

**PERINGKASAN TEKS CERITA ISLAM BERBAHASA INDONESIA
MENGUNAKAN *LONG SHORT TERM MEMORY* (LSTM)**

SKRIPSI

Oleh :
AISYA GUSTI SAVILA
NIM. 210605110045



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025**

**PERINGKASAN TEKS CERITA ISLAM BERBAHASA INDONESIA
MENGUNAKAN *LONG SHORT TERM MEMORY* (LSTM)**

SKRIPSI

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

Oleh :
AISYA GUSTI SAVILA
NIM. 210605110045

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025**

HALAMAN PERSETUJUAN

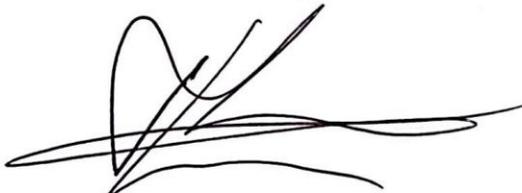
**PERINGKASAN TEKS CERITA ISLAM BERBAHASA INDONESIA
MENGUNAKAN *LONG SHORT TERM MEMORY* (LSTM)**

SKRIPSI

**Oleh :
AISYA GUSTI SAVILA
NIM. 210605110045**

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 14 Juni 2025

Pembimbing I,



Supriyono, M.Kom
NIP. 19841010 201903 1 012

Pembimbing II,



Roro Inda Melani, M.T, M.Sc
NIP. 19780925 200501 2 008

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




Drs. Fachrul Kurniawan, M.MT., IPU
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Aisyah Gusti Savila
NIM : 210605110045
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika
Judul Skripsi : Peringkasan Teks Cerita Islam Berbahasa Indonesia
Menggunakan *Long Short Term Memory (LSTM)*

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, 23 Juni 2025

Yang membuat pernyataan,



Aisyah Gusti Savila
NIM. 210605110045

MOTTO

"Proses kecilmu tetap terlihat, waktu hanya sedang menguji kesabaranmu."

HALAMAN PERSEMBAHAN

Dengan penuh rasa syukur kepada Allah SWT atas segala rahmat dan kemudahan-Nya, sehingga skripsi ini dapat terselesaikan dengan baik. Karya ini penulis persembahkan kepada:

Ibunda tercinta, Sitti Umamah dan Ayahanda tercinta, Hamery,
Yang selalu mengalirkan kasih sayang, usaha terbaik, do'a-do'a tulus, dukungan baik materi maupun non materi, dan nasehat yang tiada henti.

Kakak dan Adik tercinta,
Yang menjadi salah satu motivasi dan dorongan untuk terus maju hingga skripsi ini terselesaikan.

Segenap keluarga besar,
Yang selalu mengiringi perjalanan penulis dengan do'a dan dukungan.

Aster Teknik Informatika angkatan 2021,
Yang telah berjuang bersama dari awal hingga akhir.

teruntuk diri sendiri,
Terima kasih sudah berusaha dan bertahan sejauh ini.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillah rabbilalamin, segala puji dan syukur senantiasa penulis panjatkan ke hadirat Allah subhanahu wa ta'ala atas berkat Rahmat, serta hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Peringkasan Teks Cerita Islam Berbahasa Indonesia Menggunakan *Long Short Term Memory (LSTM)*” dengan baik dan lancar. Shalawat serta salam tetap tercurahkan kepada Nabi Muhammad SAW, yang telah membawa kita dari zaman kebodohan menuju zaman kebenaran yakni Islam dan zaman yang penuh dengan ilmu pengetahuan sebagaimana yang di rasakan pada saat ini. Dan semoga kita semua mendapat syafaatnya di hari akhir kelak, Aamiin.

Penulis mengucapkan rasa terima kasih yang begitu besar kepada seluruh pihak yang memberikan dukungan dan motivasi kepada penulis sehingga dapat menyelesaikan skripsi ini. Ucapan terima kasih ini penulis disampaikan kepada:

1. Prof. Dr. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Hj. Sri Harini, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Ir. Fachrul Kurniawan, M.MT., IPU., selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.

4. Supriyono, M.Kom selaku Dosen Pembimbing 1 yang telah membimbing serta memberikan arahan serta motivasi dalam penulisan skripsi dari awal hingga akhir.
5. Roro Inda Melani, M.T, M.Sc selaku Dosen Pembimbing 2 yang telah membimbing serta memberikan arahan serta motivasi dalam penulisan skripsi dari awal hingga akhir.
6. Dr. Irwan Budi Santoso, M.Kom dan Okta Qomaruddin Aziz, M.Kom selaku Dosen penguji yang telah meluangkan waktunya untuk menguji dan dengan sabar memberi arahan dan saran dalam menyelesaikan skripsi ini.
7. Segenap civitas akademik Program Studi Teknik Informatika, dan seluruh dosen yang telah memberikan ilmu serta arahan semasa kuliah.
8. Kedua orang tua penulis, serta keluarga besar yang telah memberikan banyak dukungan, doa serta selalu menjadi semangat sehingga penulis mampu menyelesaikan masa studi hingga mencapai gelar sarjana.
9. Kepada Kakak dan Adik Penulis yang sudah mendukung secara finansial dan emosional, serta doa yang tiada henti sehingga penulis bisa melewati proses ini sehingga selesai dengan baik.
10. Terkhusus kepada teman-teman penulis mutiara, fitria, sarah, chesha, atun, friska dan meideana yang telah memberikan bantuan yang sangat besar serta yang menjadi teman diskusi dan teman belajar.

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

Wassalamualaikum Warahmatullahi Wabarakatuh

Malang, 23 Juni 2025

Penulis

DAFTAR ISI

HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xiv
ABSTRAK	xv
ABSTRACT	xvi
المخلص	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Pernyataan Masalah.....	4
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
BAB II TINJAUAN PUSTAKA	6
2.1 Penelitian Terkait.....	6
2.2 Cerita Islam.....	8
2.3 Platform Digital	10
2.4 Peringkasan Teks.....	13
2.5 <i>Long Short Term Memory (LSTM)</i>	15
BAB III METODELOGI PENELITIAN	17
3.1 Prosedur Penelitian	17
3.2 Pengumpulan Data.....	18
3.3 Desain Sistem	20
3.3.1 Transkripsi Video Cerita Islami	21
3.3.2 Preprocessing	22
3.3.3 Ekstraksi Fitur dengan <i>Word2Vec</i>	23
3.3.4 <i>Long Short-Term Memory (LSTM)</i>	25
3.4 Skenario Pengujian	35
3.5 Evaluasi ROUGE.....	37
BAB IV HASIL DAN PEMBAHASAN	41
4.1 Tahapan Uji Coba.....	41
4.1.1 Data Pengujian	41
4.1.2 Hasil Preprocessing	43
4.2 Hasil Pengujian.....	44
4.2.1 Pengujian Rasio 90:10.....	46
4.2.2 Pengujian Rasio 80:20.....	52
4.2.3 Pengujian Rasio 70:30.....	58

4.3	Pembahasan	63
4.4	Integrasi Islam	71
BAB V KESIMPULAN DAN SARAN		74
5.1	Kesimpulan.....	74
5.2	Saran	75
DAFTAR PUSTAKA		
LAMPIRAN		

DAFTAR GAMBAR

Gambar 2.1 Struktur Memori Cell LSTM (Bastian Sianturi et al., 2023)	16
Gambar 3.1 Desain penelitian alur proses.....	17
Gambar 3.2 Blok Diagram Desain Sistem	20
Gambar 3.3 <i>Preprocessing</i> Data	22
Gambar 3.4 Arsitektur Jaringan LSTM	26
Gambar 3.5 <i>Flowchart</i> Inisialisasi Bobot LSTM.....	30
Gambar 3.6 <i>Flowchart</i> LSTM <i>Forward</i>	32
Gambar 3.7 <i>Flowchart</i> LSTM <i>Step</i>	34
Gambar 4.1 Grafik <i>Train Loss</i> dan <i>Train Accuracy</i> Model LSTM 90:10	46
Gambar 4.2 Visualisasi Panjang Ringkasan Aktual dan Prediksi LSTM (90:10)	47
Gambar 4.3 Grafik ROUGE <i>Precision</i> Sampel (Rasio 90:10)	48
Gambar 4.4 Grafik ROUGE Sampel (Rasio 90:10).....	49
Gambar 4.5 Grafik ROUGE F1 <i>Score</i> Sampel (Rasio 90:10)	50
Gambar 4.6 Grafik <i>Train Loss</i> dan <i>Train Accuracy</i> Model LSTM (Rasio 80:20)	52
Gambar 4.7 Visualisasi Panjang Ringkasan Aktual dan Prediksi LSTM (Rasio 80:20)	53
Gambar 4.8 Grafik ROUGE <i>Precision</i> Sampel (80:20).....	54
Gambar 4.9 Grafik ROUGE Sampel (80:20).....	55
Gambar 4.10 Grafik ROUGE F1 <i>Score</i> per Sampel (80:20)	56
Gambar 4.11 Grafik <i>Train Loss</i> dan <i>Train Accuracy</i> Model LSTM (Rasio 70:30).....	58
Gambar 4.12 Visualisasi Panjang Ringkasan Aktual dan Prediksi LSTM (Rasio 70:30).....	59
Gambar 4.13 Grafik ROUGE <i>Precision</i> Sampel (Rasio 70:30)	60
Gambar 4.14 Grafik ROUGE Sampel (Rasio 70:30).....	61
Gambar 4.15 Grafik ROUGE F1 <i>Score</i> Sampel (Rasio 70:30)	62

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait	7
Tabel 3.1 Skenario Pengujian	37
Tabel 4.1 Rasio Pembagian Data	43
Tabel 4.2 Contoh Data yang telah diproses	44
Tabel 4.3 Hasil Nilai ROUGE Seluruh Rasio	63
Tabel 4.4 Hasil Perbandingan Perbedaan Preprocessing Terhadap Ringkasan	65
Tabel 4.5 Perbandingan Hasil Peneliti dan Peneliti Sebelumnya	67
Tabel 4.6 Hasil ringkasan teks pada 3 skenario	67

ABSTRAK

Savila, Aisya Gusti. 2025. **Peringkasan Teks Cerita Islam Berbahasa Indonesia Menggunakan Long Short Term Memory (LSTM)**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Supriyono, M.Kom (II) Roro Ina Melani, M.T., M. Sc.

Kata Kunci: LSTM, Peringkasan Teks, ROUGE-L, ROUGE-N, Teknologi Digital, Teks Cerita Islam

Panjangnya narasi dalam cerita-cerita sering menjadi kendala bagi banyak pembaca, terutama mereka yang memiliki keterbatasan waktu atau kesulitan memahami isi cerita secara keseluruhan. Dalam hal ini, peringkasan menjadi solusi yang membantu, namun peringkasan manual tidak selalu efisien dalam memenuhi kebutuhan informasi yang cepat dan ringkas. Penelitian ini bertujuan untuk mengembangkan sistem peringkasan teks otomatis pada cerita-cerita Islam menggunakan algoritma *Long Short Term Memory* (LSTM). Penelitian ini menggunakan tiga skenario pembagian data pelatihan dan pengujian, yaitu 90:10, 80:20, dan 70:30. Hasil pengujian menunjukkan bahwa akurasi pelatihan tertinggi dicapai pada skenario 80:20 dengan nilai sebesar 89,44%. Hal ini tidak sepenuhnya menunjukkan bahwa proporsi data latih yang lebih kecil akan selalu menghasilkan akurasi lebih tinggi, karena peningkatan ini dapat dipengaruhi oleh variasi data, kondisi *overfitting*, dan kinerja *early stopping*. Oleh karena itu, rasio pembagian data memberikan pengaruh terhadap proses pelatihan. Meskipun akurasi pelatihan tertinggi diperoleh pada skenario 80:20, kualitas ringkasan secara semantik dengan performa terbaik terdapat pada skenario 90:10. Pada skenario 90:10 tersebut, nilai evaluasi ROUGE-1 mencapai *precision* sebesar 0,4147, sebesar 0,2516, dan *F1-score* sebesar 0,3027. Sedangkan ROUGE-2, mencapai *precision* sebesar 0,1022, sebesar 0,0568, dan *F1-score* sebesar 0,0684. Sementara itu, pada ROUGE-L dicapai *precision* sebesar 0,2017, sebesar 0,1209, dan *F1-score* sebesar 0,1459.

ABSTRACT

Savila, Aisya Gusti. 2025. **Indonesian Islamic Story Text Summarization Using Long Short Term Memory (LSTM)**. Thesis. Department of Informatics Engineering, Faculty of Science and Technology, State Islamic University Maulana Malik Ibrahim Malang. Advisor: (I) Supriyono, M.Kom (II) Roro Inda Melani, M.T., M. Sc.

The length of narratives in stories often becomes a challenge for many readers, especially those with limited time or difficulty in comprehending the entire content. In this context, summarization becomes a helpful solution. However, manual summarization is not always efficient in meeting the need for quick and concise information. This study aims to develop an automatic text summarization system for Islamic stories using the Long Short Term Memory (LSTM) algorithm. The study employs three training and testing data split scenarios: 90:10, 80:20, and 70:30. The experimental results show that the highest training accuracy was achieved in the 80:20 scenario, with a value of 89.44%. This does not entirely indicate that a smaller training data proportion will always yield higher accuracy, as this increase may be influenced by data variation, overfitting conditions, and early stopping performance. Therefore, the data split ratio has an impact on the training process. Although the highest training accuracy was obtained in the 80:20 scenario, the best semantic summary quality was found in the 90:10 scenario. In this 90:10 scenario, the ROUGE-1 evaluation achieved a precision of 0.4147, recall of 0.2516, and F1-score of 0.3027. Meanwhile, ROUGE-2 achieved a precision of 0.1022, recall of 0.0568, and F1-score of 0.0684. For ROUGE-L, a precision of 0.2017, recall of 0.1209, and F1-score of 0.1459 were obtained.

Key words: Digital Technology, Islamic Stories, LSTM, ROUGE-L, ROUGE-N, Text Summarization

الملخص

سافيليا، آيسيا غوستي. 2025. تلخيص نصوص القصص الإسلامية باللغة الإندونيسية باستخدام ذاكرة طويلة المدى قصيرة المدى، أطروحة. قسم هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية (LSTM)، رورو إندا ميلاني، ماجستير في الهندسة (II) سوربونو، ماجستير في علوم الكمبيوتر (I): مالانج. المشرفون. ماجستير في العلوم

الكلمات المفتاحية: التكنولوجيا الرقمية، نصوص القصص، ROUGE-L، ROUGE-N، تلخيص النصوص، LSTM، الإسلامية.

غالبًا ما تُشكّل طول السرد في القصص تحديًا لكثير من القراء، خاصةً أولئك الذين يعانون من ضيق الوقت أو صعوبة في فهم محتوى القصة بشكل كامل. في هذا السياق، تُعدّ عملية التلخيص حلاً مساعدًا، إلا أن التلخيص اليدوي لا يكون دائمًا فعالاً في تلبية الحاجة إلى المعلومات السريعة والموجزة. يهدف هذا البحث إلى تطوير نظام تلخيص تلقائي للنصوص في القصص استخدم البحث ثلاث سيناريوهات لتقسيم بيانات التدريب. (LSTM) الإسلامية باستخدام خوارزمية الذاكرة طويلة وقصيرة المدى والاختبار، وهي 90:10 و80:20 و70:30. أظهرت نتائج الاختبار أن أعلى دقة تدريب تم تحقيقها في سيناريو 80:20 بنسبة بلغت 89.44%. ومع ذلك، لا يدل هذا تمامًا على أن نسبة بيانات التدريب الأقل تؤدي دائمًا إلى دقة أعلى، إذ قد تتأثر هذه وبالتالي، فإن (early stopping) وأداء التوقف المبكر، (overfitting) الزيادة بتنوع البيانات، وحالات الإفراط في التعلم نسبة تقسيم البيانات تؤثر على عملية التدريب. وعلى الرغم من تحقيق أعلى دقة تدريب في سيناريو 80:20، فإن أفضل جودة دقة (precision) دقة ROUGE-1 للتلخيص من الناحية الدلالية تحققت في سيناريو 90:10. ففي هذا السيناريو، حقق تقييم دقة بنسبة ROUGE-2 بمقدار 0.3027. بينما حقق F1 بنسبة 0.2516، ودرجة (recall) بنسبة 0.4147 واسترجاع فقد تم الوصول إلى دقة بنسبة ROUGE-L بمقدار 0.0684. أما في F1 واسترجاعًا بنسبة 0.0568، ودرجة، 0.1022 بمقدار F1 0.1459 واسترجاع بنسبة 0.1209، ودرجة، 0.2017.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Peringkasan teks merupakan proses penting dalam penyampaian informasi, terlebih di era modern ini ketika jumlah konten digital meningkat secara eksponensial. Dalam konteks cerita-cerita Islam berbahasa Indonesia, teks-teks tersebut sering kali mengandung pesan moral, nilai-nilai kehidupan, dan ajaran agama yang sangat relevan bagi pembaca dari berbagai kalangan (AL-Hafidh et al., 2022). Namun, panjangnya narasi dalam cerita-cerita ini sering menjadi kendala bagi banyak pembaca, terutama mereka yang memiliki keterbatasan waktu atau kesulitan memahami isi cerita secara keseluruhan. Dalam hal ini, peringkasan menjadi solusi yang membantu, namun peringkasan manual tidak selalu efisien dalam memenuhi kebutuhan informasi yang cepat dan ringkas (Zakawaly et al., 2023). Hadits yang diriwayatkan oleh Imam Muslim dari Abu Hurairah menggambarkan pentingnya menyelesaikan kesulitan orang lain, yang merepresentasikan nilai-nilai moral dalam Islam:

مَنْ نَفَسَ عَنْ مُؤْمِنٍ كُرْبَةً مِنْ كُرْبِ الدُّنْيَا نَفَسَ اللَّهُ عَنْهُ كُرْبَةً مِنْ كُرْبِ يَوْمِ الْقِيَامَةِ وَمَنْ يَسَّرَ عَلَىٰ مُعْسِرٍ يَسَّرَ اللَّهُ عَلَيْهِ فِي الدُّنْيَا وَالْآخِرَةِ

"Siapa yang menyelesaikan kesulitan seorang mukmin dari berbagai kesulitan-kesulitan dunia, niscaya Allah akan memudahkan kesulitan-kesulitannya pada hari kiamat. Siapa yang memudahkan orang yang sedang kesulitan niscaya Allah mudahkan baginya di dunia dan akhirat." (HR. Muslim).

Hadits ini menggambarkan pentingnya membantu orang lain dalam kesulitan, yang merepresentasikan nilai-nilai moral dan ajaran Islam yang sering kita temukan dalam berbagai cerita berbahasa Indonesia. Nilai-nilai seperti ini harus tetap dijaga dan ditampilkan dengan akurat dalam proses peringkasan teks.

Peringkasan manual memerlukan keterampilan analisis yang mendalam serta waktu yang untuk memastikan pesan utama tetap terjaga tanpa menghilangkan elemen penting. Namun, seiring bertambahnya jumlah teks yang harus diringkas, metode ini menjadi semakin sulit diterapkan secara efisien (Utomo et al., 2022). Dalam konteks cerita Islam berbahasa Indonesia, yang sering kali kaya akan nilai moral dan budaya, tantangan ini semakin kompleks karena proses manual membutuhkan perhatian khusus terhadap keutuhan pesan (Setiadi B, 2017)

Selain itu, penyajian cerita Islam dalam format video berdurasi panjang menambah beban kognitif audiens. Materi yang kompleks dan memerlukan regulasi diri yang tinggi sering kali menyebabkan penurunan tingkat keterlibatan audiens, sehingga pesan moral yang ingin disampaikan menjadi kurang efektif (Hughes et al., 2018). Oleh karena itu, diperlukan pendekatan baru yang lebih efisien untuk meringkas teks dari video-video tersebut tanpa kehilangan esensi maknanya.

Model kecerdasan buatan seperti *Long Short Term Memory* (LSTM) menawarkan solusi yang lebih praktis dan efektif. LSTM, sebagai bagian dari jaringan saraf tiruan, memiliki kemampuan memahami konteks temporal dalam teks, sehingga dapat menghasilkan ringkasan yang padat namun tetap bermakna (Aldi et al., 2018). Keunggulan ini menjadikannya alternatif yang menjanjikan

untuk menggantikan metode manual, terutama dalam pengolahan teks panjang dan kompleks (Sandag, 2023).

Penelitian sebelumnya menunjukkan bahwa LSTM unggul dalam menghasilkan ringkasan yang lebih baik dibandingkan metode tradisional maupun model lain seperti *Transformer* dan RNN. Misalnya, pada kasus transkrip TEDx, LSTM terbukti mampu memberikan efisiensi waktu sekaligus menjaga kualitas hasil (Bayat & Isik, 2023). Namun, penerapan model ini pada Bahasa Indonesia menghadirkan tantangan tersendiri. Bahasa Indonesia memiliki struktur sintaksis dan semantik yang unik, sehingga model LSTM memerlukan adaptasi khusus dengan data pelatihan yang relevan agar dapat menghasilkan ringkasan yang akurat (Alfhi Saputra, 2021).

Pada penelitian terdahulu, banyak algoritma yang telah digunakan untuk meringkas teks berita ataupun dokumen. Algoritma LSTM memiliki akurasi yang lebih tinggi daripada algoritma lainnya seperti *Transformer* atau RNN (Bayat & Isik, 2023). Namun implementasi LSTM masih belum dieksplorasi dalam konteks peringkasan cerita islami. Oleh karena itu, penelitian ini mengusulkan pengembangan sistem peringkasan teks otomatis berbasis LSTM, khususnya untuk video yang berisi cerita-cerita Islam berbahasa Indonesia. Sistem ini bertujuan untuk menghasilkan ringkasan yang tidak hanya padat dan efisien, tetapi juga tetap mempertahankan pesan moral, nilai-nilai kehidupan, serta ajaran agama yang terkandung dalam teks.

1.2 Pernyataan Masalah

Berdasarkan latar belakang di atas maka masalah yang diangkat adalah seberapa akurat algoritma LSTM dalam menghasilkan peringkasan teks?

1.3 Batasan Masalah

Batasan masalah dari Ringkasan Teks Cerita Islami Anak Berbahasa Indonesia Menggunakan LSTM dengan adalah:

- a. Penelitian ini terbatas pada cerita Islam dalam bahasa Indonesia yang diunggah pada berbagai platform digital, seperti youtube.
- b. Evaluasi performa model dilakukan menggunakan metrik ROUGE-N dan ROUGE-L untuk mengukur sejauh mana ringkasan yang dihasilkan oleh model mencerminkan elemen naratif kompleks dan nilai-nilai moral dari cerita Islam berbahasa Indonesia.

1.4 Tujuan Penelitian

Adapun Tujuan dari penelitian ini adalah mengukur tingkat akurasi algoritma LSTM dalam cerita Islam berbahasa Indonesia, sehingga mampu menghasilkan ringkasan yang bermakna, relevan, dan sesuai dengan konteks cerita.

1.5 Manfaat Penelitian

Manfaat yang ditawarkan dalam penelitian ini adalah peneliti berharap dapat memberikan manfaat melalui penerapan Metode LSTM dalam meringkas teks cerita Islam berbahasa Indonesia secara otomatis sehingga mampu membantu pembaca memahami inti cerita dengan lebih cepat tanpa kehilangan pesan moral

yang terkandung. Selain itu, penelitian ini juga bertujuan untuk meningkatkan efektivitas distribusi cerita-cerita Islam melalui media digital, sehingga pesan keagamaan dapat lebih mudah diakses dan dimanfaatkan oleh masyarakat luas di era informasi saat ini.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian terkait *Text Summarization* menunjukkan berbagai inovasi dalam pendekatan berbasis deep learning. Penelitian oleh (Song et al., 2019) mengembangkan model LSTM-CNN (ATSDL) yang berhasil meningkatkan kualitas semantik dan struktur sintaksis ringkasan dibandingkan metode sebelumnya, terutama pada dataset CNN dan DailyMail. Penelitian oleh (Sirohi et al., 2021) memanfaatkan model LSTM dengan pendekatan encoder-decoder untuk menghasilkan ringkasan yang relevan secara real-time, mempertahankan informasi penting dari teks asli. Penelitian oleh (Tawong et al., 2024) membandingkan performa model LSTM dan GRU pada teks berita ekonomi berbahasa Thailand. Model LSTM menunjukkan performa terbaik pada teks pendek hingga menengah dengan panjang maksimum 500 kata, mencapai ROUGE-1 Recall sebesar 20,7. Sementara itu, model GRU lebih unggul pada teks panjang dengan panjang maksimum 700 kata, mencapai ROUGE-1 Recall tertinggi sebesar 26,1 serta menawarkan efisiensi komputasi yang lebih baik. Hasil ini menyoroti kemampuan model berbasis LSTM dan GRU dalam menghasilkan ringkasan berkualitas tinggi sesuai dengan kebutuhan berbagai panjang teks. Penelitian oleh (Sreenivasulu et al., 2022) menggunakan algoritma *TextRank* untuk menghasilkan ringkasan ekstraktif pada dokumen teks berbahasa Hindi. Algoritma ini bekerja dengan membangun matriks kesamaan antar kalimat, yang kemudian dikonversi menjadi

graf untuk menghitung peringkat setiap kalimat menggunakan pendekatan mirip PageRank. Model ini mampu menghasilkan ringkasan yang relevan dengan skor akurasi tinggi, yaitu *Precision* 0.6854, *Recall* 0.9497, dan *F1-score* 0.7962, yang menunjukkan keandalan metode ini untuk bahasa dengan sumber daya terbatas seperti Hindi. Penelitian oleh (Karthi et al., 2024) memperluas inovasi ini dengan menggabungkan kemampuan ekstraktif dari model BERT dan kemampuan abstraktif dari LSTM dalam sebuah model hibrida. Model ini dilatih menggunakan dataset BBC News dan dioptimasi dengan *Particle Swarm Optimization* (PSO) untuk meningkatkan efisiensi konvergensi. Hasil evaluasi menunjukkan performa yang signifikan dengan ROUGE-1 0.6714, ROUGE-2 0.5642, dan ROUGE-L 0.6714, yang mencerminkan kemampuan model ini dalam menangkap esensi teks asli.

Tabel 2.1 Penelitian Terkait

Peneliti	Judul	Data	Metode	Hasil
(Song et al., 2019)	<i>Abstractive Text Summarization Using LSTM-CNN Based Deep Learning</i>	CNN dan DailyMail.	Metode ATS berbasis LSTM-CNN	Model ATSDL menghasilkan ringkasan dengan kualitas semantik dan struktur sintaksis yang lebih baik dibandingkan metode sebelumnya pada dataset CNN dan DailyMail.
(Sirohi et al., 2021)	<i>Text Summarization Approaches Using Machine Learning & LSTM</i>	Dokumen seperti blog, artikel, dan laporan	Model LSTM dengan <i>encoder-decoder</i>	Teknik ATS berbasis LSTM menghasilkan ringkasan yang mempertahankan informasi penting dari teks asli, relevan untuk aplikasi real-time.
(Tawong et al., 2024)	<i>Economic News Using LSTM and GRU Models for Text</i>	ThaiSum	Model LSTM dan GRU	Model LSTM dengan panjang input maksimum 500 kata mencapai performa terbaik dengan ROUGE-1 Recall

Peneliti	Judul	Data	Metode	Hasil
	<i>Summarization in Deep Learning</i>			sebesar 20,7. Sementara itu, model GRU menunjukkan hasil yang lebih unggul, khususnya dengan panjang input maksimum 700 kata, mencapai ROUGE-1 Recall tertinggi sebesar 26,1.
(Sreenivasulu et al., 2022)	<i>Text Summarization Using Natural Language Processing</i>	<i>Hindi text document</i>	<i>Algoritma TeksRank</i>	Menghasilkan ringkasan ekstraktif untuk dokumen teks Hindi dengan akurasi (Precision 0.6854, Recall 0.9497, F1-score 0.7962).
(Kartha et al., 2024)	<i>NLP-Based Automatic Summarization using BERT-LSTM Hybrid Model</i>	BBC News	Hybrid model: BERT (ekstraktif) dan LSTM (abstraktif), dioptimasi dengan Particle Swarm Optimization	Meningkatkan kualitas ringkasan teks dengan ROUGE-1:0.6714, ROUGE-2:0.5642, ROUGE-L: 0.6714 pada dataset BBC News.

2.2 Cerita Islam

Cerita Islam merupakan bagian penting dari tradisi literasi Islam yang bertujuan untuk mendidik umat melalui kisah-kisah yang penuh dengan nilai moral, etika, dan ajaran agama (Hanifaa, U; Ardianti, S; Putri, 2023). Cerita-cerita ini bisa berupa kisah nabi dan rasul, kisah hidup sahabat nabi, serta kisah-kisah inspiratif dari kehidupan umat Muslim di berbagai zaman. Teks cerita Islam sering kali diwarnai dengan pesan-pesan yang mendalam terkait dengan moralitas, ketakwaan, keadilan, dan kepatuhan kepada Allah SWT. Penting untuk mengolah teks cerita Islam dalam bentuk yang mudah dipahami oleh berbagai kalangan (Nofitayanti & Fitri, 2022). Oleh karena itu, teknik penyaringan teks otomatis (*text summarization*) dapat digunakan untuk mempermudah pembaca dalam memahami inti dari cerita-

cerita tersebut. Namun, tantangan utama dalam meringkas teks cerita Islam adalah menjaga agar pesan moral dan ajaran yang terkandung dalam teks tidak hilang atau terdistorsi selama proses penyaringan.

Selain itu, penggunaan metode penyaringan otomatis juga memungkinkan pemanfaatan cerita Islam dalam berbagai platform digital, yang semakin berkembang pesat. Dengan format yang lebih pendek dan mudah dicerna, cerita-cerita Islam dapat dengan cepat disebarkan kepada audiens yang lebih luas. Hal ini tentunya mempermudah akses pembelajaran agama dan meningkatkan kesadaran spiritual di kalangan masyarakat. Meskipun penyaringan dapat meningkatkan efisiensi dalam penyebaran informasi, tetap perlu perhatian ekstra untuk memastikan kualitas narasi tetap utuh, terutama dalam mempertahankan substansi ajaran yang sangat kental dalam teks-teks agama. Penyaringan otomatis memiliki tantangan khusus, yaitu bagaimana menjaga keseimbangan antara ringkasan yang efektif dengan penghilangan makna yang tidak diinginkan.

Dalam konteks cerita Islam, hal ini menjadi semakin penting karena setiap cerita tidak hanya berfungsi sebagai narasi, tetapi juga sebagai sarana untuk mentransmisikan nilai-nilai agama yang mendalam. Tidak hanya panjang teks yang perlu diperhatikan, tetapi juga konteks dan pesan moral yang terkandung dalam setiap kalimat. Setiap elemen dalam cerita Islam, baik itu kisah nabi, sahabat, maupun peristiwa penting dalam sejarah Islam, memiliki dimensi spiritual yang harus dipertahankan. Oleh karena itu, penerapan teknik penyaringan teks memerlukan model yang dapat mendeteksi nuansa dan kedalaman yang ada dalam cerita-cerita tersebut. Tanpa pendekatan yang hati-hati, ada risiko bahwa informasi

penting seperti nilai keadilan, ketakwaan, dan ketulusan hati bisa hilang dalam proses pemotongan teks, yang pada akhirnya dapat mengurangi makna moral yang terkandung di dalamnya.

Sebagai langkah penting berikutnya, pemahaman tentang bagaimana cerita-cerita Islam berfungsi sebagai sarana untuk mendidik umat sangatlah krusial. Cerita-cerita ini sering kali mengandung pelajaran hidup yang tidak hanya berlaku pada konteks sejarah, tetapi juga relevan dalam kehidupan sehari-hari umat Muslim. Misalnya, kisah-kisah pengorbanan para sahabat nabi dalam menghadapi ujian hidup atau keteladanan nabi dalam bersikap adil dan bijaksana. Penyaringan otomatis yang tidak memperhatikan kedalaman konteks ini dapat mengurangi nilai pendidikan yang terkandung dalam cerita tersebut. Untuk itu, penyaringan teks harus dilakukan dengan cermat, menjaga agar setiap aspek penting yang membawa pesan moral tetap terjaga dan tidak hilang dalam proses pemangkasan teks.

2.3 Platform Digital

Platform digital telah menjadi sarana utama untuk mendistribusikan dan mengakses teks cerita Islam (Saraçoğlu, 2022). Berbagai platform seperti website, aplikasi mobile, dan e-book kini memudahkan umat Muslim di seluruh dunia untuk mengakses berbagai kisah dan cerita yang dapat memperkaya pemahaman agama mereka. Selain itu, *platform-platform* ini juga sering kali dilengkapi dengan fitur untuk membagikan konten, termasuk fitur berbagi ringkasan cerita atau pesan-pesan moral yang terkandung dalam teks (Nurdin et al., 2019).

Keunggulan *platform digital* terletak pada kemampuannya untuk menyajikan informasi dengan cara yang interaktif dan mudah diakses kapan saja

dan di mana saja. Pengguna dapat dengan mudah menemukan cerita-cerita Islam yang relevan dengan topik yang mereka minati, tanpa perlu terbatas oleh ruang dan waktu. Selain itu, banyak platform digital yang memungkinkan pembaca untuk memilih jenis cerita yang ingin mereka baca, misalnya kisah nabi, kisah sahabat, atau cerita inspiratif lainnya. Dengan adanya fitur pencarian dan kategori, proses penemuan cerita-cerita yang sesuai dengan kebutuhan pembaca menjadi lebih efisien dan menyenangkan. Hal ini membuka peluang besar untuk menyebarkan ajaran Islam ke berbagai kalangan, dari anak-anak hingga orang dewasa, melalui konten yang lebih bervariasi dan mudah diakses.

Keberadaan fitur berbagi di *platform digital* juga memberikan dampak positif dalam memperluas jangkauan cerita Islam. Pengguna dapat membagikan cerita atau ringkasan cerita yang mereka temukan kepada teman atau keluarga, memperkenalkan mereka pada nilai-nilai agama yang terkandung dalam cerita-cerita tersebut. Di samping itu, fitur diskusi dan komentar yang tersedia pada banyak platform memungkinkan pembaca untuk saling berbagi pendapat dan interpretasi, memperkaya pemahaman mereka tentang pesan moral dan spiritual dalam cerita Islam. Proses ini tidak hanya memperluas akses terhadap cerita Islam, tetapi juga menciptakan komunitas pembaca yang aktif dan terlibat dalam pembelajaran agama.

Penggunaan teknologi LSTM untuk meringkas teks cerita Islam juga sangat berpotensi dalam konteks *platform digital*. Teknologi ini memungkinkan platform untuk menghasilkan ringkasan otomatis dari teks panjang dengan tetap mempertahankan esensi pesan moral dan spiritual yang terkandung dalam cerita.

Dengan demikian, pembaca dapat menghemat waktu dalam memahami inti dari cerita tanpa kehilangan makna yang terkandung di dalamnya. LSTM memungkinkan proses ini dilakukan secara cepat dan akurat, sehingga dapat digunakan untuk menghasilkan ringkasan cerita yang relevan dan informatif bagi pembaca yang sibuk atau mereka yang ingin mendapatkan pemahaman singkat tetapi mendalam tentang ajaran Islam.

Selain itu, penggunaan LSTM dalam platform digital juga memungkinkan peningkatan pengalaman pengguna secara keseluruhan. Dengan adanya ringkasan otomatis, pengguna tidak hanya mendapatkan cerita secara lebih efisien, tetapi juga mendapatkan paparan yang lebih terfokus pada nilai-nilai inti yang ingin disampaikan dalam cerita Islam. Ini sangat penting mengingat keterbatasan waktu yang dimiliki oleh banyak pengguna platform digital. Dengan demikian, teknologi ini tidak hanya mempermudah akses terhadap konten, tetapi juga membantu menjaga kualitas dan kedalaman pesan yang ingin disampaikan, memastikan bahwa cerita Islam tetap dapat dipahami dengan baik meskipun dalam bentuk yang lebih ringkas.

Dengan semakin berkembangnya teknologi *digital*, *platform-platform* ini diharapkan dapat terus berinovasi dalam cara mereka menyajikan cerita-cerita Islam kepada umat Muslim di seluruh dunia. *Platform digital* tidak hanya menjadi tempat untuk mengakses teks, tetapi juga menjadi ruang untuk pengembangan komunitas yang saling berbagi pengetahuan, pengalaman, dan pemahaman tentang ajaran agama. Di masa depan, integrasi teknologi penyaringan teks otomatis seperti LSTM akan semakin memperkaya pengalaman pengguna, memungkinkan umat

Muslim untuk lebih mudah dan cepat mengakses, memahami, dan berbagi nilai-nilai agama yang terkandung dalam cerita-cerita Islam. Dengan cara ini, *platform digital* tidak hanya berfungsi sebagai alat distribusi konten, tetapi juga sebagai sarana untuk memperdalam pemahaman agama dan mempererat hubungan antar umat Muslim.

2.4 Peringkasan Teks

Peringkasan teks atau *text summarization* adalah teknik dalam pemrosesan bahasa alami yang bertujuan untuk menghasilkan ringkasan dari teks yang panjang dan kompleks, namun tetap mempertahankan informasi yang penting (Das & Roy, 2024). Teknik peringkasan teks dibagi menjadi dua jenis, yaitu *extractive summarization* dan *abstractive summarization* (Raihanunnisa et al., 2023). *Extractive summarization* memilih kalimat-kalimat penting dari teks untuk disusun ulang menjadi ringkasan, dan sering digunakan pada teks yang bersifat faktual serta tidak memerlukan interpretasi kompleks (Zhao et al., 2023). Sebaliknya, *abstractive summarization* menghasilkan ringkasan dengan menulis ulang teks menggunakan kalimat-kalimat baru yang menyampaikan inti dari teks asli (Zaman et al., 2024). Pendekatan ini lebih kompleks karena membutuhkan pemahaman konteks yang lebih mendalam serta kemampuan untuk menggeneralisasi informasi. Pada teks cerita Islam, pendekatan abstraktif lebih cocok diterapkan karena cerita-cerita tersebut sering kali mengandung pesan moral dan ajaran yang perlu disampaikan dalam bentuk ringkasan yang tetap mendalam (Karim et al., 2024). Model LSTM merupakan metode yang sangat efektif untuk *abstractive*

summarization karena kemampuannya dalam memahami urutan kata dan konteks yang terkandung dalam teks.

Peringkasan teks sangat relevan dalam konteks cerita Islam karena sering kali teks-teks tersebut panjang dan mengandung banyak informasi yang bisa jadi tidak semuanya relevan untuk audiens tertentu. Dalam cerita-cerita Islam, banyak pesan moral dan ajaran yang harus disampaikan dengan tepat dan mendalam. Proses peringkasan ini memungkinkan penyampaian pesan yang lebih ringkas namun tetap utuh, sehingga audiens dapat menangkap makna inti dari cerita tanpa harus membaca seluruh teks. Dengan memanfaatkan teknik seperti *abstractive summarization*, kita dapat merangkum cerita Islam dalam bentuk yang lebih padat tanpa kehilangan kedalaman nilai-nilai yang terkandung di dalamnya.

Di sisi lain, penerapan *abstractive summarization* dengan model seperti LSTM memberikan keuntungan besar dalam meringkas teks secara otomatis. LSTM mampu mempertahankan konteks dan urutan kata dalam cerita yang sangat penting dalam teks Islam, di mana makna sering kali tergantung pada hubungan antar kalimat dan kata. Dengan menggunakan LSTM, ringkasan yang dihasilkan tidak hanya mengurangi panjang teks tetapi juga menjaga kesinambungan pesan moral dan ajaran yang ingin disampaikan. Oleh karena itu, model ini sangat efektif dalam merangkum teks cerita Islam yang mengandung pesan moral, ketakwaan, dan ajaran agama.

Meskipun *extractive summarization* dapat digunakan untuk mengidentifikasi kalimat-kalimat penting dalam teks, namun untuk cerita Islam yang kompleks, pendekatan abstraktif lebih cocok karena dapat menghindari

hilangnya konteks dan makna dari cerita yang lebih dalam. *Abstractive summarization* memungkinkan penyusunan ulang kalimat dalam bentuk yang lebih ringkas namun tetap memuat seluruh pesan yang ada dalam teks asli. Dengan kata lain, metode ini lebih fleksibel dalam menangani teks yang mengandung pesan keagamaan, yang sering kali membutuhkan pemahaman lebih dalam tentang konteks dan nuansa bahasa.

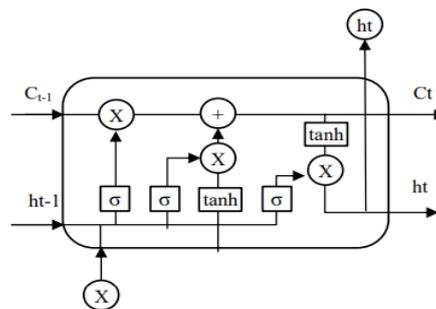
Selanjutnya, penerapan model LSTM dalam *abstractive summarization* memberikan kelebihan dalam hal konsistensi dan kualitas ringkasan. LSTM mampu memahami dan menangkap informasi jangka panjang dalam teks yang diperlukan untuk membuat ringkasan yang tetap mengandung nilai-nilai penting. Dalam hal cerita Islam, yang sering kali melibatkan aspek naratif dan ajaran yang tidak dapat dipisahkan, kemampuan LSTM untuk mempertahankan informasi kritis dalam struktur ringkasan sangat berguna. Dengan demikian, teknologi ini berperan besar dalam mempermudah proses pemahaman terhadap teks cerita Islam yang panjang, membantu pembaca untuk mendapatkan esensi dari cerita dengan lebih cepat dan efektif tanpa kehilangan makna atau nilai yang terkandung dalam cerita tersebut.

2.5 Long Short Term Memory (LSTM)

Long Short-Term Memory adalah jenis jaringan saraf tiruan yang dirancang untuk mengatasi keterbatasan dari *Recurrent Neural Networks* (RNNs) dalam menangani data berurutan, seperti teks atau urutan kata (Kong et al., 2024). LSTM memiliki kemampuan untuk mengingat informasi dalam jangka panjang, yang membuatnya sangat efektif untuk tugas-tugas yang memerlukan pemahaman konteks dan urutan, seperti *text summarization* (Bayat & Isik, 2023).

LSTM terdiri dari tiga komponen utama; *input gate*, *forget gate*, dan *output gate*, yang bekerja bersama untuk mengatur informasi yang masuk, disimpan, dan dikeluarkan dari memori (*LiteLSTM Architecture for Deep Recurrent Neural Networks*, 2022). Keunggulan LSTM dalam abstractive summarization adalah kemampuannya untuk memproses informasi yang lebih kompleks dan menghasilkan ringkasan yang lebih alami dan koheren (Khan et al., 2023).

LSTM menggunakan mekanisme gerbang (*gates*) untuk mengatur apa yang harus disimpan atau dihapus dari memori. Berikut gambar dari arsitektur LSTM.



Gambar 2.1 Struktur Memori Cell LSTM (Bastian Sianturi et al., 2023)

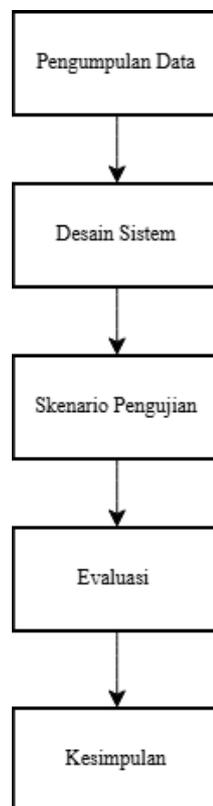
Pada LSTM, terdapat tiga jenis gerbang atau *gates unit*, yaitu *forget gate*, *input gate*, dan *output gate*. *Forget gate* berperan dalam mengontrol sejauh mana nilai yang ada tetap dipertahankan dalam sel memori. *Input gate* berfungsi untuk mengatur seberapa banyak informasi yang perlu disimpan dalam sel memori, sehingga menghindari penyimpanan data yang tidak diperlukan. Sementara itu, *output gate* menentukan seberapa banyak informasi yang ada dalam sel memori yang akan digunakan untuk menghasilkan output. Setiap gerbang ini memiliki persamaan yang berbeda, yang mengatur bagaimana informasi diproses dan disimpan dalam jaringan (Wiranda & Sadikin, 2019).

BAB III

METODELOGI PENELITIAN

3.1 Prosedur Penelitian

Desain penelitian yang akan dilakukan dalam penelitian ini ditunjukkan pada gambar 3.1.



Gambar 3.1 Desain penelitian alur proses

Pada Gambar 3.1 Proses yang digambarkan terdiri dari lima tahapan utama. Tahap pertama adalah Pengumpulan Data, di mana data yang dibutuhkan untuk proyek dikumpulkan. Langkah ini melibatkan pencarian, pengumpulan, dan pengolahan data mentah agar siap digunakan pada tahap berikutnya. Setelah data

terkumpul, tahap berikutnya adalah Desain Sistem, yang berfokus pada perancangan sistem secara menyeluruh, termasuk metode, struktur, dan mekanisme kerja yang akan digunakan untuk menyelesaikan permasalahan yang dihadapi. Setelah desain sistem selesai, proses dilanjutkan ke tahap Implementasi dan Pengujian, di mana sistem yang telah dirancang diimplementasikan. Pada tahap ini, kode atau algoritma dibuat berdasarkan desain, kemudian dilakukan pengujian untuk memastikan bahwa sistem berfungsi sesuai dengan yang direncanakan. Selanjutnya, hasil dari implementasi dan pengujian dievaluasi pada tahap Evaluasi. Dalam evaluasi, dilakukan analisis terhadap hasil yang diperoleh, seperti efektivitas, efisiensi, dan kesesuaian dengan tujuan awal. Tahap terakhir adalah Kesimpulan, di mana seluruh proses dirangkum, termasuk temuan, hasil akhir, serta kesimpulan tentang keberhasilan proyek. Tahapan ini juga dapat mencakup saran untuk pengembangan lebih lanjut di masa depan.

3.2 Pengumpulan Data

Data dalam penelitian ini diperoleh dari platform digital seperti YouTube yang menyajikan cerita Islami berbahasa Indonesia dalam bentuk video. Pengumpulan data dari *YouTube* menggunakan API *Google Cloud Console* untuk mengekstraksi metadata seperti judul, deskripsi, dan transkrip video yang mengandung cerita Islami dengan struktur naratif yang jelas, pesan moral, dan nilai-nilai keagamaan Islam. Dalam proses penelitian ini, ringkasan manual yang dijadikan acuan sebagai *ground truth* disusun oleh dua orang validator yang memiliki latar belakang pendidikan tinggi dalam bidang Pendidikan Agama Islam (PAI) dan pengalaman sebagai tenaga pendidik. Mereka membaca dan merangkum

cerita secara menyeluruh untuk menghasilkan ringkasan yang bermakna, relevan, dan sesuai dengan nilai-nilai keislaman.

Validator pertama adalah Mufti Umma Rosyidah, S.Pd, seorang guru aktif di SDN Pandaan I, Kabupaten Pasuruan, Jawa Timur. Beliau merupakan lulusan S1 Pendidikan Agama Islam dari Universitas Islam Malang (UNISMA). Dengan pengalaman langsung mengajar mata pelajaran agama Islam di tingkat dasar, beliau memiliki kemampuan dalam menyusun ringkasan yang tidak hanya informatif tetapi juga mudah dipahami oleh berbagai kalangan. Beliau juga dikenal teliti dan memiliki pemahaman mendalam terhadap konten-konten keislaman yang dikaji.

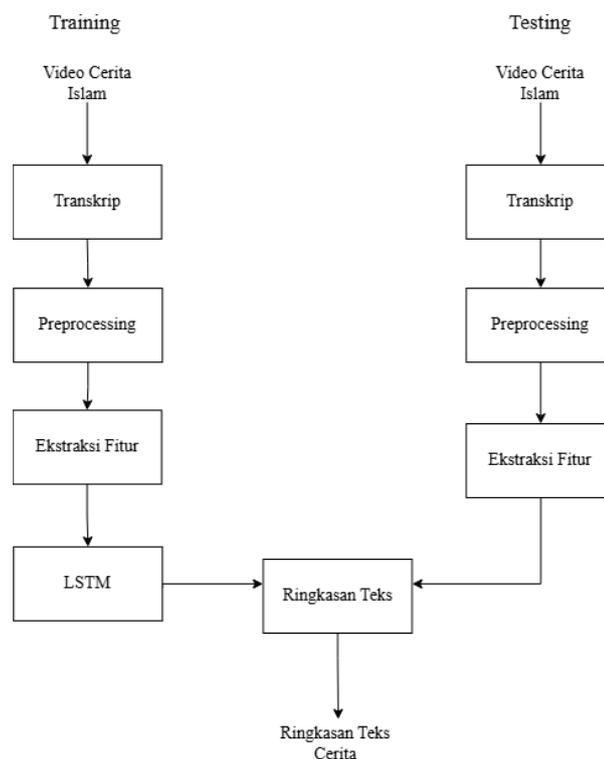
Validator kedua adalah Alivia Lailaturrohma, S.Pd, lulusan S1 Pendidikan Agama Islam dari Universitas Islam As-Syafiiyah. Beliau memiliki pengalaman sebagai staf di pesantren dan saat ini aktif mengajar di MI Al Ikhlas Pondok Gede.

Kedua validator ini telah memberikan kontribusi besar dalam memastikan bahwa ringkasan yang digunakan sebagai acuan benar-benar mencerminkan isi cerita secara utuh serta relevan dengan konteks keislaman, sehingga dapat digunakan sebagai standar pembanding yang andal dalam proses evaluasi model. Prosedur yang mereka lakukan dalam proses meringkas dimulai dengan menonton video berdurasi sekitar 20 menit yang berisi cerita Islami. Validator pertama kemudian menyimak isi video secara menyeluruh dan membuat ringkasan teks berdasarkan apa yang didengarkan, tanpa mengandalkan transkrip otomatis. Ringkasan yang dihasilkan bersifat manual, menggambarkan inti cerita dengan runtut dan bahasa yang baik, serta tetap menjaga makna dan pesan-pesan keislaman yang terkandung di dalamnya. Contohnya, apabila dalam video diceritakan tentang

keteladanan Nabi Ibrahim dalam menghadapi ujian, maka ringkasan yang dibuat akan menyoroti nilai-nilai keimanan, kepatuhan, dan pengorbanan yang menjadi inti dari cerita tersebut. Data yang sudah dikumpulkan melalui *crawling* maupun ringkasan manual digabungkan menjadi satu dataset tunggal yang digunakan untuk melatih dan menguji model LSTM dalam menghasilkan ringkasan teks yang bermakna, relevan, dan mencerminkan nilai moral serta narasi Islami.

3.3 Desain Sistem

Desain sistem pada penelitian menggunakan metode LSTM yang ditunjukkan pada gambar 3.2



Gambar 3.2 Blok Diagram Desain Sistem

3.3.1 Transkripsi Video Cerita Islami

Pengambilan transkrip otomatis dari masing-masing video di dalamnya, menerjemahkan transkrip tersebut ke dalam bahasa Indonesia, lalu menyimpannya ke dalam file CSV di Google Drive. Proses diawali dengan meng-*mount* *Google Drive* agar file CSV bisa disimpan langsung ke direktori tersebut. Kemudian, API Key YouTube diatur untuk bisa mengakses YouTube Data API dan membangun koneksi ke layanan tersebut menggunakan *build*.

Selanjutnya, fungsi *get_videos_from_playlist()* akan mengambil semua video dari playlist yang diberikan melalui *playlist_id*, termasuk informasi *videoId*, *title*, dan *link*-nya. Untuk setiap video yang didapat, fungsi *get_transcript()* akan mencoba mengambil transkrip menggunakan pustaka *YouTubeTranscriptApi*, dengan mencoba beberapa bahasa secara bergantian seperti Bahasa Indonesia, Inggris, Arab, dan Melayu. Teks transkrip yang diambil akan diproses terlebih dahulu dengan menghapus kata seperti "tepuk tangan", "musik", atau "tertawa", serta menghapus karakter non-alfabet.

Setelah transkrip diperoleh, fungsi *translate_to_indonesian()* akan mencoba menerjemahkan teks ke bahasa Indonesia menggunakan pustaka *deep_translator*. Jika proses penerjemahan gagal atau teks kosong, transkrip asli tetap disimpan. Semua data (judul video, tautan, dan transkrip yang sudah diterjemahkan) kemudian disimpan ke dalam file CSV menggunakan pustaka *csv*, dengan nama file FULL KISAH.csv yang disimpan di folder *Google Drive /content/drive/My Drive/CRAWLING DATA/*. Setiap kali sebuah video berhasil diproses. Setelah

semua proses selesai, sistem akan menampilkan pesan bahwa data berhasil disimpan.

3.3.2 Preprocessing

Pada proses preprocessing data yang telah dikumpulkan akan di input untuk di proses mulai dari *cleaning data*, *lowercasing*, selanjutnya dilakukan tokenisasi. Tujuan dari proses preprocessing ini untuk membersihkan data agar data lebih siap untuk diolah dan masuk ke dalam sistem.



Gambar 3.3 *Preprocessing Data*

a. *Cleaning Data*

Pada tahap pertama dalam pemrosesan yaitu *cleaning data* dengan menghilangkan semua tanda baca seperti titik, koma, tanda seru, dan lainnya dihapus dari teks. Selanjutnya, semua angka dihilangkan dari teks karena biasanya tidak memberikan informasi penting dalam konteks pemrosesan ringkasan teks, kemudian dilakukan pembersihan spasi berlebihan, yaitu mengganti spasi ganda atau tab menjadi satu spasi serta menghapus spasi di awal atau akhir kalimat, terakhir dilakukan penghapusan pada tanda kutip tunggal (‘) dan ganda (“) karena dianggap tidak menambah makna penting dalam kalimat. Dengan tahapan *cleaning* ini, teks menjadi lebih bersih, seragam, dan siap untuk diproses lebih lanjut seperti tokenisasi.

b. *Lowercasing*

Tahapan kedua dalam pemrosesan data yaitu *lowercasing* proses ini bertujuan untuk menyamakan format kata agar tidak terjadi duplikasi yang disebabkan oleh perbedaan kapitalisasi. Misalnya, kata "*Sekolah*", "*sekolah*", dan "*SEKOLAH*" sebenarnya merujuk pada kata yang sama, namun jika tidak diubah menjadi huruf kecil, ketiganya akan dianggap sebagai token yang berbeda dalam analisis teks. Dengan menerapkan *lowercasing*, seluruh teks seperti "*Saya Belajar di Sekolah*" akan diubah menjadi "*saya belajar di sekolah*". Proses ini membantu meningkatkan konsistensi data dan sangat bermanfaat dalam tahap-tahap selanjutnya

c. Tokenisasi

Tahapan ketiga pada pemrosesan yaitu tokenisasi dimana teks dibagi menjadi bagian-bagian yang lebih kecil, yaitu token, yang bisa berupa kata, frasa, atau karakter. Tokenisasi memungkinkan komputer untuk memahami teks secara terstruktur dan memudahkan analisis lebih lanjut. Tujuan dari proses ini adalah membantu untuk menyederhanakan teks dan mengubahnya menjadi unit-unit yang lebih mudah dianalisis oleh algoritma pemrosesan bahasa alami. Tokenisasi adalah langkah awal yang penting sebelum melakukan proses lainnya.

3.3.3 Ekstraksi Fitur dengan *Word2Vec*

Pada tahapan ini ekstraksi fitur dilakukan dengan mengubah data berupa karakter atau kata menjadi representasi numerik berupa vektor. Teknik ini dikenal sebagai *word embedding*, dengan salah satu metode yang sering digunakan adalah *Word2Vec*. Feature extraction menggunakan *Word2Vec* adalah proses penting

dalam mengonversi teks cerita Islam menjadi representasi numerik yang dapat dipahami oleh model pembelajaran mesin seperti LSTM (Fares et al., 2017). *Word2Vec* bekerja dengan merepresentasikan setiap kata dalam bentuk vektor berdimensi rendah yang mencerminkan hubungan semantik antar kata berdasarkan konteks penggunaannya dalam teks. Dalam proses ini, algoritma *Word2Vec* mempelajari hubungan antar kata melalui dua pendekatan utama: *Continuous Bag of Words* (CBOW), yang memprediksi kata target berdasarkan kata-kata di sekitarnya, dan *Skip-gram*, yang memprediksi kata-kata konteks berdasarkan kata target (Aufa & Qoiriah, 2023).

Penelitian ini menggunakan pendekatan *Skip-gram* untuk mengidentifikasi hubungan semantik antar kata (Mikolov et al., 2013). Dalam arsitektur *Skip-gram*, model diberikan kata target (w_t) dan bertujuan untuk memprediksi kata-kata konteks yang berada di sekitarnya dalam jendela. Sebagai contoh, jika *window size* ditetapkan menjadi 2, maka model akan mempertimbangkan dua kata di sebelah kiri dan dua kata di sebelah kanan dari kata target (w_t). Proses ini dilakukan dengan menghitung probabilitas kondisional menggunakan rumus *softmax*, yang mengukur kemungkinan munculnya kata-kata konteks berdasarkan kata target. Sebelum menghitung probabilitas kondisional dilakukan perhitungan rata-rata log probabilitas dari kata-kata konteks yang muncul di sekitar kata target (w_t). Dengan persamaan 3.1

$$J = \frac{1}{T} \sum_{t=1}^T \sum_{-k \leq j \leq k, j \neq 0} \log P(w_{t+j} | w_t) \quad (3.1)$$

Keterangan :

- a) w_t : kata target pada posisi T
- b) w_{t+j} : kata-kata konteks dalam jendela k di sekitar w_t
- c) T : total keseluruhan kata target

Selanjutnya dilakukan perhitungan probabilitas kondisional menggunakan rumus softmax yang menggambarkan hubungan antara kata target (w_t) dan kata-kata konteks (w_c) pada persamaan 3.2

$$P(w_c | w_t) = \frac{\exp(v_{w_o}^T \cdot v_{w_t})}{\exp(v_w^T \cdot v_{w_t})} \quad (3.2)$$

Keterangan :

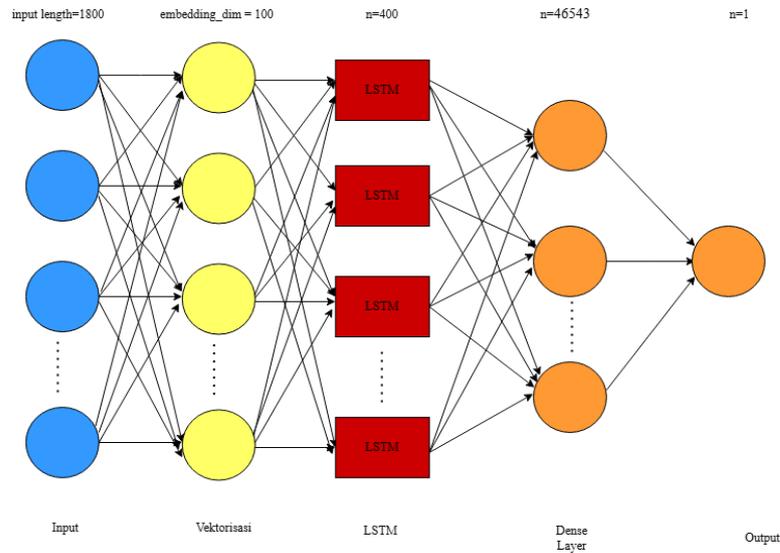
- a) v_{w_t} : vektor dari kata konteks
- b) v_{w_o} : vektor dari kata target
- c) V : ukuran kosakata

Fungsi softmax ini mengonversi hasil perkalian vektor menjadi probabilitas yang menggambarkan seberapa besar hubungan antara kata target dan kata-kata konteks. Dengan mengulang proses ini untuk seluruh kata dalam teks, model dapat mempelajari hubungan semantik yang ada dan menghasilkan representasi vektor untuk setiap kata dalam kosakata.

3.3.4 *Long Short-Term Memory (LSTM)*

Peringkasan teks berbasis LSTM merupakan salah satu pendekatan yang efektif dalam menangani permasalahan peringkasan otomatis. LSTM, sebagai bagian dari jaringan saraf tiruan, memiliki kemampuan unik untuk memahami konteks temporal dalam data sekuensial, seperti teks, sehingga mampu menangkap hubungan kompleks antar kata dalam sebuah dokumen. Untuk mendapatkan suatu model yang optimal dilakukan pembuatan arsitektur jaringan yang ada pada gambar

3.4



Gambar 3.4 Arsitektur Jaringan LSTM

Pada Gambar 3.4 menunjukkan arsitektur LSTM, dimana model ini menerima input dengan panjang nilai maksimal 1800, setiap input terdiri dari 1800 token atau kata yang akan diproses oleh model. Nilai ini mewakili data input dalam bentuk urutan teks yang terdiri dari kata-kata atau token yang sudah dipisahkan. Setiap kata atau token dalam input tersebut kemudian direpresentasikan menggunakan teknik embedding dengan dimensi 100 dan jendela konteks sebesar 10, dimana setiap kata atau token diubah ke dalam bentuk vektor. Dimensi *embedding* ini memungkinkan model untuk menangkap makna semantik dari setiap kata. Setelah *embedding* dilakukan, data input yang sudah berubah menjadi vektor ini diteruskan ke lapisan LSTM yang memiliki 400 unit. Lapisan LSTM ini berfungsi untuk menangkap pola urutan dan dependensi jangka panjang antar kata dalam teks. Keluaran dari lapisan LSTM kemudian diteruskan ke lapisan *dense layer* untuk menyaring dan menginterpretasikan informasi yang telah diekstraksi oleh LSTM. Dimana jumlah target kelas atau label yang ingin diprediksi adalah

sebanyak 46.543 kelas, yang kemungkinan merepresentasikan kata-kata atau token output dalam skenario. Kemudian keluaran dari dense layer dimasukkan ke dalam lapisan *softmax*. Fungsi softmax akan mengubah skor mentah (logit) dari *dense layer* menjadi distribusi probabilitas, di mana jumlah total dari seluruh nilai outputnya adalah 1. Nilai tertinggi dari hasil *softmax* menunjukkan kelas yang paling mungkin menjadi target output, misalnya kata yang akan diprediksi berikutnya dalam urutan teks.

Langkah pertama dalam proses LSTM adalah menentukan informasi mana yang perlu disimpan atau dihapus dalam sel memori melalui *forget gate*. *Forget gate* ini berfungsi untuk memutuskan apakah informasi dari sel sebelumnya harus dipertahankan atau dihapus. Proses ini dilakukan dengan menerapkan fungsi *sigmoid*, fungsi ini mengeluarkan nilai antara 0 dan 1, di mana nilai 0 menunjukkan bahwa informasi harus benar-benar dilupakan, dan nilai 1 berarti bahwa informasi harus dipertahankan (van der Westhuizen & Lasenby, 2018). Berikut persamaan

3.3

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (3.3)$$

Keterangan :

- a) f_t : nilai *forget gate*
- b) σ : fungsi *sigmoid*
- c) W_{fx} : bobot *forget gate* yang didapat selama pelatihan
- d) h_{t-1} : *hidden state* dari waktu sebelumnya
- e) x_t : input pada waktu t
- f) b_f : *forget gate bias*

Langkah selanjutnya adalah memperbarui *cell state* dengan menentukan informasi yang akan disimpan ke dalamnya. Ini dilakukan melalui *input gate layer* pada persamaan (3.4), yang menentukan nilai yang akan diperbarui menggunakan

fungsi aktivasi *sigmoid*, dan *tanh layer*, yang menghasilkan kandidat nilai baru menggunakan fungsi aktivasi *tanh* pada persamaan (3.5).

$$i_t = \sigma(W_i \times [h_{t-1}, x_t] + b_i) \quad (3.4)$$

$$\tilde{C}_t = \tanh(W_C \times [h_{t-1}, x_t] + b_C) \quad (3.5)$$

Keterangan :

- a) i_t : nilai *input gate*
- b) σ : fungsi *sigmoid*
- c) \tilde{C}_t : nilai *memory candidate*
- d) W_C : bobot memori kandidat

Langkah berikutnya adalah memperbarui *cell state* lama menjadi *cell state* baru dengan mengintegrasikan informasi dari forget gate dan input gate. Proses ini dilakukan dengan cara mengalikan nilai forget gate dengan cell state sebelumnya, lalu menambahkan hasilnya dengan nilai dari input gate sesuai dengan persamaan (3.6).

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \quad (3.6)$$

Keterangan :

- a) C_t : nilai *memory candidate*
- b) f_t : nilai *forget gate*
- c) C_{t-1} : *cell state* dari waktu sebelumnya
- d) i_t : nilai *input gate*
- e) \tilde{C}_t : nilai *memory candidate*

Langkah terakhir dalam proses LSTM adalah menentukan *output* dari *cell state*. Ini dilakukan dengan menggunakan *output gate* pada persamaan (3.7), yang menentukan bagian dari *cell state* yang akan menjadi *output*. *Output* ini kemudian dimasukkan ke dalam *tanh layer* untuk diberi fungsi aktivasi *tanh*, dan hasilnya dikalikan dengan *output* dari *output gate* seperti pada persamaan (3.8).

$$o_t = \sigma(W_o \times [h_{t-1}, x_t] + b_o) \quad (3.7)$$

$$h_t = o_t \times \tanh(C_t) \quad (3.8)$$

Keterangan :

- a) o_t : nilai *output gate*
- b) σ : fungsi *sigmoid*
- c) h_t : nilai *hidden state*
- d) C_t : *cell state*

Setelah keluaran dari lapisan LSTM terakhir, data diteruskan ke lapisan *Dense layer*. Lapisan ini bertugas untuk mengonversi representasi fitur yang kompleks dari LSTM menjadi skor mentah (*logits*). Skor ini adalah nilai linear yang belum diterjemahkan ke dalam probabilitas. Pada lapisan *Dense layer*, perhitungan dilakukan dengan menggunakan bobot (*weights*) dan bias (*bias*) pada persamaan 3.9.

$$z = W \cdot h + b \quad (3.9)$$

Keterangan :

- a) z : skor mentah
- b) W : Matriks bobot
- c) h : vektor output
- d) b : *bias*

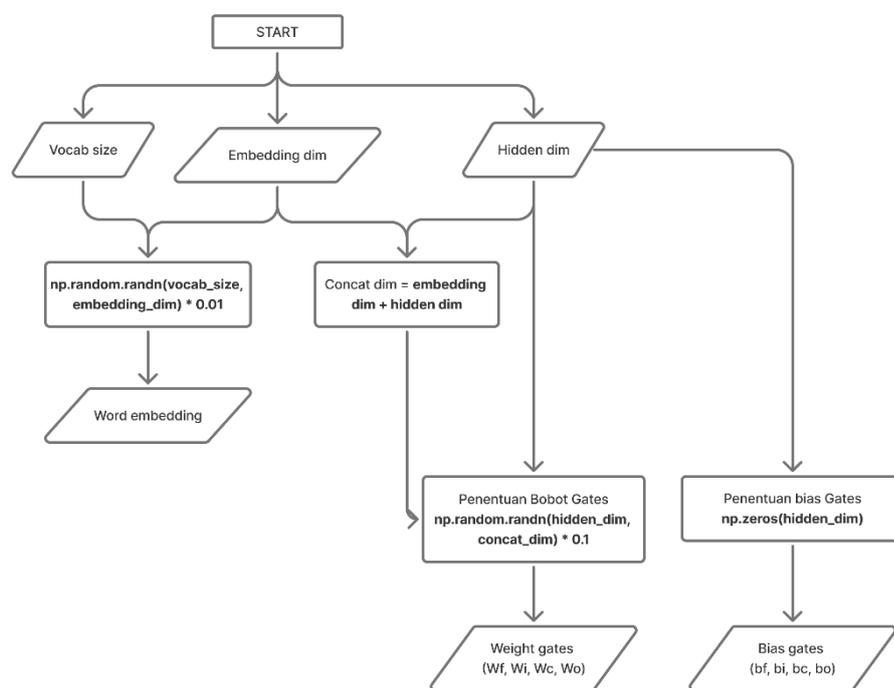
Hasil dari *Dense Layer* diteruskan ke lapisan *Softmax* untuk dikonversi menjadi distribusi probabilitas. *Softmax* memastikan bahwa semua nilai keluaran berada dalam rentang $[0,1]$ dan totalnya berjumlah 1. Probabilitas ini dihitung menggunakan persamaan 3.10.

$$P(y_i) = \frac{e^{z_i}}{\sum_{j=1}^n e^{z_j}} \quad (3.10)$$

Keterangan :

- a) $P(y_i)$: Probabilitas keluaran untuk kelas i
- b) e^{z_i} : Eksponensial dari skor z_i
- c) $\sum_{j=1}^n e^{z_j}$: Jumlah eksponensial dari seluruh skor mentah untuk semua kelas n

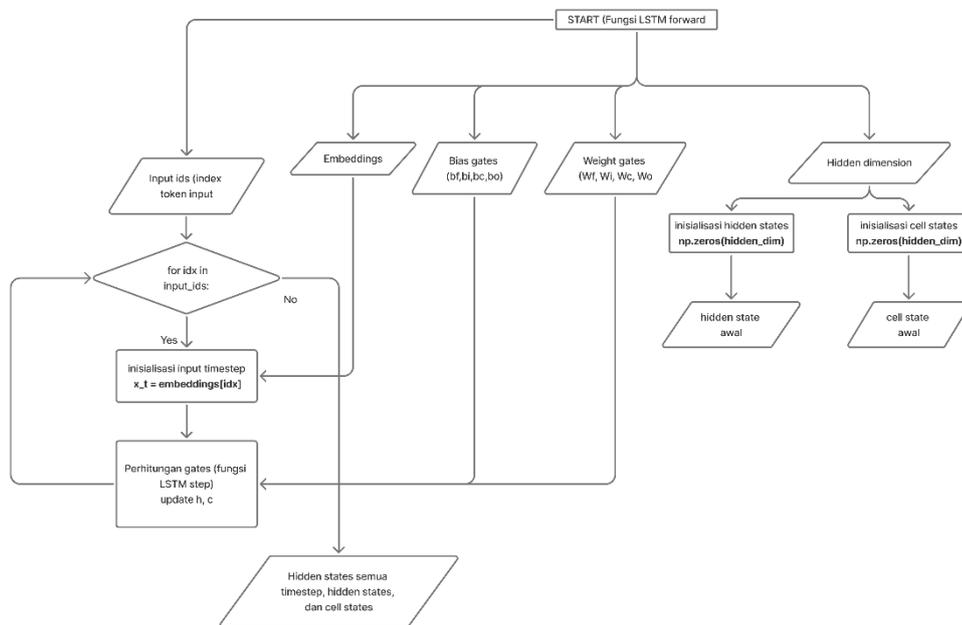
Probabilitas ini menunjukkan seberapa besar kemungkinan setiap kata dipilih sebagai bagian dari ringkasan. Model iteratif memilih kata dengan probabilitas tertinggi sebagai bagian dari keluaran hingga ringkasan lengkap dihasilkan. Pendekatan ini memungkinkan model untuk menangkap konteks teks secara dinamis, menghasilkan ringkasan yang relevan dan bermakna.



Gambar 3.5 *Flowchart* Inisialisasi Bobot LSTM

Gambar 3.5 menampilkan *flowchart* yang menggambarkan proses awal inisialisasi parameter model LSTM secara runtut sebelum digunakan dalam pelatihan. Proses dimulai dari titik "START" yang merupakan pemanggilan fungsi inisialisasi model. Selanjutnya, terdapat tiga parameter utama yang harus ditentukan terlebih dahulu, yaitu *vocab size* (jumlah kosakata unik), *embedding dimension* (ukuran vektor representasi kata), dan *hidden dimension* (jumlah unit

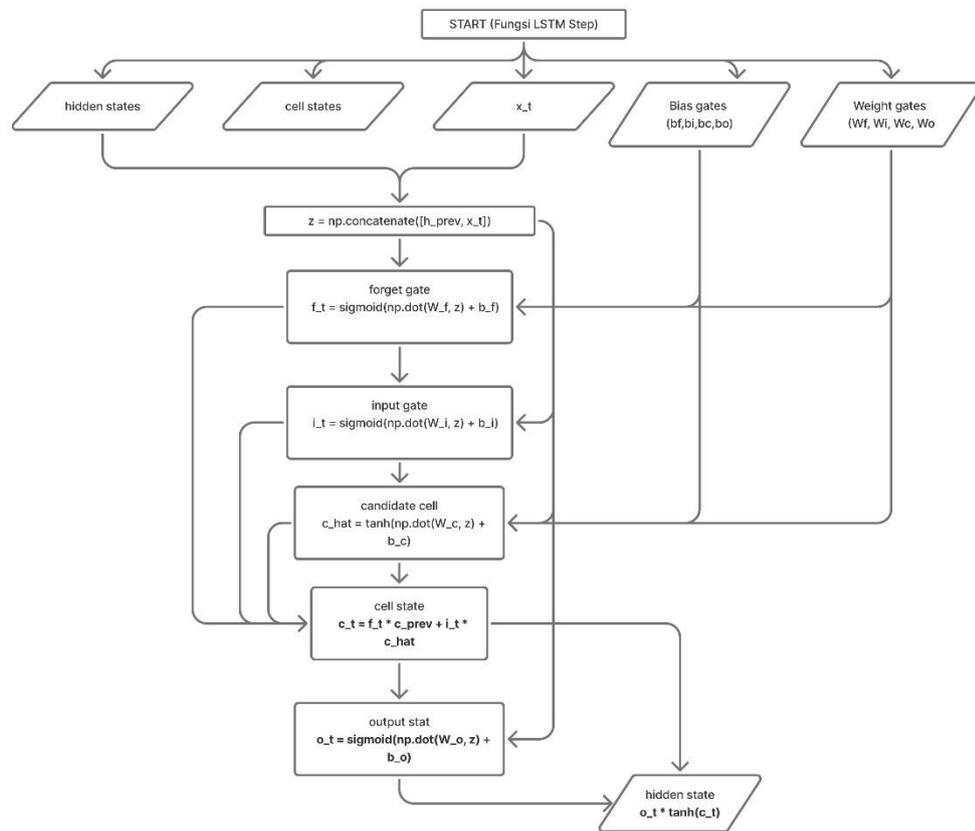
dalam memori tersembunyi LSTM). Ketiga parameter ini menjadi dasar dalam pembentukan struktur model. Dari *vocab size* dan *embedding dim*, dilakukan inisialisasi *word embedding* menggunakan fungsi `np.random.randn (vocab_size, embedding_dim) * 0.01`, yang menghasilkan vektor representasi untuk setiap kata secara acak dan kecil untuk menjaga kestabilan saat pelatihan. Lalu dihitung nilai *concat dim*, yaitu hasil penjumlahan antara *embedding dim* dan *hidden dim* karena input ke setiap gate LSTM merupakan gabungan dari input saat ini dan *hidden state* sebelumnya. Berdasarkan dimensi ini, bobot untuk masing-masing gate (*forget*, *input*, *candidate*, *output*) diinisialisasi menggunakan `np.random.randn(hidden_dim, concat_dim) * 0.1`, sedangkan bias setiap gate diatur dengan `np.zeros(hidden_dim)` untuk memberikan kondisi awal yang netral. Semua parameter inisialisasi ini termasuk bobot (W_f , W_i , W_c , W_o) dan bias (b_f , b_i , b_c , b_o) digunakan pada tahap perhitungan dalam *LSTM Forward* dan *LSTM Step* untuk menghasilkan representasi hidden state dari input teks secara berurutan. Flowchart ini menggambarkan keseluruhan alur logis dan teknis yang saling berkaitan dalam pembentukan dasar model LSTM.



Gambar 3.6 *Flowchart LSTM Forward*

Gambar 3.6 menampilkan *flowchart* proses LSTM Forward yang menggambarkan alur komputasi dari fungsi `lstm_forward` secara sistematis dan terstruktur. Proses ini diawali dari inisialisasi beberapa komponen penting, yaitu *embeddings*, bobot dan bias setiap gate (*forget*, *input*, *cell*, dan *output*), serta dimensi hidden state yang akan digunakan sepanjang proses. *Embedding* digunakan untuk mengonversi indeks token input ke dalam bentuk vektor representasi berdimensi tetap. Secara paralel, bobot untuk setiap gate (W_f , W_i , W_c , W_o) dan bias (b_f , b_i , b_c , b_o) juga diinisialisasi sebagai bagian dari parameter utama model. Kemudian, ditentukan ukuran hidden dimension, yang selanjutnya digunakan untuk menginisialisasi nilai awal dari *hidden state* dan *cell state* dengan vektor nol menggunakan `np.zeros(hidden_dim)`. Setelah seluruh parameter awal dipersiapkan,

proses dilanjutkan ke bagian inti, yaitu pengolahan input sequence token yang telah diubah menjadi indeks (`input_ids`). Token-token ini diproses satu per satu menggunakan perulangan `for idx in input_ids`. Untuk setiap indeks token, dilakukan *ekstraksi embedding* dengan `x_t = embeddings[idx]` sebagai representasi vektor dari token pada timestep tersebut. Kemudian, *embedding* ini dimasukkan ke dalam fungsi `lstm_step`, yaitu fungsi komputasi utama yang menangani perhitungan semua gate dalam LSTM dan memperbarui nilai *hidden state* (`h`) dan *cell state* (`c`) berdasarkan rumus aktivasi *sigmoid* dan *tanh*. Proses ini berlangsung secara berulang untuk setiap timestep hingga seluruh input selesai diproses. Pada akhir proses, diperoleh keluaran berupa *hidden state* pada setiap timestep serta *hidden state* dan *cell state* akhir. Nilai-nilai ini kemudian digunakan dalam tahap selanjutnya, misalnya untuk decoding dalam arsitektur *Seq2Seq* atau klasifikasi. *Flowchart* ini menggambarkan dengan jelas bagaimana setiap bagian dari input diproses secara berurutan dan bagaimana parameter LSTM digunakan untuk mempertahankan informasi temporal sepanjang urutan data.



Gambar 3.7 *Flowchart LSTM Step*

Gambar 3.7 menampilkan *flowchart* dari fungsi `lstm_step` yang merepresentasikan alur proses internal sel LSTM untuk satu timestep input. Proses dimulai dari *START* yang menerima input berupa hidden state sebelumnya (h_{prev}), cell state sebelumnya (c_{prev}), serta input saat ini (x_t) yang diperoleh dari hasil embedding. Semua input ini digabungkan menjadi satu vektor z menggunakan operasi `np.concatenate([h_prev, x_t])`, sebagai bentuk representasi gabungan konteks masa lalu dan input saat ini. Langkah pertama dalam *flowchart* ini adalah perhitungan forget gate sesuai dengan Persamaan (3.3) $f_t = \sigma(W_f \cdot z + b_f)$, di mana fungsi sigmoid menentukan seberapa banyak informasi dari cell state sebelumnya

yang akan dilupakan. Kemudian, input gate dihitung untuk menentukan informasi baru yang akan ditambahkan ke memori, menggunakan Persamaan (3.4) $i_t = \sigma(W_i \cdot z + b_i)$ dan dilanjutkan dengan perhitungan kandidat memori baru atau *candidate cell state* sesuai Persamaan (3.5) $\hat{c}_t = \tanh(W_c \cdot z + b_c)$. Ketiga komponen ini kemudian digunakan untuk memperbarui cell state dengan rumus pada Persamaan (3.6) $c_t = f_t * c_{prev} + i_t * \hat{c}_t$, yang memadukan informasi lama yang dipertahankan dan informasi baru yang ditambahkan. Selanjutnya, output gate dihitung berdasarkan Persamaan (3.7) $o_t = \sigma(W_o \cdot z + b_o)$ yang berfungsi memilih bagian dari cell state yang akan digunakan sebagai output. Output akhir dari LSTM cell pada timestep ini adalah $h_t = o_t * \tanh(c_t)$, sesuai Persamaan (3.8), yang merupakan hidden state baru dan akan diteruskan ke timestep berikutnya. Dengan demikian, flowchart ini secara lengkap menggambarkan bagaimana satu unit LSTM memproses input berdasarkan kombinasi historis dan kontekstual melalui jalur perhitungan gate-gate internal untuk menghasilkan state baru yang relevan.

3.4 Skenario Pengujian

Dalam penelitian ini, dilakukan pengujian menggunakan tiga skenario pembagian data untuk melatih dan menguji model. Skenario 1 menggunakan pembagian data sebesar 90% untuk pelatihan dan 10% untuk pengujian. Skenario 2 menggunakan pembagian 80% data untuk pelatihan dan 20% untuk pengujian, sedangkan skenario 3 menggunakan 70% data untuk pelatihan dan 30% untuk pengujian. Setiap skenario dirancang dengan penyesuaian parameter pelatihan masing-masing, seperti jumlah epoch, hidden units, dan batch size. Tujuan dari

variasi skenario ini adalah untuk menganalisis pengaruh proporsi data pelatihan dan pengujian terhadap performa model secara menyeluruh.

Penelitian oleh (Purnomo et al., 2021), penggunaan pembagian data dengan skenario 80% untuk pelatihan dan 20% untuk pengujian telah diterapkan menggunakan model LSTM dalam tugas peramalan beban listrik jangka pendek. Evaluasi kinerja dilakukan dengan menggunakan metrik RMSE (*Root Mean Square Error*) dan MAPE (*Mean Absolute Percentage Error*). Hasil penelitian menunjukkan bahwa metode LSTM dengan skenario ini mampu menghasilkan nilai evaluasi yang unggul, dengan RMSE sebesar 0,041866 dan MAPE sebesar 1,7387, dibandingkan dengan metode lain seperti ARIMA. Hal ini menunjukkan keunggulan LSTM dalam menangkap pola data yang kompleks dan menghasilkan prediksi yang akurat.

Pendekatan ini merujuk pada penelitian sebelumnya (Sakhare, 2023) yang menggunakan pembagian data sebesar 70% untuk pelatihan dan 30% untuk pengujian, di mana pembagian data yang seimbang antara pelatihan dan pengujian bertujuan untuk menghindari *overfitting* sekaligus memastikan bahwa model dievaluasi pada data yang belum pernah dilihat selama pelatihan. Evaluasi model dilakukan menggunakan metrik ROUGE (ROUGE-N dan ROUGE-L) yang menjadi standar dalam mengukur kualitas rangkuman teks berdasarkan kesamaan n-gram dan keselarasan struktur teks. Penelitian sebelumnya menunjukkan bahwa dengan model LSTM yang dikombinasikan dengan Huber loss function dan Adam optimizer, serta strategi pembagian data seperti ini, hasil pengujian menunjukkan

nilai ROUGE yang unggul. Hasil pengujian tersebut memberikan validasi bahwa pendekatan ini dapat menangkap informasi.

Berdasarkan keberhasilan skenario pembagian data pada penelitian sebelumnya, penelitian ini bertujuan untuk mengujinya dalam konteks evaluasi ROUGE, khususnya pada tugas rangkuman teks otomatis. Dengan menerapkan pembagian data 90%-10%, 80%-20% dan 70% dan 30% yang sama, penelitian ini akan membuktikan apakah metode LSTM juga dapat menghasilkan performa yang baik dalam tugas evaluasi ROUGE-N, seperti yang ditunjukkan pada tabel 3.1. Hasil penelitian ini diharapkan memberikan wawasan baru terkait efektivitas pembagian data dalam tugas evaluasi yang berbeda.

Tabel 3.1 Skenario Pengujian

Skenario	Presentase Data Latih	Presentase Data Uji
Skenario 1	90%	10%
Skenario 2	80%	20%
Skenario 3	70%	30%

3.5 Evaluasi ROUGE

Evaluasi model dalam penelitian ini menggunakan metrik ROUGE (*-Oriented Understudy for Gisting Evaluation*), yang dirancang untuk membandingkan hasil ringkasan otomatis dengan ringkasan referensi buatan manual. ROUGE merupakan metode evaluasi yang telah banyak digunakan di berbagai konferensi besar, termasuk *Document Understanding Conference (DUC)*, untuk menilai efektivitas ringkasan otomatis. Metrik ini mengukur kesesuaian antara ringkasan model dan referensi berdasarkan tingkat kesamaan elemen seperti kata, urutan kata, atau pasangan kata (Song et al., 2019).

ROUGE terdiri dari beberapa jenis, salah satunya adalah ROUGE-N, yang mengukur tingkat kesamaan berbasis *n-gram*. Dalam ROUGE-N, *n-gram* merujuk pada urutan kata sepanjang *n* dalam teks. Dua jenis ROUGE-N yang paling umum digunakan adalah ROUGE-1, mengukur kesamaan pada level unigram ($n = 1$), yaitu kata individu. ROUGE-1 digunakan untuk menilai sejauh mana kata-kata penting dalam ringkasan referensi tercakup dalam ringkasan otomatis. ROUGE-2, mengukur kesamaan pada level bigram ($n = 2$), yaitu pasangan kata berurutan. ROUGE-2 memberikan gambaran tentang kualitas struktur lokal teks dan hubungan antar kata dalam ringkasan.

Dalam penelitian ini, metrik ROUGE-N dipilih untuk mengevaluasi kualitas ringkasan otomatis karena mampu mengukur sejauh mana urutan *n-gram* dari teks referensi tercakup dalam ringkasan. Dengan mempertimbangkan *n-gram*, metrik ini tidak hanya fokus pada kata-kata kunci individu, tetapi juga pola urutan kata yang lebih panjang, sehingga lebih efektif dalam menilai kemampuan model untuk menangkap konteks naratif yang kompleks.

Pada pengukuran ROUGE, tiga metrik utama yang digunakan adalah , *precision*, dan *F1-score*. merupakan Proporsi *n-gram* dalam ringkasan referensi yang juga muncul dalam ringkasan otomatis. *Precision* merupakan Proporsi *n-gram* dalam ringkasan otomatis yang sesuai dengan *n-gram* dalam referensi. Lalu, *F1-score* merupakan rata-rata harmonis dari dan *precision* untuk memberikan keseimbangan antara keduanya. Adapun rumus untuk ROUGE-N adalah:

$$ROUGE(N) = \frac{\sum_{S \in Ref} \sum_{n-gram \in S} Count_{match}(n-gram)}{\sum_{S \in Ref} \sum_{n-gram \in S} Count(n-gram)} \quad (3.11)$$

Keterangan :

- a) n : panjang n-gram
- b) $Count_{match}(n - gram)$: jumlah n-gram yang cocok antara ringkasan otomatis dan referensi
- c) $Count(n - gram)$: jumlah total n-gram dalam referensi.

Dengan pendekatan ini, penggunaan ROUGE-N memberikan evaluasi yang sederhana namun efektif dalam memastikan kualitas ringkasan otomatis yang relevan dan informatif.

Selain ROUGE-N jenis evaluasi lainnya dalam pengukuran ringkasan otomatis adalah ROUGE-L. Berbeda dengan ROUGE-N yang fokus pada kesamaan n-gram, ROUGE-L mengukur kemiripan berdasarkan urutan kata terpanjang yang sama antara ringkasan sistem dan ringkasan referensi. Ukuran ini dikenal sebagai Longest Common Subsequence. LCS yaitu urutan kata terpanjang yang muncul dalam urutan yang sama antara ringkasan sistem dan ringkasan referensi. ROUGE-L mempertimbangkan struktur kalimat dan urutan kata, sehingga mampu menangkap keselarasan dan kelancaran dalam penyusunan ringkasan. Dalam pengukuran ROUGE-L, terdapat tiga metrik utama yang sangat penting, yaitu $precision$, dan $F1-score$. Ketiga metrik ini saling melengkapi dan memberikan gambaran menyeluruh terhadap kualitas ringkasan. $recall$ mengukur seberapa besar bagian dari ringkasan referensi yang berhasil ditangkap oleh ringkasan otomatis, sementara $precision$ menunjukkan seberapa relevan informasi yang disajikan dalam ringkasan sistem terhadap referensi. Kedua metrik ini penting untuk memastikan bahwa ringkasan tidak hanya lengkap, tetapi juga akurat. Untuk menyeimbangkan $recall$ dan $precision$, digunakanlah $F1-score$, yaitu rata-rata harmonis dari keduanya. Dengan demikian, $F1-score$ memberikan penilaian yang lebih

seimbang dalam mengevaluasi performa sistem ringkasan otomatis berdasarkan kesesuaian urutan kata. Adapun rumus untuk ROUGE-L adalah:

$$Recal = \frac{LCS(X, Y)}{m} \quad (3.12)$$

$$Precision = \frac{LCS(X, Y)}{n} \quad (3.13)$$

$$F1 = \frac{(1 + \beta^2) + ROUGE - L_{Precision} \cdot ROUGE - L_{Recall}}{ROUGE - L_{Recall} + \beta^2 \cdot ROUGE - L_{Precision}} \quad (3.14)$$

Keterangan :

- a) X : ringkasan refrensi
- b) Y : ringkasan otomatis
- c) m : panjang ringkasan refrensi
- d) n : panjang ringkasan sistem
- e) β : jumlah total n-gram dalam referensi.

Dalam penelitian ini, metrik ROUGE-N dan ROUGE-L dipilih untuk mengevaluasi kualitas ringkasan otomatis karena keduanya mampu menangkap aspek penting dari kesesuaian antara ringkasan sistem dan ringkasan referensi. ROUGE-N mengukur kemiripan berdasarkan n-gram, sehingga dapat menilai seberapa baik model mempertahankan urutan kata-kata pendek yang bermakna. Sementara itu, ROUGE-L berfokus pada *Longest Common Subsequence* (LCS), yang mencerminkan kemampuan model dalam mempertahankan struktur dan urutan kata secara keseluruhan. Dengan kombinasi keduanya, evaluasi menjadi lebih komprehensif, mencakup baik kesesuaian lokal maupun kelancaran struktur kalimat dalam ringkasan.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Tahapan Uji Coba

Pada tahapan ini, proses pengujian sistem dimulai dengan mempersiapkan data dan melakukan berbagai tahapan penting untuk memastikan model dapat menghasilkan ringkasan teks yang efektif. Tahapan pengujian bertujuan untuk mengevaluasi bagaimana model LSTM dapat melakukan peringkasan teks cerita Islam berbahasa Indonesia dengan akurat, efektif, dan sesuai dengan konteks cerita.

4.1.1 Data Pengujian

Dalam penelitian ini, data pengujian yang digunakan berasal dari proses *crawling* yang dilakukan menggunakan API YouTube. Data yang diambil adalah transkrip teks dari video yang diunggah pada kanal *Hidayah Official* di YouTube, yang berisi cerita-cerita Islami berbahasa Indonesia. Proses *crawling* ini bertujuan untuk mengekstraksi berbagai metadata yang mencakup judul video, tautan (*link*) video, serta hasil transkrip dari narasi yang ada dalam video tersebut.

Sebanyak 300 data transkrip video dipilih dan diambil dari kanal tersebut. Pemilihan data didasarkan pada kriteria bahwa video yang diambil mengandung cerita Islami dengan pesan moral yang jelas dan relevansi dengan nilai-nilai agama Islam. Teks yang diambil mencakup kisah-kisah nabi, sahabat, serta ajaran Islam yang dapat memberikan nilai pembelajaran bagi umat Muslim.

Setelah data transkrip berhasil dikumpulkan melalui proses *crawling*, langkah berikutnya adalah mengumpulkan data hasil ringkasan manusia. Data ini

diperoleh melalui proses validasi oleh seorang validator, yaitu guru agama Islam. Ini berperan penting dalam menghasilkan ringkasan teks yang sesuai dengan makna dan pesan yang terkandung dalam cerita Islam, serta memastikan bahwa hasil ringkasan tetap mencerminkan esensi dan nilai-nilai yang terkandung dalam teks asli. Sebanyak 300 data ringkasan manusia dikumpulkan, jumlah yang setara dengan data transkrip yang diambil melalui *crawling*. Data hasil ringkasan manusia ini bertujuan untuk menyediakan *ground truth* yang akurat, yang nantinya akan digunakan sebagai pembandingan dalam mengevaluasi kualitas ringkasan otomatis yang dihasilkan oleh model LSTM. Dengan melibatkan validator dalam proses ini, diharapkan kualitas ringkasan yang dihasilkan dapat lebih terjamin dan mencerminkan makna serta pesan yang ingin disampaikan dalam cerita Islam secara lebih tepat. Berdasarkan analisis terhadap 300 data ringkasan manusia yang telah dikompilasi, panjang ringkasan bervariasi antara 20 hingga 900 kata. Variasi ini disesuaikan dengan durasi masing-masing video yang diringkas, yaitu antara 2 menit hingga 30 menit. Namun, karena keterbatasan sumber daya (kemampuan komputasi mesin) dalam proses pelatihan dan inferensi model, peneliti menetapkan batas maksimum panjang ringkasan otomatis yang akan dihasilkan model LSTM adalah 500 kata. Pembatasan ini diterapkan untuk menjaga efisiensi komputasi serta menghindari *overfitting* pada model dengan tetap mempertahankan kualitas informasi yang diringkas.

Setelah data transkrip dan hasil ringkasan manusia terkumpul, tahap berikutnya dalam proses persiapan data adalah pembagian dataset menjadi dua bagian utama, yaitu data pelatihan (*training*) dan data pengujian (*testing*).

Pembagian data yang tepat sangat penting untuk memastikan bahwa model dapat dilatih dengan baik dan dievaluasi secara objektif. Pada Tabel 4.1 menunjukkan tiga skenario pembagian data latih dan data uji dengan perbandingan yang berbeda.

Tabel 4.1 Rasio Pembagian Data

Rasio	Data	
	Training	Testing
90:10	270	30
80:20	240	60
70:30	210	90

Setelah data terbagi menjadi data latih dan data uji, tahap berikutnya adalah memproses kedua dataset tersebut melalui tahapan preprocessing. Proses preprocessing ini akan memastikan bahwa data yang digunakan dalam pelatihan dan pengujian siap untuk diolah oleh model LSTM.

4.1.2 Hasil Preprocessing

Pada penelitian ini, tahap preprocessing mencakup cleaning, lowercasing, dan tokenisasi, sebagaimana telah dijelaskan pada BAB 3. Hasil dari tahapan ini menghasilkan dataset yang lebih terstruktur dan siap digunakan.

Data yang diproses terdiri dari dua kolom utama yaitu, *Transcript* yang berisi teks asli dan Ringkasan manusia yang berisi ringkasan manual. Setiap teks pada kolom ini telah diproses untuk menghasilkan versi yang lebih bersih dan terstruktur, yaitu:

1. Kolom *processed_transcriptes* berisi transkrip yang telah dibersihkan dan diproses, siap untuk melatih model.

2. Kolom *processed_summaries* berisi ringkasan yang juga telah diproses, dengan penambahan token *start* di awal dan *end* di akhir setiap ringkasan untuk menandakan batasan pada model.

Tabel 4.2 Contoh Data yang telah diproses

<i>processed transcripts</i>	<i>processed summaries</i>
ternyata ada kunci agar dapat menjawab pertanyaan hidup dengan baik	start kebiasaan menyebut nama allah melalui doa yang dipraktikkan nabi muhammad saw end
katakan di hadapan semua orang bahwa musa telah menjadi nabi	start karun adalah kerabat nabi musa yang diingatkan untuk tidak sombong end

Proses *preprocessing* memastikan bahwa data yang digunakan dalam pelatihan dan pengujian model LSTM berada dalam format yang konsisten dan siap untuk analisis lebih lanjut.

4.2 Hasil Pengujian

Pengujian dilakukan menggunakan metode LSTM untuk menghasilkan ringkasan teks secara abstraktif. Model LSTM ini diuji dengan berbagai rasio pembagian data untuk melihat seberapa baik performanya dalam menghasilkan ringkasan yang akurat. Adapun sejumlah parameter yang ditentukan dalam proses pembentukan model dan diterapkan pada seluruh skenario adalah sebagai berikut:

- a. *Embedding Dim* : 100

Parameter ini menunjukkan dimensi vektor representasi kata yang digunakan dalam proses *embedding*. Setiap kata dalam teks akan diubah menjadi vektor berdimensi 100 melalui proses *Word Embedding* dengan *Word2Vec*, yang bertujuan untuk menangkap makna semantik dari kata-kata dalam bentuk numerik.

b. *Batch Size* : 32

Batch size adalah jumlah sampel data yang diproses sekaligus dalam satu iterasi pelatihan. Dengan *batch size* 32, model akan memperbarui bobotnya setelah melihat 32 data *input* sekaligus. Ukuran ini dipilih untuk menyeimbangkan kecepatan pelatihan dan kestabilan pembelajaran.

c. *Epoch* : 280

Epoch adalah jumlah total putaran seluruh data pelatihan yang digunakan untuk melatih model. Dengan 280 *epoch*, model dilatih menggunakan seluruh dataset sebanyak 280 kali, yang bertujuan untuk meningkatkan akurasi dan kemampuan generalisasi model.

d. *Optimizer* : Adam

Adam (*Adaptive Moment Estimation*) adalah algoritma optimasi yang digunakan untuk mempercepat proses pelatihan model dengan melakukan penyesuaian adaptif terhadap *learning rate*. *Optimizer* ini sangat populer karena efisien dan bekerja dengan baik pada banyak jenis jaringan saraf.

e. *Dropout* : 0.2

Dropout adalah teknik regularisasi yang digunakan untuk mencegah *overfitting*. Dengan nilai 0.2, sebanyak 20% unit dari lapisan tertentu akan secara acak dinonaktifkan selama pelatihan untuk memastikan bahwa model tidak terlalu bergantung pada neuron tertentu.

f. *Hidden Units* : 400

Hidden units pada lapisan LSTM sebanyak 400 menunjukkan kapasitas memori model dalam menyimpan informasi kontekstual dari *input*. Semakin besar

nilai *hidden state*, semakin tinggi pula kemampuan model dalam menangkap pola kompleks dari data, meskipun juga meningkatkan risiko *overfitting* dan beban komputasi.

g. *Activation function : softmax*

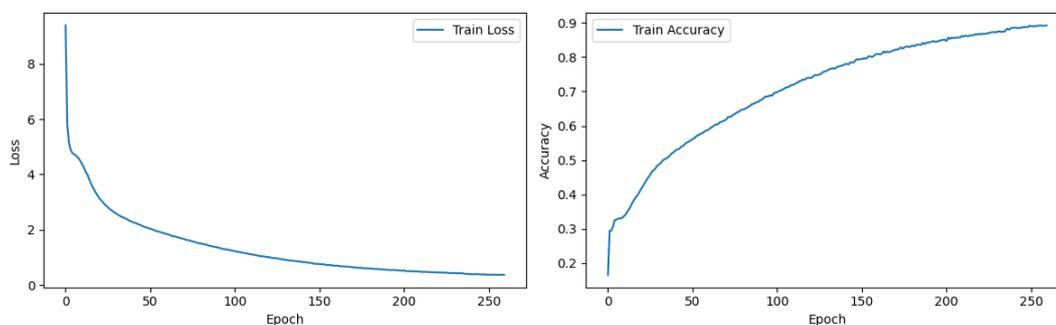
Softmax digunakan sebagai fungsi aktivasi pada *output layer*, terutama ketika model diharapkan mengklasifikasikan token atau kata yang paling sesuai dalam urutan ringkasan. Fungsi ini mengubah *output* menjadi distribusi probabilitas sehingga model dapat memilih kata dengan kemungkinan tertinggi.

h. *Loss : Spars Categorical Crossentropy*

Fungsi *loss* ini digunakan untuk menghitung kesalahan prediksi model dalam kasus klasifikasi dengan label integer. *Sparse Categorical Crossentropy* cocok digunakan dalam kasus *sequence-to-sequence* karena efisien dalam menangani output kelas besar seperti kosakata teks.

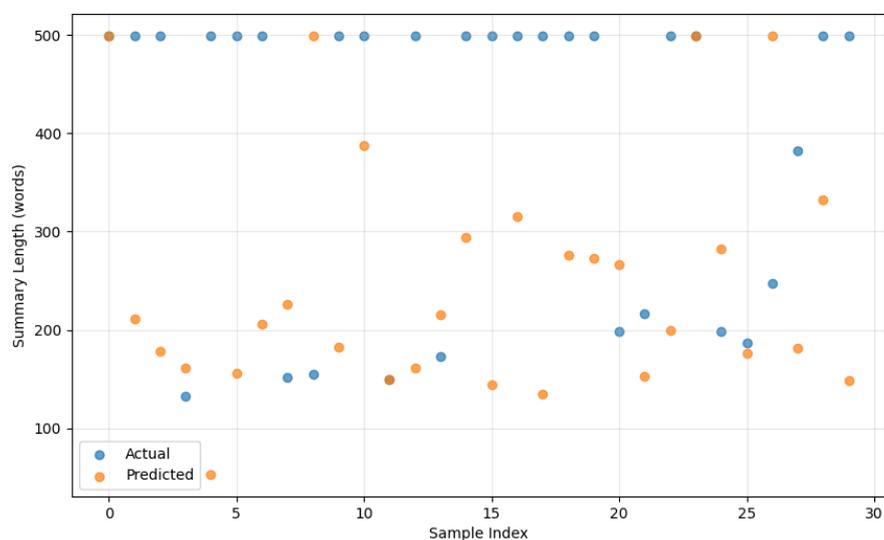
4.2.1 Pengujian Rasio 90:10

Pengujian dengan rasio 90:10 dilakukan dengan membagi dataset menjadi 90% untuk *training* sebanyak 270 data dan 10% untuk *testing* sebanyak 30 data dilakukan pengujian.



Gambar 4.1 Grafik *Train Loss* dan *Train Accuracy* Model LSTM 90:10

Pada Gambar 4.1 menampilkan performa pelatihan model LSTM dalam menyelesaikan tugas peringkasan teks cerita Islam dengan skema pembelajaran selama 228 epoch menggunakan 90% data latih (270 data). Pada grafik sebelah kiri, terlihat bahwa nilai *training loss* menunjukkan penurunan yang signifikan dari awal pelatihan, dimulai dari angka 10,0120 pada *epoch* pertama dan menurun secara konsisten hingga mencapai 0,4879 pada *epoch* ke-228. Sementara itu, grafik di sebelah kanan menunjukkan *training accuracy* yang terus meningkat dari nilai awal 0.1060 (1,06%) hingga mencapai 0.8582 (85,82%) pada akhir *epoch* ke-228. Kecepatan pembelajaran (*learning rate*) mulai pada 0.0010 dan menurun secara adaptif, yang membantu stabilitas pelatihan pada tahap akhir, sehingga menghindari *overfitting*. Hasil ini mencerminkan bahwa model berhasil belajar dari data pelatihan, meskipun ada beberapa fluktuasi pada nilai akurasi di pertengahan epoch.

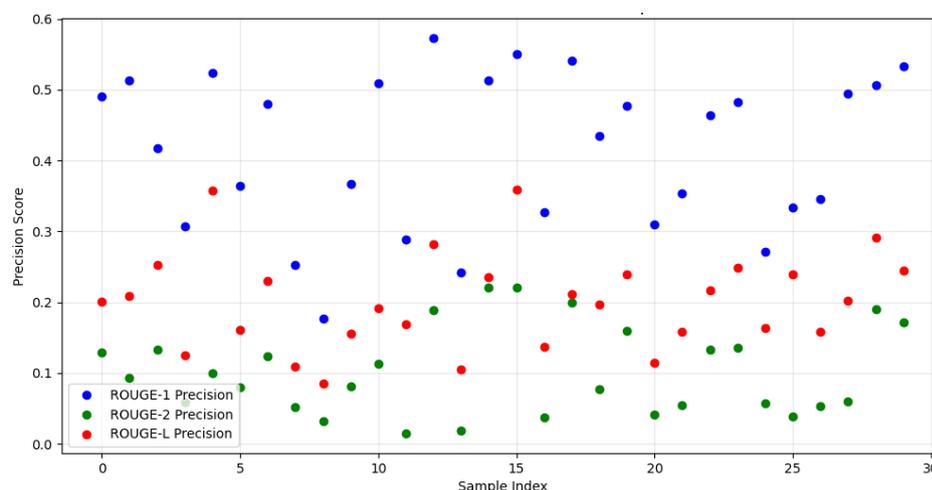


Gambar 4.2 Visualisasi Panjang Ringkasan Aktual dan Prediksi LSTM (90:10)

Pada Gambar 4.2 menyajikan visualisasi perbandingan antara panjang ringkasan *actual* (ringkasan hasil manusia) dan ringkasan *predicted* (hasil prediksi

model LSTM) untuk 30 data uji. Terlihat bahwa ringkasan aktual berwarna biru sedangkan ringkasan predicted warna *orange*. Dimana pada grafik diatas menunjukkan bahwa ringkasan *actual* mendekati panjang maksimum (500 kata), sedangkan ringkasan *predicted* lebih bervariasi untuk panjang kata-nya. Titik-titik yang berdekatan antara kedua warna pada beberapa indeks data, seperti indeks ke-1, 12, 24, dan 26 menunjukkan prediksi yang relatif akurat dalam hal panjang. Namun, penyimpangan signifikan tetap terjadi pada beberapa sampel.

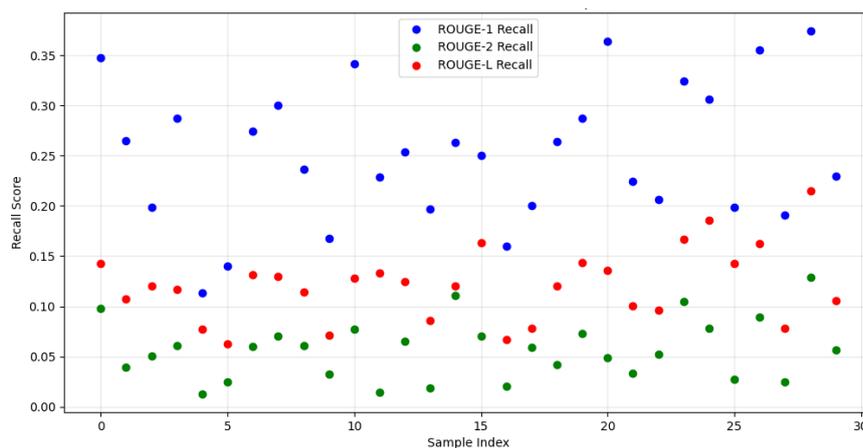
Setelah dilakukan analisis panjang ringkasan, tahapan evaluasi selanjutnya adalah mengukur kualitas isi ringkasan yang dihasilkan oleh model LSTM berdasarkan kesamaan kata dan urutan kalimat terhadap ringkasan manual. Evaluasi ini dilakukan dengan menggunakan metrik ROUGE yang terdiri dari ROUGE-1, ROUGE-2, dan ROUGE-L. Masing-masing metrik dianalisis dalam tiga aspek evaluasi yaitu *precision*, , dan *F1-score*, terhadap 30 data uji.



Gambar 4.3 Grafik ROUGE *Precision* Sampel (Rasio 90:10)

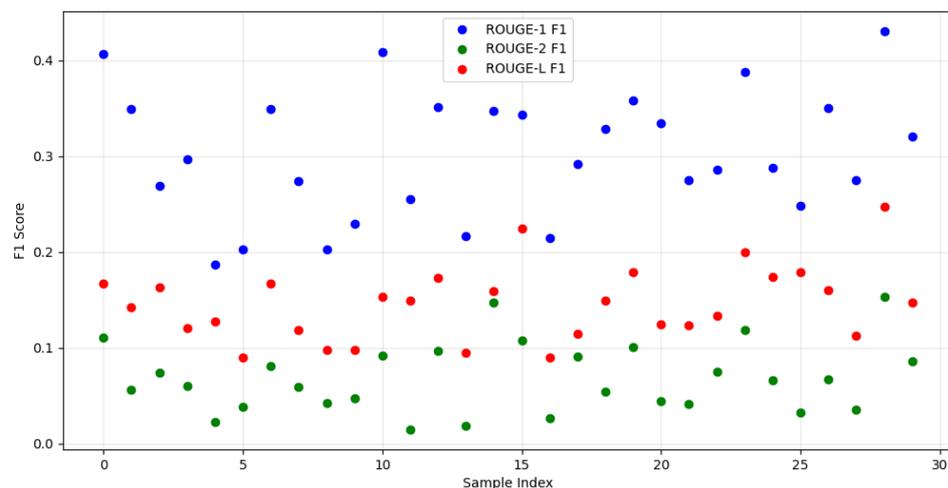
Gambar 4.3 menampilkan nilai *precision* dari ROUGE-1, ROUGE-2, dan ROUGE-L untuk 30 sampel hasil ringkasan, di mana setiap titik menunjukkan skor

precision dari masing-masing metrik pada satu sampel. Terlihat bahwa *ROUGE-1 Precision* (biru) memiliki nilai tertinggi dan paling stabil di antara ketiganya, dengan sebagian besar skor berada di kisaran 0.3 hingga 0.5, yang menunjukkan bahwa sebagian besar kata-kata penting dalam ringkasan model berhasil cocok dengan ringkasan referensi. *ROUGE-2 Precision* (hijau), yang mengukur kecocokan bigram, memiliki nilai paling rendah (rata-rata di bawah 0.2), menandakan bahwa urutan pasangan kata yang sesuai antara ringkasan model dan referensi masih jarang terjadi. Sementara itu, *ROUGE-L Precision* (merah), yang mengukur urutan kata terpanjang yang cocok, menunjukkan performa sedang (sekitar 0.15–0.3), mengindikasikan bahwa model mampu menangkap alur kalimat, tapi belum optimal. Secara keseluruhan, precision yang rendah pada ROUGE-2 dan ROUGE-L menunjukkan bahwa meskipun kata-kata kunci sudah mulai cocok, susunan dan struktur kalimat model masih kurang sesuai dengan referensi manusia.



Gambar 4.4 Grafik ROUGE Sampel (Rasio 90:10)

Gambar 4.4 menunjukkan nilai ROUGE Recall (ROUGE-1, ROUGE-2, dan ROUGE-L) dari 30 sampel ringkasan. ROUGE-1 Recall (biru) memiliki skor tertinggi dibandingkan metrik lainnya, umumnya berada di kisaran 0.15 hingga 0.35, yang mengindikasikan bahwa model mampu mengambil sebagian kata penting dari ringkasan referensi, namun masih belum secara menyeluruh. ROUGE-2 Recall (hijau) menunjukkan performa terendah, sebagian besar hanya mencapai 0.05 hingga 0.1, yang berarti model sangat sedikit menangkap pasangan kata berurutan dari ringkasan referensi. ROUGE-L Recall (merah) cenderung berada pada kisaran 0.1 hingga 0.15, memperlihatkan bahwa urutan kata yang cocok dari referensi masih terbatas. Nilai recall yang relatif rendah di semua metrik ini menunjukkan bahwa ringkasan yang dihasilkan oleh model cenderung tidak lengkap masih banyak informasi dari referensi yang belum berhasil dimasukkan oleh model dan ini menjadi indikator bahwa model perlu ditingkatkan agar lebih menyerap keseluruhan isi penting dari cerita aslinya.



Gambar 4.5 Grafik ROUGE F1 Score Sampel (Rasio 90:10)

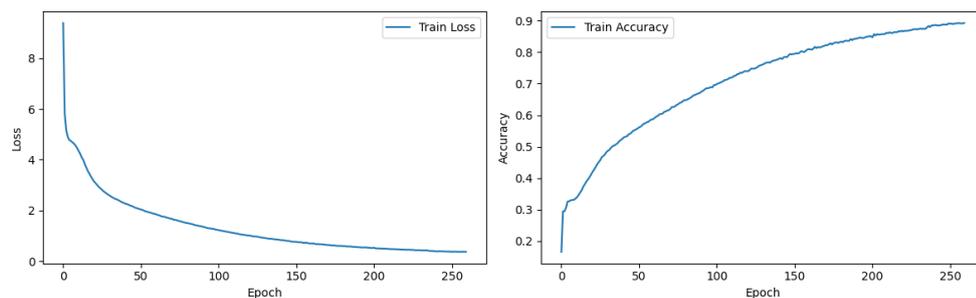
Gambar 4.5 ini menunjukkan nilai ROUGE F1-score (gabungan precision dan recall) untuk 30 sampel, mencakup ROUGE-1 (biru), ROUGE-2 (hijau), dan ROUGE-L (merah). Nilai ROUGE-1 F1 mendominasi, berkisar antara 0.2 hingga 0.4, menandakan bahwa meskipun tidak sempurna, model baik dalam menghasilkan kata-kata penting yang sesuai dengan referensi, meski belum menyeluruh. ROUGE-2 F1 yang sangat rendah (kebanyakan di bawah 0.1) mengindikasikan bahwa kemampuan model dalam menangkap urutan pasangan kata yang benar masih sangat lemah. Sementara ROUGE-L F1, yang berkisar antara 0.1 hingga 0.2, memperlihatkan bahwa struktur urutan kata dari model masih kurang mencerminkan struktur ringkasan referensi. Nilai-nilai ini menunjukkan bahwa performa model masih terbatas dalam menyajikan ringkasan yang akurat dan runtut, di mana kata-kata penting sudah mulai terwakili namun belum lengkap atau terstruktur dengan baik, dan hubungan antar kata (konteks kalimat) masih perlu ditingkatkan untuk menghasilkan ringkasan yang lebih koheren dan bermakna.

Berdasarkan hasil evaluasi menggunakan metrik ROUGE (*Precision*, , dan *F1-score*), performa model menunjukkan bahwa ROUGE-1 consistently memiliki skor tertinggi pada ketiga metrik, menandakan bahwa model mampu menangkap kata-kata penting dari ringkasan referensi, meskipun belum menyeluruh. Namun, nilai secara keseluruhan lebih rendah dibanding *Precision*, yang mengindikasikan bahwa model sering melewatkan informasi penting dari referensi meskipun kata-kata yang dihasilkan sebagian sudah tepat. ROUGE-2 yang sangat rendah di semua metrik menandakan kelemahan model dalam menangkap urutan pasangan kata atau konteks antar kata, yang penting untuk koherensi ringkasan. ROUGE-L berada di

tingkat sedang, menunjukkan bahwa sebagian urutan kalimat sesuai dengan referensi, tetapi belum konsisten. Secara keseluruhan, model masih perlu ditingkatkan dari sisi pemahaman konteks dan kelengkapan informasi agar mampu menghasilkan ringkasan yang lebih utuh, runtut, dan bermakna seperti ringkasan manusia.

4.2.2 Pengujian Rasio 80:20

Pengujian dengan rasio 80:20 dilakukan dengan membagi dataset menjadi 80% untuk *training* sebanyak 240 data dan 20% untuk *testing* sebanyak 60 data untuk pengujian.

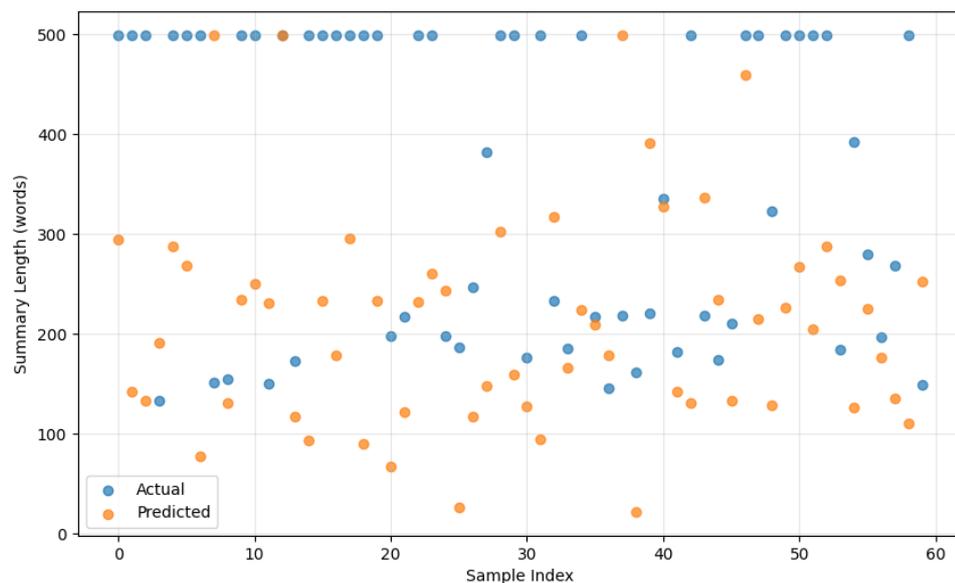


Gambar 4.6 Grafik *Train Loss* dan *Train Accuracy* Model LSTM (Rasio 80:20)

Gambar 4.6 menunjukkan Proses pelatihan model ini dilakukan selama 280 epoch, dengan perkembangan performa yang ditandai oleh peningkatan akurasi dan penurunan *loss*. Pada epoch pertama, model menunjukkan akurasi awal sebesar 0.0896 dengan nilai *loss* sebesar 10.1044, yang mencerminkan ketidaktepatan awal dalam memprediksi keluaran. Selama pelatihan, akurasi model meningkat secara bertahap hingga mencapai 89.44% pada epoch 260, dengan *loss* yang menurun signifikan menjadi 0.3589. Kecepatan pembelajaran (*learning rate*) dimulai pada 0.0010 dan secara bertahap berkurang untuk mencegah *overfitting* serta menjaga

stabilitas model di akhir pelatihan. Meski ada fluktuasi kecil dalam akurasi pada beberapa epoch, tren keseluruhan menunjukkan bahwa model mampu belajar pola dari data pelatihan dengan baik. Hasil ini menegaskan keberhasilan model dalam memahami data secara bertahap, menghasilkan prediksi dengan tingkat kesalahan yang minimal pada tahap akhir pelatihan.

Setelah model menyelesaikan pelatihan pada data latih, evaluasi dilanjutkan pada 60 data uji untuk mengukur sejauh mana model mampu menghasilkan ringkasan dengan panjang yang mendekati referensi. Visualisasi berikut memperlihatkan perbandingan antara panjang ringkasan aktual dengan hasil prediksi model.

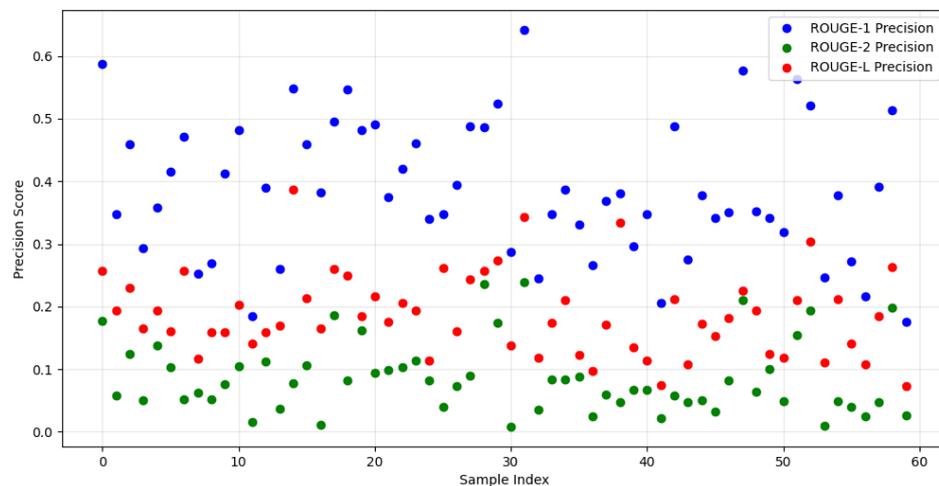


Gambar 4.7 Visualisasi Panjang Ringkasan Aktual dan Prediksi LSTM (Rasio 80:20)

Gambar 4.7 menunjukkan sebaran perbandingan jumlah kata antara ringkasan *actual* dan ringkasan *predicted* model untuk 60 sampel data uji. Titik biru mewakili panjang ringkasan manual (*actual*), sementara titik *orange*

menunjukkan panjang ringkasan hasil *predicted*. Sebagian besar ringkasan aktual berada pada nilai maksimum, yakni 500 kata, sementara prediksi model cenderung bervariasi untuk panjang kata-nya. Titik-titik yang berdekatan antara kedua warna pada beberapa indeks data, seperti indeks ke-13, 36, dan 41 menunjukkan prediksi yang relatif akurat dalam hal panjang kata hasil aktual dan hasil prediksi.

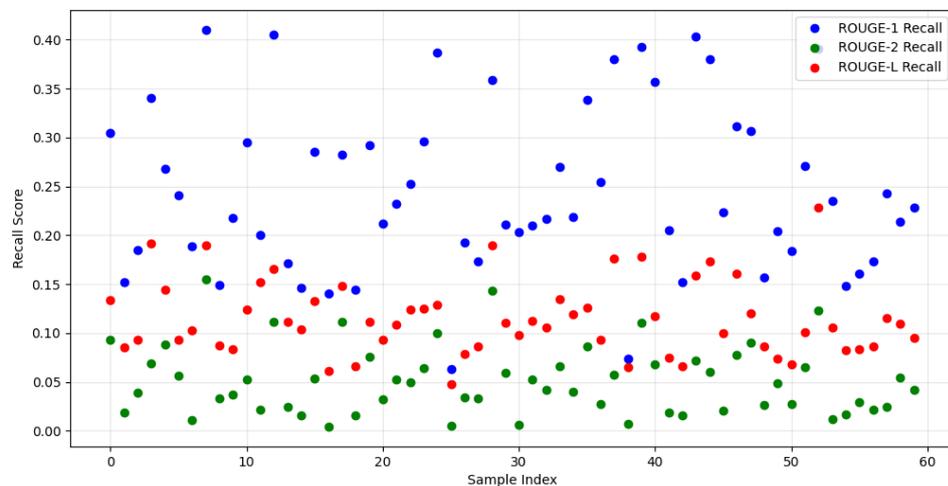
Selanjutnya mengukur kualitas isi ringkasan yang dihasilkan oleh model LSTM berdasarkan kesamaan kata dan urutan kalimat terhadap ringkasan manual. Evaluasi ini dilakukan dengan menggunakan metrik ROUGE yang terdiri dari ROUGE-1, ROUGE-2, dan ROUGE-L. Masing-masing metrik dianalisis dalam tiga aspek evaluasi yaitu *precision*, , dan *F1-score*, terhadap 60 data uji.



Gambar 4.8 Grafik ROUGE *Precision* Sampel (80:20)

Gambar 4.8 menunjukkan nilai ROUGE Precision untuk 60 sampel, meliputi ROUGE-1 (biru), ROUGE-2 (hijau), dan ROUGE-L (merah). ROUGE-1 Precision tampak dominan dengan skor berkisar antara 0.2 hingga 0.6, menunjukkan bahwa model baik dalam menghasilkan kata-kata yang relevan dengan ringkasan referensi. ROUGE-L Precision cenderung berada di kisaran 0.15

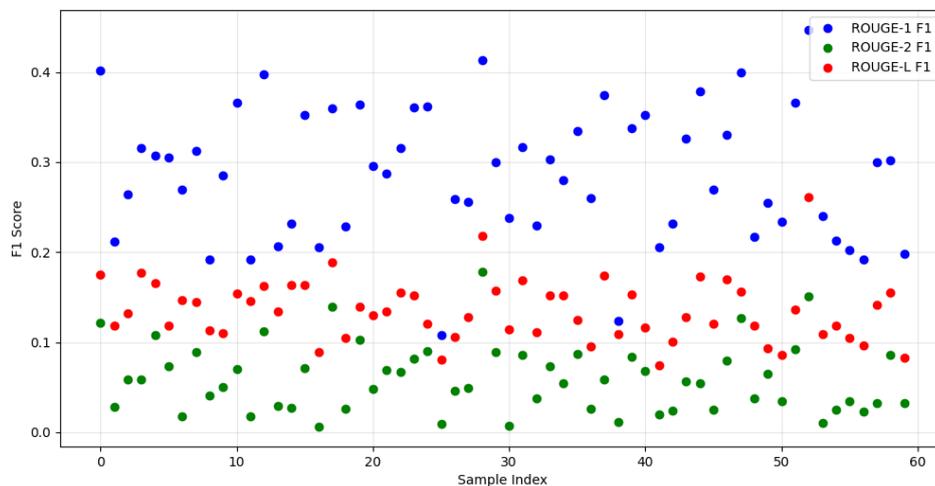
hingga 0.35, yang menandakan bahwa model juga mampu mempertahankan urutan kata yang bermakna meskipun tidak konsisten. Sebaliknya, ROUGE-2 Precision menunjukkan skor paling rendah, mayoritas hanya berada di antara 0.05 hingga 0.15, yang mengindikasikan kelemahan model dalam menyusun bigram (pasangan kata berurutan) yang sesuai dengan referensi. Nilai precision seperti ini berarti bahwa meskipun model mulai mampu memilih kata-kata yang tepat, susunan kata dan konteks antar kata masih sering meleset, terutama dalam membentuk hubungan antar kata dalam kalimat. Hal ini menunjukkan bahwa model belum sepenuhnya memahami struktur kalimat, sehingga perlu penguatan di aspek pemahaman konteks dan kohesi teks.



Gambar 4.9 Grafik ROUGE Sampel (80:20)

Gambar 4.9 memperlihatkan ROUGE Recall untuk 60 sample ROUGE-1 (biru) yang *consistently* menunjukkan performa tertinggi berada sekitar 0.15 hingga 0.4, yang berarti model mampu menangkap sebagian besar kata penting dari ringkasan referensi, meskipun belum menyeluruh. ROUGE-L Recall (merah) menunjukkan skor sedang berada sekitar 0.1 hingga 0.2, menandakan bahwa

sebagian struktur atau urutan kata berhasil ditiru oleh model, namun belum kuat secara konsisten. Sementara itu, ROUGE-2 (hijau) berada di level paling rendah berada antara 0.03 hingga 0.12, mencerminkan kelemahan signifikan model dalam menangkap bigram atau pasangan kata berurutan dari referensi. Nilai-nilai recall seperti ini menunjukkan bahwa model sering kali tidak menyertakan banyak informasi penting dari ringkasan asli, terutama dalam bentuk urutan kata yang benar, sehingga ringkasan model cenderung kurang lengkap atau hanya menangkap sebagian kecil isi dari sumber aslinya. Hal ini mengindikasikan bahwa model masih perlu diperbaiki agar bisa lebih menyerap keseluruhan pesan dan struktur dari cerita yang diringkas.



Gambar 4.10 Grafik ROUGE F1 Score per Sampel (80:20)

Gambar 4.10 menampilkan ROUGE F1-score untuk 60 sampel, dengan ROUGE-1 (biru) memiliki nilai tertinggi, berkisar antara 0.2 hingga 0.4, menunjukkan bahwa model baik dalam menghasilkan kata-kata penting yang relevan dan sebagian sesuai dengan referensi, walaupun belum sepenuhnya akurat atau lengkap. ROUGE-L F1 (merah) cenderung berada di kisaran 0.1 hingga 0.2,

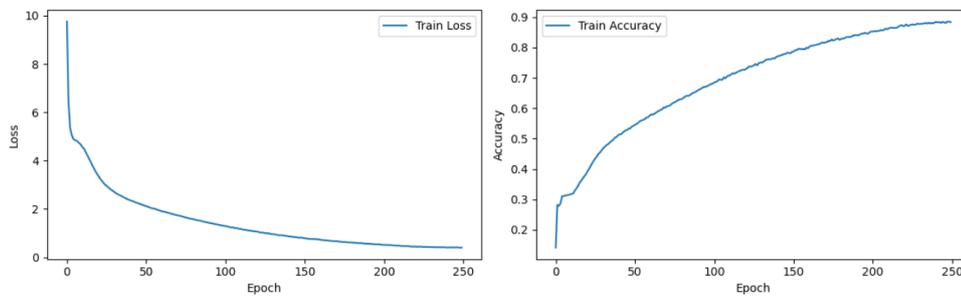
menandakan bahwa model masih memiliki kemampuan terbatas dalam menangkap urutan kata yang bermakna dan selaras dengan struktur ringkasan referensi. Sementara itu, ROUGE-2 F1 (hijau) memiliki skor terendah, sekitar 0.03 hingga 0.15, yang menunjukkan kelemahan model dalam menjaga pasangan kata berurutan (*bigram*), sehingga koherensi kalimat masih rendah. Nilai-nilai ini secara keseluruhan mengindikasikan bahwa model mulai mampu menghasilkan ringkasan yang mengandung kata-kata kunci dari teks asli, namun masih kesulitan dalam membentuk ringkasan yang runtut, lengkap, dan menyerupai struktur logis dari ringkasan manusia. Maka, peningkatan pada pemahaman konteks dan alur kalimat menjadi aspek penting yang perlu diperbaiki.

Berdasarkan hasil pengujian model dengan pembagian data pelatihan 80:20 dan jumlah sampel sebanyak 60 data uji, diperoleh gambaran performa model dari segi ROUGE Precision, Recall, dan F1-score. Secara umum, ROUGE-1 menunjukkan hasil paling tinggi di ketiga metrik, dengan nilai *precision* mencapai sekitar 0.2 hingga 0.6, 0.15 hingga 0.4, dan *F1-score* 0.2 sampai dengan 0.4, yang mengindikasikan bahwa model mampu menangkap kata-kata penting dari referensi. ROUGE-L berada di posisi menengah pada semua metrik berada antara 0.1 hingga 0.3, memperlihatkan kemampuan model dalam mempertahankan sebagian urutan atau struktur kata, namun belum optimal. Sebaliknya, ROUGE-2 consistently memiliki skor terendah di precision, recall, maupun F1 (rata-rata di bawah 0.15), menandakan bahwa model masih sangat lemah dalam membentuk hubungan antar kata secara berurutan atau membangun konteks kalimat yang padu. Secara keseluruhan, model menunjukkan performa yang baik dalam

mengidentifikasi kata kunci, tetapi masih perlu peningkatan dalam aspek kelengkapan informasi dan koherensi kalimat agar ringkasan yang dihasilkan lebih menyerupai ringkasan manusia dari sisi isi dan struktur.

4.2.3 Pengujian Rasio 70:30

Pengujian dengan rasio 70:30 dilakukan dengan membagi dataset menjadi 70% sebanyak 210 data untuk *training* dan 30% untuk *testing* sebanyak 90 data untuk pengujian.

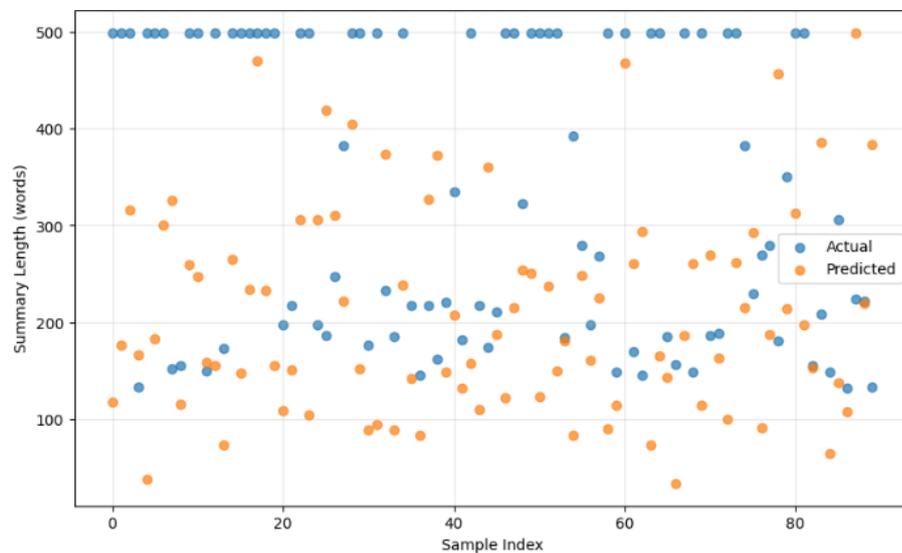


Gambar 4.11 Grafik *Train Loss* dan *Train Accuracy* Model LSTM (Rasio 70:30)

Berdasarkan Gambar 4.11, menunjukkan proses pelatihan model LSTM selama 250 epoch dengan proporsi data latih sebesar 70% (210 data) dari total dataset. grafik sebelah kiri menunjukkan penurunan nilai *loss* secara konsisten dari awal hingga akhir pelatihan sebanyak 250 *epoch*. Pada *epoch* pertama, nilai *loss* masih sangat tinggi yakni 10.3007, namun terus menurun hingga mendekati nilai 0.4131 pada *epoch* ke-250. Penurunan ini menunjukkan bahwa model berhasil mempelajari pola-pola dalam data secara bertahap dan stabil. Sementara itu, grafik sebelah kanan menampilkan peningkatan nilai *Accuracy* pelatihan seiring bertambahnya *epoch*. Akurasi pada awal pelatihan hanya berada di kisaran 0.0760

(7.6%), namun terus meningkat secara signifikan dan mencapai 0.8805 (88%) pada *epoch* terakhir.

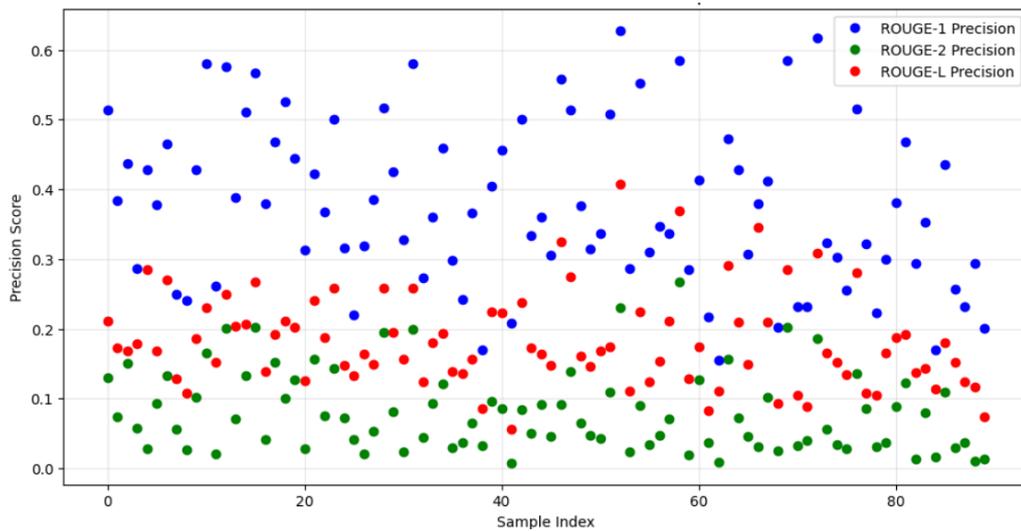
Setelah diperoleh hasil pelatihan model menggunakan arsitektur LSTM sebagaimana ditampilkan pada Gambar 4.15, langkah selanjutnya adalah melakukan evaluasi terhadap performa model dalam memprediksi panjang ringkasan dari data uji.



Gambar 4.12 Visualisasi Panjang Ringkasan Aktual dan Prediksi LSTM (Rasio 70:30)

Gambar 4.12 menunjukkan plot perbandingan antara panjang ringkasan *actual* dan hasil *predicted* oleh model LSTM pada data uji dengan rasio pembagian data 70:30. Titik-titik biru merepresentasikan panjang ringkasan hasil manusia (*aktual*), sedangkan titik-titik *orange* menggambarkan hasil prediksi model (*predicted*). Sebagian besar titik *orange* terlihat berdekatan dengan titik biru, yang menandakan bahwa model berhasil dalam memprediksi panjang ringkasan. Titik-titik yang berdekatan antara kedua warna pada beberapa indeks data, seperti indeks ke-12, 54, 83, dan 89.

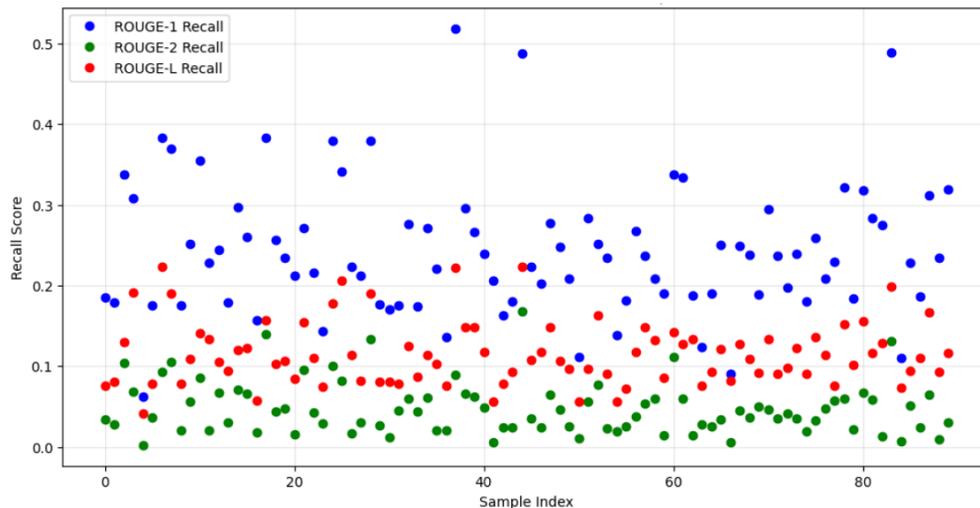
Setelah dilakukan pelatihan dan pengujian menggunakan metode LSTM dengan rasio pembagian data 70:30, langkah berikutnya adalah mengevaluasi kualitas hasil ringkasan yang dihasilkan oleh model.



Gambar 4.13 Grafik ROUGE *Precision* Sampel (Rasio 70:30)

Gambar 4.13 menampilkan sebaran skor *precision* dari masing-masing metrik ROUGE terhadap 90 data uji. Terlihat bahwa ROUGE-1 *Precision* (biru) memiliki performa tertinggi, dengan skor yang bervariasi antara 0.2 hingga lebih dari 0.6, menunjukkan bahwa model mampu menghasilkan kata-kata penting yang cocok dengan referensi, meskipun terdapat fluktuasi antar sampel. ROUGE-L *Precision* (merah) berada di tingkat menengah, berkisar antara 0.15 hingga 0.35, yang menunjukkan bahwa model mampu menangkap sebagian struktur atau urutan kata yang relevan dalam ringkasan referensi, walaupun belum konsisten. Sementara itu, ROUGE-2 *Precision* (hijau) tetap yang terendah, umumnya hanya mencapai 0.05 hingga 0.2, yang mencerminkan kelemahan model dalam menghasilkan bigram atau pasangan kata yang sesuai, sehingga koherensi antarkata masih rendah.

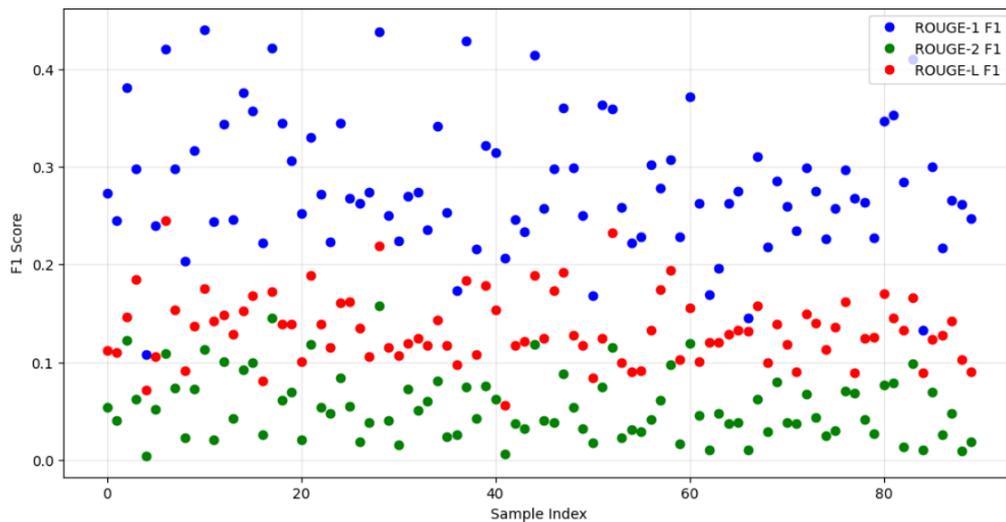
Nilai-nilai ini menunjukkan bahwa meskipun model akurat dalam memilih kata-kata penting (terutama secara individual), kemampuan dalam menjaga keterkaitan kata dalam konteks kalimat dan struktur teks masih perlu ditingkatkan agar kualitas ringkasan lebih informatif dan koheren.



Gambar 4.14 Grafik ROUGE Sampel (Rasio 70:30)

Gambar 4.14 menunjukkan grafik ROUGE. Skor *ROUGE-1* (biru) kembali menunjukkan performa tertinggi, dengan skor yang bervariasi antara 0.2 hingga lebih dari 0.6, menunjukkan bahwa model mampu menghasilkan kata-kata penting yang cocok dengan referensi, meskipun terdapat fluktuasi antar sampel. *ROUGE-L Precision* (merah) berada di tingkat menengah, berkisar antara 0.15 hingga 0.35, yang menunjukkan bahwa model mampu menangkap sebagian struktur atau urutan kata yang relevan dalam ringkasan referensi, walaupun belum konsisten. Sementara itu, *ROUGE-2 Precision* (hijau) tetap yang terendah, umumnya hanya mencapai 0.05 hingga 0.2, yang mencerminkan kelemahan model dalam menghasilkan bigram atau pasangan kata yang sesuai, sehingga koherensi antarkata masih rendah.

Nilai-nilai ini menunjukkan bahwa meskipun model akurat dalam memilih kata-kata penting (terutama secara individual), kemampuan dalam menjaga keterkaitan kata dalam konteks kalimat dan struktur teks masih perlu ditingkatkan agar kualitas ringkasan lebih informatif dan koheren.



Gambar 4.15 Grafik ROUGE F1 Score Sampel (Rasio 70:30)

Gambar 4.15 menggambarkan skor *F1* dari masing-masing metrik ROUGE pada 90 data uji. Skor ROUGE-1 F1 (biru) memiliki performa tertinggi dengan skor berkisar antara 0.2 hingga 0.4, menunjukkan bahwa model baik dalam menyeimbangkan antara precision dan recall dalam mengenali kata-kata penting dari ringkasan referensi, meskipun masih terdapat variasi antar sampel. ROUGE-L F1 (merah) cenderung berada di tengah, dengan nilai antara 0.1 hingga 0.25, yang berarti model memiliki kemampuan sedang dalam mempertahankan struktur kalimat yang sesuai dengan ringkasan manusia, namun belum optimal. Sementara itu, ROUGE-2 F1 (hijau) menunjukkan skor paling rendah, berkisar 0.02 hingga 0.12, mencerminkan bahwa model masih sangat terbatas dalam menghasilkan

bigram atau pasangan kata yang terstruktur dan bermakna secara berurutan. Nilai F1-score ini menunjukkan bahwa meskipun model sudah bisa mengambil kata penting, ia masih belum mampu membentuk ringkasan yang runtut, lengkap, dan menyerupai gaya bahasa alami manusia. Maka, penguatan pada aspek konteks, alur naratif, dan struktur kalimat sangat diperlukan untuk meningkatkan kualitas ringkasan secara keseluruhan.

4.3 Pembahasan

Tabel 4.3 Hasil Nilai ROUGE Seluruh Rasio

Skenario	Metriks	Precision	Recall	F1
SKENARIO 1 (90:10)	ROUGE-1	0.4147	0.2516	0.3027
	ROUGE-2	0.1022	0.0568	0.0684
	ROUGE-L	0.2017	0.1209	0.1459
SKENARIO 2 (80:20)	ROUGE-1	0.3871	0.2450	0.2863
	ROUGE-2	0.0858	0.0511	0.0595
	ROUGE-L	0.1874	0.1147	0.1353
SKENARIO 3 (70:30)	ROUGE-1	0.3762	0.2434	0.2805
	ROUGE-2	0.0812	0.0478	0.0550
	ROUGE-L	0.1823	0.1159	0.1341

Tabel 4.3 menunjukkan bahwa skenario 90:10 menghasilkan nilai ROUGE tertinggi dibandingkan skenario 80:20 dan 70:30. Pada ketiga jenis metrik ROUGE, yaitu ROUGE-1, ROUGE-2, dan ROUGE-L, serta pada ketiga aspek evaluasi yaitu *Precision*, *Recall*, dan *F1-score*. Dengan nilai ROUGE-1 *Precision* sebesar 0.4147, yang merupakan nilai tertinggi dibandingkan skenario lainnya. Hal ini menunjukkan bahwa ketika data pelatihan lebih banyak (90%), model memiliki kemampuan yang lebih baik dalam menghasilkan kata-kata yang relevan dengan ringkasan referensi. ROUGE-1 juga tertinggi di skenario ini, yaitu 0.2516, diikuti oleh *F1-score* sebesar 0.3027, menandakan keseimbangan yang baik antara ketepatan dan kelengkapan informasi yang dihasilkan model. Pada metrik ROUGE-2 dan ROUGE-L, Skenario

l juga mencatat skor tertinggi, meskipun nilainya masih tergolong rendah (misalnya ROUGE-2 F1 hanya 0.0684), menunjukkan bahwa model masih kesulitan membentuk hubungan antarkata atau struktur kalimat yang kompleks. Seiring penurunan proporsi data pelatihan pada Skenario 2 (80:20) dan Skenario 3 (70:30), terlihat bahwa semua nilai metrik mengalami penurunan, baik pada precision, recall, maupun F1-score. Hal ini mengindikasikan bahwa semakin sedikit data pelatihan yang digunakan, maka kemampuan model dalam memahami konteks, menangkap informasi penting, dan menyusun ringkasan yang bermakna semakin menurun. Penurunan ini dikarenakan karena jumlah data yang lebih sedikit tidak memberikan keragaman pola yang cukup kepada model untuk dipelajari, sehingga model cenderung *overfitting* terhadap data pelatihan yang terbatas. Oleh karena itu, dapat disimpulkan bahwa ketersediaan data pelatihan yang lebih banyak berkontribusi signifikan terhadap peningkatan kualitas ringkasan yang dihasilkan oleh model.

Sebelum dilakukan pengujian akhir terhadap tiga skenario pembagian data, peneliti terlebih dahulu melakukan eksplorasi preprocessing dengan lima tahapan, yaitu *cleaning*, *lowercasing*, *tokenisasi*, *stopword removal*, dan *stemming*. Pendekatan ini diharapkan mampu menyederhanakan data dan mengurangi redundansi linguistik agar proses pelatihan model menjadi lebih efisien. Akan tetapi, setelah dilakukan pelatihan dan evaluasi, ditemukan beberapa kekurangan signifikan baik dari sisi teknis maupun kualitas hasil ringkasan. Dari hasil pelatihan model, akurasi training memang mencapai 87,09%, namun kualitas ringkasan yang dihasilkan sangat jauh dari memadai secara naratif. Hasil prediksi menunjukkan

panjang ringkasan yang menyimpang tajam dari versi manual sebagian besar terlalu pendek, tidak utuh, dan kehilangan unsur naratif serta nilai religius. Berikut tabel perbandingan hasil yang dihasilkan oleh preprocessing dengan 5 tahapan dan 3 tahapan tanpa *stemming* dan *stopword removal*.

Tabel 4.4 Hasil Perbandingan Perbedaan Preprocessing Terhadap Ringkasan

Hasil Ringkasan model LSTM (5 Tahapan)	Hasil Ringkasan model LSTM (3 Tahapan tanpa stemming dan stopword)
umar aslam ganti naik unta bahkan umar perintah aslam baca alquran ia tuntun unta jalan penuh berkah tiba palestina umar tampil sederhana pakai tambal buat abu ubaidah amr bin ash kejut coba beri pakai lebih layak umar tolak sampai pesan sangat kuat mulia umat islam datang agama bukan harta kendaranya pakai mewah umar tegas siapa cari mulia dari islam allah hina hendak masuk baitul maqdis gilir umar tuntun unta aslam tunggang aslam sempat tolak khawatir sangka khalifah duduk tempat umar sikeras aslam tetap atas unta...	sementara aslam menunggangnya aslam sempat menolak karena khawatir disangka khalifah oleh penduduk setempat namun umar bersikeras agar aslam tetap di atas unta sesampainya di sana seorang pendeta nasrani bertanya siapa pemimpin kaum muslimin dan dijawab bahwa umar adalah orang yang menuntun unta sang pendeta takjub karena pemimpin muslim ternyata sangat sederhana dan bahkan disebutkan dalam kitab injil sebagai pemimpin yang adil umar kemudian bertemu dengan para pemimpin nasrani dan menerima penyerahan kunci kota dengan syarat keselamatan warga kristen dan tempat ibadah mereka...

Berdasarkan tabel 4.4 perbandingan hasil ringkasan, pendekatan lima tahapan preprocessing menghasilkan ringkasan yang cenderung terpotong, minim struktur, dan kehilangan makna naratif akibat hilangnya kata-kata penting karena proses *stemming* dan *stopword removal*. Sebaliknya, ringkasan dari tiga tahapan preprocessing (tanpa *stemming* dan *stopword*) tampak lebih utuh, naratif, dan mampu menangkap konteks cerita secara lengkap, termasuk tokoh, peristiwa, dan nilai-nilai yang terkandung. Oleh karena itu, peneliti memutuskan untuk menggunakan tiga tahapan preprocessing sebagai pendekatan utama karena lebih

efektif dalam mempertahankan kualitas semantik dan struktur cerita Islami (Santosa et al., 2022).

Berdasarkan hasil pengujian tiga skenario pada tabel 4.3 rasio pembagian data (90:10, 80:20, dan 70:30), dapat disimpulkan bahwa perbedaan rasio memang memberikan pengaruh terhadap akurasi pelatihan model, namun tidak secara langsung mencerminkan kualitas hasil ringkasan yang dihasilkan. Skenario 90:10 menghasilkan akurasi pelatihan tertinggi sebesar 85.82% , diikuti oleh 80:20 sebesar 89.44%, dan 70:30 sebesar 88%. Hal ini tidak sepenuhnya menunjukkan bahwa proporsi data latih yang lebih kecil akan selalu menghasilkan akurasi lebih tinggi, karena peningkatan ini dapat dipengaruhi oleh variasi data, kondisi *overfitting*, dan kinerja *early stopping*. Oleh karena itu, rasio pembagian data memberikan pengaruh terhadap proses pelatihan, namun tidak selalu linier terhadap kualitas hasil ringkasan. Namun, dari segi evaluasi kualitas ringkasan menggunakan metrik ROUGE, skenario 90:10 justru menunjukkan performa terbaik dalam menghasilkan ringkasan yang relevan, utuh, dan mendekati ringkasan manual. Hal ini mengindikasikan bahwa akurasi pelatihan yang tinggi tidak selalu berbanding lurus dengan kemampuan model dalam memahami dan merekonstruksi narasi teks, sehingga pemilihan rasio data tidak hanya mempertimbangkan akurasi, tetapi juga keseimbangan antara performa pelatihan dan kualitas semantik hasil keluaran model. Dengan demikian, rasio 90:10 dipilih sebagai konfigurasi optimal dalam penelitian ini.

Tabel 4.5 Perbandingan Hasil Peneliti dan Peneliti Sebelumnya

Peneliti	ROUGE-1	ROUGE-2	ROUGE-L
(Regundwar et al., 2024)	25.64	8.77	24.05
(Karuna et al., 2023)	27.71	1.82	27.27
Peneliti	30.27	6.84	14.59

Berdasarkan perbandingan Tabel 4.5 dengan penelitian-penelitian sebelumnya, model yang digunakan oleh peneliti dalam studi ini menghasilkan skor ROUGE-1 sebesar 30.27, ROUGE-2 sebesar 6.84, dan ROUGE-L sebesar 14.59 pada dataset cerita Islam berbahasa Indonesia. Meskipun nilai metrik masih lebih rendah dibandingkan penelitian internasional, namun skor ROUGE-1 menghasilkan hasil yang lebih tinggi dibanding pada peneliti sebelumnya, serta menunjukkan performa yang stabil pada teks naratif keagamaan. Hal ini menunjukkan bahwa model LSTM dengan preprocessing sederhana (*cleaning, lowercasing, tokenisasi*) efektif untuk kasus teks naratif Islami, meskipun masih memiliki ruang untuk peningkatan performa, terutama pada struktur kalimat dan pasangan kata (ROUGE-2). Selain itu, skor ROUGE-L yang masih rendah juga mengindikasikan bahwa model belum optimal dalam mempertahankan urutan dan keselarasan antar kalimat yang mencerminkan struktur teks asli secara keseluruhan.

Tabel 4. 6 Hasil ringkasan teks pada 3 skenario

Skenario	Actual Summary	Predicted
90:10	orang miskin neraka jahim memiliki api yang sangat besar yang diperuntukkan bagi orang-orang yang menentang ayat-ayat Allah dan berusaha melemahkan kemauan orang untuk berimmaneraka jahanam yang sangat gelap dan penuh dengan api yang sangat panas juga memiliki tempat yang dalam dan luas penghuninya adalah	air di ringan timbangan dalam setiap aktivitas bisa menjaga agar tidak lagi selalu berlindung kepada Tuhan agar tidak terjerumus ke dalam neraka yang mengerikan seperti jahannam sa'ir saqar huthomah jahim ladza dan wail masing-masing disiapkan untuk jenis dosa tertentu neraka memiliki kedalaman dan panas luar

	<p>orang-orang yang melakukan dosa besar orang-orang munafik dan mereka yang mengikuti setan neraka jahanam digambarkan memiliki batu yang dilemparkan selama tahun yang akhirnya baru mencapai dasarnya dalam neraka ini para penghuni akan merasakan siksaan yang sangat besar di mana mereka dipasang rantai belenggu di leher mereka kemudian diseret untuk dibakar di dalam api yang sangat panas makanan dan minuman di neraka tidak akan memberikan rasa kenyang atau kepuasan penghuni neraka hanya</p>	<p>biasa dijaga oleh malaikat yang tidak akan durhaka kepada Allah semoga kita diselamatkan dari siksaan yang sangat pedih tempat neraka yang mengerikan dalam neraka akan berlangsung di akhirat meskipun mereka yang menolak untuk tidak bertobat dan neraka akan menerima balasan yang sangat berat di neraka meskipun mereka akan tidak dimasukkan ke tempat yang neraka di surga ini adalah pengingat bagi umat Islam untuk selalu memperbaiki diri dan menjalankan perintah Allah dengan ikhlas..</p>
80:20	<p>diberikan air yang mendidih dan nanah yang hanya akan menambah penderitaan mereka buah zakum yang merupakan makanan penghuni neraka digambarkan sangat buruk dan panas yang jika jatuh ke dunia bisa menyebabkan kerusakan yang sangat besar pakaian penghuni neraka pun akan terbuat dari api neraka itu sendiri yang semakin menambah penderitaan mereka siksaan di neraka tidak hanya terbatas pada api dan panas tetapi juga dapat berupa siksaan yang lebih berat salah satu contoh siksaan yang paling ringan di neraka adalah Abu Thalib yang akan memakai sandal yang terbuat dari api neraka sandal itu sangat panas hingga otaknya mendidih namun meski siksaan yang dialami oleh orang-orang di neraka itu sangat berat ada</p>	<p>yang ringan timbangan amal baiknya dan mereka akan dibakar dengan api panas yang sangat dalam sementara itu neraka jahanam adalah tempat yang paling buruk dan dasar yang sangat dalam kehidupan umat Islam tidak hanya sebagai peringatan bagi penghuni neraka untuk menjaga iman dan amal perbuatannya dengan mengikuti perintah Allah untuk menghindari siksaan neraka yang mengerikan ada juga cerita tentang neraka dan neraka adalah pengingat bagi kita semua untuk tetap teguh dalam ajaran Islam dan tidak mengikuti ajaran yang salah untuk menghindari neraka neraka adalah tempat yang sangat penting dan kehidupan akhirat yang penuh dengan amal perbuatan kita akan dihitung di dunia ini dan</p>

	yang paling ringan di antara mereka yang masih merasa	akhirat kelak akan dihitung di akhirat kelak...
70:30	bahwa siksaannya lebih pedih dari orang lain meskipun neraka selalu digambarkan dengan api yang sangat panas ada juga bentuk siksaan lainnya seperti siksaan dingin dalam alquran neraka juga digambarkan dengan adanya air yang sangat dingin yang menyiksa penghuni neraka air ini disebut gasak yang menurut beberapa ulama adalah air busuk yang sangat dingin dan tak tertahankan gambaran tentang luasnya neraka juga sangat mengerikan neraka jahanam digambarkan memiliki kedalaman yang sangat dalam bahkan jika sebuah batu dilemparkan ke dalam neraka batu tersebut membutuhkan waktu hingga tahun untuk mencapai dasarnya di dalam neraka setiap penghuni akan merasakan siksaan yang berbedabeda tergantung pada dosa yang telah mereka lakukan neraka bukanlah tempat yang hanya diperuntukkan bagi orang-orang kafir saja tetapi juga bagi mereka yang melakukan dosa besar munafik serta yang mengabaikan ajaran agama neraka menjadi tempat yang disediakan oleh allah untuk memberikan balasan kepada mereka yang telah mendurhakainya dan menolak perintahnya sebagai umat manusia kita diajarkan untuk selalu	dalam islam pandangan mengenai urutan masuk neraka dan surga bervariasi di kalangan ulama beberapa ulama berpendapat bahwa sebagian besar manusia bahkan yang beriman mungkin akan mengalami penyucian di neraka sebelum akhirnya diampuni dan masuk surga sementara yang lain berpendapat bahwa dosa besar atau kekafiran yang belum diampuni allah dapat menyebabkan seseorang masuk neraka terlebih dahulu sebagian besar umat islam berkeyakinan bahwa kiamat tidak dapat keluar dari tempat yang allah di mana allah akan masuk surga sementara yang kafir yang masuk neraka adalah amal perbuatan mereka yang mengikuti dajjal akan datang azab yang akan datang sebagian orang yang beriman dan beramal saleh serta tunduk kepada ajarannya tanpa menyombongkan diri dan merasa istimewa hanya karena identitas kelompok atau keturunan

	<p>berusaha menjaga amal baik dan menghindari perbuatan dosa agar kita tidak termasuk dalam golongan yang menjadi penghuni nerakakeberadaan neraka dalam islam tidak hanya sekedar sebagai tempat balasan bagi orang-orang yang berbuat jahat tetapi juga sebagai peringatan bagi umat manusia untuk senantiasa berbuat baik taat kepada allah dan mengikuti ajaran yang telah disampaikan oleh para nabi dan rasulnya setiap orang yang beriman dan beramal shaleh akan mendapatkan ganjaran yang setimpal di akhirat sedangkan mereka yang durhaka akan mendapatkan balasan yang setimpal di neraka oleh karena itu kita harus selalu menjaga amal kita dan menghindari perbuatan yang dapat membawa kita ke dalam neraka agar kita bisa memperoleh surga yang telah disediakan bagi hambahambanya yang beriman dan beramal baik</p>	
--	--	--

Tabel 4.6 menunjukkan hasil ringkasan otomatis yang dihasilkan oleh model LSTM pada tiga skenario pembagian data latih dan uji, yaitu 90:10, 80:20, dan 70:30. Setiap baris dalam tabel menampilkan perbandingan antara ringkasan asli (*actual summary*) yang ditulis oleh manusia dan ringkasan yang dihasilkan oleh model (*predicted summary*) berdasarkan data uji dari masing-masing skenario. Hasil tersebut memberikan gambaran bahwa semakin besar proporsi data pelatihan, seperti pada skenario 90:10, maka ringkasan yang dihasilkan cenderung lebih

informatif, koheren, dan mendekati ringkasan referensi. Sebaliknya, pada skenario 70:30, ringkasan yang dihasilkan tampak lebih umum, kurang spesifik, dan kehilangan beberapa informasi penting dari teks aslinya. Hal ini menunjukkan bahwa jumlah data pelatihan sangat memengaruhi kemampuan model dalam memahami konteks cerita dan menghasilkan ringkasan yang relevan. Dengan demikian, penggunaan proporsi data pelatihan yang lebih besar dapat meningkatkan kualitas ringkasan yang dihasilkan oleh model, baik dari segi isi, struktur kalimat, maupun kesesuaian makna.

4.4 Integrasi Islam

Dalam Islam, ilmu pengetahuan memiliki kedudukan yang tinggi, sebagaimana disebutkan dalam Surat *Al-Baqarah* ayat 269:

يُؤْتِي الْحِكْمَةَ مَنْ يَشَاءُ وَمَنْ يُؤْتَ الْحِكْمَةَ فَقَدْ أُوتِيَ خَيْرًا كَثِيرًا وَمَا يَذَّكَّرُ إِلَّا أُولُو الْأَلْبَابِ

"Dia (Allah) menganugerahkan hikmah kepada siapa yang Dia kehendaki. Siapa yang dianugerahi hikmah, sungguh dia telah dianugerahi kebaikan yang banyak. Tidak ada yang dapat mengambil pelajaran (darinya), kecuali ululalbab." (Q.S. *Al-Baqarah*: 269)

Ayat ini menunjukkan bahwa hikmah merupakan anugerah besar yang diberikan kepada hamba yang dipilih oleh Allah. Hikmah tidak hanya berarti kecerdasan intelektual, tetapi juga pemahaman yang mendalam, ketepatan dalam mengambil keputusan, serta kemampuan mengolah ilmu untuk kebaikan.

Menurut Tafsir Ibnu Katsir, Al-Qur'an Surat *Al-Baqarah* ayat 269 yang berbunyi "يُؤْتِي الْحِكْمَةَ مَنْ يَشَاءُ" dijelaskan bahwa hikmah adalah pemahaman yang mendalam terhadap Al-Qur'an dan *As-Sunnah* yang Allah berikan kepada siapa

yang Dia kehendaki. Hikmah mencakup ilmu tentang ayat-ayat yang nasikh dan mansukh, muhkam dan mutasyabih, halal dan haram, serta berbagai cabang ilmu syar'i lainnya. Para ulama memiliki pendapat berbeda dalam menjelaskan makna hikmah. Ibnu Abbas menyebutkan bahwa hikmah adalah pengetahuan tentang Al-Qur'an. Mujahid mengatakan bahwa hikmah adalah ketepatan dalam ucapan. Abu Aliyah menjelaskan bahwa hikmah berarti rasa takut kepada Allah. Ibrahim an-Nakha'i memaknai hikmah sebagai pemahaman. Zaid bin Aslam mengartikan hikmah sebagai akal. Imam Malik menegaskan bahwa hikmah adalah pemahaman tentang agama yang ditanamkan Allah dalam hati hamba-Nya sebagai bentuk rahmat dan karunia. Sementara itu As-Suddi berpendapat bahwa hikmah adalah kenabian. Jumhur ulama menyimpulkan bahwa hikmah tidak terbatas pada kenabian saja tetapi mencakup segala bentuk pemahaman akal ketakwaan dan keilmuan yang Allah anugerahkan kepada hamba-Nya sebagai derajat tinggi dalam agama (Ishaq, 2008).

Penelitian ini berusaha menerapkan prinsip tersebut dalam bidang teknologi, khususnya dalam pemrosesan teks cerita Islam berbahasa Indonesia. Dengan menggunakan algoritma LSTM, penelitian ini bertujuan untuk mengukur tingkat akurasi dalam merangkum cerita agar tetap bermakna, relevan, dan sesuai dengan konteks aslinya. Hal ini sejalan dengan konsep hikmah dalam Islam, di mana teknologi tidak hanya digunakan untuk otomatisasi, tetapi juga untuk memastikan bahwa informasi yang dihasilkan tetap memiliki nilai dan manfaat. Dengan demikian, penelitian ini diharapkan dapat menjadi kontribusi dalam

pengembangan ilmu yang mendukung penyebaran cerita Islam secara lebih efektif dan berkualitas.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini berhasil mengembangkan sistem peringkasan teks otomatis menggunakan algoritma LSTM untuk cerita-cerita Islam. Hasil penelitian menunjukkan bahwa model mampu menghasilkan ringkasan yang relevan dengan isi cerita aslinya, terutama dalam mengenali kata-kata penting. Namun, kemampuan dalam menyusun kalimat yang lengkap dan runtut masih perlu ditingkatkan. Kualitas ringkasan yang dihasilkan juga dipengaruhi oleh rasio pembagian data pelatihan dan pengujian. Semakin besar proporsi data pelatihan, model cenderung bekerja lebih baik dalam memahami dan menyusun isi ringkasan.

Berdasarkan hasil pengujian tiga skenario pembagian data pelatihan dan pengujian (90:10, 80:20, dan 70:30), dapat disimpulkan bahwa algoritma LSTM mampu menghasilkan ringkasan teks cerita Islam dengan tingkat akurasi yang bervariasi, tergantung pada jumlah data pelatihan yang digunakan. Nilai akurasi pada pelatihan tertinggi diperoleh pada skenario 80:20 sebesar 89,44%. Hal ini menunjukkan bahwa meskipun model dinilai akurat secara pelatihan, kemampuan dalam menghasilkan ringkasan yang bermakna tetap perlu dilihat dari hasil evaluasi isi ringkasan itu sendiri. Hasil evaluasi menggunakan metrik ROUGE menunjukkan bahwa skenario 90:10 memberikan performa terbaik. Pada metrik ROUGE-1, diperoleh nilai *precision* sebesar 0,4147, sebesar 0,2516, dan *F1-score* sebesar 0,3027. Pada metrik ROUGE-2, nilai *precision*, , dan *F1-score* masing-masing

tercatat sebesar 0,1022; 0,0568; dan 0,0684. Sedangkan pada ROUGE-L, *precision* mencapai 0,2017, sebesar 0,1209, dan F1-score sebesar 0,1459. Nilai-nilai ini menunjukkan bahwa model LSTM mampu menghasilkan ringkasan yang relevan dengan teks asli, terutama dalam memilih kata-kata penting, meskipun masih memiliki keterbatasan dalam menyusun kalimat yang utuh dan runtut. Dengan demikian, dapat disimpulkan bahwa algoritma LSTM memiliki tingkat akurasi yang baik dalam menghasilkan peringkasan teks cerita Islam, terutama ketika didukung oleh rasio data pelatihan yang optimal, seperti pada skenario 90:10.

5.2 Saran

Dalam penelitian ini, disarankan agar pengembangan selanjutnya memperluas cakupan dataset yang digunakan untuk meningkatkan performa dan kemampuan generalisasi model. Penambahan variasi dalam tema cerita, panjang teks, dan sumber video dari berbagai platform selain *YouTube* dapat membantu model untuk memahami konteks yang lebih beragam. Dengan memperkaya jenis dan isi data, model diharapkan dapat menghindari bias, menangkap lebih banyak pola naratif, serta menghasilkan ringkasan yang lebih tepat dan bermakna, terutama dalam konteks cerita-cerita Islami. Selain itu, sumber daya komputasi menjadi salah satu kendala utama dalam proses pelatihan dan evaluasi model, khususnya pada saat mengolah dataset yang besar atau menjalankan model deep learning seperti LSTM dengan parameter yang kompleks. Waktu pemrosesan yang lama dan keterbatasan memori menjadi tantangan dalam eksperimen ini. Oleh karena itu, disarankan agar penelitian selanjutnya dapat memanfaatkan fasilitas komputasi yang lebih

mumpuni, seperti GPU atau *cloud computing*, agar proses pelatihan berjalan lebih efisien dan optimal.

DAFTAR PUSTAKA

- AL-Hafidh, D. F., Rozi, I. F., & Putri, I. K. (2022). Peringkasan Teks Otomatis pada Portal Berita Olahraga menggunakan metode Maximum Marginal Relevance. *Jurnal Informatika Polinema*, 8(3), 21–30. <https://doi.org/10.33795/jip.v8i3.519>
- Aldi, M. W. P., Jondri, & Aditsania, A. (2018). Analisis dan Implementasi Long Short Term Memory Neural Network Untuk Prediksi Harga Bitcoin. *E-Proceeding of Engineering*, 5(2), 3548–3555.
- Alfhi Saputra, M. (2021). Peringkasan Teks Otomatis Bahasa Indonesia secara Abstraktif Menggunakan Metode Long Short-Term Memory. *E-Proceeding of Engineering : Vol.8, No.2 April 2021* |, 8(2), 3474–3488.
- Ammar, A. N., & Suyanto, S. (2020). Peringkasan Teks Ekstraktif Menggunakan Binary Firefly Algorithm. *Indonesia Journal on Computing (Indo-JC)*, 5(2), 31–42. <https://doi.org/10.21108/indojc.2020.5.2.440>
- Aufa, M. J., & Qoiriah, A. (2023). Analisis Sentimen Pengguna Platform Belajar Online Coursera menggunakan Random Forest dengan Metode Ekstraksi Fitur Word2vec. *Journal of Informatics and Computer Science (JINACS)*, 04, 244–255. <https://doi.org/10.26740/jinacs.v4n02.p244-255>
- Bastian Sianturi, T., Cholissodin, I., & Yudistira, N. (2023). Penerapan Algoritma Long Short-Term Memory (LSTM) berbasis Multi Fungsi Aktivasi Terbobot dalam Prediksi Harga Ethereum. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 7(3), 1101–1107. <http://j-ptiik.ub.ac.id>
- Bayat, S., & Isik, G. (2023). Assessing the Efficacy of Lstm, Transformer, and Rnn Architectures in Text Summarization. *International Conference on Applied Engineering and Natural Sciences*, 1(1), 813–820. <https://doi.org/10.59287/icaens.1099>
- Das, A., & Roy, S. (2024). *Bodo Language Text Summarization Using Lstm*. 1–16.
- Hanifaa, U; Ardianti, S; Putri, G. (2023). *Implementasi Nilai Keislaman Pada Kisah Nabi Ismail As. dalam Pendidikan*. 1(2), 56–65.
- Hughes, C., Costley, J., & Lange, C. (2018). The effects of self-regulated learning and cognitive load on beginning to watch and completing video lectures at a cyber-university. *Interactive Technology and Smart Education*, 15(3), 220–237. <https://doi.org/10.1108/ITSE-03-2018-0018>
- Ishaq, A. S. A. bin M. bin A. bin. (2008). *Tafsir Ibnu Katsir Jilid 8.pdf*.

- Karim, A. A., Usama, M., Ibrahim, M. A., Hatem, Y., Wael, W., Mazrua, A. T., El-Monayer, G. K., Elbanhaw, M., Foad, K., & Moawad, I. F. (2024). Arabic Abstractive Summarization Using the Multilingual T5 Model. *2024 6th International Conference on Computing and Informatics (ICCI)*, 223–228. <https://doi.org/10.1109/ICCI61671.2024.10485135>
- Kartha, R. S., Agal, S., Odedra, N. D., Nanda, C. S. K., Rao, V. S., Kuthe, A. M., & Taloba, A. I. (2024). NLP-Based Automatic Summarization using Bidirectional Encoder Representations from Transformers-Long Short Term Memory Hybrid Model: Enhancing Text Compression. *International Journal of Advanced Computer Science and Applications*, 15(5), 1223–1236. <https://doi.org/10.14569/IJACSA.2024.01505124>
- Karuna, G., Akshith, M., Dinesh, P. S., Vardhan, B. V., Bisht, Y. S., & Narsaiah, M. N. (2023). Automated Abstractive Text Summarization using Deep Learning. *E3S Web of Conferences*, 430. <https://doi.org/10.1051/e3sconf/202343001021>
- Khan, A., Ahmad, I., Ali, Q. E., Khan, M. O., & Sana, U. (2023). *Automated Abstractive Text Summarization Using Multidimensional Long Short-Term Memory*. 21(02), 1–11.
- Kong, Y., Wang, Z., Nie, Y., Zhou, T., Zohren, S., Liang, Y., Sun, P., & Wen, Q. (2024). *Unlocking the Power of LSTM for Long Term Time Series Forecasting*. <http://arxiv.org/abs/2408.10006>
- Kroon, A. C., Trilling, D., & Raats, T. (2021). Guilty by Association: Using Word Embeddings to Measure Ethnic Stereotypes in News Coverage. *Journalism and Mass Communication Quarterly*, 98(2), 451–477. <https://doi.org/10.1177/1077699020932304>
- LiteLSTM Architecture for Deep Recurrent Neural Networks*. (2022). <https://doi.org/10.1109/iscas48785.2022.9937585>
- Nofitayanti, N., & Fitri, A. (2022). Model Pembelajaran Kisah Qurani. *Journal on Education*, 4(4), 1420–1430. <https://doi.org/10.31004/joe.v4i4.1835>
- Nurdin, A., Sidik, I., Najmy, & Ummah, M. S. (2019). New Media dan Komunikasi Dakwah. In *Sustainability (Switzerland)* (Vol. 11, Issue 1). http://scioteca.caf.com/bitstream/handle/123456789/1091/RED2017-Eng-8ene.pdf?sequence=12&isAllowed=y%0Ahttp://dx.doi.org/10.1016/j.regscuirbeco.2008.06.005%0Ahttps://www.researchgate.net/publication/305320484_SISTEM_PEMBETUNGAN_TERPUSAT_STRATEGI_MELESTARI
- Purnomo, H., Suyono, H., & Hasanah, R. N. (2021). Peramalan Beban Jangka Pendek Sistem Kelistrikan Kota Batu Menggunakan Deep Learning Long Short-Term Memory. *Transmisi*, 23(3), 97–102.

<https://doi.org/10.14710/transmisi.23.3.97-102>

- Quoc, V. N., Thanh, H. Le, & Minh, T. L. (2020). Abstractive Text Summarization Using LSTMs with Rich Features. *Communications in Computer and Information Science*, 1215 CCIS, 28–40. https://doi.org/10.1007/978-981-15-6168-9_3
- Raihanunnisa, F., Arhami, M., & Hidayat, R. (2023). Pendekatan Hybrid Pada Sistem Peringkat Teks Artikel Berita Bahasa Inggris Menggunakan Natural Language Processing. *Telematika Mkom*, 15(2), 86–92. <https://journal.budiluhur.ac.id/index.php/telematika/>
- Regundwar, S., Bhagwat, R., Bhosale, S., Chougale, R., & Abbu, S. (2024). *INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING Sequence-to-Sequence Abstractive Text Summarization Model for Headline Generation with Attention*. 12(3), 842–851.
- Sakhare, D. Y. (2023). A Sequence-to-Sequence Text Summarization Using Long Short-Term Memory Based Neural Approach. *International Journal of Intelligent Engineering and Systems*, 16(2), 142–151. <https://doi.org/10.22266/ijies2023.0430.12>
- Sandag, G. A. (2023). Perbandingan Algoritma LSTM Untuk Analisis Sentimen Pengguna Twitter Terhadap Layanan Rumah Sakit Saat Pandemi Covid-19. *TeKa : Jurnal Teknologi Informasi Dan Komunikasi (Journal Information and Communication Technology) (e-Journal)*, 13(01), 31–40. <https://doi.org/10.36342/teika.v13i01.3063>
- Saraçoğlu, T. N. (2022). Islamic Studies in the Context of Digital Humanities. *Islam Tetkikleri Dergisi*, 12(2), 835–891. <https://doi.org/10.26650/iuitd.2022.1123964>
- Setiadi B, S. (2017). Peringkasan Kumpulan Berita Secara Otomatis Menggunakan Metode Maximum Marginal Relevance. *Prosiding Seminar Nasional Teknologi Dan Informatika*, 123, 153–160.
- Sirohi, N. K., Bansal, M., & Rajan, D. S. N. R. (2021). *Text Summarization Approaches Using Machine Learning & LSTM*. 11(4), 5010–5026. <https://doi.org/10.47059/REVISTAGEINTEC.V11I4.2526>
- Song, S., Huang, H., & Ruan, T. (2019). Abstractive text summarization using LSTM-CNN based deep learning. *Multimedia Tools and Applications*, 78(1), 857–875. <https://doi.org/10.1007/s11042-018-5749-3>
- Sreenivasulu, G., Thulasi Chitra, N., Sujatha, B., & Venu Madhav, K. (2022). Text Summarization Using Natural Language Processing. *Lecture Notes in Networks and Systems*, 191, 653–663. <https://doi.org/10.1007/978-981-16->

- Tawong, K., Pholsukkarn, P., Noawaroongroj, P., & Siriborvornratanakul, T. (2024). Economic news using LSTM and GRU models for text summarization in deep learning. *Journal of Data, Information and Management*, 6(1), 29–39. <https://doi.org/10.1007/s42488-023-00111-y>
- Utomo, M. S., Wibowo, J. S., & Wahyudi, E. N. (2022). Text Summarization Pada Artikel Berita Menggunakan Vector Space Model Dan Cosine Similarity. *Jurnal Dinamika Informatika*, 14(1), 11–24. <https://doi.org/10.35315/informatika.v14i1.9163>
- van der Westhuizen, J., & Lasenby, J. (2018). *The unreasonable effectiveness of the forget gate*. 1–15. <http://arxiv.org/abs/1804.04849>
- Wiranda, L., & Sadikin, M. (2019). Penerapan Long Short Term Memory pada Data Time Series untuk Memprediksi Penjualan Produk PT. Metiska Farma. *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, 8(3), 184–196.
- Zakawaly, G., Hayatin, N., & Nastiti, V. R. S. (2023). Improvisasi Algoritma Dijkstra Pada Peringkasan Teks Otomatis Untuk Artikel Politik. *Jurnal Repositor*, 5(2), 709–716. <https://repositor.umm.ac.id/index.php/repositor/article/view/1437>
- Zaman, F., Afzal, M., Teh, P. S., Sarwar, R., Kamiran, F., & Naif, R. (2024). *Intelligent Abstractive Summarization of Scholarly Publications with Transfer Learning*. 3(3).
- Zhao, H., Zhang, W., Huang, M., Feng, S., & Wu, Y. (2023). A Multi-Granularity Heterogeneous Graph for Extractive Text Summarization. *Electronics (Switzerland)*, 12(10), 1–12. <https://doi.org/10.3390/electronics12102184>

LAMPIRAN

LAMPIRAN

Lampiran 1 Hasil Preprocessing

Tahapan	Sebelum	Sesudah
Cleaning Data	Seorang Muslim, selalu membantu saudaranya dalam kebaikan dan menjauhi keburukan untuk meraih ridha Allah.	Seorang Muslim selalu membantu saudaranya dalam kebaikan dan menjauhi keburukan untuk meraih ridha Allah
Lowercasing	Seorang Muslim selalu membantu saudaranya dalam kebaikan dan menjauhi keburukan untuk meraih ridha Allah.	seorang muslim selalu membantu saudaranya dalam kebaikan dan menjauhi keburukan untuk meraih ridha allah
Tokenisasi	seorang muslim selalu membantu saudaranya dalam kebaikan dan menjauhi keburukan untuk meraih ridha allah.	["seorang", "muslim", "selalu", "membantu", "saudaranya", "dalam", "kebaikan", "dan", "menjauhi", "keburukan", "untuk", "meraih", "ridha", "allah"]

Lampiran 2 Ekstraksi fitur dengan Word2Vec

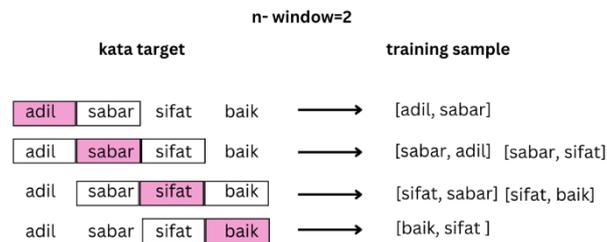
Pada tahapan pertama, diterapkan teknik *one-hot encoding* pada lapisan input, di mana setiap kolom diisi dengan nilai 0, kecuali satu kolom yang bernilai 1 untuk mewakili kata tertentu, sehingga menghasilkan vektor biner. Berikut adalah ilustrasi penerapan one-hot encoding pada input kata yang dapat dilihat pada Tabel.

Contoh Kata: “adil sabar sifat baik”.

id	adil	sabar	sifat	baik
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

Vektor yang merepresentasikan sebuah kata pada arsitektur model skip-gram digunakan untuk memprediksi kata-kata di sekitarnya ketika diberikan satu

kata sebagai input, dengan tujuan untuk mempelajari hubungan semantik antar kata. Kata-kata yang berada di sekitar kata input tersebut disebut sebagai *window*, dan jumlah kata dalam window ini dapat ditentukan dengan parameter n_window . Sebagai contoh, ketika jumlah window = 2, hal ini digambarkan pada Gambar dibawah.



Selanjutnya agar model dapat memahami hubungan antar kata, model harus mampu memberikan probabilitas tertinggi pada kata-kata di sekitarnya yang akan diprediksi. Probabilitas tertinggi diperoleh menggunakan fungsi softmax. Sebagai contoh diambil training sample yaitu (adil, sabar). Model diminta untuk memprediksi kata 'sabar' ketika diberikan kata input kata 'adil', dengan asumsi jumlah fitur adalah 4, maka dimensi matriks bobot W dan W' masing-masing adalah 4×4 dan 4×4 , dengan nilai bobot yang ditentukan sebagai berikut.

$$W = \begin{bmatrix} 0.2 & 0.3 & 0.5 & 0.8 \\ 0.4 & 0.5 & 0.2 & 0.4 \\ 0.6 & 0.4 & 0.1 & 0.3 \\ 0.4 & 0.3 & 0.5 & 0.2 \end{bmatrix}$$

$$W' = \begin{bmatrix} 0.4 & 0.5 & 0.1 & 0.3 \\ 0.2 & 0.3 & 0.5 & 0.8 \\ 0.3 & 0.2 & 0.2 & 0.7 \\ 0.7 & 0.2 & 0.5 & 0.1 \end{bmatrix}$$

Pada tabel diatas, input vector kata adil adalah $X = [1 0 0 0]$, sehingga masing-masing hasil pembobotan (W) untuk input ke hidden layer dan pembobotan (W') untuk hidden layer ke output.

$$\begin{array}{ccc}
 X & W & h \\
 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} x \begin{bmatrix} 0.2 & 0.3 & 0.4 & 0.5 \\ 0.5 & 0.8 & 0.2 & 0.4 \\ 0.6 & 0.4 & 0.4 & 0.3 \\ 0.1 & 0.3 & 0.5 & 0.2 \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.3 \\ 0.4 \\ 0.5 \end{bmatrix} \\
 W' & h & W' x h & y_{pred} \\
 \begin{bmatrix} 0.4 & 0.5 & 0.2 & 0.3 \\ 0.1 & 0.3 & 0.5 & 0.8 \\ 0.3 & 0.2 & 0.7 & 0.2 \\ 0.2 & 0.7 & 0.5 & 0.1 \end{bmatrix} x \begin{bmatrix} 0.2 \\ 0.3 \\ 0.4 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 0.46 \\ 0.71 \\ 0.5 \\ 0.5 \end{bmatrix} & \begin{bmatrix} 0.2005 \\ 0.2758 \\ 0.2460 \\ \mathbf{0.2777} \end{bmatrix}
 \end{array}$$

Kata dengan nilai probabilitas tertinggi di atas merujuk pada kata input, yaitu adil. Pada contoh ini, kata-kata sekitar yang akan diprediksi direpresentasikan dalam bentuk vektor $[0 1 0 0]$. Vektor-vektor tersebut kemudian akan dikurangkan dengan nilai probabilitas hasil perhitungan fungsi Softmax untuk menghitung nilai loss atau error.

$$\begin{array}{ccc}
 y_{pred} & y_{target\ w+1} & loss_{w+1} \\
 \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0.2005 \\ 0.2758 \\ 0.2460 \\ 0.2777 \end{bmatrix} = \begin{bmatrix} -0.2005 \\ 1.2881 \\ -0.2460 \\ -0.2777 \end{bmatrix}
 \end{array}$$

Nilai loss digunakan untuk memperbaiki matriks bobot pada lapisan input dan output. Hasil ekstraksi fitur Word2Vec dari teks ulasan yang telah diproses ditampilkan pada Tabel berikut.

Kata	Fitur 1	Fitur 2	Fitur 3	Fitur 4
adil	-0.01337175	0.00595563	0.12753819	0.2252019
sabar	-0.23252526	-0.17792478	0.16145732	0.22430672
sifat	0.07188095	0.02477654	-0.20712055	-0.23622298
baik	-0.18804094	-0.09840259	-0.18778956	-0.02325106

Lampiran 3 Mekanisme Perhitungan LSTM

Langkah pertama yaitu menghitung forget gate. Misalnya, bobot $W_f = 0.2$, hidden state $h_{t-1} = 0$, dan input $x_t = 0.3$ bias $b_f = 0.5$, maka:

$$f_t = \sigma(0.2(0 + 0.3) + 0.5)$$

$$f_t = \sigma(0.2 \times 0.3 + 0.5)$$

$$f_t = \sigma(0.56)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

$$\sigma(0.56) = \frac{1}{1 + e^{-0.56}} = \frac{1}{1 + 0.571} = \frac{1}{1.571} = 0.637$$

Langkah selanjutnya adalah memperbarui *cell state* dilakukan melalui input gate. Misalnya, bobot $W_i = 0.4$, hidden state $h_{t-1} = 0$, dan input $x_t = 0.3$ bias $b_i = 0.4$, maka:

$$i_t = \sigma(0.4 \times (0 + 0.3) + 0.4)$$

$$i_t = \sigma(0.4 \times 0.3 + 0.4)$$

$$i_t = \sigma(0.52)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

$$\sigma(0.52) = \frac{1}{1 + e^{-0.52}} = \frac{1}{1 + 0.594} = \frac{1}{1.594} = 0.627$$

Memori kandidat adalah informasi baru yang akan dipertimbangkan untuk dimasukkan ke dalam sel memori. Misalnya, bobot $W_c = 0.2$, hidden state $h_{t-1} = 0$, dan input $x_t = 0.3$ dan bias $b_c = 0.5$, maka:

$$C \sim_t = \tanh(0.2 \times (0 + 0.3) + 0.5)$$

$$C \sim_t = \tanh(0.2 \times 0.3 + 0.5)$$

$$C \sim_t = \tanh \tanh(0.56)$$

$$\tanh(0.56) = \frac{1.751 - 0.571}{1.751 + 0.571} = \frac{1.18}{2.322} = 0.508$$

Cell state diperbarui dengan menggabungkan informasi dari forget gate dan input gate. Misalnya, cell state $c_{t-1} = 0$, maka:

$$C_t = 0.637 \times 0 + 0.627 \times 0.508 = \mathbf{0.318}$$

Langkah terakhir dalam proses LSTM adalah menentukan *output* dari *cell state*. Misalnya, bobot $W_o = 0.5$, hidden state $h_{t-1} = 0$, dan input $x_t = 0.3$ dan bias $b_o = 0.4$, maka:

$$o_t = \sigma(0.5 \times (0 + 0.3) + 0.4)$$

$$o_t = \sigma(0.5 \times 0.3 + 0.4)$$

$$o_t = \sigma(0.55)$$

$$\sigma(0.55) = \frac{1}{1 + e^{-0.55}} = \frac{1}{1 + 0.577} = \frac{1}{1.577} = 0.634$$

$$h_t = 0.634 \times 0.318$$

$$h_t = 0.201$$

Setelah mendapatkan nilai h_t diteruskan ke lapisan *Dense Layer* untuk dihitung dengan bobot dan bias. Dengan perhitungan sebagai berikut : Misalkan diketahui bobot (weight) $W = [0.1 \ 0.2 \ 0.5]$ hidden state = $[-0.2 \ 0.5 \ 0.3]$, serta bias = $[0.5 \ 0.1 \ 0.1]$

$$z_1 = 0.1 \cdot -0.2 + 0.5 = 0.48$$

$$z_2 = 0.2 + 0.5 + 0.1 = 0.8$$

$$z_3 = 0.5 \cdot 0.3 + 0.1 = 0.25$$

Hasil perhitungan dari skor mentah (*logits*) pada persamaan 3.9 akan dilakukan aktivasi *softmax* yang mengonversi hasil tersebut menjadi probabilitas yang bernilai antara 0 dan 1, serta memastikan bahwa jumlah semua probabilitas sama dengan 1.

Lampiran 4 Biodata Validator

Nama Lengkap : Alivia Lailaturrohima, S.Pd

TTL : Pamekasan, 20 Desember 2000

Alamat : Jl. 25 Swadaya, Duren Sawit, Jakarta Timur, DKI Jakarta

Jenis Kelamin : Perempuan

Agama : Islam

Warga Negara : Indonesia

Pendidikan : Universitas Islam As-Syafiiyah

Pekerjaan : Guru di MI Al Ikhlas Pondok Gede.

Nama Lengkap : Mufti Umma Rosyidah, S.Pd

TTL : Pasuruan, 05 Mei 1998

Alamat : Jl. Taman Dayu, Dsn. Sukorame, Ds. Durensewu, Kec. Pandaan,
Kab. Pasuruan, Prov. Jatim.

Jenis Kelamin : Perempuan

Agama : Islam


```

        'link':
f"https://www.youtube.com/watch?v={video['snippet']['resourceId']
}['videoId']}"
    }
    all_videos.append(video_data)

    next_page_token = response.get('nextPageToken')
    if not next_page_token:
        break

    time.sleep(1)    # Menunggu sebentar agar tidak
melanggar batasan API
    except Exception as e:
        print(f"Error fetching videos: {e}")
        break

    return all_videos

# Ambil transkrip dari video YouTube
def get_transcript(video_id):
    try:
        for lang in ['id', 'en', 'ar', 'ms']: # Coba beberapa
bahasa
            try:
                transcript_list = yta.get_transcript(video_id,
languages=[lang])
                transcript = ' '.join([entry['text'] for entry in
transcript_list])
                return preprocess_text(transcript)
            except Exception as e:
                print(f"Error fetching transcript for language
{lang}: {e}")
                continue

        return "No transcript available" # Jika tidak ada
transkrip
    except Exception as e:
        return f"Error fetching transcript: {str(e)}"

# Terjemahkan transkrip ke bahasa Indonesia
def translate_to_indonesian(transcript):
    try:
        if not transcript or transcript.isspace() or
len(transcript) < 10:
            return "Transkrip tidak tersedia"

        translated = GoogleTranslator(source='auto',
target='id').translate(transcript)
        return translated if translated and not
translated.isspace() else transcript
    except Exception as e:
        print(f"Terjadi kesalahan saat menerjemahkan: {e}")
        return transcript # Jika gagal menerjemahkan, kembalikan
teks asli

# Simpan hasil video dan transkrip dalam file CSV

```

```

def save_videos_with_transcripts(playlist_id, output_dir="data",
filename="/content/drive/My Drive/CRAWLING DATA/FULL KISAH.csv"):
    import os
    os.makedirs(output_dir, exist_ok=True)
    csv_filename = os.path.join(output_dir, filename)

    videos = get_videos_from_playlist(playlist_id)

    if not videos:
        print("Playlist kosong atau tidak dapat diakses.")
        return

    with open(csv_filename, 'w', newline='', encoding='utf-8') as
csvfile:
        fieldnames = ['Title', 'Link', 'Transcript']
        writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
        writer.writeheader()

        for video in videos:
            video_id = video['videoId']
            transcript = get_transcript(video_id)

            # Jika transkrip tersedia, lakukan terjemahan ke
bahasa Inggris
            if "No transcript available" not in transcript:
                transcript = translate_to_english(transcript)

            # Tulis data ke dalam file CSV
            writer.writerow({'Title': video['title'], 'Link':
video['link'], 'Transcript': transcript})
            print(f"Processed: {video['title']}")

        print(f"✅ Data berhasil disimpan di {csv_filename}")

# ID playlist YouTube (GANTI dengan yang benar)
playlist_id = 'PLo_gTQ2VHaHtttdJP4peAOAB8bj5k39uvN'

# Jalankan proses crawling dan simpan hasilnya
save_videos_with_transcripts(playlist_id)

```

Lampiran 6 Kode Inisialisasi bobot LSTM

```
embedding_dim = 16
hidden_dim = 32

Initialize embeddings as random normal matrix of size [vocab_size
x embedding_dim], scaled by 0.01

Set concat_dim to hidden_dim + embedding_dim

Initialize:
    W_f as random normal matrix of size [hidden_dim x concat_dim],
scaled by 0.1
    b_f as zero vector of size [hidden_dim]

    W_i as random normal matrix of size [hidden_dim x concat_dim],
scaled by 0.1
    b_i as zero vector of size [hidden_dim]

    W_c as random normal matrix of size [hidden_dim x concat_dim],
scaled by 0.1
    b_c as zero vector of size [hidden_dim]

    W_o as random normal matrix of size [hidden_dim x concat_dim],
scaled by 0.1
    b_o as zero vector of size [hidden_dim]
```

Lampiran 7 Kode *Forward* LSTM

```
def lstm_forward(input_ids, embeddings, W_f, b_f, W_i, b_i, W_c,
b_c, W_o, b_o, hidden_dim):
    h = np.zeros(hidden_dim)
    c = np.zeros(hidden_dim)
    H = []
    for idx in input_ids:
        x_t = embeddings[idx]
        h, c = lstm_step(x_t, h, c, W_f, b_f, W_i, b_i, W_c, b_c,
W_o, b_o)
        H.append(h.copy())
    return np.array(H), h, c
```

Lampiran 8 Kode STEP LSTM

```
def lstm_step(x_t, h_prev, c_prev, W_f, b_f, W_i, b_i, W_c, b_c,
W_o, b_o):
    z = np.concatenate([h_prev, x_t])
    f_t = sigmoid(np.dot(W_f, z) + b_f)
    i_t = sigmoid(np.dot(W_i, z) + b_i)
    c_hat = tanh(np.dot(W_c, z) + b_c)
    c_t = f_t * c_prev + i_t * c_hat
    o_t = sigmoid(np.dot(W_o, z) + b_o)
```

```
h_t = o_t * tanh(c_t)
```

```
return h_t, c_t
```

Lampiran 9 Dataset

<https://www.kaggle.com/datasets/aisyagustisavila/data-ringkasan/data>