4.12 didapatkan nilai recall sebesar 0.69907:

$$Recall = \frac{151}{216} = 0.69907$$

Selanjutnya dilakukan juga perhitungan *precision* menggunakan rumus 3.2 perhitungan *precision* dilakukan untuk mengetahui persentase jumlah kata benar yang berhasil diprediksi oleh ringkasan system. Kemudian juga dilakukan perhitungan *F-measure* menggunakan rumus 3.3 untuk mengukur performa efisiensi algoritma yang digunakan dalam peringkasan teks otomatis multi-dokumen. Perhitungan precision dan F1 ditunjukkan sebagai berikut..

$$precision = \frac{151}{206} = 0.74257$$

$$f - measure = \frac{(2 * 0.74257 * 0.69907)}{(0.74257 + 0.69907)} = 0.7177033$$

Tabel 4.13 menampilkan Hasil Rouge-1 pada uji coba ke 1 menggunakan model Word2vec CBOW dengan vector 100 dan window 5 pada 50 data uji yang dibagi menjadi 5-fold.

Tabel 4. 13 Hasil ROUGE-1 fold-1 pada model CBOW vektor 100 window 5

Dokumen	Recall	Precision	F-measure
1	0.437853	0.74519	0.5516
2	0.375	0.6355	0.47169
3	0.30546	0.83333	0.44705
4	0.4501	0.61946	0.52141
5	0.6944	0.74257	0.71770
6	0.40463	0.64876	0.4984
7	0.41516	0.95833	0.5793
8	0.55700	0.76	0.6428
9	0.29508	0.679245	0.41142
10	0.24099	0.764285	0.36643
<b>Rata-rata</b>	<b>0.41758</b>	<b>0.738678</b>	<b>0.52079</b>

Tabel 4. 14 Hasil rata-rata rouge-1 model CBOW

Fold	Vektor	Dimensi	Recall	Precision	F-measure	Waktu(s)
1	100	5	0.41758	0.73867	0.52079	1.6592
		10	0.45418	0.66326	0.516951	1.3839
	200	5	0.505097	0.682321	0.56471	1.3938
		10	0.473203	0.650098	0.52620	1.4239
	<b>Rata-rata</b>		<b>0.46251505</b>	<b>0.6835894</b>	<b>0.53216585</b>	<b>1.4652</b>
2	100	5	0.4566526	0.683977	0.5305259	2.0837
		10	0.415421	0.67097	0.503744	2.0025
	200	5	0.53698	0.692726	0.58869	2.0655
		10	0.484844	0.702833	0.564160	2.0573
	<b>Rata-rata</b>		<b>0.4734744</b>	<b>0.6876265</b>	<b>0.54653</b>	<b>2.0522</b>
3	100	5	0.5640282	0.7253065	0.612556	1.6390
		10	0.50063	0.729922	0.575230	1.3978
	200	5	0.539699	0.708046	0.599508	1.4194
		10	0.532063	0.69872644	0.5829980	1.4464
	<b>Rata-rata</b>		<b>0.5341.</b>	<b>0.7155.</b>	<b>0.592573</b>	<b>1.4757</b>
4	100	5	0.50063193	0.72992219	0.5752308	1.4942
		10	0.450960	0.71425	0.52481960	1.8861
	200	5	0.483686	0.770490	0.5669907	1.4627
		10	0.52695	0.633159	0.5350408	2.2325
	<b>Rata-rata</b>		<b>0.490556983</b>	<b>0.711955298</b>	<b>0.55052049</b>	<b>1.7689</b>
5	100	5	0.505891	0.6935040	0.537457	2.0440
		10	0.46482	0.74350	0.537457	2.061
	200	5	0.476258	0.787109	0.54223402	1.4299
		10	0.4702627	0.672898	0.50135328	1.4050
	<b>Rata-rata</b>		<b>0.479307925</b>	<b>0.72425275</b>	<b>0.529625325</b>	<b>1.7349</b>

Pada Tabel 4.14, menunjukan nilai tertinggi untuk rata-rata recall adalah pada fold-3 dengan penggunaan vektor sebesar 100 dan *window* sebesar 5, mencapai nilai 0.56402826007 dan waktu peringkasan 1.6390 detik. Sementara itu, precision tertinggi tercatat pada fold-4 dengan penggunaan vektor sebesar 200 dan *window* sebesar 5, dengan nilai mencapai 0.770490 dan waktu 1.4627 detik. Sedangkan untuk F-measure, nilai tertinggi terlihat pada fold-3 dengan vektor sebesar 100 dan *window* sebesar 5, dengan nilai mencapai 0.61255659 waktu peringkasan 1.6390 detik. Nilai terendah untuk rata-rata recall adalah sebesar 0.415421 pada vektor 200 *window* 10, dan fold ke-4. *Precision* terendah adalah 0.633159 pada vektor 200 dan *window* 10, dan fold ke-4. Terakhir, untuk F-measure, nilai terendah adalah 0.50135328 dan terjadi pada penggunaan vektor sebesar 200, *window* sebesar 10, dan fold ke-5.

### 4.3.2 Hasil Uji Coba Menggunakan Model *Skip-gram*

Hasil uji coba kedua dilakukan menggunakan ekstraksi fitur model *Skip-gram*. Pada tiap model digunakan vektor 100 dan 200 dimensi dengan masing-masing nilai *window* sebanyak 5 dan 10. Tabel 4.15 adalah hasil rata-rata recall, precision, dan f-measure tiap fold-nya.

Tabel 4. 15 Hasil rata-rata rouge-1 menggunakan model *Skip-gram*

Fold	vector	window	Recall	Precision	F-measure	Waktu(s)
1	100	5	0.43548	0.68087	0.52463	3.7847
	100	10	0.33272	0.69979	0.442109	2.2907
	200	5	0.46897	0.66590	0.5477	2.2291
	200	10	0.45145	0.66432	0.52930	2.3899
	<b>Rata-rata</b>		<b>0.422155</b>	<b>0.67772</b>	<b>0.510935</b>	<b>2.6736</b>
2	100	5	0.45665	0.68397	0.53052	3.3836
	100	10	0.397416	0.69393	0.472542	4.1672
	200	5	0.411359	0.7268	0.503511	6.6457
	200	10	0.48141	0.69956	0.558532	3.9630
	<b>Rata-rata</b>		<b>0.43670875</b>	<b>0.701065</b>	<b>0.5162762</b>	<b>4.5399</b>

3	100	5	0.55121	0.720588	0.612686	3.9066
	100	10	0.43317	0.73742	0.52406	4.6234
	200	5	0.39559	0.64418	0.64418	2.3087
	200	10	0.460465	0.73865	0.53658	2.4109
	<b>Rata-rata</b>		<b>0.460109</b>	<b>0.71021</b>	<b>0.579377</b>	<b>3.3124</b>
4	100	5	0.388703	0.60030	0.455101	2.2987
	100	10	0.35568	0.74467	0.428244	2.3188
	200	5	0.3327	0.752544	0.44210	6.8910
	200	10	0.34254	0.71067	0.420594	3.2383
	<b>Rata-rata</b>		<b>0.354906</b>	<b>0.59709</b>	<b>0.43651</b>	<b>3.6867</b>
5	100	5	0.45448	0.67517	0.520395	2.2835
	100	10	0.38108	0.763212	0.45063	2.3889
	200	5	0.417391	0.786736	0.588957	1.5197
	200	10	0.35867	0.7794	0.46758	3.5281
	<b>Rata-rata</b>		<b>0.398077</b>	<b>0.75113</b>	<b>0.479535</b>	<b>2.4301</b>

Pada Tabel 4.15, menunjukan nilai tertinggi untuk rata-rata recall adalah pada fold-3 dengan vektor 100 dan *window* 5, mencapai nilai 0.55121 dan waktu 3.38367 detik. Sementara itu, precision tertinggi terdapat pada fold-4 dengan vektor 200 dan *window* 5, dengan nilai mencapai 0.786736 dan waktu 6.891076 detik. Sedangkan untuk F-measure, nilai tertinggi terdapat pada fold-3 dengan vektor 100 dan *window* 5, dengan nilai mencapai 0.612686 dan waktu 3.38367 detik. Nilai terendah untuk rata-rata recall adalah sebesar 0.3327 pada vektor 200, *window* 10, dan fold ke-4. Precision terendah adalah 0.64418, terdapat pada vector 200, *window* 10, dan fold ke-4. Terakhir, untuk F-measure, nilai terendah adalah 0.420594 dan terdapat pada 200, *window* sebesar 10, dan fold ke-5 dengan waktu 3.528156 detik.

#### 4.4 Pembahasan

Pada penelitian ini didapatkan hasil evaluasi menggunakan Rouge-1 berupa *recall*, *precision*, dan *f-measure* yang ditunjukkan pada tabel 4.16. Tabel 4.16 menunjukan hasil nilai rata-rata rouge tiap fold pada model CBOW dan *Skip-gram*

Tabel 4. 16 Hasil Rata-rata Rouge-1 tiap fold

Uji coba	Fold	Recall	Precision	F-measure	Waktu (s)
CBOW	1	0.46251505	0.6835894	0.53216585	1.4652
	2	0.4734744	0.6876265	0.54653	2.0522
	3	0.5341	0.7155	0.592573	1.4757
	4	0.490556983	0.711955298	0.55052049	1.7689
	5	0.479307925	0.72425275	0.529625325	1.7349
	<b>Rata-rata</b>	<b>0.487990872</b>	<b>0.70458479</b>	<b>0.550282933</b>	<b>1.6994</b>
Skip-gram	1	0.422155	0.67772	0.510935	2.6736
	2	0.436709	0.701065	0.516276	4.5399
	3	0.460109	0.71021	0.579377	3.3124
	4	0.354906	0.59709	0.43651	3.6867
	5	0.398077	0.75113	0.479535	2.4301
	<b>Rata-rata</b>	<b>0.414391</b>	<b>0.687443</b>	<b>0.504527</b>	<b>3.3285</b>

Untuk uji coba pertama digunakan model CBOW dengan 5-fold didapatkan nilai rata-rata *F-measure* tertinggi sebesar 0.592573 yaitu pada fold-3 dengan waktu 1.4757 detik, nilai rata-rata *recall* tertinggi sebesar 0.5341 pada fold ke-3 dengan waktu 1.4757 detik dan nilai rata-rata *precision* tertinggi sebesar 0.72425275 pada fold ke-5 dengan waktu 1.7349 detik.

Untuk uji coba kedua digunakan model *Skip-gram* dengan 5-fold didapatkan nilai rata-rata *F-measure* tertinggi sebesar 0.579377 yaitu pada fold ke-3 dengan waktu 3.3124 detik, nilai rata-rata *recall* tertinggi sebesar 0.460109 pada fold ke-3 dengan waktu 3.3124 dan nilai rata-rata presisi tertinggi sebesar 0.75113 pada fold ke-5 dengan waktu 2.4301.

Dari penjelasan di atas, dapat disimpulkan bahwa ekstraksi fitur memiliki dampak signifikan terhadap kinerja peringkasan dokumen. Model Continuous Bag of Words (CBOW) menunjukkan kinerja yang lebih baik dan mempunyai waktu relatif cepat dalam meringkas dibandingkan dengan model *Skip-gram*, dengan waktu rata-rata 1.6994 detik dan rata-rata nilai *f-measure* sebesar 0.550282933 sedangkan dari model *Skip-gram* memiliki rata-rata waktu dalam meringkas

3.3285 detik dan nilai rata-rata *f-measure* sebesar 0.504527. Kemudian, dari hasil uji coba diatas menunjukan nilai rouge-1 terbaik dari masing-masing model terdapat pada fold-3 dengan nilai rata-rata f-measure dari model CBOW dan *Skip-gram* adalah 0.592573 dan 0.579377.

Hasil eksperimen yang sudah diuraikan menunjukkan bahwa penerapan metode RNN dapat menghasilkan sistem peringkasan multi dokumen secara otomatis. Sistem tersebut dapat menghemat waktu dan tenaga yang diperlukan untuk membaca serta memahami banyak dokumen secara manual. Selain itu, sistem ini juga dapat membantu pengguna dalam memperoleh informasi yang lebih ringkas dan menyeluruh dari berbagai sumber yang berbeda.

#### **4.5 Integrasi Penelitian**

Dari hasil penelitian yang dilakukan, sistem peringkasan teks berita berbahasa Indonesia multi-dokumen menunjukkan kontribusi yang signifikan dalam bidang Natural Language Processing (NLP). Penelitian ini menggarisbawahi pentingnya pengembangan algoritma peringkasan teks untuk memproses informasi dari berbagai sumber. Sistem peringkasan teks yang dikembangkan memiliki tujuan utama untuk mendapatkan inti informasi dan esensi dari berita berbahasa Indonesia. Dengan menyajikan ringkasan berita, diharapkan dapat memudahkan pembaca atau pengguna dalam memahami konten berita tanpa harus membaca seluruh dokumen. Hasil pengujian sistem menunjukkan bahwa pengguna dapat dengan mudah mengakses ringkasan berita dari banyak dokumen sekaligus (multi-dokumen) secara otomatis. Proses ini

memberikan keuntungan signifikan, mengingat waktu yang dibutuhkan relatif lebih singkat dibandingkan dengan meringkas secara manual.

Sistem peringkasan teks otomatis ini selaras dengan ajaran islam tentang memanfaatkan waktu dengan sebaik-baiknya. Pada Qur'an surat al-ashr ayat 1-3.

وَالْعَصْرِ ﴿١﴾ إِنَّ الْإِنْسَانَ لَفِي خُسْرٍ ﴿٢﴾ إِلَّا الَّذِينَ آمَنُوا وَعَمِلُوا الصَّالِحَاتِ وَتَوَّصَوْا بِالْحَقِّ وَتَوَّصَوْا  
بِالصَّبْرِ ﴿٣﴾

*“Demi masa. Sesungguhnya manusia itu benar-benar dalam kerugian, kecuali orang-orang yang beriman dan beramal saleh dan saling menasihati dalam kebenaran dan saling menasihati dalam kesabaran.”* (QS. Al-ashr 104: ayat 1-3).

Menurut tafsir Ibnu Katsir menjelaskan bahwa ayat ini mengingatkan manusia tentang pentingnya memanfaatkan waktu. Waktu adalah anugerah dari Allah SWT yang harus dimanfaatkan sebaik-baiknya. Manusia berada dalam kerugian jika tidak memanfaatkan waktu dengan sebaik-baiknya. Selain ayat diatas terdapat ayat tentang shalat qashar, yaitu meringkas shalat untuk orang-orang yang bepergian Allah SWT menyebutkan dalam Surat An-nisa' ayat 101.

وَإِذَا ضَرَبْتُمْ فِي الْأَرْضِ فَلَيْسَ عَلَيْكُمْ جُنَاحٌ أَنْ تَقْصُرُوا مِنَ الصَّلَاةِ إِنْ خِفْتُمْ أَنْ يُفْتِنَكُمْ الَّذِينَ كَفَرُوا إِنَّ الْكَافِرِينَ كَانُوا  
لَكُمْ عَدُوًّا مُّبِينًا ﴿١٠١﴾

*”Apabila kamu bepergian di bumi, maka tidak dosa bagimu untuk mengqashar salat jika kamu takut diserang orang-orang yang kafur. Sesungguhnya orang-orang kafir itu adalah musuh yang nyata bagimu.”* (QS. An-Nisa' 4: ayat 101)

Tafsir jalalain memberikan penjelasan mengenai ayat ini yaitu (Dan jika kamu melakukan perjalanan) atau bepergian (di muka bumi, maka tidak ada salahnya kamu) (apabila mengqashar shalat) dengan membuat yang empat rakaat menjadi dua (jika kamu khawatir akan diperangi) atau mendapatkan luka dari

orang-orang kafir (menyatakan peristiwa yang terjadi kala itu maka mafhumnya tidak berlaku). Menurut keterangan dari sunnah, yang dimaksud dengan suatu perjalanan panjang ialah empat pos atau dua marahlah. Dan dari firman-Nya, “Maka tidak salahnya kamu.” Ditarik kesimpulan bahwa mengqashar shalat itu merupakan keringanan dan bukan kewajiban. Dan ini pendapat Imam Syafi’i. (Sesungguhnya orang-orang kafir itu bagi kamu adalah musuh yang nyata) maksud jelas dan terang permusuhannya terhadap kamu.”. (Al-Mahalli *et al.*, 1469)

Ibnu Katsir menjelaskan bahwa ayat al-ashr mengingatkan manusia tentang pentingnya memanfaatkan waktu. Pada ayat kedua tafsir jalalain menjelaskan tentang meringkas shalat karena adanya kepentingan. Dengan peringkasan teks berita otomatis karena dokumen teks yang tersebar di internet sangat banyak. Dengan peringkasan teks tersebut dapat memudahkan seseorang dalam mendapatkan inti dan juga dapat lebih menghemat waktu saat membaca berita

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Dalam eksperimen ini, penelitian menggunakan model CBOW dan *Skip-gram* untuk ekstraksi fitur pada peringkasan dokumen. Hasil menunjukkan bahwa CBOW memiliki akurasi yang lebih tinggi dan waktu peringkasan yang lebih singkat dibandingkan dengan *Skip-gram*, dengan nilai rata-rata F-measure sebesar 0.550282933 dan waktu peringkasan 1.6994 detik, sedangkan untuk model *Skip-gram* mendapat nilai rata-rata f-measure sebesar 0.504527 dan rata-rata waktu 3.3825 detik. Nilai terbaik untuk metrik rouge-1 dari kedua model terdapat pada fold-3, dengan nilai rata-rata F-measure dari CBOW dan *Skip-gram* sebesar 0.592573 dan 0.579377. Dari hasil tersebut menunjukkan bahwa penerapan metode RNN dalam sistem peringkasan multi dokumen dapat efisien menghemat waktu dan tenaga, serta membantu pengguna memperoleh informasi yang lebih ringkas dan menyeluruh dari berbagai sumber.

#### **5.2 Saran**

Dalam pengembangan sistem peringkasan teks otomatis berbahasa Indonesia, terdapat saran yang dapat diterapkan untuk mencapai hasil yang lebih maksimal, yaitu penambahan jumlah dataset. Dataset yang lebih banyak akan membuat model yang lebih optimal karena model akan belajar dengan lebih banyak berita dengan berbagai topik yang berbeda

## DAFTAR PUSTAKA

- Adelia, R., Suyanto, S., & Wisesty, U. N. (2019). *Indonesian Abstractive Text Summarization Using Bidirectional Gated Recurrent Unit*. *Procedia Computer Science*, 157, 581–588. <https://doi.org/10.1016/j.procs.2019.09.017>
- Adytoma, W. W., Huda, A. F., Maylawati, D. S. A., Arianti, N. D., Darmalaksana, W., Rahman, A., & Ramdhani, M. A. (2020, October 15). Automatic Text Summarization for Hadith with Indonesian Text using Bellman-Ford Algorithm. *6th International Conference on Computing, Engineering, and Design, ICCED 2020*. <https://doi.org/10.1109/ICCED51276.2020.9415864>
- Akaresti, A. K., Fauzi, M. A., & Abdurrachman Bachtiar, F. (2019). *Peringkasan Multi-Dokumen Berbasis Clustering pada Sistem Temu Kembali Berita Online Menggunakan Metode K-Means* (Vol. 3, Issue 1). <http://j-ptiik.ub.ac.id>
- Al-Mahalli, I. J., & As-Sayuti, I. J. (1469). *Tafsir Jalalain*. Sinar Baru Algensindo.
- Anjar Prabowo, D., Fhadli, M., Ainun Najib, M., Agus Fauzi, H., Cholissodin, I., & Studi Informatika, P. (2016). TF-IDF-Enhanced Genetic Algorithm untuk Extractive Automatic Text Summarization. In *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)* (Vol. 3, Issue 3).
- Gupta, S., & Gupta, S. K. (2019). Abstractive summarization: An overview of the state of the art. *Expert Systems with Applications*, 121, 49–65. <https://doi.org/10.1016/j.eswa.2018.12.011>
- Hanunggul, P. M. (2019). *The Impact of Local Attention in LSTM for Abstractive Text Summarization*.
- Hochreiter, S., & Schmidhuber, J. (1997). *Long Short-Term Memory*.
- Holton, A. E., & Chyi, H. I. (2012). News and the overloaded consumer: Factors influencing information overload among news consumers. *Cyberpsychology, Behavior, and Social Networking*, 15(11), 619–624. <https://doi.org/10.1089/cyber.2011.0610>
- Ivanedra, K., & Mustikasari, M. (2019). *Implementasi Metode Recurrent Neural Network Pada Text Summarization dengan Teknik Abstraktif*. 6(4), 377–382. <https://doi.org/10.25126/jtiik.201961067>

- Jain, A., Kulkarni, G., & Shah, V. (2018). *Natural Language Processing*. [www.ijcseonline.org](http://www.ijcseonline.org)
- Jain, G., Sharma, M., & Agarwal, B. (2019). Optimizing semantic LSTM for spam detection. *International Journal of Information Technology (Singapore)*, 11(2), 239–250. <https://doi.org/10.1007/s41870-018-0157-5>
- Kalangi, P. K., Praveen Kumar, K., Padma, T. N. S., Pruthvi, B., & Goud, R. (2020). Automatic Text Summarization Using LSTM Based On Sentence Semantics. In *International Journal of All Research Education and Scientific Methods (IJARESM)* (Vol. 8, Issue 6). [www.ijaresm.com](http://www.ijaresm.com)
- Khan, A., & Salim, N. (2014). A Review on Abstractive Summarization Methods. *Journal of Theoretical and Applied Information Technology*, 10(1). [www.jatit.org](http://www.jatit.org)
- Kurniawan, K., & Louvan, S. (2019). IndoSum: A New Benchmark Dataset for Indonesian Text Summarization. *Proceedings of the 2018 International Conference on Asian Language Processing, IALP 2018*, 215–220. <https://doi.org/10.1109/IALP.2018.8629109>
- Liu, P., Qiu, X., & Huang, X. (2016). *Recurrent Neural Network for Text Classification with Multi-Task Learning*. <http://arxiv.org/abs/1605.05101>
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). *Distributed Representations of Words and Phrases and their Compositionality*.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013b). *Efficient Estimation of Word Representations in Vector Space*. <http://arxiv.org/abs/1301.3781>
- Mustaqhfiri, M., Abidin, Z., & Kusumawati, D. R. (2012). *Peringkasan Teks Otomatis Berita Berbahasa Indonesia Menggunakan Metode Maximum Marginal Relevance*. 134–147. <https://doi.org/10.18860/mat.v0i0.1578>
- Najibullah, A. (2015). Indonesian Text Summarization Based on Naïve Bayes Method. *Proceeding of the International Seminar and Conference 2015: The Golden Triangle (Indonesia-India-Tiongkok)*. [www.isc.unwahas.ac.id](http://www.isc.unwahas.ac.id)
- Naufal Ammar, A. (2020). Peringkasan Teks Ekstraktif Menggunakan Binary Firefly Algorithm. *Ind. Journal on Computing*, 5(2), 32–42. <https://doi.org/10.21108/indojc.2020.5.2.440>
- Rahman, M. M., & Siddiqui, F. H. (2019). An Optimized Abstractive Text Summarization Model using Peephole Convolutional LSTM. *Symmetry*, 11(10). <https://doi.org/10.3390/sym11101290>
- Ramanujam, N., & Kaliappan, M. (2016). An automatic multidocument text summarization approach based on naïve Bayesian classifier using timestamp

strategy. *Scientific World Journal*, 2016.  
<https://doi.org/10.1155/2016/1784827>

Rossana Ramirez-Avila, M., Paola, J., Universidad, B., & Grande, C. (2021). The Effect of Summarizing Narrative Texts to Improve Reading Comprehension. *Available Online Journal of Foreign Language Teaching and Learning*, 6(2).

Saputra, I. P. G. H. (2017). Peringkasan Teks Otomatis Untuk Dokumen Bahasa Bali Berbasis Metode Ekstraktif. *Jurnal Ilmiah ILMU KOMPUTER Universitas Udayana*, X(1), 33–38.

Shah, C., & Jivani, A. (2019). An Automatic Text Summarization on Naive Bayes Classifier Using Latent Semantic Analysis. In *Data, Engineering and Applications* (pp. 171–180). Springer Singapore. [https://doi.org/10.1007/978-981-13-6347-4\\_16](https://doi.org/10.1007/978-981-13-6347-4_16)

Shini, R. S., & Kumar, V. D. A. (2021). Recurrent Neural Network based Text Summarization Techniques by Word Sequence Generation. *Proceedings of the 6th International Conference on Inventive Computation Technologies, ICICT 2021*, 1224–1229. <https://doi.org/10.1109/ICICT50816.2021.9358764>

## **LAMPIRAN**

**Tabel Hasil Rouge-1 model CBOW vector 100 window 5**

Fold	Dokumen	Recall	Precision	F-measure
1	1	0.437853	0.74519	0.5516
	2	0.375	0.6355	0.47169
	3	0.30546	0.833333	0.447058
	4	0.4501	0.619469	0.52141
	5	0.6944	0.742574	0.717703
	6	0.40463	0.64876	0.4984
	7	0.415162	0.95833333	0.57934
	8	0.55700	0.76	0.6428
	9	0.295081	0.679245	0.4114285
	10	0.24099	0.76428571	0.3664383
	<b>Rata-rata</b>	<b>0.4175802407099</b>	<b>0.7386786806</b>	<b>0.52079587</b>
2	11	0.647059	0.733333	0.6875
	12	0.589372	0.824324	0.687324
	13	0.415	0.62406	0.498498
	14	0.497925	0.674157	0.572792
	15	0.310662	0.673307	0.425157
	16	0.238806	0.447552	0.311436
	17	0.588832	0.594872	0.591837
	18	0.537234	0.95283	0.687075
	19	0.637555	0.528986	0.578218
	20	0.621212	0.745455	0.677686
	<b>Rata-rata</b>	<b>0.508366</b>	<b>0.679888</b>	<b>0.571752</b>
3	21	0.708696	0.806931	0.75463
	22	0.648084	0.609836	0.628378
	23	0.716102	0.692623	0.704167
	24	0.657971	0.585052	0.619372
	25	0.395954	0.622727	0.484099
	26	0.362283	0.9125	0.51865
	27	0.75	0.841629	0.793177
	28	0.621495	0.786982	0.694517
	29	0.684073	0.755043	0.717808
	30	0.623656	0.711656	0.664756
	<b>Rata-rata</b>	<b>0.616831</b>	<b>0.732498</b>	<b>0.657955</b>
4	31	0.426752	0.54918	0.480287
	32	0.617021	0.783784	0.690476
	33	0.217391	1	0.357143
	34	0.333333	0.660714	0.443114
	35	0.434343	0.632353	0.51497
	36	0.608939	0.703226	0.652695
	37	0.906977	0.735849	0.8125
	38	1	0.601307	0.75102
	39	0.575472	0.743902	0.648936
	40	0.487047	0.618421	0.544928
	<b>Rata-rata</b>	<b>0.560727</b>	<b>0.702874</b>	<b>0.589607</b>
5	41	0.20915	1	0.345946
	42	0.164251	0.790698	0.272
	43	0.13	0.722222	0.220339
	44	0.352697	0.648855	0.456989
	45	0.224265	0.77707	0.348074
	46	0.253731	0.731183	0.376731
	47	0.121827	0.857143	0.213333
	48	0.531915	0.534759	0.533333
	49	0.248908	0.791667	0.378738
	50	0.631313	0.726744	0.675676
	<b>Rata-rata</b>	<b>0.286806</b>	<b>0.758034</b>	<b>0.382116</b>

**Tabel Hasil Rouge-1 model CBOW vector 100 window 10**

Fold	Dokumen	Recall	Precision	F-measure
1	1	0.310734	0.753425	0.44
	2	0.365	0.544776	0.437126
	3	0.157556	0.680556	0.255875
	4	0.382637	0.395349	0.388889
	5	0.662037	0.756614	0.706173
	6	0.420103	0.586331	0.489489
	7	0.155235	0.716667	0.255193
	8	0.680782	0.715753	0.69783
	9	0.36612	0.665012	0.472247
	10	0.47973	0.705298	0.571046
	<b>Rata-rata</b>	<b>0.397993</b>	<b>0.651978</b>	<b>0.471387</b>
2	11	0,477124183	0,7019230769	0,5680933804
	12	0,5845410628	0,8066666667	0,6778711436
	13	0,5	0,5681818182	0,5319148886
	14	0,4190871369	0,480952381	0,4478935649
	15	0,4099264706	0,6501457726	0,5028184845
	16	0,1268656716	0,4857142857	0,2011834287
	17	0,3908629442	0,5789473684	0,4666666619
	18	0,4946808511	0,6690647482	0,5688073346
	19	0,5982532751	0,7043701799	0,6469893693
	20	0,7222222222	0,7009803922	0,7114427811
	<b>Rata-rata</b>	<b>0.47235638</b>	<b>0.634694668</b>	<b>0.53236810</b>
3	21	0,6260869565	0,6457399103	0,6357615844
	22	0,7735191638	0,5677749361	0,6548672518
	23	0,5148305085	0,81	0,629533674
	24	0,3913043478	0,5510204082	0,4576271138
	25	0,3034682081	0,7394366197	0,4303278647
	26	0,3573200993	0,9290322581	0,5161290282
	27	0,5282258065	0,7844311377	0,6313252964
	28	0,8504672897	0,7109375	0,7744680801
	29	0,6579634465	0,8051118211	0,7241379261
	30	0,6370967742	0,7095808383	0,671388097
	<b>Rata-rata</b>	<b>0.56402826007</b>	<b>0.7253065429</b>	<b>0.61255659</b>
4	31	0,2484076433	0,78	0,3768115905
	32	0,6489361702	0,6703296703	0,6594594545
	33	0,3719806763	0,7475728155	0,4967741891
	34	0,6396396396	0,9726027397	0,7717391256
	35	0,4444444444	0,4731182796	0,4583333283
	36	0,4134078212	0,5	0,4525993834
	37	0,6744186047	0,7837837838	0,724999995
	38	1	1	1
	39	0,641509434	0,5074626866	0,5666666617
	40	0,5544041451	0,5544041451	0,5544041401
	<b>Rata-rata</b>	<b>0.56371485788</b>	<b>0.6989274</b>	<b>0.60617878</b>
5	41	0,1909090909	1	0,3206106843
	42	0,4672897196	0,5555555556	0,5076142082
	43	0,4173913043	1	0,5889570511
	44	0,7724550898	0,5443037975	0,6386138565
	45	0,7971014493	0,5472636816	0,6489675468
	46	0,4561403509	0,8863636364	0,6023165978
	47	0,3868613139	0,4491525424	0,4156862695
	48	0,5123966942	0,6458333333	0,5714285665
	49	0,510373444	0,6473684211	0,5707656563
	50	0,648	0,5547945205	0,5977859729
	<b>Rata-rata</b>	<b>0.515891</b>	<b>0.6830635</b>	<b>0.5462746</b>

**Tabel Hasil Rouge-1 model CBOV vector 200 window 5**

Fold	Dokumen	Recall	Precision	F-measure
1	1	0.251412	0.773913	0.379531
	2	0.28	0.589474	0.379661
	3	0.495177	0.608696	0.546099
	4	0.636656	0.414226	0.501901
	5	0.509259	0.728477	0.599455
	6	0.546392	0.738676	0.628148
	7	0.306859	0.615942	0.409639
	8	0.527687	0.746544	0.618321
	9	0.569672	0.634703	0.600432
	10	0.47973	0.744755	0.583562
		<b>Rata-rata</b>	<b>0.460284</b>	<b>0.659541</b>
2	11	0.75817	0.90625	0.825623
	12	0.536232	0.716129	0.61326
	13	0.46	0.476684	0.468193
	14	0.506224	0.525862	0.515856
	15	0.244485	0.796407	0.374121
	16	0.253731	0.459459	0.326923
	17	0.553299	0.784173	0.64881
	18	0.37234	0.804598	0.509091
	19	0.604803	0.738667	0.665066
	20	0.388889	0.740385	0.509934
		<b>Rata-rata</b>	<b>0.467817</b>	<b>0.694861</b>
3	21	0.56087	0.854305	0.677165
	22	0.599303	0.609929	0.604569
	23	0.792373	0.712381	0.750251
	24	0.57971	0.520833	0.548697
	25	0.384393	0.615741	0.47331
	26	0.354839	0.888199	0.507092
	27	0.512097	0.74269	0.606205
	28	0.747664	0.723982	0.735632
	29	0.5953	0.811388	0.686747
	30	0.58871	0.760417	0.663636
		<b>Rata-rata</b>	<b>0.571526</b>	<b>0.723986</b>
4	31	0.477707	0.614754	0.537634
	32	0.531915	0.757576	0.625
	33	0.371981	0.777778	0.503268
	34	0.513514	0.780822	0.619565
	35	0.484848	0.533333	0.507937
	36	0.597765	0.664596	0.629412
	37	0.732558	0.414474	0.529412
	38	0.717391	0.795181	0.754286
	39	0.716981	0.655172	0.684685
	40	0.393782	0.608	0.477987
		<b>Rata-rata</b>	<b>0.553844</b>	<b>0.660169</b>
5	41	0,1909090909	1	0,3206106843
	42	0,4672897196	0,5555555556	0,5076142082
	43	0,4173913043	1	0,5889570511
	44	0,7724550898	0,5443037975	0,6386138565
	45	0,7971014493	0,5472636816	0,6489675468
	46	0,4561403509	0,8863636364	0,6023165978
	47	0,3868613139	0,4491525424	0,4156862695
	48	0,5123966942	0,6458333333	0,5714285665
	49	0,510373444	0,6473684211	0,5707656563
	50	0,648	0,5547945205	0,5977859729
		<b>Rata-rata</b>	<b>0.515891</b>	<b>0.6830635</b>

**Tabel Hasil Rouge-1 model CBOW vector 200 window 10**

Fold	Dokumen	Recall	Precision	F-measure
1	1	0.265537	0.94	0.414097
	2	0.31	0.861111	0.455882
	3	0.604502	0.681159	0.640545
	4	0.21865	0.4	0.282744
	5	0.634259	0.611607	0.622727
	6	0.420103	0.765258	0.542429
	7	0.635379	0.761905	0.692913
	8	0.566775	0.75	0.64564
	9	0.193989	0.628319	0.296451
	10	0.236486	0.596591	0.33871
	<b>Rata-rata</b>	<b>0.408568</b>	<b>0.699595</b>	<b>0.493214</b>
2	11	0.477124183	0.7019230769	0.5680933804
	12	0.5845410628	0.8066666667	0.6778711436
	13	0,5	0,5681818182	0,5319148886
	14	0,4190871369	0,480952381	0,4478935649
	15	0,4099264706	0,6501457726	0,5028184845
	16	0,1268656716	0,4857142857	0,2011834287
	17	0,3908629442	0,5789473684	0,4666666619
	18	0,4946808511	0,6690647482	0,5688073346
	19	0,5982532751	0,7043701799	0,6469893693
	20	0,7222222222	0,7009803922	0,7114427811
	<b>Rata-rata</b>	<b>0.47235638</b>	<b>0.634694668</b>	<b>0.53236810</b>
3	21	0.465217	0.690323	0.555844
	22	0.61324	0.61324	0.61324
	23	0.716102	0.737991	0.726882
	24	0.634783	0.576316	0.604138
	25	0.5	0.733051	0.594502
	26	0.215881	0.878788	0.346614
	27	0.604839	0.909091	0.726392
	28	0.915888	0.787149	0.846652
	29	0.663185	0.744868	0.701657
	30	0.712366	0.681234	0.696452
	<b>Rata-rata</b>	<b>0.60415</b>	<b>0.735205</b>	<b>0.641237</b>
4	31	0.343949	0.519231	0.413793
	32	0.691489	0.942029	0.797546
	33	0.31401	0.878378	0.462633
	34	0.405405	0.703125	0.514286
	35	0.393939	0.684211	0.5
	36	0.27933	0.37037	0.318471
	37	0.94186	0.496933	0.650602
	38	0.771739	0.5	0.606838
	39	0.839623	0.712	0.770563
	40	0.38342	0.578125	0.461059
	<b>Rata-rata</b>	<b>0.536476</b>	<b>0.63844</b>	<b>0.549579</b>
5	41	0.190909	1	0.320611
	42	0.158879	1	0.274194
	43	0.686957	1	0.814433
	44	0.335329	0.736842	0.460905
	45	0.630435	0.58	0.604167
	46	0.596491	0.842975	0.69863
	47	0.29927	1	0.460674
	48	0.297521	0.72	0.421053
	49	0.518672	0.600962	0.556793
	50	0.6	0.412088	0.488599
	<b>Rata-rata</b>	<b>0.431446</b>	<b>0.789287</b>	<b>0.510006</b>

**Tabel Hasil Rouge-1 model Skip-gram vector 100 window 5**

Fold	Dokumen	Recall	Precision	F-measure
1	1	0.3841808	0.723404	0.501845
	2	0.6	0.710059	0.650406
	3	0.6173633	0.834783	0.709797
	4	0.4501608	0.443038	0.446571
	5	0.3611111	0.586466	0.446991
	6	0.4123711	0.627451	0.497667
	7	0.599278	0.817734	0.691667
	8	0.5765472	0.73444	0.645985
	9	0.4562842	0.606171	0.520655
	10	0.3986486	0.553125	0.463351
	<b>Rata-rata</b>	<b>0.48559452</b>	<b>0.663667</b>	<b>0.557494</b>
2	11	0.2352941	0.818182	0.365482
	12	0.6183575	0.752941	0.679045
	13	0.53	0.630952	0.576087
	14	0.3858921	0.510989	0.439716
	15	0.2242647	0.685393	0.33795
	16	0.2910448	0.423913	0.345133
	17	0.2588832	0.662338	0.372263
	18	0.5053191	0.698529	0.58642
	19	0.5240175	0.670391	0.588235
	20	0.5858586	0.70303	0.639118
	<b>Rata-rata</b>	<b>0.41589317</b>	<b>0.655666</b>	<b>0.492945</b>
3	21	0.1173913	0.658537	0.199262
	22	0.45644599	0.642157	0.533605
	23	0.50423729	0.741433	0.600252
	24	0.54492754	0.717557	0.61944
	25	0.29190751	0.647436	0.40239
	26	0.38957816	0.907514	0.545139
	27	0.44758065	0.973684	0.61326
	28	0.4953271	0.972477	0.656347
	29	0.60052219	0.839416	0.700152
	30	0.40053763	0.764103	0.525573
	<b>Rata-rata</b>	<b>0.42484554</b>	<b>0.786431</b>	<b>0.539542</b>
4	31	0.2866242	0.865385	0.430622
	32	0.18085106	1	0.306306
	33	0.29951691	0.738095	0.426117
	34	0.26126126	1	0.414286
	35	0.05050505	0.166667	0.077519
	36	0.33519553	0.408163	0.368098
	37	0.48837209	1	0.65625
	38	0.52173913	0.432432	0.472906
	39	0.79245283	0.617647	0.694215
	40	0.24352332	0.618421	0.349442
	<b>Rata-rata</b>	<b>0.34600414</b>	<b>0.684681</b>	<b>0.419576</b>
5	41	0.209150327	1	0.3459459
	42	0.260869565	0.8852459	0.4029851
	43	0.12	1	0.2142857
	44	0.161825726	0.7222222	0.2644068
	45	0.099264706	0.6835443	0.1733547
	46	0.03358209	0.9	0.0647482
	47	0.314720812	0.7209302	0.4381625
	48	0.186170213	0.4605263	0.2651515
	49	0.137554585	1	0.2418426
	50	0.373737374	0.8222222	0.5138889
	<b>Rata-rata</b>	<b>0.18968754</b>	<b>0.819469</b>	<b>0.292477</b>

**Tabel Hasil Rouge-1 model Skip-gram vector 100 window 10**

Fold	Dokumen	Recall	Precision	F-measure
1	1	0.4265537	0.857955	0.569811
	2	0.62	0.601942	0.610837
	3	0.4019293	0.661376	0.5
	4	0.4276527	0.390029	0.407975
	5	0.3564815	0.846154	0.501629
	6	0.4690722	0.608696	0.52984
	7	0.4043321	0.817518	0.541063
	8	0.4169381	0.752941	0.536688
	9	0.4562842	0.677485	0.545306
	10	0.3671171	0.541528	0.437584
	<b>Rata-rata</b>	<b>0.43463608</b>	<b>0.675562</b>	<b>0.518073</b>
2	11	0.248366	0.826087	0.38191
	12	0.4347826	0.703125	0.537313
	13	0.59	0.702381	0.641304
	14	0.3692946	0.577922	0.450633
	15	0.0661765	0.9	0.123288
	16	0.2947761	0.593985	0.394015
	17	0.2893401	0.791667	0.423792
	18	0.6170213	0.690476	0.651685
	19	0.3820961	0.601375	0.46729
	20	0.5959596	0.694118	0.641304
	<b>Rata-rata</b>	<b>0.38878129</b>	<b>0.708114</b>	<b>0.471253</b>
3	21	0.34347826	0.822917	0.484663
	22	0.55400697	0.713004	0.623529
	23	0.64194915	0.772959	0.701389
	24	0.68115942	0.61039	0.643836
	25	0.50289017	0.725	0.593857
	26	0.18858561	0.817204	0.306452
	27	0.41532258	0.837398	0.555256
	28	0.60747664	0.844156	0.706522
	29	0.63707572	0.862191	0.732733
	30	0.28225807	0.729167	0.406977
	<b>Rata-rata</b>	<b>0.48542026</b>	<b>0.773439</b>	<b>0.575521</b>
4	31	0.1974522	0.476923	0.279279
	32	0.5851064	1	0.738255
	33	0.3140097	0.613208	0.415335
	34	0.0630631	0.466667	0.111111
	35	0	0	0
	36	0.4022346	0.510638	0.45
	37	0.4883721	1	0.65625
	38	0.7826087	0.514286	0.62069
	39	0.5943396	0.715909	0.649485
	40	0.3626943	0.714286	0.4811
	<b>Rata-rata</b>	<b>0.37898807</b>	<b>0.601192</b>	<b>0.44015</b>
5	41	0.109090909	1	0.1967213
	42	0.317757009	0.6296296	0.4223602
	43	0.2	1	0.3333333
	44	0.556886228	0.2664756	0.3604651
	45	0.615942029	0.5743243	0.5944056
	46	0.502923977	0.8349515	0.6277372
	47	0.547445255	0.6410256	0.5905512
	48	0.512396694	0.5486726	0.5299145
	49	0.53526971	0.5584416	0.5466102
	50	0.528	0.519685	0.5238095
	<b>Rata-rata</b>	<b>0.44257118</b>	<b>0.657321</b>	<b>0.472591</b>

**Tabel Hasil Rouge-1 model Skip-gram vector 200 window 5**

Fold	Dokumen	Recall	Precision	F-measure
1	1	0.4124294	0.695238	0.51773
	2	0.61	0.616162	0.613065
	3	0.2604502	0.476471	0.336798
	4	0.585209	0.389722	0.467866
	5	0.5925926	0.677249	0.632099
	6	0.435567	0.53481	0.480114
	7	0.3140794	0.861386	0.460317
	8	0.7687296	0.749206	0.758842
	9	0.6502732	0.644986	0.647619
	10	0.6396396	0.741514	0.68682
	<b>Rata-rata</b>	<b>0.52689701</b>	<b>0.638674</b>	<b>0.560127</b>
2	11	0.4183007	0.627451	0.501961
	12	0.4975845	0.730496	0.591954
	13	0.52	0.679739	0.589235
	14	0.4024896	0.587879	0.477833
	15	0.3658088	0.805668	0.503161
	16	0.1492537	0.481928	0.22792
	17	0.5482234	0.885246	0.677116
	18	0.6542553	0.81457	0.725664
	19	0.5676856	0.562771	0.565217
	20	0.6919192	0.655502	0.673219
	<b>Rata-rata</b>	<b>0.48155208</b>	<b>0.683125</b>	<b>0.553328</b>
3	21	0.47391304	0.762238	0.58445
	22	0.44250871	0.604762	0.511066
	23	0.75423729	0.763948	0.759062
	24	0.69855073	0.621134	0.657572
	25	0.65317919	0.766102	0.705148
	26	0.36476427	0.835227	0.507772
	27	0.37903226	0.77686	0.509485
	28	0.78504673	0.870466	0.825553
	29	0.64490862	0.854671	0.735119
	30	0.43548387	0.786408	0.560554
	<b>Rata-rata</b>	<b>0.56316247</b>	<b>0.764182</b>	<b>0.635578</b>
4	31	0.3630573	0.438462	0.397213
	32	0.5106383	0.727273	0.6
	33	0.2608696	0.586957	0.361204
	34	0.4414414	1	0.6125
	35	0.3737374	1	0.544118
	36	0.4692737	0.485549	0.477273
	37	0.5	0.895833	0.641791
	38	0.5543478	0.401575	0.465753
	39	0.8962264	0.650685	0.753968
	40	0.2694301	0.452174	0.337662
	<b>Rata-rata</b>	<b>0.4639022</b>	<b>0.663851</b>	<b>0.519148</b>
5	41	0.209150327	1	0.3459459
	42	0.183574879	0.7755102	0.296875
	43	0.14	0.6829268	0.2323651
	44	0.22406639	0.4655172	0.302521
	45	0.154411765	0.6043165	0.2459736
	46	0.634328358	0.609319	0.6215722
	47	0.106598985	0.4285714	0.1707317
	48	0.234042553	0.8461538	0.3666667
	49	0.189956332	0.7565217	0.3036649
	50	0.393939394	0.8478261	0.537931
	<b>Rata-rata</b>	<b>0.2470069</b>	<b>0.701666</b>	<b>0.342425</b>

**Tabel Hasil Rouge-1 model Skip-gram vector 200 window 10**

Fold	Dokumen	Recall	Precision	F-measure
1	1	0.4830508	0.919355	0.633333
	2	0.475	0.904762	0.622951
	3	0.4855305	0.608871	0.54025
	4	0.4919614	0.42033	0.453333
	5	0.4814815	0.590909	0.530612
	6	0.4278351	0.551495	0.481858
	7	0.5090253	0.783333	0.617068
	8	0.713355	0.765734	0.738617
	9	0.4754098	0.643253	0.54674
	10	0.5630631	0.642674	0.60024
		<b>Rata-rata</b>	<b>0.51057126</b>	<b>0.683072</b>
2	11	0.3986928	0.743902	0.519149
	12	0.5362319	0.711538	0.61157
	13	0.605	0.823129	0.697406
	14	0.3526971	0.52795	0.422886
	15	0.2886029	0.773399	0.420348
	16	0.3134328	0.571429	0.404819
	17	0.1878173	0.822222	0.305785
	18	0.6808511	0.882759	0.768769
	19	0.3995633	0.6	0.479685
	20	0.7373737	0.705314	0.720988
		<b>Rata-rata</b>	<b>0.45002629</b>	<b>0.716164</b>
3	21	0.35217391	0.716814	0.472303
	22	0.52264808	0.652174	0.580271
	23	0.57838983	0.745902	0.651551
	24	0.65797101	0.698462	0.677612
	25	0.56647399	0.768627	0.652246
	26	0.33995037	0.872611	0.489286
	27	0.48790323	0.916667	0.636842
	28	0.62149533	0.836478	0.713137
	29	0.61357702	0.867159	0.718654
	30	0.46505376	0.711934	0.562602
		<b>Rata-rata</b>	<b>0.52056365</b>	<b>0.778683</b>
4	31	0.3248408	0.383459	0.351724
	32	0.4361702	0.82	0.569444
	33	0.2512077	0.52	0.338762
	34	0.3783784	1	0.54902
	35	0.2929293	1	0.453125
	36	0.0391061	0.205882	0.065728
	37	0.5232558	0.625	0.56962
	38	0.7065217	1	0.828025
	39	0.7830189	0.721739	0.751131
	40	0.3160622	0.559633	0.403974
		<b>Rata-rata</b>	<b>0.40514911</b>	<b>0.683571</b>
5	41	0	0	0
	42	0.2990654	0.780488	0.432432
	43	0.4347826	0.833333	0.571429
	44	0.4491018	0.290698	0.352941
	45	0.4057971	0.54902	0.466667
	46	0.4619883	0.858696	0.60076
	47	0.4379562	0.495868	0.465116
	48	0.8181818	0.614907	0.702128
	49	0.659751	0.742991	0.698901
	50	0.384	0.521739	0.442396
		<b>Rata-rata</b>	<b>0.43506243</b>	<b>0.568774</b>