TEXT SUMMARIZATION UMPAN BALIK PENGGUNA WEBSITE SIBAYAR PONDOK PESANTREN SABILURROSYAD DENGAN METODE BI-LSTM

SKRIPSI

Oleh :

SAYYED AAMIR HASSAN
NIM. 210605110098

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

TEXT SUMMARIZATION UMPAN BALIK PENGGUNA WEBSITE SIBAYAR PONDOK PESANTREN SABILURROSYAD DENGAN METODE BI-LSTM

SKRIPSI

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

Oleh :
SAYYED AAMIR HASSAN
NIM. 210605110098

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

HALAMAN PERSETUJUAN

TEXT SUMMARIZATION UMPAN BALIK PENGGUNA WEBSITE SIBAYAR PONDOK PESANTREN SABILURROSYAD DENGAN METODE BI-LSTM

SKRIPSI

Oleh :
SAYYED AAMIR HASSAN
NIM. 210605110098

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 16 April 2025

Pembimbing I,	Pembimbing II,
Supriyono, M. Kom	Dr. Zainal Abidin, M.Kom
NIP. 19841010 201903 1 012	NIP. 19760613 200501 1 004

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

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

HALAMAN PENGESAHAN

TEXT SUMMARIZATION UMPAN BALIK PENGGUNA WEBSITE SIBAYAR PONDOK PESANTREN SABILURROSYAD DENGAN METODE BI-LSTM

SKRIPSI

Oleh :
SAYYED AAMIR HASSAN
NIM. 210605110098

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 16 Mei 2025

Susunan Dewan Penguji

Ketua Penguji	: Fatchurrochman, M.Kom NIP. 19700731 200501 1 002	( )
Anggota Penguji I	: Khadijah Fahmi Hayati Holle, M.Kom NIP. 19900626 202203 2 002	( )
Anggota Penguji II	: Supriyono, M.Kom NIP. 19841010 201903 1 012	( )
Anggota Penguji III	: Dr. Zainal Abidin, M.Kom NIP. 19760613 200501 1 004	( )

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

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

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini :

Nama	: SAYYED AAMIR HASSAN
NIM	: 210605110098
Fakultas / Prodi	: Sains dan Teknologi / Teknik Informatika
Judul Skripsi	: *Text Summarization* Umpan Balik Pengguna Website Sibayar Pondok Pesantren Sabilurrosyad dengan Metode Bi-LSTM

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan 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 di kemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 16 April 2025
Yang membuat pernyataan,

Sayyed Aamir Hassan
NIM. 210605110098

MOTTO

. . . . وَفَوْقَ كُلِّ ذِيْ عِلْمٍ عَلِيْمٌ ﴿٧٦﴾

". . . dan di atas setiap orang yang berpengetahuan, ada yang lebih mengetahui."

(QS. Yusuf: 76)

KATA PENGANTAR

Segala puji bagi Allah SWT, yang maha pengasih lagi maha penyayang, yang telah memberi karunia kepada seluruh anak cucu Adam berupa ilmu dan amal. Limpahan shalawat dan salam senantiasa tercurah kepada baginda Nabi Muhammad SAW, nabi akhir zaman dan utusan bagi seluruh alam. Segala puji penulis haturkan kepada Allah SWT. atas limpahan rahmat dan karunianya sehingga penulis diberi waktu dan kemampuan untuk menyelesaikan skripsi ini dengan baik –sebagai syarat menyelesaikan studi sarjana Teknik Informatika pada Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Apresiasi sebesar-besarnya kepada seluruh pihak yang telah berkontribusi baik secara pikiran maupun materi, secara langsung maupun tidak langsung, jasmani maupun rohani, dan segala bentuk dukungan lainnya. Apresiasi khusus penulis berikan kepada :

1. Supriyono, M.Kom, wali dosen sekaligus dosen pembimbing I, yang telah memberikan ilmu, saran, kritik, bimbingan, dan waktunya atas penulisan skripsi ini hingga selesai.
2. Dr. Zainal Abidin, M.Kom, dosen pemimbing II, yang telah memberikan ilmu, saran, kritik, bimbingan, dan waktunya atas penulisan skripsi ini hingga selesai.

3. Seluruh civitas akademik Program Studi Teknik Informatika atas pengalaman dan ilmunya yang sangat berharga dan semoga kelak di kemudian hari apa yang telah diajarkan bermanfaat bagi seluruh pihak, terkhusus diri saya sendiri.
4. Kedua orang tua penulis, Abah Moch. Hasan dan Ibu Hanik Mahmudah yang menjadi motivasi utama penulis dalam menyelesaikan studi ini, yang telah memberikan segala dukungan dari awal sampai akhir, yang telah membesarkan saya hingga menjadi diri saya hari ini. Semoga Allah SWT panjangkan umurnya dan diberikan kesehatan serta rejeki yang tidak terputus.
5. Seluruh teman-teman senasib seperjuangan akademis khususnya rekan-rekan angkatan 2021 Teknik Informatika UIN Malang, yang telah berkenan untuk berdiskusi, *sharing* informasi, menunjukkan cara mengurus skripsi dan tolong-menolong dalam hal kebaikan.
6. Seluruh kawan-kawanku Pondok Gasek kamar 11A dan sekitarnya, yang telah membersamai penulis dan meramaikan harinya, terima kasih atas segala bentuk adu nasib, dan manifestasinya yang menjadi motivasi penulis dalam proses pengerjaan skripsi ini.
7. Tim SiBayar Ponpes Gasek, yang berjuang mengembangkan *website* SiBayar dari mulai tahap *groundbreaking* hingga jadi seperti saat ini.
8. Seseorang dari Wates-Joyosuko, yang telah berjasa dalam bantuan moral dan logistik skripsi penulis dari awal hingga akhir. Terima kasih telah hadir dan saling mendukung dalam dinamika perjalanan akademis penulis.

9. Seluruh pihak yang terlibat namun tidak bisa penulis sebutkan satu persatu. Semoga Allah SWT membalas kebaikan dan mengampuni segala dosa kita semua.

اذا تم أمر بدا نقصه

Apabila telah selesai suatu urusan, kelak tampak kekurangannya.

Sama halnya dengan skripsi ini, sehingga perlu adanya bimbingan dari pihak-pihak yang lebih mumpuni. Semoga ilmu dan skripsi ini dapat memberikan sesuatu yang bermanfaat bagi khazanah ilmu pengetahuan manusia.

Malang, 16 April 2025

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Hassan, Sayyed Aamir. 2025. ***Text Summarization* Umpan Balik Pengguna *Website* SiBayar Pondok Pesantren Sabilurrosyad dengan Metode Bi-LSTM**. Skripsi. Program Studi Teknik Informatika. Fakultas Sains dan Teknologi. Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing : (I) Supriyono, M.Kom (II) Dr. Zainal Abidin, M.Kom.

Kata Kunci: *Bidirectional Long Short Term Memory, Text Summarization,* Umpan Balik

SiBayar adalah *website* yang sedang dikembangkan oleh Pondok Pesantren Sabilurrosyad untuk mempermudah administrasi dan manajemen kepesantrenan. Untuk meningkatkan fungsionalitas *website,* diperlukan umpan balik dari pengguna terhadap fitur-fitur yang ada. Namun, mengelola dan menganalisis sejumlah besar umpan balik pengguna secara manual dapat menjadi proses yang sangat memakan waktu. Oleh karena itu, diperlukan pendekatan otomatis seperti *text summarization* untuk merangkum dan menganalisis data tersebut. Penelitian ini bertujuan untuk menghasilkan ringkasan otomatis dari umpan balik pengguna pada *website* SiBayar Pondok Pesantren Sabilurrosyad menggunakan metode *Bidirectional Long Short-Term Memory* (Bi-LSTM), dengan fokus pada identifikasi parameter terbaik melalui *hyperparameter tuning* serta evaluasi akurasi ringkasannya. Hasil pengujian *hyperparameter tuning* menunjukkan bahwa konfigurasi yang memberikan performa terbaik adalah yang menggunakan algoritma optimasi Nadam, jumlah lapisan 1 dan ukuran *batch* 1, serta *variational dropout* dengan *dropout rate* 0.5. Evaluasi kualitas ringkasan model dilakukan menggunakan metrik ROUGE yang menunjukkan bahwa model Bi-LSTM mencapai skor ROUGE-1 sebesar 0.6221, skor ROUGE-2 sebesar 0.5462, dan skor ROUGE-L sebesar 0.660. Secara keseluruhan, model Bi-LSTM di penelitian ini memiliki performa yang baik dalam melakukan peringkasan teks, namun kesesuaian pasangan dan urutan kata masih perlu ditingkatkan untuk hasil yang lebih optimal.

ABSTRACT

Hassan, Sayyed Aamir. 2025. **Text Summarization on User Feedback on the SiBayar Website of Sabilurrosyad Islamic Boarding School Using the Bi-LSTM Method.** Thesis. Department of Informatics Engineering Faculty of Science and Technology. Maulana Malik Ibrahim State Islamic University Malang. Supervisor: (I) Supriyono, M.Kom (II) Dr. Zainal Abidin, M.Kom.

Key words: Bidirectional Long Short Term Memory; Text Summarization; User Feedback

SiBayar is a website developed by the Sabilurrosyad Islamic Boarding School to facilitate its administration and management. To improve the functionality of the website, user feedback is needed on the existing features. However, managing and analyzing a large amount of user feedback manually can be a very time-consuming process. Therefore, an automated approach such as text summarization is needed to summarize and analyze the data. This study aims to generate an automated summary of user feedback on the SiBayar website of the Sabilurrosyad Islamic Boarding School using the Bidirectional Long Short-Term Memory (Bi-LSTM) method, focusing on identifying the best parameters through hyperparameter tuning and evaluating the accuracy in full. The results of the hyperparameter tuning test show that the configuration that provides the best performance is the one using the Nadam algorithm optimization, the number of layers 1 and batch size 1, and the variational dropout with a dropout rate of 0.5. The model summary quality evaluation was performed using the ROUGE metric which showed that the Bi-LSTM model achieved a ROUGE-1 score of 0.6221, a ROUGE-2 score of 0.5462, and a ROUGE-L score of 0.660. Overall, Bi-LSTM model in this study has good performance in summarizing text, but the suitability of word pairs and sequences still needs to be improved for more optimal results.

مستخلص البحث

حسن، سيد أمير. ٢٠٢٥. **تلخيص النصوص لتعليقات المستخدمين في موقع SiBayar معهد سبيل الرشاد باستخدام طريقة Bi-LSTM**، أطروحة. قسم هندسة المعلوماتية. كلية العلوم والتكنولوجيا. جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف: (أولا) سوبريونو، الماجستير (ثانيا) د. زين العابدين, الماجستير.

الكلمات المفتاحية : *Bidirectional Long Short Term Memory* ,تلخيص النصوص ,تعليقات المستخدمين

يتم تطوير موقع SiBayar من قبل معهد سبيل الرشاد الإسلامية لتسهيل الإدارة وإدارة شؤون المدرسة. ولتحسين وظائف الموقع، هناك حاجة إلى ملاحظات من المستخدمين حول الميزات الموجودة. ومع ذلك، يمكن أن يكون إدارة وتحليل كمية كبيرة من ملاحظات المستخدمين يدويًا عملية تستغرق وقتًا طويلاً. لذلك، هناك حاجة إلى نهج آلي مثل تلخيص النصوص لتلخيص وتحليل البيانات. تهدف هذه الدراسة إلى إنشاء ملخص تلقائي لملاحظات المستخدمين على معهد سبيل الرشاد الإسلامية باستخدام طريقة Bi-LSTM، مع التركيز على تحديد أفضل المعلمات من خلال ضبط المعلمات الفائقة وتقييم دقة الملخص. SiBayar. تظهر نتائج اختبار ضبط المعلمات الفائقة أن التكوين الذي يوفر أفضل أداء هو التكوين الذي يستخدم خوارزمية تحسين Nadam وعدد الطبقات 1 وحجم الدفعة 1 وvariational dropout بمعدل dropout يبلغ 0.5. تم إجراء تقييم جودة ملخص النموذج باستخدام مقياس ROUGE الذي أظهر أن نموذج Bi-LSTM حقق درجة ROUGE-1 وهي0.6221، ودرجة ROUGE-2 وهي0.5462، ودرجة ROUGE-L وهي 0.660 بشكل عام، فإن Bi-LSTM قادر على إجراء تلخيص للنصوص، ولكن لا يزال يتعين تحسين ملاءمة أزواج الكلمات والتسلسلات للحصول على نتائج أكثر مثالية

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pondok pesantren adalah institusi pendidikan Islam berbasis asrama di Indonesia yang berperan sebagai pusat pembelajaran agama dan pengembangan sosial masyarakat (Siregar, 2018). Umumnya pondok pesantren memiliki sejumlah komponen inti, seperti ulama atau kyai, santri, literatur keagamaan klasik, masjid, serta asrama untuk para siswa (Julhadi, 2019). Seiring dengan berjalannya waktu, pesantren telah berkembang dari sistem tradisional menjadi lebih modern, di mana beberapa pesantren tetap mempertahankan nilai-nilai tradisional sambil mengadopsi kurikulum dan metode pengajaran modern. Perkembangan ini meliputi integrasi kurikulum pendidikan pesantren dan kurikulum formal nasional, pendekatan pengajaran baru dengan memadukan bahasa Arab dan Inggris, serta sistem manajemen yang lebih tertata (Zarkasyi, 2015).

Pondok Pesantren Sabilurrosyad merupakan pondok pesantren di Kota Malang yang didirikan pada 1995 oleh K.H. Marzuqi Mustamar. Pondok Pesantren didirikan untuk mempertahankan ajaran agama Islam serta melindungi masyarakat setempat dari pengaruh kristenisasi. Kepemimpinan kyai di Pondok Pesantren Sabilurrosyad ditandai dengan teladan yang baik, kerendahan hati, dan perlakuan terhadap santri seperti keluarga (Kusuma Pertiwi et al., 2018). Pondok Pesantren Sabilurrosyad aktif dalam melakukan kegiatan dan kampanye dalam rangka mencegah radikalisme beragama dengan mempromosikan paham Islam moderat

(*wasathiyah)* (Yoyok Amirudin, 2020).

Seiring dengan perkembangan teknologi, pesantren mulai mengadopsi teknologi digital dalam meningkatkan proses belajar dan manajemen kepengurusan pesantren (Marsum & Syahroni, 2020). Pondok pesantren dapat menggunakan aplikasi *mobile* untuk mengelola jadwal kegiatan dan optimalisasi manajemen kepesantrenan (Nawangnugraeni et al., 2023) . Penggunaan teknologi di pesantren menunjukkan banyak dampak positif seperti meningkatkan aksesibilitas pendidikan, menambah khazanah sumber belajar dan meningkatkan pengawasan dan evaluasi hasil belajar santri (Muchasan et al., 2024). Namun, belum semua pesantren mengintegrasikan sistem informasi secara penuh ke dalam proses akademiknya—sebagian karena ingin menjaga tradisi, dan beberapa masih mengandalkan perangkat dasar seperti Microsoft Office dan Google (Maimunah & Junadi, 2023).

Pondok Pesantren Sabilurrosyad sedang mengembangkan sebuah *website* sistem informasi akademik bernama SiBayar yang dirancang khusus untuk mengelola berbagai aspek manajemen pesantren, termasuk administrasi umum, pembayaran syahriah (iuran bulanan), serta pembayaran untuk keperluan kegiatan pondok bagi para santri. *Website* ini bertujuan untuk menciptakan sebuah sistem yang efisien dan terpusat, di mana segala kebutuhan dan layanan pondok dapat diakses melalui satu pintu, yaitu aplikasi *website* SiBayar. Pengembangan *website* SiBayar dilakukan santri-santri pondok sendiri. Santri yang terlibat dalam pembuatan *website* ini memiliki latar belakang dalam pemrograman *web*, dan juga

terdapat sejumlah santri yang memiliki keahlian di bidang akuntansi yang turut serta dalam merancang sistem pembukuan keuangan pada *website* SiBayar.

Banyaknya pihak yang akan terlibat dalam rancang bangun *website* SiBayar membuat diperlukannya pengujian berdasarkan umpan balik pengguna, baik itu dari sisi santri maupun pengurus pondok. Pengujian perangkat lunak merupakan tahap penting dalam pengembangan *software* untuk menemukan kesalahan dan memastikan kualitas produk. Umpan balik pengguna berperan penting dalam tahap pengujian *website* agar keperluan dari pengguna dapat diakomodir secara efektif oleh pengembang perangkat lunak dalam meningkatkan usabilitas pengguna (Munawar, 2019).

Pemahaman yang tepat dan teliti terhadap umpan balik dapat memastikan pengambilan keputusan dalam pengembangan website SiBayar sesuai dengan harapan semua pengguna. Oleh sebab itu, maka diperlukan bantuan teknologi NLP (*natural language processing*) dalam menyaring teks umpan balik pengguna yang sangat banyak. Salah satu implementasi teknologi NLP tersebut *adalah text summarization* atau teknik peringkasan teks. *Text Summarization* mampu mengatasi masalah dalam mengatasi konten atau teks dari pengguna dalam jumlah besar secara manual, dengan cara menyediakan cara yang lebih efisien untuk mengekstrak informasi yang penting dari data teks (Kothari et al., 2020). Dengan adanya peringkasan teks pada umpan balik pengguna, diharapkan pengembang perangkat lunak mampu mengetahui komponen-komponen mana saja yang perlu diperbaiki atau ditingkatkan, sehingga dapat lebih cepat merespons kebutuhan

pengguna dan meningkatkan kualitas serta efektivitas perangkat lunak secara keseluruhan.

Konsep *text summarization*, bekerja dengan memilih informasi yang penting dan menyusunnya kembali dengan cara yang jelas dan mudah dipahami, tanpa mengurangi makna dari informasi tersebut. Hasilnya dari *text summarization* berupa intisari dari suatu informasi yang disajikan dalam bentuk lebih singkat. Penyajian intisari informasi secara singkat, selaras dengan salah satu sunnah Nabi dalam hadis berikut.

عن أبي هريرة رضي الله عنه أن رسول الله صلى الله عليه وسلم قال : بعثت بجوامع الكلم

"Dari Abu Hurairah RA, sesungguhnya Rasul SAW bersabda, Aku diutus dengan jawami'ul kalim (kalimat-kalimat yang ringkas, namun penuh makna)(HR. Bukhari : 6845)."

Dikutip dari Fathul Bari Syarh Shahih Bukhari, hadis ini menggambarkan salah satu keutamaan Nabi Muhammad SAW yaitu *jawami'ul kalim*, kemampuan berbicara dengan kata-kata yang sedikit namun sangat kaya makna (القليل اللفظ الكثير المعاني) (al-'Asqalāni, 2010). Cara Nabi Muhammad SAW berbicara, sebagaimana dalam *jawaami'ul Kalim*, dapat menjadi pedoman bagi umat Islam dalam hal bertutur kata maupun menyampaikan informasi secara bijaksana dan efektif.

Berdasarkan penelitian terdahulu terkait tugas peringkasan teks, beberapa peneliti telah menggunakan metode seperti *Hierarchial Clustering* dengan *C-Means, Multi-Marginal Relevance, CLSA, Long Short Term Memory* dan *Bidirectional Long Short Term Memory (*Bi-LSTM). Namun, apabila diamati dari beberapa metode tersebut, tugas peringkasan teks terkait topik *software engineering*

masih dilakukan oleh metode peringkasan ekstraktif. Penggunaan metode abstraktif yang telah teruji dalam menghasilkan ringkasan teks seperti Bi-LSTM masih kurang dieksplorasi terkait dengan topik tersebut.

Walaupun Bi-LSTM dalam beberapa penelitian menunjukkan nilai yang cukup baik, diperlukan suatu pendekatan tambahan untuk meningkatkan performa metode tersebut. Salah satu cara untuk meningkatkan performa model adalah dengan menentukan komponen parameter yang sesuai dengan kebutuhan atau tujuan dilatihnya algoritma dengan *hyperparameter tuning*. Penentuan parameter yang tepat dalam melakukan pelatihan Bi-LSTM dapat mempengaruhi akurasi model secara signifikan (Reimers & Gurevych, 2017).

Dengan demikian, penelitian ini mengusulkan Bi-LSTM sebagai metode guna menghasilkan ringkasan umpan balik pengguna *website* SiBayar Pondok Pesantren Sabilurosyad. Melalui implementasi metode Bi-LSTM, penelitian ini bermaksud untuk melakukan tugas peringkasan teks serta mengidentifikasi kombinasi parameter terbaik model dalam melakukan tugas tersebut. Hasil dari metode Bi-LSTM kemudian dievaluasi akurasinya dalam merepresentasikan umpan balik pengguna dalam bentuk ringkasan.

1.2 Pernyataan Masalah

Berdasarkan uraian latar belakang di atas, maka penelitian ini maka penelitian ini difokuskan untuk menjawab pertanyaan-pertanyaan berikut.

a. Apa konfigurasi terbaik dari model Bi-LSTM dalam tugas peringkasan teks umpan balik pengguna melalui pengujian *hyperparameter tuning*?

b. Bagaimana performa model Bi-LSTM dalam menghasilkan ringkasan teks umpan balik pengguna, yang dievaluasi menggunakan metrik ROUGE-N dan ROUGE-L?

1.3 Batasan Masalah

Penelitian ini memiliki batasan-batasan yang perlu diperjelas untuk memastikan fokus dan ruang lingkup yang jelas dalam analisis yang dilakukan, antara lain:

a. Penelitian ini menggunakan pendekatan abstraktif dalam melakukan tugas peringkasan teks.
b. Data teks umpan balik pengguna dalam bahasa Indonesia dan dikumpulkan melalui kuesioner.

1.4 Tujuan Penelitian

Berdasarkan permasalahan yang diidentifikasi pada latar belakang, penelitian ini bertujuan untuk :

a. Untuk mengidentifikasi konfigurasi terbaik dari Bi-LSTM dalam membuat model peringkasan teks umpan balik pengguna melalui pengujian *hyperparameter tuning*.
b. Untuk mengevaluasi performa model Bi-LSTM dalam menghasilkan ringkasan teks umpan balik pengguna menggunakan metrik ROUGE.

1.5 Manfaat Penelitian

Diharapkan hasil dari penelitian ini mampu memberikan manfaat baik dari segi teoritis maupun praktis, sebagai berikut :

a. Secara teoritis, penelitian ini berkontribusi pada pengembangan pengetahuan tentang bagaimana penerapan metode Bi-LSTM dalam meringkas teks umpan balik pengguna perangkat lunak.
b. Secara praktis, penelitian ini menawarkan alat bantu yang efisien untuk menganalisis umpan balik pengguna, yang dapat diadopsi oleh pengembang perangkat lunak SiBayar di Pondok Pesantren Sabilurrosyad.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terkait

Penelitian oleh (Jindal & Kaur, 2020) menganalisis metode peringkasan teks laporan bug perangkat lunak menggunakan *Rapid Automatic Keyword Extraction* (RAKE) dan *Term Frequency-inverse Document Frequency (*TF-IDF) untuk ekstraksi kata kunci, Fuzzy C-means clustering untuk ekstraksi kalimat, serta rule-engine berbasis pengetahuan domain untuk seleksi final. Hierarchical clustering diterapkan untuk menghilangkan redundansi dan melakukan perangkingan ulang. Penelitian tersebut menggunakan dua dataset dari *Apache Project Bug Report Corpus (APBRC)*, yang terdiri dari 21 laporan bug, dan Bug Report Corpus (BRC), yang mencakup 36 laporan bugs. Hasil penelitian menghasilkan skor rata-rata 78,22% untuk presisi, 82,18% untuk recall, 80,10% untuk f-score, dan 81,66% untuk presisi piramida.

Penelitian yang dilakukan oleh (Tripathy & Ashok, 2023) meneliti peringkasan teks menggunakan metode Abstractive Summarization berbasis deep learning, yaitu bidirectional Long Short-Term Memory (LSTM) dan *Pointer Generator mode*. Penelitian ini menggunakan dua dataset dari Amazon Fine Food Review dan CNN/Daily Mail dengan total dataset sejumlah 121,782 dengan 80% dataset sebagai data latih 12% sebagai data uji dan 8% sisanya sebagai data validasi. Metode LSTM, yang dilengkapi dengan Bahdanau *Attention Model Decoder* dan *Conceptnet Numberbatch embeddings,* serta metode *Pointer Generator*, yang

mengatasi masalah pengulangan kata dan ketidakmampuan jaringan untuk menyalin fakta, dibandingkan dalam penelitian ini. Hasil menunjukkan bahwa penggunaan berbagai parameter, seperti Cosine Similarity sebesar 0.29277002 dan *training epoch* yang tepat, membantu menganalisis efisiensi model dalam menghasilkan ringkasan yang lebih baik. Skor ROUGE-1 pada penelitian ini sebesar 0.21058, ROUGE-2 0.37963, dan ROUGE-L 0.38653

Penelitian (Yuliska & Syaliman, 2022) menggunakan metode *Bidirectional Long Short Term Memory Network* (Bi-LSTM) untuk peringkasan teks otomatis berbasis kueri dengan pendekatan ekstraktif. Penelitian ini menggunakan dataset DUC 2005-2007, yang terdiri dari artikel berita berbahasa Inggris yang dikelompokkan dalam 145 topik dengan total 104.773 kalimat. Dataset ini mencakup empat ringkasan referensi per topik. Dalam eksperimen, DUC 2005 digunakan sebagai data training, DUC 2006 sebagai data validation, dan DUC 2007 sebagai data testing. Hasil eksperimen menunjukkan bahwa Bi-LSTM mampu menghasilkan ringkasan dengan kinerja yang baik, dibuktikan dengan skor ROUGE-1 sebesar 43.53, ROUGE-2 sebesar 11.40, dan ROUGE-L sebesar 18.67.

Penelitian oleh (Sari & Fatonah, 2021) berfokus pada peringkasan teks otomatis pada modul pembelajaran berbahasa Indonesia. Objek penelitian ini melibatkan sepuluh file modul pembelajaran yang berasal dari para dosen Universitas Mercu Buana, dengan lima file berformat .docx dan lima file lainnya berformat .pdf. Dalam penelitian ini, metode *Term Frequency-Inverse Document Frequency* (TF-IDF) digunakan untuk pembobotan kata, sedangkan metode *Cross Latent Semantic Analysis* (CLSA) diterapkan untuk proses peringkasan teks. Hasil

penelitian menunjukkan bahwa pengujian akurasi dilakukan dengan membandingkan ringkasan manual yang dibuat oleh manusia dengan hasil ringkasan dari sistem otomatis. Nilai rata-rata f-measure, precision, dan recall tertinggi diperoleh pada tingkat kompresi 20%, dengan nilai masing-masing 0.3853, 0.432, dan 0.3715.

Penelitian oleh (Saputra et al., 2021) melakukan peringkasan teks abstraktif menggunakan metode Long Short Term Memory (LSTM). Dataset yang digunakan dalam penelitian ini adalah INDOSUM yang merupakan dataset *benchmark* untuk peringkasan teks Bahasa Indonesia. Penelitian ini menggunakan skenario pengujian berupa implementasi data *preprocessing* yang berbeda seperti pembagian *stemming*, dan *stopword removal*. Setiap skenario pengujian memiliki *hyperparameter* yang ditentukan sehingga tidak terdapat konfigurasi terhadap masing-masing skenario. Hasil penelitian ini menghasilkan rata-rata skor ROUGE-1 sebesar 0.13826 dan maksimum skor 0.50485 pada skenario tanpa *stopword* dan *stemming.*

Penelitian oleh (Shaliha, 2021) menggunakan dataset dalam Bahasa Indonesia dalam melakukan peringkasan abstraktif dengan metode *Bidirectional Long Short Term Memory* (Bi-LSTM). Dalam meningkatkan performa model, penelitian tersebut memakai *Local Attention* saat memprediksi hasil ringkasan. Hasil penelitian menunjukkan bahwa model yang dibangun memiliki skor ROUGE-1 0.06236, dan skor ROUGE-2 0.00684.

Penelitian yang dilakukan (Jiang, et al., 2021) mengeksplorasi penggunaan *Bidirectional* LSTM disertai dengan *Attention Based* guna melakukan tugas

peringkasan teks. Dataset yang digunakan dalam penelitain ini adalah LCTS *short-text corpus* yang diambil dari *website Weibo* dan *Ttnew Long Text Corpus* yang digunakan pada NLPCC 2017-2018 untuk peringkasan teks yang terdiri dari lebih dari 2 juta teks. Penelitian ini melakukan pengujian *hyperparameter tuning* dengan mekanisme *early stopping* pada saat *epoch* pelatihan. Hasil menujukkan metode yang digunakan memiliki skor ROUGE-1 0.2547, ROUGE-2 0.809, ROUGE-L 0.2903 pada dataset LCSTS dan ROUGE-1 0.2527, ROUGE-2 0.803, ROUGE-L 0.2904 untuk dataset *Ttnews*.

Dalam penelitian (Karo Karo et al., 2024) peringkasan teks pada ulasan aplikasi *Digital Library System* diimplementasikan menggunakan metode *Maximum Marginal Relevance* (MMR). Objek penelitian ini adalah ulasan pengguna aplikasi *Digital Library System* yang merupakan aplikasi perpustakaan mobile dari Universitas Negeri Medan (Unimed). Metode MMR digunakan dalam proses peringkasan teks, sementara evaluasi hasil ringkasan dilakukan menggunakan metrik precision, recall, dan F1-score. Penelitian ini berhasil mengidentifikasi 30 ulasan dengan skor MMR tertinggi, dan dari ulasan tersebut, dihasilkan ringkasan yang terdiri dari 10 ulasan dengan peringkat MMR tertinggi. Hasil ringkasan yang dihasilkan menunjukkan akurasi sebesar 30,51%, recall sebesar 56,25%, dan F1-score sebesar 39,56%.

Tabel 2.1 Penelitian Terkait

No.	Sumber	Dokumen	Pendekatan	Metode	Hasil
1	(Jindal & Kaur, 2020)	*Apache Project Bug Report Corpus (APBRC)*, dan Bug Report Corpus (BRC)	Ekstraktif	Fuzzy C-means clustering	78,22% presisi, 82,18% recall, 80,10% f-score, dan 81,66% presisi piramida.
2	(Saputra et al., 2021)	INDOSUM (Teks berita)	Abstraktif	Long Short Term Memory (LSTM)	ROUGE-1 sebesar 0.13846

No.	Sumber	Dokumen	Pendekatan	Metode	Hasil
3	(Jiang et al., 2021)	LCSTS Corpus	Abstraktif	CNN-Bi-LSTM	ROUGE-1 sebesar 0.2527, ROUGE-2 sebesar 0.0803, dan ROUGE-L sebesar 30.56
4	(Shaliha, 2021)	Teks Bahasa Indonesia	Abstraktif	*Bidirectional Long Short-Term Memory (LSTM)*	ROUGE-1 0.06236 dan and ROUGE-2 0.00684.
5	(Tripathy & Ashok, 2023)	Amazon Fine Food Review dan CNN/Daily Mail	Abstraktif	*Bidirectional Long Short-Term Memory (LSTM)*	Cosine Similarity sebesar 0.29277002
6	(Yuliska & Syaliman, 2022)	DUC (*Document Understanding Conference)* 2005-2007	Abstraktif	*Bidirectional Long Short Term Memory Network (Bi-LSTM)*	Skor ROUGE-1 sebesar 43.53, ROUGE-2 sebesar 11.40, dan ROUGE-L sebesar 18.67.
7	(Y. M. Sari & Fatonah, 2021)	10 file modul pembelajaran yang dosen Universitas Mercu Buana	Ekstraktif	*Term Frequency-Inverse Document Frequency* (TF-IDF), dan *Cross Latent Semantic Analysis* (CLSA)	Nilai *f-measure, precision,* dan *recal l* masing-masing 0.3853, 0.432, dan 0.3715.
8	(Karo Karo et al., 2024)	Ulasan aplikasi *Digital Library System*	Ekstraktif	*Maximum Marginal Relevance* (MMR)	Akurasi sebesar 30,51%, recall sebesar 56,25%, dan F1-score sebesar 39,56%.
	Penelitian ini	Umpan Balik Pengguna *website* SiBayar	Abstraktif	*Bidirectional Long Short Term Memory Network (Bi-LSTM)*	

Berdasarkan Tabel 2.1, peringkasan dokumen dalam konteks yang berhubungan dengan *software engineering* masih dieksplorasi pada peringkasan ekstraktif. Sedangkan untuk peringkasan abstraktif menggunakan Bi-LSTM masih belum dieksplorasi. Dengan demikian penelitian ini memilih fokus pada

peringkasan teks Bi-LSTM dengan pendekatan abstraktif dengan mengeksplorasi dokumen pada konteks *software engineering*.

2.2 Umpan Balik Pengguna

Umpan balik pengguna adalah informasi yang diberikan oleh seseorang yang telah menggunakan suatu produk atau layanan mengenai bagaimana mereka melihat dan berinteraksi dengan produk tersebut (Kanev et al., 2023). Umpan balik ini bisa mencakup berbagai jenis, seperti komentar, saran, keluhan, atau pujian. Komentar meliputi aspek-aspek spesifik dari produk, sementara saran dan keluhan bisa memberikan informasi tentang area yang perlu diperbaiki atau fitur yang mungkin diperlukan (Yan et al., 2023).

Dalam pengujian perangkat lunak, umpan balik pengguna sangat penting untuk perkembangan perangkat lunak, karena memberikan informasi berharga untuk menentukan kebutuhan dan meningkatkan kualitas perangkat lunak (Supriyono, 2019). Umpan balik ini dapat memberikan informasi tentang bug, kesalahan, atau masalah fungsional yang dialami pengguna di dunia nyata (Martens, 2020)

2.3 *Text Summarization*

Text Summarization adalah proses merangkum dokumen yang panjang menjadi versi yang lebih singkat sambil tetap mempertahankan informasi penting (Thange et al., 2023). Tujuan utama dari text summarization adalah untuk membantu pengguna memahami inti dari sebuah dokumen tanpa harus membaca keseluruhan teks (Raundale & Shekhar, 2021). Ada dua pendekatan utama dalam

merangkum teks: ekstraktif dan abstraktif (Sohail et al., 2020). Proses peringkasan teks melibatkan beberapa tahap, seperti praproses, ekstraksi fitur, pembobotan kalimat dan *summary extraction (Varade et al., 2021)*.

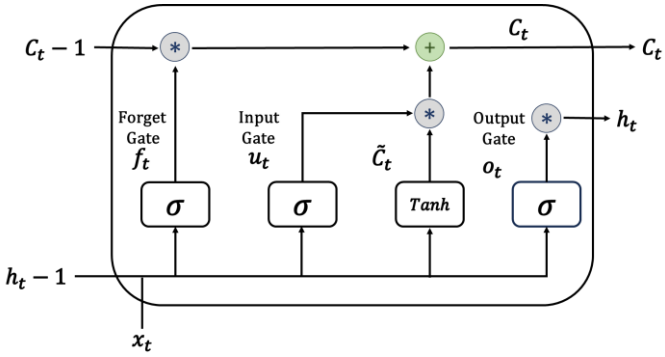
Metode ekstraktif memilih kalimat-kalimat langsung dari teks sumber untuk disertakan dalam ringkasan (El-Kassas et al., 2021). Teknik ini menggunakan algoritma untuk menentukan kalimat mana yang paling relevan berdasarkan frekuensi kata atau hubungan antar kalimat (Suleiman & Awajan, 2020). Metode abstraktif menghasilkan ringkasan dengan membuat kalimat baru yang menggambarkan informasi utama dari teks sumber. Proses tersebut melibatkan pendalaman konten teks dan penyampaian informasi dengan cara yang lebih ringkas dan terstruktur (Patel & Mangaokar, 2020).

Berdasarkan jumlah sumber dokumennya *text summarization* terbagi menjadi dua jenis utama, yaitu *single document summarization* dan *multi document summarization*. *Single document summarization* adalah proses menghasilkan ringkasan dari satu dokumen tunggal, dengan tujuan menyajikan inti informasi secara ringkas tanpa mengubah makna utama dari teks sumber. Sementara itu, multi *document summarization* merupakan teknik merangkum informasi dari beberapa dokumen yang saling berkaitan atau membahas topik yang sama, untuk menghasilkan satu ringkasan terpadu yang merepresentasikan keseluruhan isi dokumen.

2.4 *Long Short-Term Memory*

Long Short-Term Memory (LSTM) pertama kali diperkenalkan pada 1997 oleh Hochreiter dan Schmidhuber, sebagai alternatif dari *Recurrent Neural Network*

(RNN). Jaringan RNN Metode LSTM mengatasi masalah tersebut dengan menerapkan konsep *memory cell* yang dapat menyimpan informasi dalam jangka waktu yang lebih lama (Okut, 2021). LSTM bisa belajar menghubungkan jeda waktu yang lebih dari 1000 langkah waktu dengan menjaga aliran error tetap stabil melalui *constant error carrousels* (CEC) . Dengan keunggulan itu, LSTM mampu mempelajari data sekuensial panjang dengan baik, sehingga LSTM mampu bekerja dengan baik dalam mengolah pemodelan bahasa seperti analisis sentimen dan pemrosesan bahasa alami. LSTM menerapkan konsep *memory cell* yang meliputi empat gerbang: *input gate, forget gate,* dan *output gate* (Yu et al., 2019).



Gambar 2.1 Struktur gerbang LSTM (Hasan et al., 2019)

Forget gate memiliki peran mengontrol informasi lama yang perlu dihapus dari sel memori (Greff et al., 2017a). Fungsinya adalah menentukan bagian mana dari memori yang tidak lagi relevan dan dapat dilupakan. Jika forget gate memberikan nilai mendekati 1, berarti informasi lama tetap disimpan dalam memori, sementara nilai mendekati 0 berarti informasi tersebut dihapus. Seperti *input gate, forget gate* juga menggunakan fungsi *sigmoid* pada persamaan 2.1 untuk memutuskan informasi mana yang harus dilupakan atau disimpan (Qian et al., 2016). Perhitungan *forget gate* dinyatakan dalam Persaman 2.2 berikut.

$$\sigma(x)=\frac{1}{1+e^{-x}} \tag{2.1}$$

$$f_t = \sigma(W_{fx}\cdot x_t+W_{fh}\cdot h_{t-1}+b_f) \tag{2.2}$$

Keterangan :

e	: bilangan Euler (bernilai sekitar 2,718)
f_t	: nilai *forget gate*
σ	: fungsi *sigmoid*
W_{fx}	: bobot *forget gate* yang didapat selama pelatihan
h_{t-1}	: *hidden state* dari waktu sebelumnya
x_t	: input pada waktu t
b_f	: *forget gate bias*

Input gate berfungsi mengontrol informasi baru yang akan dimasukkan ke dalam sel memori. Gerbang ini menentukan seberapa banyak data dari input saat ini yang akan disimpan dalam memori LSTM. Jika gerbang ini terbuka, informasi baru dapat diperbarui di sel memori. Sebaliknya, jika tertutup, data baru tersebut tidak akan diterima (Qian et al., 2016). Fungsi aktivasi *sigmoid* digunakan di input gate untuk menghasilkan nilai antara 0 dan 1, di mana nilai 0 menunjukkan bahwa informasi tidak akan dimasukkan, dan nilai 1 menunjukkan bahwa informasi akan diterima sepenuhnya. Berikut Persamaan 2.3 dan 2.4 yang menyatakan perhitungan *input gates*

$$i_t = \sigma(W_{ix}\cdot x_t+W_{ih}\cdot h_{t-1}+b_f) \tag{2.3}$$

$$\tilde{C}_t = \tanh(W_{\tilde{C}x}\cdot x_t+W_{\tilde{C}}\cdot h_{t-1}+b_{\tilde{C}x}) \tag{2.4}$$

Keterangan :

i_t	: nilai *input gate*
σ	: fungsi *sigmoid*
$\tilde{C}_t$	: nilai *memory candidate*
W_c	: bobot memori kandidat

Selanjutnya, informasi pada *cell state* awal diperbarui ke *cell state* yang baru dengan persamaan berikut.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (2.5)$$

Keterangan :

C_t : nilai *cell state* baru
$\tilde{C}_t$: nilai *memory candidate*
C_{t-1} : nilai *cell state* sebelumnya

Output gate bertugas mengontrol informasi yang akan dikeluarkan dari *cell state* sebagai *output* (Greff et al., 2017b). Gerbang ini memutuskan bagian mana dari informasi dalam sel memori yang relevan untuk digunakan sebagai *output* pada langkah waktu saat ini. Selain itu, informasi ini juga berperan dalam perhitungan di langkah waktu berikutnya. *Output gate* menggunakan fungsi *sigmoid* untuk memilih informasi yang akan diteruskan ke *output*, dan fungsi *tanh* digunakan untuk menentukan nilai output yang akan dikeluarkan yang berkisar dari -1 sampai 1. Fungsi tanh dinyatakan dalam persamaan 2.6. Perhitungan *output gate* dinyatakan dalam persamaan 2.7 dan 2.8 berikut.

$$tanh(z) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.6)$$

$$o_t = \sigma(W_{ox} \cdot x_t + W_{oh} \cdot h_{t-1} + b_o) \quad (2.7)$$

$$h_t = o_t \cdot tanh(C_t) \quad (2.8)$$

Keterangan :

o_t : nilai *output gate*
σ : fungsi *sigmoid*
h_t : nilai *hidden state*
C_t : *cell state*

2.5 *Bidirectional Long Short-Term Memory*

Bidirectional Long Short-Term Memory (Bi-LSTM) adalah varian dari *Long Short-Term Memory* (LSTM) yang memungkinkan jaringan saraf untuk mempertimbangkan konteks sekuensial dari kedua arah, baik dari masa lalu (*forward*) maupun masa depan (*backward*) (Sutskever et al., 2014). Dalam arsitektur standar LSTM, informasi hanya mengalir dalam satu arah, yaitu dari masa lalu ke masa depan. Bi-LSTM memperluas kemampuan ini dengan menggabungkan dua LSTM, satu memproses input secara maju (*forward*) dan yang lainnya secara mundur (*backward*). Hal ini memungkinkan model untuk menangkap konteks dari seluruh urutan data, sehingga meningkatkan kemampuan untuk memahami informasi yang bergantung pada keseluruhan konteks, baik sebelum maupun sesudah titik tertentu dalam urutan (Siami-Namini et al., 2019).

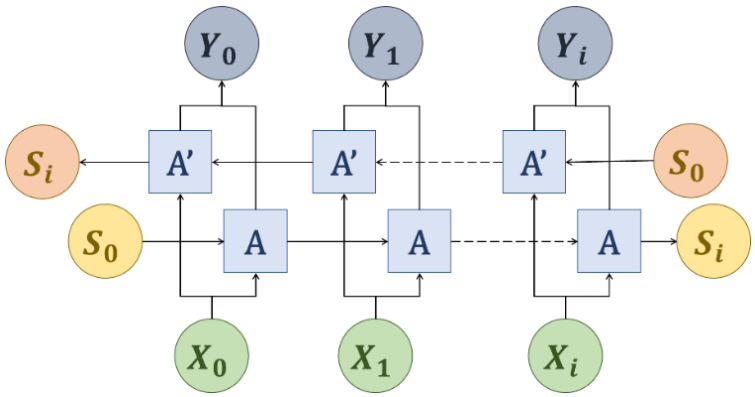
Mekanisme kerja Bi-LSTM melibatkan dua lapisan LSTM yang berjalan secara paralel. Lapisan pertama memproses urutan input dari awal hingga akhir, sementara lapisan kedua memprosesnya dari akhir hingga awal. Setelah itu, hasil dari kedua arah ini digabungkan atau dirata-rata untuk menghasilkan output akhir (Zhang et al., 2018). Penggabungan sekuens tersebut dinyatakan dalam persamaan 2.9 berikut (Ghojogh & Ghodsi, 2023).

$$y_t = \sigma(\overrightarrow{h_t}, \overleftarrow{h_t}) \tag{2.9}$$

Keterangan :

y_t : vektor gabungan
σ : fungsi *concatenation*
$\overrightarrow{h_t}$: *hidden states* pada *forward pass*
$\overleftarrow{h_t}$: *hidden states* pada *backward pass*

Dengan cara ini, Bi-LSTM dapat mempertimbangkan informasi dari seluruh urutan, yang membuatnya sangat cocok untuk tugas-tugas yang membutuhkan pemahaman penuh dari konteks, seperti pemrosesan bahasa alami (NLP), di mana makna suatu kata sering kali dipengaruhi oleh kata-kata di sekitarnya, baik sebelum maupun sesudahnya (Mohammad Masum et al., 2019).



Gambar 2.2 Lapisan-lapisan Bi-LSTM (Imrana et al., 2021)

Diagram pada Gambar 2.2 menunjukkan struktur lapisan Bi-LSTM dimana A dan A'adalah *nodes* berupa lapisan Bi-LSTM. Simbol X_i bertindak sebagai input dan Y_i sebagai *output* yang didapat dari gabungan *node* A dan A' (Cui et al., 2020). Pemrosesan dua arah dapat diamati dari dua arah panah *hidden state* S_0 ke S_i yang berbeda, yang masing-masing bergerak maju dari arah depan dan belakang. Pemrosesan pada Bi-LSTM melibatkan beberapa jaringan yang terdiri dari *cell state, forget gate, input gate* dan *output gate* seperti halnya jaringan pada lapisan LSTM.

Perbedaan utama antara LSTM biasa dan Bi-LSTM adalah arah aliran informasi (Huang et al., 2015a). LSTM standar hanya melihat data dari satu arah

(forward) dan biasanya digunakan ketika urutan masa depan tidak tersedia atau tidak relevan. Sebaliknya, Bi-LSTM melihat data dari dua arah, yang memungkinkan pemahaman yang lebih kaya tentang konteks. Meskipun Bi-LSTM lebih kuat dalam menangkap informasi konteks penuh, hal ini juga berarti bahwa Bi-LSTM memerlukan lebih banyak memori dan waktu komputasi dibandingkan dengan LSTM biasa karena ada dua lapisan yang harus dilatih secara bersamaan.

Bi-LSTM sangat berguna untuk tugas-tugas di mana konteks dari kedua arah sangat penting. Contohnya, dalam pemrosesan teks, seperti *named entity recognition* (NER), *parsing* bahasa alami, terjemahan mesin, dan analisis sentimen, di mana makna atau hubungan antara kata-kata dapat dipengaruhi oleh kata-kata di depan maupun di belakangnya (Nallapati et al., 2016). Namun, jika data bersifat sekuensial dengan aliran waktu atau dalam kasus pemrosesan *real-time* di mana informasi masa depan tidak tersedia, LSTM standar lebih tepat digunakan (Józefowicz et al., 2015).

BAB III

DESAIN DAN IMPLEMENTASI

3.1 Pengumpulan Data

Penelitian ini menggunakan data umpan balik pengguna *website* SiBayar Pondok Pesantren Sabilurrosyad Gasek Malang, yang berfokus pada fitur *herregistrasi* dan syahriah. Peneliti mengumpulkan data umpan balik *website* pada semester genap tahun ajaran pesantren pada bulan Februari 2025. Responden adalah santri Pondok Pesantren Sabilurrosyad, dengan pengumpulan data melalui kuesioner daring (*Google Form*) yang berisi 6 pertanyaan seputar pengalaman, keluhan maupun saran pengguna dalam menggunakan fitur di *website* yang tertera dalam Lampiran 1.

3.2 Input Data

Input sistem adalah jawaban responden terhadap 5 pertanyaan yang digabungkan menjadi 3 kategori: fitur *login*, pengisian form daftar ulang, dan keseluruhan pengalaman pengguna. Sebelum digabungkan, peneliti menyeleksi jawaban kuesioner untuk memastikan kesesuaian jawaban responden dengan pertanyaan dalam kuesioner. Setiap responden menghasilkan 3 data teks yang mewakili pendapat mereka tentang masing-masing fitur. Kemudian umpan balik tersebut diringkas oleh ahli untuk menjadi fitur ringkasan manusia. Tabel 3.1 adalah contoh data yang akan digunakan serta dua fitur *user_feedback* (umpan balik pengguna) dan *human_summary* (ringkasan ahli) yang akan menjadi input bagi sistem.

Tabel 3.1 Contoh *input data*

No.	*Feedback* Pengguna	Ringkasan Manusia
1	Website sibayar terlalu banyak form yang harus saya inputkan. Setiap ingin melanjutkan ke form berikutnya, kok harus simpan data dulu ya. Jadinya proses daftar ulang agak ribet karena banyak data yang harus saya masukkan. Menurut saya daftar ulang itu ya tinggal bayar saja tidak usah input banyak data.	Proses input form ribet banyak data daftar ualng tidak usah input banyak data
2	Akses sangat mudah dan rincian syahriah bisa dilihat setiap bulannya beserta password wifi yang diberikan setiap bulan ketika setelah saya bayar.	Akses mudah, rincian syahriah dan password wifi tersedia setiap bulan
3	Mungkin tata letak menu di bawah itu sebaiknya satu saja, boleh disamping atau dibawah. Karena bisa membuat tampilan terlalu belibet.	tata letak menu di bawah sebaiknya satu saja, agar tidak belibet.

3.3 Desain Sistem

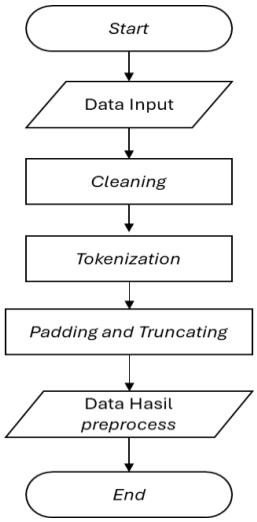
Alur sistem yang dinyatakan pada Gambar 3.1 secara garis besar dimulai dari input data, pemrosesan data, hingga hasil pemrosesan data (*output*). Data input akan melalui tahap *data preprocessing* yang meliputi beberapa langkah untuk membersihkan data agar di tahap selanjutnya proses akan berjalan lebih efektif. Selanjutnya, data hasil *preprocessing* akan diproses dalam model algoritma Bi-LSTM untuk meringkas teks. Selanjutnya hasil ringkasan dari Bi-LSTM dievaluasi untuk melihat seberapa bagus performa algoritma tersebut dalam menjalankan tugas peringkasan teks.

Gambar 3.1 Alur Desain Sistem

3.4 Data *Preprocessing*

Tahap data *preprocessing* bertujuan untuk meningkatkan kualitas data dengan mengatasi ketidakseragaman data dan inkonsistensi data agar siap untuk

analisis lebih lanjut, (Singh & Gaur, 2019). Tahap tersebut dalam penelitian ini divisualisasikan pada Gambar 3.2 yang meliputi beberapa langkah seperti pembersihan data hingga menyeragamkan panjang input.



Gambar 3.2 Tahap Preprocessing

3.4.1 *Cleaning*

Cleaning merupakan tahap pertama dalam *data preprocessing* berdasarkan diagram Gambar 3.2 yang bertujuan untuk untuk memastikan teks dalam dataset bersih dan siap digunakan untuk pemrosesan lebih lanjut. Contoh sebelum dan sesuah data *cleaning* dinyatakan pada Tabel 3.2. Dalam melakukan *data cleaning* pertama, digunakan *regex* (*regular expression*) untuk menghilangkan semua karakter yang bukan huruf, seperti angka, tanda baca, dan simbol khusus. Selain itu, diterapkan *case folding*, yaitu proses mengubah seluruh huruf dalam teks menjadi huruf kecil (*lowercase*), yang bertujuan untuk menyamakan bentuk kata yang ditulis dengan huruf besar maupun kecil. Kedua proses ini bersama-sama

memastikan bahwa teks bersih, seragam, dan siap digunakan untuk tahap tokenisasi dan proses pemodelan lebih lanjut.

Tabel 3.2 Contoh *Data Cleaning*

No.	Sebelum *cleaning*	Sesudah *cleaning*
1	Website sibayar terlalu banyak form yang harus saya inputkan. Setiap ingin melanjutkan ke form berikutnya, kok harus simpan data dulu ya. Jadinya proses daftar ulang agak ribet karena banyak data yang harus saya masukkan. Menurut saya daftar ulang itu ya tinggal bayar saja tidak usah input banyak data.	website sibayar terlalu banyak form yang harus saya inputkan setiap ingin melanjutkan ke form berikutnya kok harus simpan data dulu ya jadinya proses daftar ulang agak ribet karena banyak data yang harus saya masukkan menurut saya daftar ulang itu ya tinggal bayar saja tidak usah input banyak data
2	Akses sangat mudah dan rincian syahriah bisa dilihat setiap bulannya beserta password wifi yang diberikan setiap bulan ketika setelah saya bayar.	akses sangat mudah dan rincian syahriah bisa dilihat setiap bulannya beserta password wifi yang diberikan setiap bulan ketika setelah saya bayar
3	Mungkin tata letak menu di bawah itu sebaiknya satu saja, boleh disamping atau dibawah. Karena bisa membuat tampilan terlalu belibet.	mungkin tata letak menu di bawah itu sebaiknya satu saja, boleh disamping atau dibawah karena bisa membuat tampilan terlalu belibet

3.4.2 *Tokenization*

Tokenization adalah proses memecah teks menjadi unit-unit kecil yang disebut *token*. Tujuannya adalah untuk mengubah teks menjadi bentuk yang lebih mudah dipahami oleh model pembelajaran mesin, karena model tidak dapat langsung bekerja dengan teks mentah. Dengan memecah teks menjadi token, setiap kata dalam teks dapat dianalisis dan diproses secara individual oleh model.

Dalam proses tokenization, penelitian ini menggunakan library Keras *Tokenizer* yang memecah kalimat menjadi kata tunggal lalu melakukan vektorisasi (mengubah kata menjadi representasi numerik) sehingga berbentuk token.

Tabel 3.3 Contoh Tokenization

No.	Sebelum *tokenization*	Sesudah *tokenization*
1	website sibayar terlalu banyak form yang harus saya inputkan setiap ingin melanjutkan ke form berikutnya kok harus simpan data dulu ya jadinya proses daftar ulang agak ribet karena banyak data yang harus saya	[68,25,84,38,7,3,16,9,100, 178,31,7,826,16,111,4,92, 465,467,116,8,6,155,295,2 3,38,4,3,16,9,243,78,9,8,6,

No.	Sebelum *tokenization*	Sesudah *tokenization*
	masukkan menurut saya daftar ulang itu ya tinggal bayar saja tidak usah input banyak data	89,465,436,51,70,2,246,163,38,4]
2	akses sangat mudah dan rincian syahriah bisa dilihat setiap bulannya beserta password wifi yang diberikan setiap bulan ketika setelah saya bayar	[26,19,12,1,259,27,5,118,100,379,223,44,18,3,261,100,55,45,77,9,51]
3	mungkin tata letak menu di bawah itu sebaiknya satu saja, boleh disamping atau dibawah karena bisa membuat tampilan terlalu belibet	[60,524,364,59,14,222,89, 54,146,70,525,526,34,527, 23,5,254,73,84,528]

Pada Tabel 3.3 di atas, setiap angka yang ada pada *tokenization* ini mewakili indeks yang ada pada kamus kosa kata yang dibentuk pada proses *tokenization,* dengan setiap angka mewakili kata yang berbeda.

3.4.3 *Padding and Truncating*

Penelitian ini menggunakan metode Bi-LSTM yang membutuhkan *fixed-length input* karena data akan melalui *training* dalam bentuk *batch.* Oleh karena itu, diperlukan penyeragaman ukuran input agar algoritma bekerja dengan baik(Bier et al., 2022). *Padding* menambah panjang input ke ukuran tetap, sementara truncating memotong teks yang terlalu panjang agar sesuai. Misalkan sebuah teks memiliki 20 token dengan maksimal token yang ditentukan sebanyak 30. Maka, teks tersebut akan ditambahkan *padding* berupa token berisi angka 0 untuk menyeragamkan ukuran input. Apabila teks memiliki panjang melebihi maksimum token, maka teks akan di-*truncate* agar ukuran teks sama seperti maksimal token(Ding et al., 2024). Implementasi *padding* (apabila panjang *sequence*=50) dapat dilihat di Tabel 3.4

Tabel 3.4 Contoh *Padding*

No.	Sebelum *padding*	Sesudah *padding*
1	[[68,25,84,38,7,3,16,9,100,178,31,7,826,16,111,4,92,465,467,116,8,6,155,295,23,38,4,3,16,9,243,78,9,8,6,89,465,436,51,70,2,246,163,38,4]]	[68,25,84,38,7,3,16,9,100,178,31,7,826,16,111,4,92,465,467,116,8,6,155,295,23,38,4,3,16,9,243,78,9,8,6,89,465,436,51,70,2,246,163,38,4, 0, 0, 0]]

No.	Sebelum *padding*	Sesudah *padding*
2	[26,19,12,1,259,27,5,118,100,379,223,44,18, 3,261,100,55,45,77,9,51]	[26, 19, 12, 1, 259, 27, 5, 118, 100, 379, 223, 44, 18, 3, 261, 100, 55, 45, 77, 9, 51, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
3	[60,524,364,59,14,222,89,54,146,70,525,526, 34,527,23,5,254,73,84,528]	[60, 524, 364, 59, 14, 222, 89, 54, 146, 70, 525, 526, 34, 527, 23, 5, 254, 73, 84, 528, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Berdasarkan Tabel 3.4 di atas, *padding* menambahkan nilai 0 ke dalam data untuk memastikan panjang input pada setiap data berjumlah sama. Berikut Tabel 3.5 apabila batas panjang sekuens = 20 sementara input memiliki panjang 55 token.

Tabel 3.5 Contoh *Truncating*

No.	Sebelum *truncating*	Sesudah *truncating*
1	[[68,25,84,38,7,3,16,9,100,178,31,7,826,16,111,4,92, 465,467,116,8,6,155,295,23,38,4,3,16,9,243,78,9,8,6, 89,465,436,51,70,2,246,163,38,4]]	[68, 25, 84, 38, 7, 3, 16, 9, 100, 178, 31, 7, 826, 16, 111, 4, 92, 465, 467, 116]
2	[26,19,12,1,259,27,5,118,100,379,223,44,18,3,261,10 0,55,45,77,9,51]	[26, 19, 12, 1, 259, 27, 5, 118, 100, 379, 223, 44, 18, 3, 261, 100, 55, 45, 77, 9]
3	[60, 524, 364, 59, 14, 222, 89, 54, 146, 70, 525, 526, 34, 527, 23, 5, 254, 73, 84, 528]	[60, 524, 364, 59, 14, 222, 89, 54, 146, 70, 525, 526, 34, 527, 23, 5, 254, 73, 84, 528]

Pada Tabel 3.5 di atas, *truncating* memastikan data tidak melebihi batas panjang input untuk menjaga keseragaman panjang data.

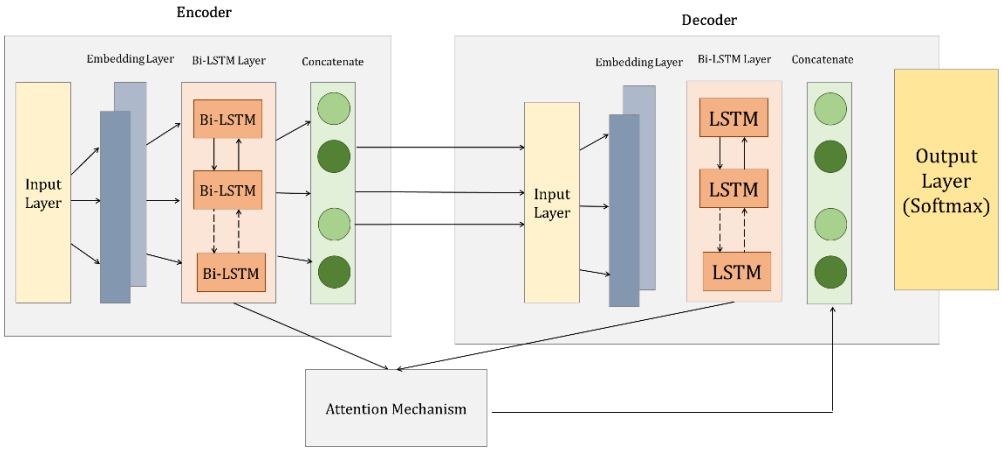
3.5 Bi-LSTM untuk Peringkasan Teks

Penelitian ini menggunakan pendekatan abstraktif dalam melakukan tugas peringkasan teks menggunakan metode Bi-LSTM. Peringkasan teks dengan pendekatan abstraktif bekerja dengan cara menangkap ide pokok yang digunakan sebagai dasar bagi pembentukan ringkasan teks. Metode Bi-LSTM dipilih karena

performanya yang bagus dalam tugas peringkasan otomatis dalam berbagai penelitian terdahulu.

Bi-LSTM memiliki dua lapisan, satu lapisan untuk memproses input dari awal hingga akhir urutan (*forward pass*), dan lapisan lainnya untuk memprosesnya dari belakang ke awal urutan (*backward pass*) (Shabanian et al., 2017). Tujuan dari diberlakukannya dua arah pemrosesan input ini adalah untuk memahami hubungan suatu kata dalam kalimat berdasarkan pola kata sebelum dan sesudahnya.

3.5.1 Arsitektur Model



Gambar 3.3 Arsitektur Bi-LSTM

Model yang dibangun dalam arsitektur di penelitian ini merupakan model *Sequence-to-Sequence (Seq2Seq)* berbasis Bi-LSTM dengan mekanisme *Attention* sebagaimana pada Gambar 3.3. Oleh karena menggunakan model *seq2seq,* model ini terdiri dari dua bagian utama, yaitu *encoder* dan *decoder*, yang bekerja secara bersamaan untuk menangkap informasi dari teks input dan menghasilkan keluaran yang lebih terstruktur(Wu & Xing, 2024).

Berikut adalah rincian input dan *output* dari setiap lapisan pada arsitektur peringkasan teks Bi-LSTM.

Tabel 3.6 Lapisan dalam arsitektur model

No.	Layer (type)	Output Shape	Param #	Connected to
1	**decoder_input**	(None, None)	0	-
2	(InputLayer)			
3	**encoder_input**	(None, 50)	0	-
4	(InputLayer)			
5	**embedding_1**	(None, None, 100)	jumlah vocab pada dataset * 100	encoder_input[0]…
6	(Embedding)			decoder_input[0]…
7	**bidirectional**	[(None, 50, 200),	160,800	embedding_1[0][0]
8	(Bidirectional)	(None, 100),		
		(None, 100),		
		(None, 100),		
		(None, 100)]		
9	**concatenate**	(None, 200)	0	bidirectional[0]…
	(Concatenate)			bidirectional[0]…
10	**concatenate_1**	(None, 200)	0	bidirectional[0]…
	(Concatenate)			bidirectional[0]…
11	**lstm_1 (LSTM)**	[(None, None, 200),	240,800	embedding_1[1][0]…
		(None, 200),		concatenate[0][0]…
		(None, 200)]		concatenate_1[0]…
12	**attention**	(None, None, 200)	0	lstm_1[0][0],
	(Attention)			bidirectional[0]…
13	**concatenate_2**	(None, None, 400)	0	lstm_1[0][0],
	(Concatenate)			attention[0][0]
14	**dense (Dense)**	(None, None, 865)	346,865	concatenate_2[0]…

Tabel 3.6 di atas berisi urutan arsitektur model dari yang paling awal hingga akhir. Selain itu pada tabel juga dirinci jumlah dimensi, jumlah input yang diterima, jumlah parameter, dan juga hubungan antar lapisan pada arsitektur.

3.5.1.1 *Input Layer*

Arsitektur model memiliki dua *input layer* yang masing-masing menjadi input bagi lapisan *encoder* dan *decoder*. Berdasarkan Tabel 3.7 *decoder_input* menerima urutan target yang akan diprediksi, dengan bentuk *output (None, None),* di mana dimensi pertama menunjukkan ukuran batch yang dapat bervariasi, sementara dimensi kedua menunjukkan panjang urutan yang juga bervariasi,

biasanya sesuai dengan panjang teks target dalam tugas seperti ringkasan atau penerjemahan. Input ini tidak memiliki parameter, melainkan hanya mendefinisikan bentuk input untuk decoder.

Sementara itu, *encoder_input* menerima urutan input dari sumber, seperti teks yang ingin diproses atau diterjemahkan, dengan panjang tetap 50 token (atau sub-token). Bentuk outputnya adalah (None, 50), yang berarti model akan menerima urutan input dengan panjang yang sudah ditentukan (50 token). Sama seperti *decoder_input*, lapisan ini juga tidak memiliki parameter dan hanya berfungsi untuk memberi bentuk pada data input yang akan diteruskan ke lapisan berikutnya, yakni *embedding layer*.

3.5.1.2 *Embedding Layer*

Setelah menerima input, data diteruskan ke *Embedding Layer*, yang bertugas mengubah urutan token (kata atau sub-kata) menjadi vektor berdimensi 100. (Satvika et al., 2021). Vektor-vektor ini merupakan representasi kontinu dari kata-kata dalam ruang vektor, yang memungkinkan model untuk memahami hubungan semantik antar kata. Penelitian ini menggunakan bantuan *Embedding* pada *library* Keras. Berdasarkan apa yang ditulis pada dokumentasi di *website* Keras, *Embedding* bekerja dengan mengubah input berupa bilangan bulat positif (yang mewakili kata atau token) menjadi *dense vector* tetap berdimensi tertentu.

Output dari *embedding* layer memiliki bentuk (None, None, 100), di mana dimensi pertama adalah ukuran *batch*, dimensi kedua adalah panjang urutan input

(*timestep*), dan dimensi ketiga adalah dimensi vektor embedding yang berjumlah 100.

Lapisan *embedding* ini menggunakan matriks *embedding* yang *trainable* (dapat dilatih), dengan total parameter sebesar jumlah *vocab* dari dataset dikalikan dengan besar dimensi pada *embedding dimesion. Embedding Layer* untuk Encoder bertugas untuk mengubah urutan token dari input sumber. Sementara itu, *Embedding Layer* untuk Decoder memiliki fungsi yang serupa, yaitu mengubah urutan token target (kata yang akan diprediksi) menjadi representasi vektorial. *Decoder* memerlukan embeddings yang terpisah karena urutan yang diprediksi berbeda dari input sumber. (Kothari et al., 2020)

Berikut adalah contoh dari *output Embedding layer* yang memiliki ukuran *batch size 1*, dengan panjang *timestep* 50 serta ukuran dimensi *embedding* sebesar 100 (1, 50, 100).

Tabel 3.7 Contoh *output Embedding Layer dengan shape* (1, 50, 100).

Timestep (t)	**Embedding**
1	[-0.0551 -1.0308 -0.52812 -0.26787 -0.10296 -0.2622 0.3692 0.30433 -0.6336 0.6306 . . . -0.2471 -0.3438 -0.7207 0.3104 -0.7552 0.1975 0.38782287 -0.1995 -0.00418 -0.1883]
2	[3.3736 1.1842 -1.2748 1.965 -7.2215 1.3425 6.1860 -3.694 1.3483 -1.3079 1.104 . . . 2.9600 1.8100 1.1270 8.1700 1.8530 -3.4330 2.3990 -8.8600 -8.8100 -3.1100 8.3000 -4.0200]
3	[0.4764 0.0571 -0.0589 0.1272 -0.6504 -0.0418 -0.0369 -0.2163 -0.0945 -0.459 . . . -0.0955 0.1809 0.5048 -0.1647 0.2716 0.0577 0.2192 -0.1762 -0.1367 0.2191]

3.5.1.3 Lapisan *Encoder*

Pada lapisan *encoder*, *Bidirectional Long Short-Term Memory* (Bi-LSTM) digunakan untuk memahami konteks teks secara lebih luas dengan membaca urutan kata dari dua arah(*forward* dan *backward pass*) (Zhu & Yu, 2017). Informasi pada *forward pass* dimulai dari awal hingga akhir sekuens. *Forward pass* memproses urutan input dari awal hingga akhir, secara bertahap memperbarui hidden state di setiap langkah waktu berdasarkan informasi yang telah dilihat hingga titik waktu tersebut. Proses ini melibatkan perhitungan pada setiap langkah waktu yang diinisasi dengan t.

Diketahui hasil *word embedding* dari kumpulan kata ‘website sibayar form input lanjut’ yang bertindak sebagai input $x_1$hingga x_5 sebagai berikut.

[-0.00719, 0.004232890431, 0.002163394678, 0.001528491210, 0.002764832145], [0.007696646, 0.007696646348, 0.009120642456, 0.003176542198, 0.004583791203], [-0.009508036273, -0.009508036829, 0.007779993653, 0.005931826401, 0.006829145702], [0.007425896112, 0.008356192045, 0.009234578192, 0.010689405784, 0.011897625341], [0.012879405612, 0.013743218945, 0.014634527109, 0.015407829681, 0.016532194817]].

Penelitian ini menerapkan Bi-LSTM dengan menggunakan *library* dari Keras *layers.Bidirectional* dengan parameter *layer* yang diisi dengan LSTM. Bi-LSTM terbentuk dari kelas *Bidirectional* yang bertindak untuk membungkus lapisan LSTM, agar LSTM dapat memproses data sekuensial dari dua arah sekaligus, yaitu dari awal ke akhir (*forward*) dan dari akhir ke awal (*backward*). (Géron, 2022)

Selain itu, penelitian ini menerapkan beberapa parameter pada kode LSTM dengan menggunakan class *Bidirectional(LSTM(hidden_units, return_state=True*

return_sequences=True, kernel_initializer=glorot_uniform())) yang memiliki kegunaan, antara lain: *hidden units* menentukan jumlah neuron dalam *hidden states* pada LSTM, yang dalam penelitian ini menggunakan 100 neuron (Reimers & Gurevych, 2017); *return_sequences=True* menghasilkan output pada setiap *timestep* untuk urutan lebih lanjut; *return_state=True* mengembalikan *output*, *hidden state*, dan *cell state* sebagai keluaran dari Bi-LSTM; *kernel_initializer* mengatur inisialisasi bobot; *Dropout* adalah teknik regularisasi yang secara acak menonaktifkan sebagian unit atau koneksi pada input selama pelatihan untuk mencegah *overfitting* (Srivastava et al., 2014); *Recurrent_dropout* adalah teknik regularisasi yang diterapkan pada koneksi *recurrent* (internal) antara *timestep* dalam lapisan LSTM..(Keras, n.d.)

Penelitian ini menggunakan *kernel_initializer* berupa distribusi *uniform Glorot Initialization* untuk menentukan *weight* awal untuk setiap *gates*. Proses inisialisasi bobot menggunakan *Glorot Initialization* agar membantu model dalam mengoptimisasi nilai *loss function* dibandingkan metode inisialisasi yang lain seperti *Random Initialization* (Glorot et al., 2011) .

Misalkan diketahui terdapat 1 neuron di LSTM, 1 input fitur, dan 1 neuron *hidden state,* yang dinyatakan oleh $n_{in} = 1$ dan $n_{out} = 1$ pada Persamaan 3.1 berikut.

$$U = \left(-\frac{\sqrt{6}}{\sqrt{n_{in}+n_{out}}}, \frac{\sqrt{6}}{\sqrt{n_{in}+n_{out}}}\right) \tag{3.1}$$

$$U = \left(-\frac{\sqrt{6}}{\sqrt{1+1}}, \frac{\sqrt{6}}{\sqrt{1+1}}\right)$$

$$U = (-1.732, 1.732)$$

Berdasarkan hasil dari distribusi Glorot Uniform, maka diketahui bobot-bobot untuk setiap *gates* pada layer *forward* dan *backward pass* yang didapat dari *sampling* acak rentang $U = (-1.732, 1.732)$ dinyatakan dalam Tabel 3.7 dan 3.8.

Tabel 3.8 Contoh Inisialisasi Bobot *Forward Pass*

No.	***Gate***	**Bobot *gate***	***Hidden state***
1	*Input Gate*	$W_{ix} = 0.5$	$W_{ih} = -0.3$
2	*Forget Gate*	$W_{fx} = -0.2$	$W_{fh} = 0.1$
3	*Output Gate*	$W_{ox} = 0.7$	$W_{oh} = -0.1$
4	*Candidate Cell*	$W_{\tilde{c}x} = 0.4$	$W_{\tilde{c}h} = -0.2$

Tabel 3.9 Contoh Inisialisasi Bobot *Backward Pass*

No.	***Gate***	**Bobot *gate***	***Hidden state***
1	*Input Gate*	$W_{ix} = 0.4$	$W_{ih} = -0.2$
2	*Forget Gate*	$W_{fx} = -0.1$	$W_{fh} = 0.2$
3	*Output Gate*	$W_{ox} = 0.6$	$W_{oh} = -0.1$
4	*Candidate Cell*	$W_{\tilde{c}x} = 0.3$	$W_{\tilde{c}h} = -0.2$

Berdasarkan hasil vector pada *word embedding*, maka diketahui input pada *timestep* pada *forward pass* dengan $t = 1$ adalah -0.00719 (x_1). Langkah berikutnya adalah menghitung x_1 terhadap semua *gate*. Rumus 3.2 menunjukkan bagaimana kalkulasi *forget gate* menggunakan fungsi sigmoid dengan $h_0 = 0$ (*hidden state* awal).

$$f_t = \sigma(W_{fx} \cdot x_t + W_{fh} \cdot h_{t-1} + b_f) \tag{3.2}$$

$$f_1 = \sigma((-0.2) \cdot (-0.00719) + 0.1 \cdot 0 + 0)$$

$$f_1 = \sigma(0.001438)$$

$$f_1 = (0.5004)$$

Setelah memilah informasi yang diperlukan di tahap *forget gate,* langkah selanjutnya adalah memperbarui informasi yang akan disimpan dala *cell state.* Sebuah fungsi *sigmoid* pada Rumus 3.3 digunakan untuk menentukan nilai mana yang akan diperbarui (*update)*. Selanjutnya, terdapat sebuah fungsi *tanh* yang

ditunjukkan pada Rumus 3.4 yang berfungsi untuk membuat vektor dari kandidat nilai *cell state (Abdelouahab et al., 2017).* Gabungan dari kedua fungsi tersebut digunakan untuk memperbarui nilai *state.*

$$i_t = \sigma(W_{ix} \cdot x_t + W_{ih} \cdot h_{t-1} + b_f) \tag{3.3}$$

$$i_1 = \sigma((-0.5) \cdot (-0.00719) + (-0.3) \cdot 0 + 0)$$

$$i_1 = \sigma(-0.003595)$$

$$i_1 = 0.4991$$

$$\tilde{C}_t = \tanh(W_{\tilde{C}x} \cdot x_t + W_{\tilde{C}} \cdot h_{t-1} + b_{\tilde{C}x}) \tag{3.4}$$

$$\tilde{C}_1 = \tanh((0.4) \cdot (-0.00719) + (-0.2) \cdot 0 + 0)$$

$$\tilde{C}_1 = \tanh(-0.002876)$$

$$\tilde{C}_1 = (-0.002876)$$

Pada tahap ini, pembaruan keadaan *cell state* C_{t-1} menjadi *cell state* baru C_t dilakukan sebagaimana pada Persamaan 3.5. Selanjutnya, nilai kandidat baru $i_t * \tilde{C}_t$ ditambahkan, yang telah disesuaikan berdasarkan seberapa besar keputusan untuk memperbarui setiap nilai keadaan.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{3.5}$$

$$C_1 = 0.5004 * 0 + 0.4991 \cdot (-0.002876) = -0.001434$$

Langkah terakhir dari *forward pass* adalah menentukan *output* mana yang perlu diambil berdasarkan *cell state.* Pertama, fungsi *sigmoid* pada rumus 3.6 digunakan untuk menentukan bagian mana dari keadaan sel yang akan dikeluarkan.

$$o_t = \sigma(W_{ox} \cdot x_t + W_{oh} \cdot h_{t-1} + b_o) \tag{3.6}$$

$$o_1 = \sigma((0.7) \cdot (-0.00719) + (-0.1) \cdot 0 + 0)$$

$$o_1 = \sigma(0.005033)$$

$$o_1 = 0.4987$$

Selanjutnya, keadaan sel tersebut diproses melalui fungsi *tanh* pada rumus 3.7 untuk membatasi nilainya antara -1 dan 1, dan kemudian dikalikan dengan output dari gerbang sigmoid.

$$h_t = o_t \cdot tanh(C_t) \tag{3.7}$$

$$h_1 = 0.4987 \cdot (-0.001434) = -0.000715$$

Nilai yang didapatkan pada h_1 dan o_1 akan digunakan sebagai input pada tahap selanjutnya hingga proses berulang sampai *timestep* terakhir.

Pada *backward pass,* input diproses dimulai dari akhir hingga awal (x_T hingga x_1) dengan perhitungan yang sama seperti *forward pass* dengan melibatkan *forget gate, input gate,* dan *output gate*. Berikut adalah perhitungan *backward pass* pada *timestep* $t = T$ dengan input $x_T = 0.01653, h_T = 0, C_{T-1} = 0$.

$$f_T = \sigma((-0.1) \cdot (0.01653) + 0.2 \cdot 0 + 0) \; = \; \sigma(0.001653) = 0.4996$$

$$i_T = \sigma((0.4) \cdot (0.01653) + (-0.2) \cdot 0 + 0) \; = \; \sigma(0.006612) = 0.5017$$

$$\tilde{C}_T = \tanh((0.3) \cdot (0.01653) + (-0.2) \cdot 0 + 0) \; = \; \tanh(0.004959) = 0.4959$$

$$C_T = 0.4996 \cdot 0 + 0.5017 \; \cdot 0.004959 = 0.002489$$

$$o_T = \sigma((0.6) \cdot (0.01653) + (-0.1) \cdot 0 + 0) \; = \; \sigma(0.009918) = 0.5024$$

$$h_T = 0.5024 \cdot \tanh(0.002489) = 0.5024 \cdot 0.002489 = 0.00125$$

Proses perhitungan *backward pass* terus berulang hingga input mencapai awal dari sekuens x_1.

3.5.1.4 *Encoder Concatenate Layer*

Bi-LSTM menggabungkan (*concatenate*) *hidden states* dari *forward* dan *backward* LSTM. Penggabungan ini berhubungan dengan penggunaan argumen

return_sequences=True dan *return_state=True* karena keduanya mempengaruhi output dan status internal yang dikembalikan oleh LSTM. *return_sequences=True* memungkinkan output pada setiap timestep dari kedua arah untuk digabungkan, sementara *return_state=True* memberikan akses ke *hidden states* dan *cell states* dari kedua arah, yang juga dapat digabungkan atau digunakan untuk pemrosesan lebih lanjut dalam model.

Pada setiap *time step*, hidden *states* $h_1$dari *forward* LSTM dan $h_T$dari *backward* LSTM digabungkan untuk membentuk vektor *output* gabungan $y_t = \sigma(\overrightarrow{h_t}, \overleftarrow{h_t})$. Hasil akhir dari lapisan Bi-LSTM perhitungan manual pada satu *timestep forward* dan *backward* adalah vektor gabungan $y_T = \sigma(0.000715, \ldots, 0.00125)$ dan untuk seluruh *timestep* dapat direpresentasikan dalam bentuk vektor $Y_T = [y_{T-n}, \ldots, y_{T-1}]$. Selain *hidden states* Bi-LSTM juga menggabungkan *cell state* dari forward dan *backward* LSTM dengan cara yang serupa. *Output* berupa penggabungan *hidden state* dan *cell state* dari dua arah pemrosesan di Bi-LSTM digunakan oleh *decoder* pada model.

Selain gabungan *hidden state* dan *cell* state Bi-LSTM *Seq2Seq* memiliki *encoder outputs* yang berisi representasi tersembunyi dari setiap kata dalam input teks, yang diperoleh dengan menggabungkan *hidden state* dari *forward* dan *backward* LSTM pada setiap *timestep*. Output ini memiliki bentuk (*batch_size, timesteps, hidden_units* * 2) dan digunakan oleh *Attention Mechanism* untuk membantu *decoder* fokus pada bagian input yang paling relevan.

Misalnya, jika input adalah *"Website Sibayar terlalu banyak form", encoder outputs* berupa vektor numerik seperti [[0.12, 0.35, 0.88, ..., -0.21], [0.45, -0.32,

0.91, ..., 0.11], ...], di mana setiap vektor mewakili makna dari kata tertentu dalam konteks kalimat. *Encoder outputs* tidak langsung digunakan oleh *decoder* untuk menghasilkan teks, tetapi berfungsi sebagai sumber informasi bagi *Attention Layer* agar ringkasan yang dihasilkan tetap akurat dan kontekstual.

3.5.1.5 Lapisan *Decoder*

Lapisan *Decoder* didahului oleh *input layer,* dan *embedding layer* serta memiliki argumen parameter yang sama dengan lapisan encoder. *Decoder* dalam model ini berfungsi untuk menghasilkan teks *output* secara sekuensial, satu kata per langkah waktu (*timestep*) (Jorge et al., 2019). Proses *decoding* dimulai dengan menerima input awal, yaitu *hidden state* (h_T, hc_t) dari *encoder* serta *decoder input* yang didapatkan dari *embedding* .

$$X_{decoder} = \{x_{decoder1}, x_{decoder2}, x_{decoderT}\} \tag{3.8}$$

Jika *decoder* memproses teks target *"saya mendaftar ulang",* dan *embedding* memiliki dimensi 100, maka input ke decoder bisa direpresentasikan sebagai:

$$X_{decoder} = \begin{bmatrix} 0.21 & \dots & -0.34 \\ -0.45 & \dots & 0.67 \\ -0.45 & \dots & 0.67 \end{bmatrix}$$

Pada setiap *timestep* t, *decoder* menerima input *embedding* dari kata sebelumnya dan hidden state dari timestep sebelumnya:

$$h_t, c_t = \text{LSTM}(x_{decoder^t}, h_{t-1}, c_{t-1}) \tag{3.9}$$

Keterangan **:**

$x_{decoder^t}$	: input embedding kata ke-t dalam decoder.
h_{t-1} , c_{t-1}	: nilai *hidden state dan context vector* dari *timestep* sebelumnya.
LSTM	: menghasilkan h_t, c_t baru untuk *timestep* berikutnya

Decoder menghasilkan *output* berupa *hidden states* dan *context vector* yang akan menjadi input bagi *Attention Layer.* Output *decoder* tersebut selanjutnya akan dilakukan *concatenation* bersamaan dengan *output* dari *Attention Layer* untuk kemudian diteruskan hasilnya ke *fully connected layer.*

3.5.1.6 *Attention Mechanism*

Attention Mechanism dalam *model Seq2Seq* Bi-LSTM dengan Luong adalah teknik yang memungkinkan *decoder* untuk lebih selektif dalam memilih informasi dari *encoder outputs* saat menghasilkan setiap kata dalam output sequence. Tidak seperti pendekatan tradisional yang hanya mengandalkan *hidden state* terakhir dari *encoder*, Luong *Attention* memungkinkan model untuk melihat kembali seluruh *hidden state* dari *encoder* dan memberi bobot lebih pada bagian yang paling relevan dengan kata yang sedang dihasilkan. Dengan cara ini, model dapat menangkap hubungan yang lebih baik antara kata-kata dalam input dan output, sehingga lebih efektif dalam menangani input *sequence* yang panjang (Luong et al., 2015). Berikut adalah persamaan *attention score* yang menjadi unsur utama dalam *attention mechanism* ini.

$$e_t, j = h_t{}^{\top}, h_{encoder,j} \tag{3.10}$$

Keterangan**:**

e_t, j	: skor perhatian antara *decoder timestep* t dan *encoder* timestep j.
h_t	: h*idden state* dari *decoder* pada timestep saat ini.LSTM
$h_{encoder,j}$	: *hidden state* dari encoder pada timestep j.
$\top$	: Transposisi (*dot product* untuk menghitung skor relevansi).

Mekanisme ini bekerja dengan membandingkan *hidden state* dari *decoder* pada tiap langkah dengan semua *hidden state* dari *encoder*, lalu menentukan

attention weight (bobot) untuk masing-masing hidden state *encoder*. Bobot ini menunjukkan seberapa besar kontribusi setiap kata dalam input terhadap kata yang akan dihasilkan di *output*.

3.5.1.7 *Concatenation* pada *Decoder* dan *Attention*

Setelah bobot dihitung, model membentuk *context vector*, yaitu gabungan dari *hidden state encoder* yang paling berpengaruh. Dengan menggabungkan kedua informasi ini, model dapat memberikan perhatian lebih pada bagian input yang relevan saat menghasilkan setiap token output, yang meningkatkan akurasi prediksi. Operasi penggabungan ini dilakukan dengan *concatenation* sebagaimana pada persamaan di bawah.

$$H_{final} = \sigma(H_{decoder}, H_{attention}). \quad (3.11)$$

Keterangan :

H_{final} : vektor gabungan dari decoder dan attention.
$H_{decoder}$: vektor output dari decoder
σ : fungsi concatenate
$H_{attention}$: vektor output dari attention

Hasil yang digabungkan ini kemudian diteruskan ke lapisan *dense* untuk menghasilkan prediksi akhir.

3.5.1.8 *Dense Layer*

Lapisan ini berfungsi untuk menghasilkan *output* akhir dari model Bi-LSTM. Fungsi aktivasi yang digunakan pada lapisan ini menggunakan *softmax*, karena *output* yang diinginkan adalah prediksi distribusi probabilitas atas kata-kata dalam kosakata untuk menghasilkan ringkasan abstraktif. Hasil dari fungsi *softmax*

bernilai rentang antara 0 dan 1 dan tidak pernah negatif. Apabila terdapat n kelas, maka total dari seluruh *output* fungsi *softmax* adalah 1 (Ther et al., 2023).

Misalnya diketahui hasil dari lapisan sebelumnya adalah adalah $z = \begin{bmatrix} 0.3 \\ 1.15 \\ 0 \end{bmatrix}, b = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.3 \end{bmatrix}$ dan $z + b = \begin{bmatrix} 0.4 \\ 1.35 \\ 0.3 \end{bmatrix}$. Selanjutnya nilai eksponen dapat dihitung untuk setiap elemen pada matriks $z + b$.

$$e^{z_1} = e^{0.4} = 1.4918$$

$$e^{z_2} = e^{1.35} = 3.8571$$

$$e^{z_3} = e^{3} = 1.3499$$

Setelah mendapat nilai eksponen untuk setiap elemen matrks, kemudian seluruh nilai eksponen dihitung jumlahnya keseluruhan.

$$total_{eksponen} = \sum_{i=1}^{3} e^{z_1} = e^{z_1} + e^{z_2} e^{z_3} = 1.4918 + 3.8571 + 1.3499$$

$$= 6.6988$$

Berdasarkan nilai total eksponen pada perhitungan persamaan di atas, maka nilai *softmax* untuk setiap elemen dapat dihitung sebagaimana persamaan berikut.

$$softmax(z_1) = \frac{e^{z_1}}{total_{eksponen}} = \frac{1.4918}{6.6988} = 0.2220$$

$$softmax(z_1) = \frac{e^{z_2}}{total_{eksponen}} = \frac{3.8571}{6.6988} = 0.5754$$

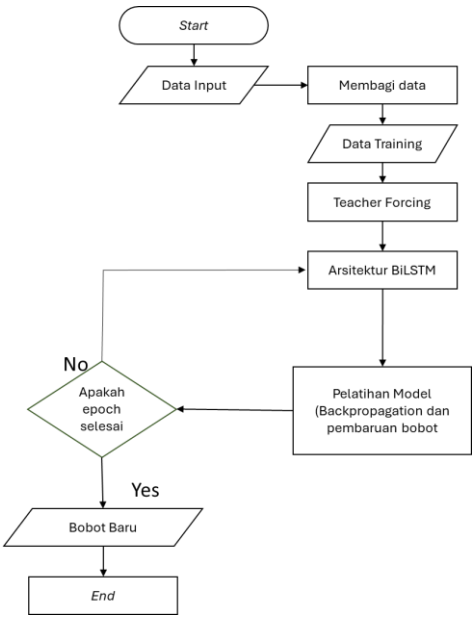
$$softmax(z_1) = \frac{e^{z_3}}{total_{eksponen}} = \frac{1.3499}{6.6988} = 0.2017$$

Output akhir dari fungsi *softmax* apabila dijumlah adalah 1, yang menunjukkan bahwa *output layer* menghasilkan distribusi probabilitas.

Dalam proses menghasilkan ringkasan teks dari representasi numerik, digunakan metode *nucleus sampling* untuk memilih kata-kata berdasarkan distribusi probabilitas yang lebih bervariasi sehingga mengurangi repetisi kata. Dalam *nucleus sampling*, ditentukan sebuah ambang batas probabilitas kumulatif p, dan pemilihan token hanya dilakukan dari kumpulan token dengan total probabilitas yang melebihi nilai p. Token-token yang berada di luar kumpulan ini akan diabaikan, sementara probabilitas token yang dipilih disesuaikan ulang agar jumlahnya menjadi 1 (Holtzman et al., 2019).

3.5.2 Alur Pelatihan

Keseluruhan alur pelatihan model divisualisasikan dalam Gambar 3.4 berikut (Alfin et al., 2024).



Gambar 3.4 Alur Model Bi-LSTM

Pelatihan dari model Bi-LSTM dimulai dengan memasukkan data hasil *text preprocessing* yang kemudian dibagi proporsi data latih dan uji-nya. Penelitian ini

membagi data pada saat proses pelatihan menjadi 80% data latih dan 20% data uji berdasarkan *Pareto Principle* (Joseph, 2022). Selanjutnya dilakukan *teacher forcing* pada data latih, agar *output* dari model sebelumnya, dapat digunakan sebagai input untuk langkah berikutnya (Ranzato et al., 2015). Selanjutnya, data latih dimasukkan ke dalam arsitektur Bi-LSTM untuk menghasilkan *output* sekuensial. Setelahnya, terdapat proses menghitung nilai *loss, backpropagation,* dan *update* bobot yang dilakukan pada fungsi *model.fit* pada kode untuk memperbarui parameter arsitektur Bi-LSTM (Huang et al., 2015b). Selama *training* performa model selama setiap *epoch* dihitung berdasarkan nilai *loss* dan akurasi prediksi *sequence* dari data latih dengan data uji (Terven et al., 2023). Proses dari arsitektur hingga evaluasi diulang sebanyak *epoch* yang diinginkan, atau berhenti jika memenuhi suatu kondisi dalam mekanisme *early stopping* (Hussein & Shareef, 2024). Hasil akhir dari pelatihan adalah bobot baru dari model yang dapat disimpan untuk *deploy* pada sistem atau digunakan pada proses evaluasi.

3.6 Evaluasi

Dalam melakukan validasi hasil ringkasan teks dari sistem, penelitian ini melibatkan ahli bahasa dalam menentukan ringkasan yang tepat dari dataset. Hasil ringkasan dari ahli kemudian dibandingkan dengan hasil ringkasan dari sistem untuk mengevaluasi seberapa bagus kualitas ringkasan yang dihasilkan dari sistem.

Penelitian ini menggunakan metode *Recall-Oriented Understudy for Gisting Evaluation* (ROUGE) untuk evaluasi hasil. *ROUGE* berisi matriks untuk menilai kualitas hasil tugas peringkasan teks (Lin, 2004). Efektivitas peringkasan teks dapat ditentukan dengan membandingkan hasil ringkasan dari sistem terhadap

ringkasan manusia. Penelitian ini menggunakan ROUGE-N dan ROUGE-L dalam menilai kualitas hasil ringkasan.

ROUGE-N mengevaluasi seberapa mirip hasil peringkasan dengan menghitung n-kata (*n-gram*) yang sama yang ada di hasil ringkasan algoritma dengan hasil ringkasan manusia. Skor ROUGE berkisar antara 0 hinga 1, yang mengindikasikan semakin tinggi skor menunjukkan tingkat kesesuaian yang lebih tinggi antara hasil ringkasan sistem dengan hasil ringkasan manusia.

ROUGE-L mengevaluasi kesamaan antara hasil yang dihasilkan oleh model dan referensi menggunakan penghitungan panjang urutan *n-gram* terpanjang yang berurutan (*longest common subsequence*/LCS).

$$ROUGE(N) = \frac{Jumlah\ n{-}gram\ yang\ sama}{Jumlah\ n{-}gram\ pada\ ringkasan\ manusia} \quad (3.12)$$

$$ROUGE - L_{Recall} = \frac{LCS\ (ringkasan\ manusia,\ \ ringkasan\ mesin)}{panjang\ ringkasan\ manusia} \quad (3.13)$$

$$ROUGE\_L_{Precision} = \frac{LCS\ (ringkasan\ manusia,\ \ ringkasan\ mesin)}{panjang\ ringkasan\ mesin} \quad (3.14)$$

$$ROUGE\_L_{F1} = \frac{(1+\beta^2)\ ROUGE\_L_{Recall} \cdot ROUGE\_L_{Precision}}{ROUGE\_L_{Recall} + \beta^2 \cdot ROUGE\_L_{Precision}} \quad (3.15)$$

Tabel 3.10 Contoh evaluasi hasil ringkasan

Teks lengkap	**Ringkasan sistem**	**Ringkasan manusia**
Website sibayar terlalu banyak form yang harus saya inputkan. Setiap ingin melanjutkan ke form berikutnya, kok harus simpan data dulu ya. Jadinya proses daftar ulang agak ribet karena banyak data yang harus saya masukkan. Menurut saya daftar ulang itu ya tinggal bayar saja tidak usah input banyak data.	form input simpan data proses daftar ulang ribet banyak data bayar tidak usah input banyak data	proses input form ribet banyak data daftar ualng tidak usah input banyak data

Misalkan diketahui hasil ringkasan sistem dan ringkasan manusia berdasarkan teks lengkap pada Tabel 3.9 di atas. Apabila diketahui $n = 1$ untuk jumlah *n-gram* pada ROUGE, maka hanya 1 kata (*unigram)* yang dijadikan perbandingan masing-masing ringkasan. Ringkasan sistem menghasilkan 16 kata (*unigram)* dengan 12 kata muncul dari 13 kata hasil ringkasan manusia (referensi). Berikut perhitungan nilai *recall, precision* dan *f1-score* apabila diketahui komposisi *unigram* kedua ringkasan tersebut.

$$ROUGE(1) = \frac{Jumlah\ unigram\ yang\ sama}{Jumlah\ unigram\ \ pada\ ringkasan\ manusia} = \frac{12}{13} = 0.9231$$

Hasil dari metrik evaluasi tersebut dapat diinterpretasikan bahwa ringkasan sistem memiliki tingkat kesesuaian yang cukup tinggi dengan ringkasan manusia.

Untuk menghitung ROUGE-L (*Longest Common Subsequence)*, pertama-tama harus mencari urutan *Longest Common Subsequence* (LCS) antara ringkasan sistem dan ringkasan Manusia. LCS adalah urutan kata yang berurutan dan sama yang muncul dalam kedua ringkasan tersebut. LCS yang ditemukan adalah urutan kata "form", "input", "banyak", "data", "daftar", "ulang", "tidak", "usah", "input", "banyak", "data", yang terdiri dari 11 kata yang sama. Dengan panjang LCS sebesar 11 dan panjang ringkasan manusia 13 kata, nilai ROUGE-L dapat dihitung menggunakan rumus :

$$ROUGE\_L_{Recall} = \frac{11}{13} = 0{,}846$$

$$ROUGE\_L_{Precision} = \frac{11}{16} = 0{,}688$$

$$ROUGE\_L_{F1} = \frac{2 \cdot ROUGE\_L_{Precision} \cdot ROUGE\_L_{Recall}}{ROUGE\_L_{precision} + ROUGE\_L_{recall}} = 0{,}759$$

Dalam hal ini, ROUGE-L-nya adalah 1, yang berarti ada kesamaan sebesar 75,9% dalam urutan kata antara ringkasan manusia dan ringkasan sistem berdasarkan *LCS*.

3.7 Skenario Pengujian

Penelitan ini melakukan pengujian *hyperparameter* untuk model Bi-LSTM dengan yang bertujuan untuk melakukan *tuning* atau optimasi algoritma dengan menemukan setiap kombinasi *hyperparameter* terbaik. (Yang & Shami, 2020). Pengujian *hyperparameter ini* melibatkan pemilihan konfigurasi parameter *optimizer* dan jumlah lapisan*, batch size,* dan *dropout. Optimizer* adalah algoritma yang digunakan untuk mengupdate bobot model selama proses pelatihan, dengan tujuan meminimalkan kesalahan model. Jumlah lapisan merujuk pada jumlah lapisan dalam model, seperti lapisan LSTM, yang menentukan seberapa dalam model dapat mempelajari pola dan hubungan dalam data. *Batch size* adalah jumlah sampel data yang diproses dalam satu iterasi pembaruan bobot. Sementara itu, *dropout* adalah teknik regularisasi yang mencegah *overfitting* dengan cara menonaktifkan secara acak sebagian neuron pada setiap iterasi pelatihan. Parameter-parameter tersebut dipilih karena memiliki pengaruh yang signifikan terhadap performa model (Reimers & Gurevych, 2017).

Tabel 3.11 Signifikansi parameter menurut (Reimers & Gurevych, 2017).

No.	Parameter	Signifikansi
1	*Optimizer*	High
2	Jumlah lapisan	Medium
3	*Batch size*	High
4	*Dropout*	High

Setiap parameter memiliki beberapa nilai atau metode yang menjadi konfigurasi. Nilai-nilai ini didapatkan dari hasil terbaik yang telah diujikan pada penelitian (Reimers & Gurevych, 2017). Berikut adalah konfigurasi dari setiap parameternya.

Tabel 3.12 Konfigurasi parameter

No.	Parameter	Konfigurasi
1	*Optimizer*	Adam, Nadam, RMSprop
2	Jumlah lapisan	1, 2, 3,4
3	*Batch size*	1, 8, 16, 32
4	*Dropout*	No Dropout, Naive, Variational
		0, 0.2, 0.5

Pengujian *hyperparameter* ini menggunakan metode *grid search* dengan melakukan pencarian menyeluruh terhadap setiap kombinasi parameter. Masing-masing kombinasi konfigurasi dipilih yang paling optimal dengan mempertimbangkan nilai akurasi data validasinya (Liashchynskyi & Liashchynskyi, 2019). Dalam menentukan akurasi, model menghitung persentase token yang diprediksi dengan benar pada setiap langkah dalam urutan *output* (Terven et al., 2023).

BAB IV

HASIL DAN PEMBAHASAN

4.1 Dataset

Penelitian ini berhasil mengumpulkan dataset berupa umpan balik pengguna *website* SiBayar Pondok Pesantren Sabilurrosyad sejumlah 114 umpan balik yang diperoleh dari 38 responden. *Website* SiBayar dapat diakses melalui https://sibayar.ponpesgasek.id/login dengan pratinjau fitur dan tampilannya dapat dilihat di Lampiran 2. Proses pengumpulan dataset dimulai dengan menyebarkan kuesioner (Lampiran 1) yang dimuat dalam *Google Form*. Responden menjawab lima pertanyaan kuesioner yang berupa isian yang berisi pengalaman mereka dalam mengakses fitur-fitur dalam SiBayar. Berikut adalah Tabel 4.1 contoh jawaban dari responden.

Tabel 4.1 Contoh jawaban responden

Pertanyaan 1 (Akses herregistrasi)	**Pertanyaan 2 (Form herregistrasi)**	**Pertanyaan 3 (Kendala herregistrasi)**	**Pertanyaan 4 (Fitur syahriah)**	**Pertanyaan 5 (Kritik dan saran)**
Akses form daftar ulang sangat mudah dan cepat. Kendalanya mungkin upload foto menggunakan hp agak repot, lebih mudah upload menggunakan laptop	Form daftar ulang sangat lengkap mengenai informasi pribadi santri. Saya suka dan setuju dengan adanya kelengkapan dan relevansi data yang ada pada website tersebut.	Kendala dalam penggunaan aplikasi untuk Saya yaitu pernah gagal dalam menyimpan data. Jadi ketika ada halaman yang belum terisi lengkap dan belum kesimpan, lalu pindah ke halaman yang lain, maka halaman yang belum lengkap tersebut hilang semua dan harus mengisi ulang lagi.	Saya suka dengan website Sibayar, karena dengan adanya website tersebut, memudahkan Saya untuk melihat tagihan syahriah dan akses WiFi, sehingga tidak perlu menggunakan buku pembayaran dan menghubungi admin untuk meminta akses WiFi.	Keamanan website lebih diperhatikan, kalau bisa per halaman ada fitur save sementara nya, sehingga tidak mudah gagal dalam penyimpana n.

Peneliti memeriksa apakah setiap jawaban dari responden sesuai dengan pertanyaan kuesioner agar data yang dihasilkan sesuai dengan konteks penelitian. Kalimat jawaban responden kemudian digabungkan berdasarkan kategori fiturnya untuk membentuk satu data. Pertanyaan 1, 2, dan 3 digabungkan menjadi 1 data; sementara pertanyaan 4 dan 5 menjadi dua data terpisah. Setiap data umpan balik pengguna tersebut akan diringkas oleh ahli bahasa, Aisyatul Azizah, S.H, M. H. sebagai patokan evaluasi model. Selanjutnya, data sudah bisa digunakan oleh sistem untuk menghasilkan ringkasan dari mesin. Berikut Tabel 4.2 contoh data yang akan menjadi input sistem.

Tabel 4.2 Contoh data yang belum diringkas dan sudah diringkas

No.	Umpan Balik Pengguna	Ringkasan Manusia
1	Form daftar ulang sangat lengkap mengenai informasi pribadi santri. Saya suka dan setuju dengan adanya kelengkapan dan relevansi data yang ada pada website tersebut. Form daftar ulang sangat lengkap mengenai informasi pribadi santri. Saya suka dan setuju dengan adanya kelengkapan dan relevansi data yang ada pada website tersebut. Kendala dalam penggunaan aplikasi untuk Saya yaitu pernah gagal dalam menyimpan data. Jadi ketika ada halaman yang belum terisi lengkap dan belum kesimpan, lalu pindah ke halaman yang lain, maka halaman yang belum lengkap tersebut hilang semua dan harus mengisi ulang lagi.	Akses form daftar ulang mudah cepat. Upload foto mudah pakai laptop daripada HP. Form sangat lengkap dan relevan, namun kendalanya data yang belum tersimpan bisa hilang jika pindah halaman, sehingga harus diisi ulang.
2	Saya suka dengan website Sibayar, karena dengan adanya website tersebut, memudahkan Saya untuk melihat tagihan syahriah dan akses WiFi, sehingga tidak perlu menggunakan buku pembayaran dan menghubungi admin untuk meminta akses WiFi.	Suka dengan Sibayar, memudahkan cek tagihan syahriah dan akses WiFi tanpa buku pembayaran atau hubungi admin
3	Keamanan website lebih diperhatikan, kalau bisa per halaman ada fitur save sementara nya, sehingga tidak mudah gagal dalam penyimpanan.	Keamanan website diperhatikan, per halaman ada fitur save , agar tidak gagal simpan.

4.2 Hasil Pengujian Parameter

Berdasarkan rincian konfigurasi parameter pada Tabel 3.9, maka jumlah keseluruhan kombinasi adalah sebanyak 240. Pengujian ini dilakukan selama 20

epoch dengan mekanisme *early stopping* yang menghentikan pelatihan model lebih awal ketika akurasi tidak meningkat selama beberapa epoch tertentu, agar mencegah *overfitting* dan menghemat waktu pelatihan pada *grid search* (Hussein & Shareef, 2024). Pada saat proses pengujian parameter, 20% data latih diambil sebagai data validasi. Berikut adalah hasil dari pencarian konfigurasi terbaik dengan *grid search*.

Tabel 4.3 10 Konfigurasi dengan performa terbaik

No.	*optimizer*	*num_layers*	*batch_size*	*dropout_type*	*dropout_rate*	*val_loss*	*val_accuracy*
1	nadam	1	1	variational	0.5	1.1449	88.74%
2	nadam	1	1	naive	0.2	1.1127	88.21%
3	adam	1	1	variational	0.5	1.2282	88.00%
4	adam	1	1	variational	0.2	1.1482	86.84%
5	adam	1	1	naive	0.2	1.2007	86.74%
6	nadam	1	1	variational	0.2	1.2043	86.42%
7	nadam	2	1	no_dropout	0	1.2972	86.32%
8	nadam	1	1	no_dropout	0	1.2785	85.89%
9	adam	1	1	no_dropout	0	1.2883	85.68%
10	nadam	1	1	naive	0.5	1.2332	85.16%

Berdasarkan Tabel 4.3, diketahui berdasarkan 10 kombinasi konfigurasi terbaik, algoritma *optimizer* yang paling sering muncul sebagai konfigurasi terbaik adalah Nadam dan Adam Sementara itu, konfigurasi jumlah *layer* dan ukuran *batch* dengan nilai 1, mendominasi 10 kombinasi teratas. Namun terdapat variasi nilai *dropout* yang berbeda-beda di tiap kombinasi baik itu konfigurasi tanpa *dropout, naive,* maupun *variational dropout* dengan tingkat *dropout* yang berbeda-beda. Akurasi tertinggi pada pengujuan parameter tercatat sebesar 88.74%.

Tabel 4.4 10 Konfigurasi dengan performa terburuk

No.	*optimizer*	*num_layers*	*batch_size*	*dropout_type*	*dropout_rate*	*val_loss*	*val_accuracy*
231	nadam	4	32	naive	0.2	2.8322	60.00%
232	rmsprop	4	32	naive	0.2	2.8843	59.68%
233	nadam	3	8	no_dropout	0	2.8273	59.68%
234	nadam	4	1	naive	0.5	3.2001	59.47%
235	rmsprop	4	32	variational	0.5	2.7213	58.53%
236	rmsprop	4	16	naive	0.2	3.6064	57.58%
237	adam	4	1	naive	0.2	2.9953	57.05%

No.	*optimizer*	*num_layers*	*batch_size*	*dropout_type*	*dropout_rate*	*val_loss*	*val_accuracy*
238	nadam	4	16	naive	0.5	3.74072	55.58%
239	nadam	4	32	no_dropout	0	3.27887	54.21%
240	rmsprop	3	32	variational	0.2	3.89651	48.11%

Berdasarkan Tabel 4.4, yang menunjukkan 10 konfigurasi dengan performa terburuk, dapat dilihat bahwa konfigurasi-konfigurasi tersebut memiliki kombinasi *optimizer* yang bervariasi. Selain itu, jumlah lapisan yang lebih besar (yaitu 3 dan 4 lapisan) dan ukuran batch yang lebih besar (16 dan 32) juga lebih sering ditemukan dalam kombinasi-konfigurasi dengan akurasi terendah. Pada 10 nilai terburuk, kombinasi *dropout* terburuk didominasi oleh *naive* dropout. Akurasi terendah dari pengujuan parameter tercatat sebesar 55.58% pada kombinasi terakhir.

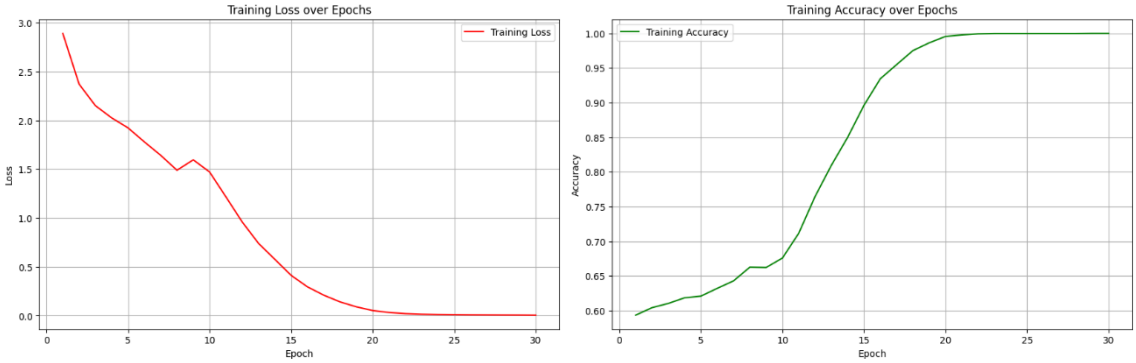
4.3 Hasil Ringkasan dan Evaluasi

Berdasarkan pengujian *hyperparameter*, diketahui parameter terbaik untuk tugas peringkasan teks pada penelitian ini menggunakan kombinasi konfigurasi parameter pada Tabel 4.5 berikut.

Tabel 4.5 Konfigurasi parameter

Parameter	Konfigurasi
Optimizer	Nadam
Jumlah lapisan	1
Batch size	1
Dropout	*Variational*
	0.5

Kemudian, model dilatih kembali berdasarakan kombinasi konfigurasi parameter terbaik guna memastikan agar model dapat bekerja dengan baik dengan seluruh data. Berikut adalah hasil dari pelatihan datanya dengan fluktuasi nilai akurasi dan *loss* sebagaimana pada Gambar 4.1.



Gambar 4.1 Grafik nilai loss dan akurasi model dari parameter terbaik

Hasil dari pelatihan digunakan untuk membuat ringkasan dari teks yang kemudian dihitung skor ROUGE-1, ROUGE-2, dan ROUGE-L untuk menentukan kualitas ringkasan dari sistem. Berikut adalah hasil ringkasan teksnya.

Tabel 4.6 Hasil ringkasan

No.	Ringkasan Manusia	Ringkasan Sistem	ROUGE-1	ROUGE-2	ROUGE-L
1	sibayar sudah bagus saran saya aplikasinya cukup dikembangkan lagi agar bisa memuat lebih banyak fitur seperti surat izin diniyah	fitur menu dan belum pembayaran agar santri yang sudah lunas untuk mudah dalam pendaftaran dan pembayaran	0.158	0.000	0.061
2	fitur bisa ditambah dengan sistem pembayaran syahriah yang otomatis mendeteksi rekening saat membayar tagihan bulanan jadi tidak perlu konfirmasi ke bendahara	sistem bisa ditambah dengan sistem pembayaran syahriah yang otomatis mendeteksi rekening saat membayar tagihan bulanan jadi	0.714	0.700	0.833

No.	Ringkasan Manusia	Ringkasan Sistem	ROUGE-1	ROUGE-2	ROUGE-L
3	saran untuk fitur daftar ulang lebih disederhanakan formnya menu di bawah atau di samping sebaiknya pilih salah satu karena keduanya isinya sama	fitur daftar ulang lebih disederhanakan formnya menu di bawah atau di samping sebaiknya pilih salah satu karena keduanya isinya sama	0.905	0.905	0.950
4	diberikan sistem otomatis yang mendeteksi pembayaran masuk dan langsung terdeteksi di sistem agar tidak perlu konfirmasi ke bendahara yang tidak cepat balas chat	sistem otomatis yang mendeteksi pembayaran masuk dan langsung terdeteksi di sistem agar	0.550	0.500	0.710
5	tambahkan validasi di nomor hp nik atau input lain biar santri tidak ngasal isi tambahkan fitur pembayaran keamanan dan diniyah biar lengkap	validasi di nomor hp nik atau input lain biar santri	0.500	0.429	0.667
6	wifi bisa buat hotspot biar santri nggak kehabisan paketan tambahkan juga fitur perizinan diniyah biar bisa di sibayar	wifi bisa buat hotspot biar santri nggak kehabisan paketan tambahkan juga fitur perizinan diniyah biar bisa di sibayar	1.000	1.000	1.000
7	tingkatkan fitur buat semua pembayaran di sibayar dan perbaiki form daftar ulang	sibayar buat semua pembayaran di sibayar dan perbaiki form daftar ulang	0.833	0.818	0.909

Untuk lebih lengkap dari Tabel 4.6, hasil ringkasan dapat dilihat pada Lampiran. Berikut adalah rincian nilai evaluasi ROUGE.

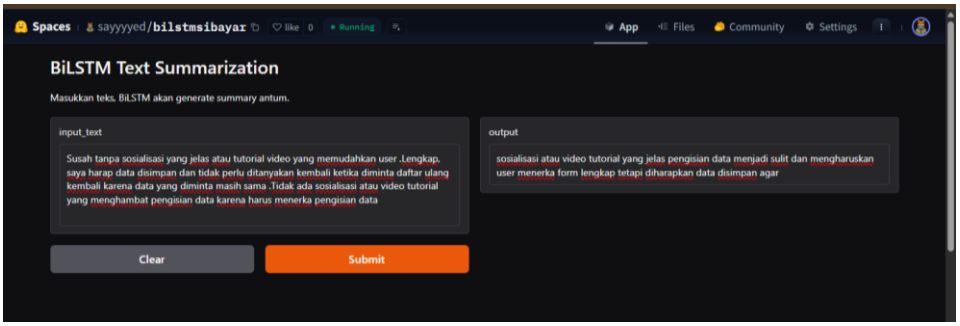
Tabel 4.7 Evaluasi nilai ROUGE

Statistik	ROUGE-1	ROUGE-2	ROUGE-L
Rata-rata	0.6221	0.5462	0.660
Min	0.0714	0	0.0606
Max	1	1	1

Tabel 4.7 menunjukkan hasil evaluasi kinerja sistem ringkasan yang memiliki rata-rata performa model dengan skor ROUGE-1 (0.6221), ROUGE 2 (0.5462) dan ROUGE-L (0.660). Terdapat variasi performa yang signifikan, terlihat dari nilai minimum yang rendah, terutama pada ROUGE-2 (0), yang menunjukkan bahwa terdapat ringkasan yang gagal menangkap kesesuaian *bigram* dengan referensi. Selain itu terdapat hasil ringkasan yang memiliki nilai skor 1 yang menandakan bahwa terdapat hasil ringkasan yang sama dengan teks referensi.

4.4 Hasil Implementasi Sistem

Model peringkasan teks dengan metode Bi-LSTM yang telah dilatih dengan parameter terbaik kemudian diterapkan dalam sistem melalui *website.* Untuk menyiapkan model untuk *deploy* ke sistem, diperlukan hasil tokenisasi dan bobot hasil *training* Bi-LSTM. Kemudian hasil-hasil dari model tersebut diunggah ke laman *huggingface* untuk dilakukan *deploy* pada *space* sehingga menghasilkan implementasi sistem berupa *website*. Berikut adalah tampilan implementasinya.



Gambar 4.2 Implementasi model peringkasan teks ke sistem

Implementasi model sebagaimana pada Gambar 4.2 dapat diakses pada laman https://huggingface.co/spaces/sayyyyed/Bi-LSTMsibayar. Untuk menghasilkan ringkasan, masukkan teks umpan balik pengguna pada *form* yang memiliki label *input_text*. Tekan tombol *submit* untuk menghasilkan ringkasan yang akan terlihat pada *form* berlabel *output.* Gunakan tombol *clear* untuk menghapus seluruh isi *input_text* untuk memulai input baru.

4.5 Pembahasan

Penelitian ini bertujuan untuk menghasilkan ringkasan otomatis dari umpan balik pengguna *website* SiBayar Pondok Pesantren Sabilurrosyad menggunakan metode Bi-LSTM dengan mencari parameter terbaik (*hyperparameter tuning)* serta mengevaluasi hasil ringkasan mesin dengan ringkasan manusia menggunakan metrik ROUGE. Data dikumpulkan melalui kuesioner yang berisi tanggapan para pengguna SiBayar terkait fitur herregistrasi, pendaftaran santri baru, *syahriah*, serta saran terhadap pengembangan SiBayar. Data yang terkumpul kemudian diringkas secara manual oleh ahli yang akan digunakan sebagai evaluasi hasil ringkasan mesin.

Data preprocessing diterapkan pada teks untuk merapikan dari karakter-karakter yang tidak perlu digunakan oleh sistem. Data yang sebelumnya berbentuk teks akan dipecah menjadi token dan diubah menjadi numerik melalui proses tokenization. Data dipastikan berukuran sama melalui proses *padding* dan *truncating.*

Data yang telah dirapikan, kemudian akan dimasukkan ke dalam model. Model Bi-LSTM meliputi *Input Layer*, *Embedding Layer,* Bi-LSTM *Layer (Encoder), Concatenation Layer, Input Layer (Decoder), Embedding Layer (Decoder), LSTM Layer (Decoder), Attention Layer, Concatenation Layer,* dan *Dense Layer*. Lapisan-lapisan pada model tersebut berguna untuk memproses data secara sekuens untuk menghasilkan vektor *output* probabilitas yang akan diterjemahkan kembali menjadi teks oleh metode *decoding nucleus sampling*.

Langkah pengujian pertama-tama menentukan parameter yang dijadikan sebagai konfigurasi yang meliputi *optimizer,* jumlah *layer*, *batch size*, dan *dropout*. Kemudian, parameter tersebut diuji dengan konfigurasi tertentu untuk dicari kombinasi terbaiknya antar parameter menggunakan metode *grid search*.

Hasil pengujian terlihat bahwa pemilihan kombinasi parameter memiliki pengaruh signifikan terhadap performa model Bi-LSTM. Optimizer *Adam* dan *Nadam* secara konsisten menunjukkan performa terbaik daripada *RMSProp* yang dibuktikan pada hasil 10 kombinasi terbaik. Kombinasi terbaik cenderung menggunakan jumlah lapisan dan *batch size* yang kecil, yaitu 1, yang menunjukkan bahwa model dengan arsitektur yang lebih sederhana justru mampu memberikan hasil yang lebih optimal pada data berkompleksitas rendah yang digunakan pada

penelitian ini. Selain itu, hasil dari pengujian menunjukkan sebaran kombinasi konfigurasi *dropout* yang bervariasi, tanpa menunjukkan dominasi salah satu tipe *dropout* ataupun juga *rate*-nya. Hal tersebut mengindikasikan konfigurasi *dropout* tidak memberi pengaruh signifikan terhadap performa model.

Berdasarkan hasil dari *hyperparameter tuning,* kombinasi terbaik yang memiliki akurasi tertinggi adalah kombinasi yang memiliki konfigurasi algoritma optimasi Nadam, 1 lapisan Bi-LSTM, *batch size* 1, dan *variational dropout* dengan *rate* 0.5. Agar model dapat melihat keseluruhan data, dilakukan proses *training* kembali, karena pada *grid search* penentuan performa didasarkan pada data validasi. Hasil dari pelatihan kemudian dievaluasi kepada data uji untuk mengukur kualitas hasil ringkasan menggunakan metrik ROUGE.

Hasil evaluasi model menghasilkan skor ROUGE-1 (0.6221) yang menunjukkan adanya 62,21% unigram yang terdapat dalam ringkasan manusia juga muncul dalam ringkasan yang dihasilkan oleh model. Nilai ROUGE-2 (0.5462) menunjukkan adanya 54,62% bigram (pasangan dua kata) dari ringkasan manusia berhasil direproduksi dalam keluaran model, yang mengindikasikan adanya kesamaan dalam pasangan kata berurutan. Skor ROUGE-L sebesar (0.660) menunjukkan secara struktur kalimat dengan urutan kata terpanjang yang sama antara referensi dan hasil model mencakup 66% dari total kata dalam referensi.

Hasil-hasil tersebut mengindikasikan bahwa model peringkasan teks pada penelitian ini memiliki kemampuan yang baik dalam menghasilkan inti kalimat sebagaimana (Givchi et al., 2022) yang mendapat skor 0.596 di mana modelnya mampu menghasilkan interpretasi dan konsep utama dengan baik pada dokumen.

Meskipun demikian, perlu adanya perbaikan dalam aspek pasangan kata dan urutan kata sehingga hasilnya dapat lebih baik.

Berikut adalah perbandingan performa Bi-LSTM terhadap hasil penelitian terdahulu.

Tabel 4.8 Hasil penelitian sebelumnya tentang peringkasan abstraktif

No.	Penelitian	Metode	*Hyperparameter Tuning*	ROUGE-1	ROUGE-2	ROUGE-L
1	(Shaliha, 2021)	Bi-LSTM	-	0.06236	0.00684	-
2	(Saputra et al., 2021)	LSTM	-	0.13846	-	-
3	(Jiang et al., 2021)	CNN-Bi-LSTM	*Convolutional layer, Epoch*	0.2547	0.0809	0.2903
4	(Tripathy & Ashok, 2023)	Bi-LSTM	-	0.21058	0.37963	0.38653
5	(Yuliska & Syaliman, 2022)	Bi-LSTM	Word2vec *tuning*	0.43	0.1140	0.1867
6	**Penelitian ini**	**Bi-LSTM**	***Optimizer,* jumlah *layer, batch size, dropout***	**0.6221**	**0.5462**	**0.660**

Dibandingkan dengan penelitian-penelitian sebelumnya pada Tabel 4.8, hasil dalam penelitian ini menunjukkan kualitas Bi-LSTM yang cukup signifikan yang dievaluasi dengan skor ROUGE. Hasil dari penelitian ini terbukti lebih baik dari penelitan sebelumnya yang menggunakan Bi-LSTM dengan tanpa *hyperparameter tuning* atau dengan *hyperparameter tuning* yang lebih sedikit. Temuan ini sejalan dengan studi sebelumnya (Reimers & Gurevych, 2017; Liashchynskyi & Liashchynskyi, 2019) yang menyatakan bahwa pemilihan kombinasi *hyperparameter* yang tepat sangat penting dalam proses *tuning* model untuk mencapai performa optimal.

Meski demikian, perlu dicatat bahwa meskipun penelitian ini menunjukkan skor ROUGE yang lebih tinggi dibandingkan dengan penelitian sebelumnya,

perbedaan dalam data yang digunakan dapat memengaruhi kinerja model. Setiap dataset memiliki karakteristik unik, termasuk ukuran, keberagaman, dan kompleksitas teks, yang dapat mempengaruhi sejauh mana model dapat menghasilkan ringkasan yang berkualitas.

4.6 Integrasi Islam

Islam mengajarkan bahwa memberikan saran atau umpan balik yang jujur adalah suatu hal yang dianjurkan oleh seorang muslim untuk menjaga hubungan *muamalah* antar manusia. Hal ini selaras dengan firman Allah SWT. dalam Surat Al-Baqarah ayat 42, yang berbunyi:

وَلَا تَلْبِسُوا الْحَقَّ بِالْبَاطِلِ وَتَكْتُمُوا الْحَقَّ وَاَنْتُمْ تَعْلَمُوْنَ ﴿٤٢﴾

"Janganlah kamu campuradukkan yang haq (kebenaran yang sempurna) dengan yang batil (salah dan sesat) dan janganlah kamu sembunyikan yang haq, sedangkan kamu mengetahui) "(QS. Al-Baqarah: 42) (Shihab, 2013)

Ayat ini mengajarkan bahwa seseorang tidak boleh mencampurkan kebenaran dengan kebatilan serta tidak boleh menyembunyikan fakta yang benar. Tafsir Al-Mishbah mengatakan bahwa mencampuradukkan perkara haq dan batil adalah ciri-ciri orang sesat dan menjadi sebuah perilaku dosa apabila seseorang dengan sengaja menyembunyikan kebenaran (Shihab, 2002). Dalam hal memberikan saran atau kritik, setiap umpan balik yang diberikan harus berdasarkan kejujuran dan fakta, bukan sekadar pujian kosong atau kritik yang menyesatkan. Selain itu salah satu bentuk tolong-menolong dalam kebaikan adalah memberikan nasihat dan informasi yang benar di antara sesama muslim.

Untuk memilah umpan balik, saran, atau kritik mana yang baik dan yang buruk, seorang Muslim diharuskan bijak menggunakan akalnya dalam memilih informasi yang bermanfaat. Seorang Muslim diingatkan untuk selalu mendengarkan dengan hati-hati, menyaring dengan kebijaksanaan, dan mengikuti yang terbaik di antara pilihan yang ada (فَيَتَّبِعُوْنَ اَحْسَنَه).

الَّذِيْنَ يَسْتَمِعُوْنَ الْقَوْلَ فَيَتَّبِعُوْنَ اَحْسَنَهٗۗ اُولٰۤىِٕكَ الَّذِيْنَ هَدٰىهُمُ اللّٰهُ وَاُولٰۤىِٕكَ هُمْ اُولُوا الْاَلْبَابِ ﴿١٨﴾

"(Yaitu) mereka yang mendengarkan perkataan (siapapun engkau) lalu mengikuti apa yang paling baik di antaranya. Mereka itulah orang-orang yang telah Allah tunjuki (jalan lebar yang lurus) dan mereka itulah Ulul Albab (orang-orang yang berakal bersih, murni, dan cerah).(QS. Az-Zumar (39:18) (Shihab, 2013)

Dalam Tafsir Jalalain, istilah *ulul albab* merujuk kepada pribadi yang menggunakan akalnya dengan baik. Seorang Muslim yang dapat menggunakan akalnya dengan baik adalah salah satu tanda bahwa ia adalah salah seorang yang diberi petunjuk oleh Allah SWT.. Semua manusia pasti memiliki akal, namun petunjuk dari Allah SWT. datang kepada hambanya yang diantara mereka bijak menggunakan akalnya dengan baik (Al-Mahalliy & As-Suyuthi, 1990).

Pemilihan informasi merupakan salah satu aspek penting dalam pemrosesan bahasa alami khususnya peringkasan teks. Metode peringkasan teks memilih informasi yang penting di antara berbagai informasi yang diterima. Dengan adanya ringkasan dari umpan balik pengguna, pengembang *website* Sibayar dapat mempersingkat waktu yang diperlukan dalam memilah kritik dan saran dari penggunanya. Pemanfaatan waktu dengan baik selaras dengan apa yang telah diperingatkan oleh QS. Al-Ashr ayat 1-3.

وَالْعَصْرِۙ ﴿١﴾ اِنَّ الْاِنْسَانَ لَفِيْ خُسْرٍۙ ﴿٢﴾ اِلَّا الَّذِيْنَ اٰمَنُوْا وَعَمِلُوا الصّٰلِحٰتِ وَتَوَاصَوْا بِالْحَقِّ وَتَوَاصَوْا بِالصَّبْرِ ﴿٣﴾

"Demi waktu. Sesungguhnya (semua) manusia (yang mukallaf yakni yang mendapat perintah beban keagamaan) itu benar-benar dalam (wadah) kerugian (kebinasaan yang besar), kecuali orang-orang yang beriman mengerjakan amal-amal saleh, serta saling berwasiat tentang kebenaran dan saling berwasiat dalam kesabaran." (QS. Al-Ashr ayat 1-3). (Shihab, 2013)

Imam Ar-Razi dalam tafsirnya mengatakan bahwa dalam Surah Al-Ashr terdapat peringatan keras yang dinyatakan dengan sumpah apabila seseorang membuang waktunya sia-sia. Peringatan tersebut berisi ancaman kerugian yang bakal dialami manusia kecuali mereka memegang empat perkara. Perkara tersebut adalah iman, amal yang baik, saling menasehati dalam kebaikan dan saling menasehati dalam kesabaran (Al-Razi, 2018).

Dalam konteks penelitian ini, umpan balik yang berkualitas memastikan data yang digunakan oleh peneliti dalam menyusun model peringkasan dapat bekerja dengan baik dan sesuai dengan pengalaman pengguna. Selain itu, bagi pengembang *website* Sibayar, umpan balik dari pengguna yang berisi pengalaman ketika menggunakan *website,* kritikan maupun saran apabila disampaikan dengan jujur, maka akan sangat membantu dalam proses pengembangan *website* Sibayar. Hal tersebut menunjukkan adanya saling menasihati dan tolong-menolong di antara sesama Muslim. Sebagai hasilnya, *website* Sibayar akan lebih *user-friendly*, dan mampu memberikan solusi yang sesuai dengan kebutuhan manajemen pondok pesantren maupun santri sebagai pengguna.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini bertujuan untuk mengidentifikasi kombinasi dari konfigurasi parameter terbaik Bi-LSTM agar menghasilkan model peringkasan teks yang baik. Parameter yang dicari konfigurasi terbaiknya meliputi *optimizer,* jumlah *layer*, *batch size,* dan *dropout*. Hasil *hyperparameter tuning* menunjukkan kombinasi konfigurasi terbaik dengan algoritma optimasi menggunakan *optimizer* Nadam, jumlah lapisan 1 dan *batch size* 1 serta *dropout rate* 0,5 menggunakan *variational dropout* terbukti memberikan hasil akurasi 88,74% pada data validasi.

Hasil evaluasi model menggunakan data uji menunjukkan skor ROUGE-1 sebesar 0.6221, ROUGE-2 sebesar 0.5462, dan ROUGE-L sebesar 0.660. Secara keseluruhan, model Bi-LSTM pada penelitian ini terbukti mampu melakukan tugas peringkasan teks baik dengan mempertahankan inti kalimat, tetapi terdapat beberapa aspek dalam hasil ringkasan, seperti kesamaan pasangan kata dan urutan kata, yang perlu diperbaiki untuk mencapai hasil yang lebih baik.

5.2 Saran

Berdasarkan hasil penelitian ini, terdapat beberapa saran yang dapat dipertimbangkan untuk pengembangan model Bi-LSTM dalam peringkasan otomatis, sebagai berikut:

a. Mengeksplorasi *hyperparameter* lain dengan variasi yang beragam atau meningkatkan jumlah parameter agar mendapatkan kombinasi parameter yang lebih lengkap.
b. Mengeksplorasi arsitektur Bi-LSTM lain dengan arsitektur yang berbeda untuk meningkatkan kemampuan model dalam menghasilkan ringkasan abstraktif.
c. Menggunakan lebih banyak data, agar model dapat menghasilkan ringkasan yang lebih baik karena memiliki informasi yang lebih bervariasi.

DAFTAR PUSTAKA

Abdelouahab, K., Pelcat, M., & Berry, F. (2017). Why TanH is a Hardware Friendly Activation Function for CNNs. *Proceedings of the 11th International Conference on Distributed Smart Cameras*, 199–201. https://doi.org/10.1145/3131885.3131937

Ahmad, D. M. (2012). The Dynamics of the Pondok Pesantren: An Islamic Educational Institution in Indonesia. In *Reaching for the Sky* (pp. 63–74). BRILL. https://doi.org/10.1163/9789401207584_006

Alfin, M., Abidin, Z., & Basid, P. M. N. S. A. (2024). Peringkasan Multi Dokumen Berbahasa Indonesia Menggunakan Metode Recurrent Neural Network. *Techno.Com*, *23*(1), 187–197. https://doi.org/10.62411/tc.v23i1.9605

al-'Asqalāni. (2010). *Fathul Bari: syarah Shahih al-Bukhari*. Pustaka Imam Asy-Syafii. https://books.google.co.id/books?id=f31nnQAACAAJ

Al-Mahalliy, I. J., & As-Suyuthi, I. J. (1990). *Terjemah Tafsir Jalalain Berikut Asbaabun Nuzul*. Penerbit Sinar Baru. https://books.google.co.id/books?id=h0R4AQAACAAJ

Al-Razi, F.-D. (2018). *The Great Exegesis Tafsir Al Kabir Volume I The Fatiha: Vol. I*. The Royal Ahl-Bayt Institute of Thought & Islamic Text Society.

Bier, A., Jastrzebska, A., & Olszewski, P. (2022). Variable-Length Multivariate Time Series Classification Using ROCKET: A Case Study of Incident Detection. *IEEE Access*, *10*, 95701–95715. https://doi.org/10.1109/ACCESS.2022.3203523

Cui, Z., Ke, R., Pu, Z., & Wang, Y. (2020). Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values. *Transportation Research Part C: Emerging Technologies*, *118*, 102674.

Ding, H., Wang, Z., Paolini, G., Kumar, V., Deoras, A., Roth, D., & Soatto, S. (2024). Fewer Truncations Improve Language Modeling. *ArXiv*, *abs/2404.10830*. https://api.semanticscholar.org/CorpusID:269187631

El-Kassas, W. S., Salama, C. R., Rafea, A. A., & Mohamed, H. K. (2021). Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, *165*, 113679. https://doi.org/10.1016/j.eswa.2020.113679

Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc."

Ghojogh, B., & Ghodsi, A. (2023). Recurrent neural networks and long short-term memory networks: Tutorial and survey. *ArXiv Preprint ArXiv:2304.11461*.

Givchi, A., Ramezani, R., & Baraani-Dastjerdi, A. (2022). Graph-based abstractive biomedical text summarization. *Journal of Biomedical Informatics*, *132*, 104099. https://doi.org/10.1016/j.jbi.2022.104099

Glorot, X., Bordes, A., & Bengio, Y. (2011). *Deep Sparse Rectifier Neural Networks*.

Greff, K., Srivastava, R. K., Koutnik, J., Steunebrink, B. R., & Schmidhuber, J. (2017a). LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(10), 2222–2232. https://doi.org/10.1109/TNNLS.2016.2582924

Greff, K., Srivastava, R. K., Koutnik, J., Steunebrink, B. R., & Schmidhuber, J. (2017b). LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(10), 2222–2232. https://doi.org/10.1109/TNNLS.2016.2582924

Hasan, M. N., Toma, R. N., Nahid, A.-A., Islam, M. M. M., & Kim, J.-M. (2019). Electricity theft detection in smart grid systems: A CNN-LSTM based approach. *Energies*, *12*(17), 3310.

Holtzman, A., Buys, J., Du, L., Forbes, M., & Choi, Y. (2019). *The Curious Case of Neural Text Degeneration*. http://arxiv.org/abs/1904.09751

Huang, Z., Xu, W., & Yu, K. (2015a). Bidirectional LSTM-CRF Models for Sequence Tagging. *ArXiv*, *abs/1508.01991*. https://api.semanticscholar.org/CorpusID:12740621

Huang, Z., Xu, W., & Yu, K. (2015b). Bidirectional LSTM-CRF Models for Sequence Tagging. *ArXiv*, *abs/1508.01991*. https://api.semanticscholar.org/CorpusID:12740621

Hussein, B. M., & Shareef, S. M. (2024). An Empirical Study on the Correlation between Early Stopping Patience and Epochs in Deep Learning. *ITM Web of Conferences*, *64*, 01003. https://doi.org/10.1051/itmconf/20246401003

Imrana, Y., Xiang, Y., Ali, L., & Abdul-Rauf, Z. (2021). A bidirectional LSTM deep learning approach for intrusion detection. *Expert Systems with Applications*, *185*, 115524. https://doi.org/10.1016/j.eswa.2021.115524

Jiang, J., Zhang, H., Dai, C., Zhao, Q., Feng, H., Ji, Z., & Ganchev, I. (2021). Enhancements of Attention-Based Bidirectional LSTM for Hybrid Automatic Text Summarization. *IEEE Access*, *9*, 123660–123671. https://doi.org/10.1109/ACCESS.2021.3110143

Jindal, S. G., & Kaur, A. (2020). Automatic Keyword and Sentence-Based Text Summarization for Software Bug Reports. *IEEE Access*, *8*, 65352–65370. https://doi.org/10.1109/ACCESS.2020.2985222

Jorge, J., Giménez, A., Iranzo-Sánchez, J., Civera, J., Sanchis, A., & Juan, A. (2019). Real-Time One-Pass Decoder for Speech Recognition Using LSTM

Language Models. *Interspeech 2019*, 3820–3824. https://doi.org/10.21437/Interspeech.2019-2798

Joseph, V. R. (2022). Optimal ratio for data splitting. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *15*(4), 531–538. https://doi.org/10.1002/sam.11583

Józefowicz, R., Zaremba, W., & Sutskever, I. (2015). An Empirical Exploration of Recurrent Network Architectures. *International Conference on Machine Learning*. https://api.semanticscholar.org/CorpusID:9668607

Julhadi, J. (2019). PONDOK PESANTREN: Ciri Khas, Perkembangan, dan Sistem Pendidikannya. *Mau'izhah*, *9*(2). https://doi.org/10.55936/mauizhah.v9i2.26

Kanev, G., Mladenova, T., & Valova, I. (2023). Leveraging User Experience for Enhancing Product Design: A Study of Data Collection and Evaluation. *2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 01–06. https://doi.org/10.1109/HORA58378.2023.10156767

Karo Karo, I. M., Dewi, S., & Perdana, A. (2024). Implementasi Text Summarization Pada Review Aplikasi Digital Library System Menggunakan Metode Maximum Marginal Relevance. *JEKIN - Jurnal Teknik Informatika*, *4*(1), 25–31. https://doi.org/10.58794/jekin.v4i1.671

Keras. (n.d.). *LSTM Layer*. Retrieved May 19, 2025, from https://keras.io/api/layers/recurrent_layers/lstm/

Kothari, K., Shah, A., Khara, S., & Prajapati, H. (2020). *A NOVEL APPROACH IN USER REVIEWS ANALYSIS USING TEXT SUMMARIZATION AND SENTIMENT ANALYSIS: SURVEY*. https://api.semanticscholar.org/CorpusID:214738395

Kusuma Pertiwi, A., Septia Anggra Cahyani, S., Chulashotud Diana, R., & Gunawan, I. (2018). The Leadership of Kyai: A Descriptive Study. *Proceedings of the 3rd International Conference on Educational Management and Administration (CoEMA 2018)*. https://doi.org/10.2991/coema-18.2018.32

Liashchynskyi, P., & Liashchynskyi, P. (2019). Grid search, random search, genetic algorithm: a big comparison for NAS. *ArXiv Preprint ArXiv:1912.06059*.

Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. *Text Summarization Branches Out*, 74–81. https://aclanthology.org/W04-1013

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *ArXiv Preprint ArXiv:1508.04025*.

Maimunah, M., & Junadi, J. (2023). IMPLEMENTASI SISTEM INFORMASI AKADEMIK DI PONDOK PESANTREN. *Al-Afkar : Manajemen Pendidikan Islam*, *11*(01), 56–70. https://doi.org/10.32520/al-afkar.v11i01.594

Marsum, M., & Syahroni, Abd. W. (2020). EFEKTIFITAS PENGGUNAAN TEKNOLOGI PADA PESANTREN MODERN DALAM MENGHADAPI REVOLUSI INDUSTRI 4.0. *Jurnal Kariman*, *8*(02), 233–242. https://doi.org/10.52185/kariman.v8i02.155

Martens, D. (2020). *Improving the Quality of User Feedback for Continuous Software Evolution*. https://api.semanticscholar.org/CorpusID:220307596

Mohammad Masum, A. K., Abujar, S., Islam Talukder, M. A., Azad Rabby, A. K. M. S., & Hossain, S. A. (2019). Abstractive method of text summarization with sequence to sequence RNNs. *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 1–5. https://doi.org/10.1109/ICCCNT45670.2019.8944620

Muchasan, A., Nur Syam, & Anis Humaidi. (2024). Pemanfaatan Teknologi di Pesantren (Dampak dan Solusi dalam Konteks Pendidikan). *INOVATIF: Jurnal Penelitian Pendidikan, Agama, Dan Kebudayaan*, *10*(1), 16–33. https://doi.org/10.55148/inovatif.v10i1.849

Munawar, Z. (2019). Meningkatkan Kinerja Individu melalui Kritik/Saran menggunakan Recommender System. *TEMATIK*, *6*(1), 20–38. https://doi.org/10.38204/tematik.v6i1.185

Nallapati, R., Zhou, B., dos Santos, C., Gulcehre, C., & Xiang, B. (2016). Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, 280–290. https://doi.org/10.18653/v1/K16-1028

Nawangnugraeni, D. A., Abdillah, M. Z., & Al'Amin, M. (2023). Implementasi Aplikasi Android untuk Sistem Penjadwalan Kegiatan Pondok Pesantren PDF Walindo. *Jurnal Pengabdian Masyarakat Progresif Humanis Brainstorming*, *6*(1), 1–8. https://doi.org/10.30591/japhb.v6i1.3728

Okut, H. (2021). Deep Learning for Subtyping and Prediction of Diseases: Long-Short Term Memory. In *Deep Learning Applications*. IntechOpen. https://doi.org/10.5772/intechopen.96180

Patel, N., & Mangaokar, N. (2020). Abstractive vs Extractive Text Summarization (Output based approach) - A Comparative Study. *2020 IEEE International Conference for Innovation in Technology (INOCON)*, 1–6. https://doi.org/10.1109/INOCON50539.2020.9298416

Qian, P., Qiu, X., & Huang, X. (2016). Bridging LSTM Architecture and the Neural Dynamics during Reading. *International Joint Conference on Artificial Intelligence*. https://api.semanticscholar.org/CorpusID:9657569

Ranganathan, J., & Abuka, G. (2022). Text Summarization using Transformer Model. *2022 Ninth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, 1–5. https://doi.org/10.1109/SNAMS58071.2022.10062698

Ranzato, M., Chopra, S., Auli, M., & Zaremba, W. (2015). *Sequence Level Training with Recurrent Neural Networks*. http://arxiv.org/abs/1511.06732

Raundale, P., & Shekhar, H. (2021). Analytical study of Text Summarization Techniques. *2021 Asian Conference on Innovation in Technology (ASIANCON)*, 1–4. https://doi.org/10.1109/ASIANCON51346.2021.9544804

Reimers, N., & Gurevych, I. (2017). Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. *ArXiv Preprint ArXiv:1707.06799*.

Saputra, A. M., Al Maki, W. F., & Andini, N. (2021). *Peringkas Teks Otomatis Bahasa Indonesia secara Abstraktif Menggunakan Metode Long Short-Term Memory*. https://repository.telkomuniversity.ac.id/pustaka/167769/peringkas-teks-otomatis-bahasa-indonesia-secara-abstraktif-menggunakan-metode-long-short-term-memory.html

Sari, Y. M., & Fatonah, N. S. (2021). Peringkasan Teks Otomatis pada Modul Pembelajaran Berbahasa Indonesia Menggunakan Metode Cross Latent Semantic Analysis (CLSA). *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, *7*(2), 153. https://doi.org/10.26418/jp.v7i2.47768

Satvika, Thada, V., & Singh, J. (2021). *A Primer on Word Embedding* (pp. 525–541). https://doi.org/10.1007/978-981-15-8530-2_42

Shabanian, S., Arpit, D., Trischler, A., & Bengio, Y. (2017). Variational Bi-LSTMs. *ArXiv*, *abs/1711.05717*. https://api.semanticscholar.org/CorpusID:27949543

Shaliha, K. M. (2021). *Implementasi Bidirectional Long Short Term Memory pada peringkasan abstraktif teks berbahasa Indonesia*. UIN Sunan Gunung Djati Bandung.

Shihab, M. Q. (2002). Tafsir al-misbah. *Jakarta: Lentera Hati*, *2*, 52–54.

Shihab, M. Q. (2013). *Al Qur'an & Maknanya* (P. J. Syarfuan, S. R. Cahyono, & A. Halim, Eds.; Cetakan II). Penerbit Lentera Hati.

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2019a). The Performance of LSTM and BiLSTM in Forecasting Time Series. *2019 IEEE International Conference on Big Data (Big Data)*, 3285–3292. https://doi.org/10.1109/BigData47090.2019.9005997

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2019b). The Performance of LSTM and BiLSTM in Forecasting Time Series. *2019 IEEE International Conference on Big Data (Big Data)*, 3285–3292. https://doi.org/10.1109/BigData47090.2019.9005997

Singh, N., & Gaur, B. (2019). Data Preprocessing: A Step-by-Step Guide for Clean and Usable Data. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *10*(2), 1148–1153. https://doi.org/10.61841/turcomat.v10i2.14384

Siregar, M. K. (2018). Pondok Pesantren Antara Misi Melahirkan Ulama Dan Tarikan Modernisasi. *Jurnal Pendidikan Agama Islam Al-Thariqah*, *3*(2), 16–27. https://doi.org/10.25299/althariqah.2018.vol3(2).2263

Sohail, A., Aslam, U., Tariq, H. I., & Jayabalan, M. (2020). *METHODOLOGIES AND TECHNIQUES FOR TEXT SUMMARIZATION: A SURVEY*. https://api.semanticscholar.org/CorpusID:226746447

Srivastava, N., Hinton, G., Krizhevsky, A., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. In *Journal of Machine Learning Research* (Vol. 15).

Suleiman, D., & Awajan, A. (2020). Deep Learning Based Abstractive Text Summarization: Approaches, Datasets, Evaluation Measures, and Challenges. *Mathematical Problems in Engineering*, *2020*, 1–29. https://doi.org/10.1155/2020/9365340

Supriyono. (2019). Penerapan ISO 9126 Dalam Pengujian Kualitas Perangkat Lunak pada E-book. *MATICS*, *11*(1), 9. https://doi.org/10.18860/mat.v11i1.7672

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. *ArXiv*, *abs/1409.3215*. https://api.semanticscholar.org/CorpusID:7961699

Terven, J., Cordova-Esparza, D. M., Ramirez-Pedraza, A., Chavez-Urbiola, E. A., & Romero-Gonzalez, J. A. (2023). Loss functions and metrics in deep learning. *ArXiv Preprint ArXiv:2307.02694*.

Thange, S., Dange, J., Karjule, V., & Sase, J. (2023). A Survey on Text Summarization Techniques. *International Journal of Scientific and Research Publications*, *13*(11), 528–535. https://doi.org/10.29322/IJSRP.13.11.2023.p14355

Ther, A., Kumar Pandit, B., Ganguly, A., Chakraborty, A., & Banerjee, A. (2023). Resource-efficient VLSI Architecture of Softmax Activation Function for Real-time Inference in Deep Learning Applications. *2023 International Symposium on Devices, Circuits and Systems (ISDCS)*, 01–05. https://doi.org/10.1109/ISDCS58735.2023.10153520

Tripathy, S. A., & Ashok, S. (2023). Abstractive method-based Text Summarization using Bidirectional Long Short-Term Memory and Pointer Generator Mode. *Journal of Applied Research and Technology*, *21*(1), 73–86. https://doi.org/10.22201/icat.24486736e.2023.21.1.1446

Varade, S., Sayyed, E., Nagtode, V., & Shinde, S. (2021). Text Summarization using Extractive and Abstractive Methods. *ITM Web of Conferences*, *40*, 03023. https://doi.org/10.1051/itmconf/20214003023

Wu, Y., & Xing, Y. (2024). Efficient Machine Translation with a BiLSTM-Attention Approach. *ArXiv Preprint ArXiv:2410.22335*.

Yan, J. (Kevin), Benbya, H., & Leidner, D. E. (2023). Feedback types and users' behavior in online innovation contests: Evidence of two underlying mechanisms. *Information & Management*, *60*(3), 103755. https://doi.org/10.1016/j.im.2023.103755

Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, *415*, 295–316. https://doi.org/10.1016/j.neucom.2020.07.061

Yoyok Amirudin. (2020). Peran Pondok Pesantren dalam Mencegah Faham Radikalisme Agama. *TABYIN: JURNAL PENDIDIKAN ISLAM*, *2*(1), 92–103. https://doi.org/10.52166/tabyin.v2i1.24

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation*, *31*(7), 1235–1270. https://doi.org/10.1162/neco_a_01199

Yuliska, Y., & Syaliman, K. U. (2022). Peringkasan Dokumen Teks Otomatis Berdasarkan Sebuah Kueri Menggunakan Bidirectional Long Short Term Memory Network. *INTECOMS: Journal of Information Technology and Computer Science*, *5*(2), 65–71. https://doi.org/10.31539/intecoms.v5i2.4729

Zhang, Y., Liu, Q., & Song, L. (2018). Sentence-State LSTM for Text Representation. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 317–327. https://doi.org/10.18653/v1/P18-1030

Zhu, S., & Yu, K. (2017). Encoder-decoder with focus-mechanism for sequence labelling based spoken language understanding. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5675–5679. https://doi.org/10.1109/ICASSP.2017.7953243

LAMPIRAN

Lampiran 1

KUESIONER

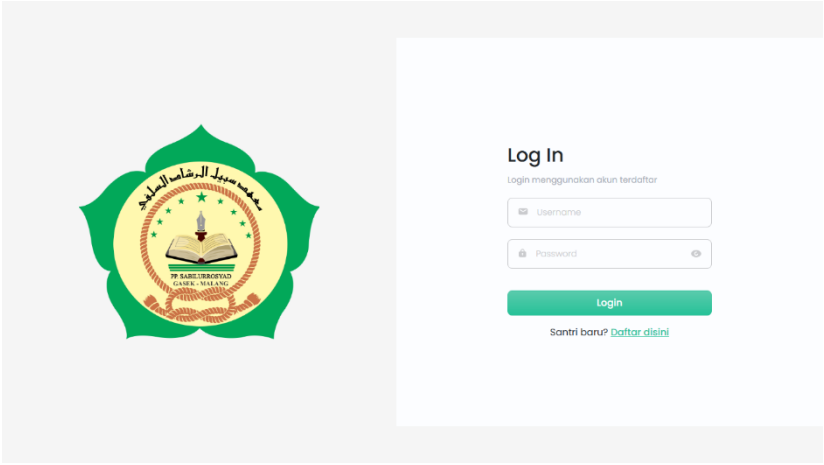
Prosedur Pengisian

1. Responden memasukkan nama lengkap.
2. Responden memasukkan nomor kamar.
3. Apabila responden ingin melihat fitur SiBayar, responden dapat melihat *preview* berupa tangkapan laman SiBayar atau mengakses tautan SiBayar secara langsung.
4. Responden harus menjawab setiap pertanyaan dan menjawab sesuai pertanyaan.
5. Responden menjawab pertanyaan paling sedikit 20 kata.

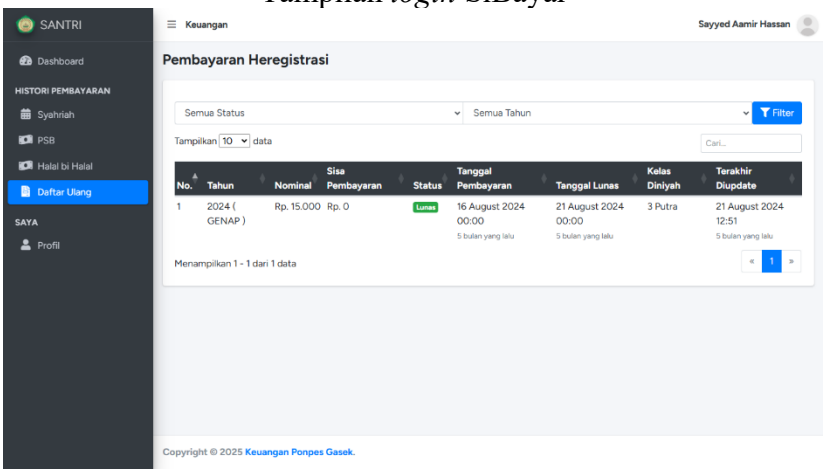
Daftar Pertanyaan

1. Jelaskan pengalaman Anda dalam mengakses form daftar ulang (SANTRI LAMA)
   Bagaimana pengalaman Anda dalam membuat Akun Santri Baru (SANTRI BARU)
2. Bagaimana pendapat Anda tentang kelengkapan dan relevansi data yang diminta dalam form daftar ulang (santri lama) atau form pembuatan akun santri baru (santri baru)?
3. Ceritakan kendala teknis (contoh: *error*, loading lambat, atau gagal menyimpan data) saat Anda mengisi form daftar ulang/form pendaftaran santri baru? Jika ada, ceritakan kendala tersebut.
4. Bagaimana pengalaman Anda dalam mengakses SiBayar untuk melihat tagihan *syahriah* akses Wi-Fi
5. Apa saran Anda untuk meningkatkan pengalaman pengguna secara keseluruhan dalam fitur-fitur yang ada dalam SiBayar?

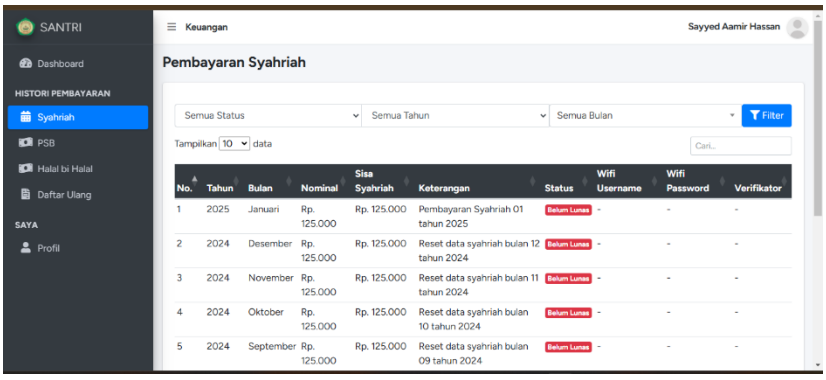
Lampiran 2



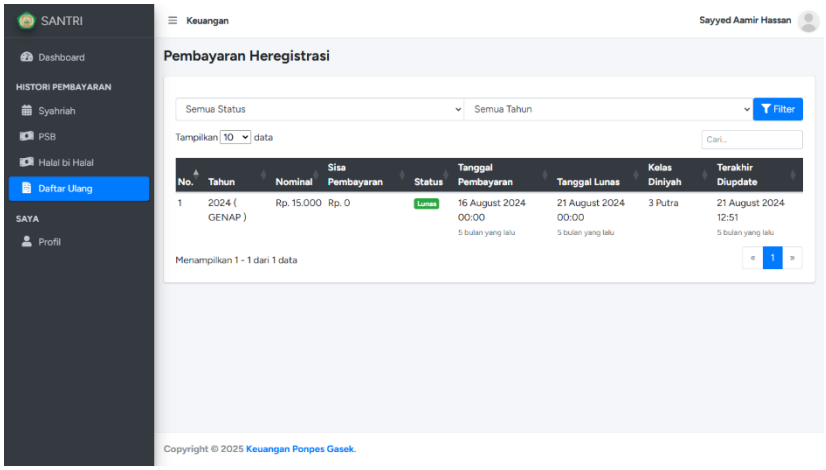
Tampilan *login* SiBayar



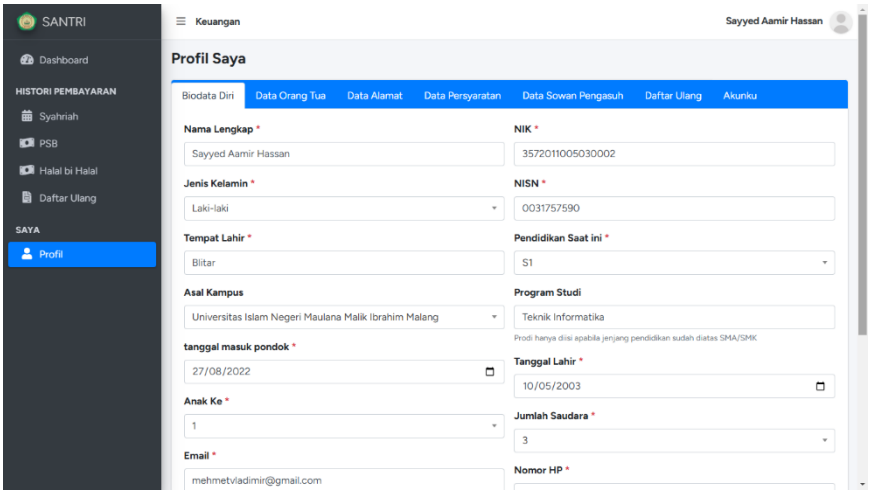
Laman daftar ulang santri



Laman *syahriah* dan akses WiFi



Laman *herregistrasi*



Laman formulir *herregistrasi*

Lampiran 3

Profil ahli sebagai pembuat ringkasan manusia

Nama : Aisyatul Azizah, M.H

Riwayat Pendidikan

1. S-1 Hukum Keluarga Islam - UIN Sunan Kalijaga Yogyakarta
2. S-2 Hukum Keluarga Islam - UIN Sunan Kalijaga Yogyakarta
3. S-3 Hukum Keluarga Islam - UIN Maulana Malik Ibrahim Malang

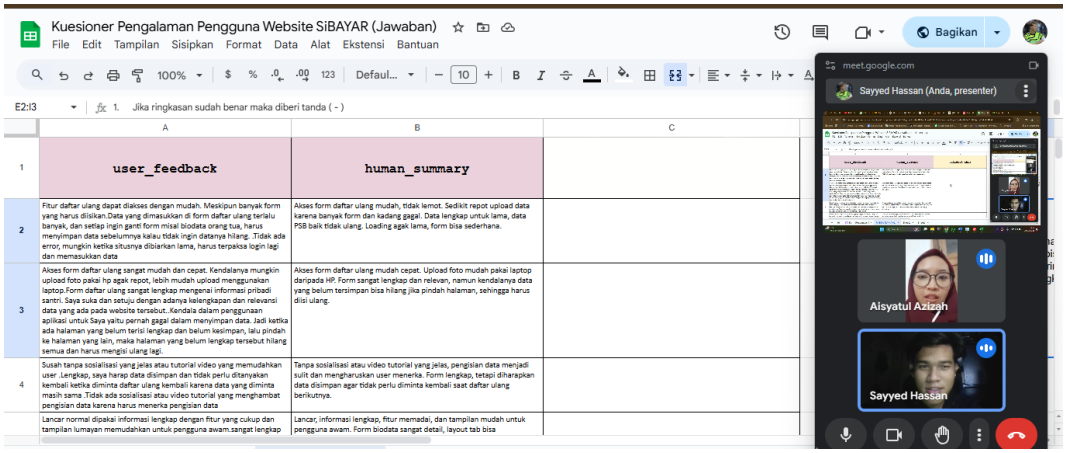
Riwayat Organisasi

1. Ketua PC IPPNU Kota Yogyakarta
2. Divisi Organisasi Pusat IPPNU
3. Ketua PC Fatayat NU Kota Blitar
4. Tim Media NU Kota Blitar

Riwayat Profesi

1. Direktur TKAL – TPAL AMM Yogyakarta
2. Penyuluh Agama Islam Kota Blitar
3. Kepala SMP Bustanul Muta'allimin Kota Blitar
4. Dosen Hukum Keluarga Islam di UNU Blitar

Dokumentasi dengan ahli



Lampiran 4

Hasil pengujian ringkasan sistem

No.	*Machine Summary*	*Reference Summary*	ROUGE-1	ROUGE-2	ROUGE-L
1	sibayar sudah bagus saran saya aplikasinya cukup dikembangkan lagi agar bisa memuat lebih banyak fitur seperti surat izin diniyah	fitur menu dan belum pembayaran agar santri yang sudah lunas untuk mudah dalam pendaftaran dan pembayaran	0.158	0.000	0.061
2	fitur bisa ditambah dengan sistem pembayaran syahriah yang otomatis mendeteksi rekening saat membayar tagihan bulanan jadi tidak perlu konfirmasi ke bendahara	sistem bisa ditambah dengan sistem pembayaran syahriah yang otomatis mendeteksi rekening saat membayar tagihan bulanan jadi	0.714	0.700	0.833
3	saran untuk fitur daftar ulang lebih disederhanakan formnya menu di bawah atau di samping sebaiknya pilih salah satu karena keduanya isinya sama	fitur daftar ulang lebih disederhanakan formnya menu di bawah atau di samping sebaiknya pilih salah satu karena keduanya isinya sama	0.905	0.905	0.950
4	diberikan sistem otomatis yang mendeteksi pembayaran masuk dan langsung terdeteksi di sistem agar tidak perlu konfirmasi ke bendahara yang tidak cepat balas chat	sistem otomatis yang mendeteksi pembayaran masuk dan langsung terdeteksi di sistem agar	0.550	0.500	0.710
5	tambahkan validasi di nomor hp nik atau input lain biar santri tidak ngasal isi tambahkan fitur pembayaran keamanan dan diniyah biar lengkap	validasi di nomor hp nik atau input lain biar santri	0.500	0.429	0.667
6	wifi bisa buat hotspot biar santri nggak kehabisan paketan tambahkan juga fitur perizinan diniyah biar bisa di sibayar	wifi bisa buat hotspot biar santri nggak kehabisan paketan tambahkan juga fitur perizinan	1.000	1.000	1.000

		diniyah biar bisa di sibayar			
7	tingkatkan fitur buat semua pembayaran di sibayar dan perbaiki form daftar ulang	sibayar buat semua pembayaran di sibayar dan perbaiki form daftar ulang	0.833	0.818	0.909
8	buat pembayaran otomatis tanpa hubungi bendahara dan sederhanakan form daftar ulang untuk santri lama	sistem bagus tapi sudah bagus pengguna tampilan bisa izin dan sosialisasi fitur wifi tapi tampilan	0.071	0.000	0.077
9	perbaiki form daftar ulang dan buat layanan syahriah serta wifi otomatis tanpa perlu hubungi bendahara	menyimpan lancar buat semua pembayaran otomatis dan password wifi otomatis tanpa perlu hubungi bendahara	0.533	0.357	0.500
10	fiturnya sudah bagus ke depan bisa ditambah fitur lain supaya santri lebih mudah dalam pendaftaran dan pembayaran	sudah bagus tapi harus tagihan sebaiknya fitur lain agar pembayaran agar pembayaran dan pendaftaran cukup lewat sibayar	0.412	0.125	0.312
11	semua sudah bagus semoga tim it sibayar menambah fitur lagi agar pembayaran dan pendaftaran cukup lewat sibayar tanpa perlu mencari pengurus	sudah sudah bagus semoga tim it sibayar menambah fitur lagi agar pembayaran dan pendaftaran cukup lewat sibayar tanpa perlu mencari pengurus	0.950	0.950	0.974
12	tambahkan fitur persuratan dan perizinan biar santri bisa izin dan bayar langsung di sibayar	sibayar persuratan dan perizinan biar santri bisa izin dan bayar langsung di sibayar	0.846	0.846	0.917
13	semua pembayaran sebaiknya lewat sibayar agar lebih mudah form daftar ulang yang terlalu banyak sebaiknya dikurangi	agar sibayar menambah panduan yang jelas agar pengguna mudah dalam akses fitur fitur	0.267	0.000	0.231
14	semua sudah bagus tambahkan fitur deteksi otomatis pembayaran seperti siakad kampus agar tanpa kirim bukti ke bendahara	sudah bagus tambahkan fitur deteksi otomatis pembayaran seperti siakad kampus agar tanpa	0.941	0.938	0.970

		kirim bukti ke bendahara			
15	permudah form daftar ulang tambah fitur pembayaran otomatis dan kirim notifikasi tunggakan ke orang tua	fitur dipermudah ulang tambah fitur pembayaran otomatis dan kirim notifikasi tunggakan ke orang tua	0.800	0.786	0.857
16	sebaiknya akses dipermudah atau dibuat aplikasi di play store dengan notifikasi langsung bukan hanya whatsapp	akses dipermudah atau dibuat aplikasi di play store dengan notifikasi langsung bukan hanya whatsapp	0.933	0.929	0.966
17	semua fitur sudah bagus tapi sebaiknya ditambah pembayaran surat izin diniyah dan pengajuan persuratan agar website lebih bermanfaat tidak hanya untuk tagihan dan daftar ulang	fitur fitur sudah bagus tapi sebaiknya ditambah pembayaran surat izin diniyah dan pengajuan persuratan agar website lebih bermanfaat	0.708	0.667	0.829
18	sudah bagus tingkatkan lagi imbauan pakai data saat daftar ulang dan sosialisasi fitur baru diperlukan	sudah bagus tapi data data	0.200	0.071	0.316
19	perbaiki form daftar ulang agar lebih efektif tambah fitur baru dan sosialisasikan di grup pondok	fitur yang halaman tambahkan petunjuk tiap dan konfirmasi di grup agar	0.333	0.071	0.308
20	fiturnya sudah baik bisa ditambah lagi atau dibuat aplikasi android agar notifikasi muncul di handphone	baik ditambah lagi atau dibuat aplikasi android agar notifikasi muncul di handphone	0.800	0.714	0.889
21	webnya sangat bagus sebaiknya semua pembayaran santri dapat dilakukan di sibayar untuk memudahkan administrasi	fitur sangat bagus sebaiknya semua pembayaran santri dapat dilakukan di sibayar untuk memudahkan administrasi	0.929	0.923	0.929
22	buat tampilan lebih simpel dan menarik serta tambahkan virtual account untuk pembayaran syahriah	respon di sibayar dan perbaiki	0.077	0.000	0.111
23	semoga fitur web ini lebih menarik dan	sibayar web ini lebih menarik dan	0.846	0.833	0.880

	digunakan sukarela seperti amal masjid keliling	digunakan sukarela seperti amal masjid keliling			

LAMPIRAN 5

Hasil pengujian *hyperparameter tuning*

No.	*optimizer*	*num_layers*	*batch_size*	*dropout_type*	*dropout_rate*	Loss	*Accuracy*
1	nadam	1	1	variational	0.5	1.144911	88.74%
2	nadam	1	1	naive	0.2	1.112714	88.21%
3	adam	1	1	variational	0.5	1.228261	88.00%
4	adam	1	1	variational	0.2	1.148281	86.84%
5	adam	1	1	naive	0.2	1.200744	86.74%
6	nadam	1	1	variational	0.2	1.204348	86.42%
7	nadam	2	1	no_dropout	0	1.297162	86.32%
8	nadam	1	1	no_dropout	0	1.278518	85.89%
9	adam	1	1	no_dropout	0	1.288385	85.68%
10	nadam	1	1	naive	0.5	1.233259	85.16%
11	adam	1	1	naive	0.5	1.330149	84.84%
12	adam	2	1	no_dropout	0	1.444617	82.95%
13	rmsprop	1	1	variational	0.5	1.128594	82.84%
14	adam	2	1	variational	0.2	1.615461	81.89%
15	nadam	2	1	variational	0.5	1.613418	81.79%
16	rmsprop	1	1	variational	0.2	1.20361	81.26%
17	adam	1	8	variational	0.2	1.881002	80.95%
18	rmsprop	1	1	naive	0.2	1.346548	79.58%
19	adam	2	1	naive	0.2	1.687256	78.63%
20	nadam	2	1	variational	0.2	1.601973	78.63%
21	adam	1	8	naive	0.5	1.856993	78.53%
22	nadam	2	1	naive	0.2	1.781457	78.21%
23	rmsprop	1	1	no_dropout	0	1.418847	78.21%
24	rmsprop	1	1	naive	0.5	1.46492	78.21%
25	adam	1	8	variational	0.5	1.999172	77.79%
26	adam	1	8	naive	0.2	2.02434	76.53%
27	adam	1	16	naive	0.2	2.137739	76.32%
28	nadam	2	1	naive	0.5	1.891327	75.37%
29	adam	2	1	naive	0.5	1.814539	74.95%
30	adam	1	8	no_dropout	0	2.104012	74.84%
31	rmsprop	1	8	no_dropout	0	2.028894	71.89%
32	rmsprop	2	1	variational	0.2	1.817903	70.95%
33	adam	1	32	naive	0.2	2.24516	70.74%
34	rmsprop	1	8	naive	0.5	2.064446	70.21%
35	rmsprop	2	1	variational	0.5	1.762544	69.89%
36	rmsprop	2	1	naive	0.2	1.810981	69.58%
37	rmsprop	1	8	naive	0.2	2.141551	69.05%

No.	*optimizer*	*num_layers*	*batch_size*	*dropout_type*	*dropout_rate*	Loss	*Accuracy*
38	rmsprop	2	1	naive	0.5	1.863546	68.84%
39	rmsprop	3	1	no_dropout	0	1.891892	68.11%
40	rmsprop	2	1	no_dropout	0	1.806352	68.00%
41	adam	2	8	variational	0.2	2.361237	67.58%
42	rmsprop	3	1	variational	0.2	1.812621	67.58%
43	rmsprop	3	1	variational	0.5	1.85587	67.26%
44	rmsprop	3	1	naive	0.2	1.779979	67.16%
45	rmsprop	1	8	variational	0.2	2.177938	66.95%
46	nadam	3	8	variational	0.5	2.270718	66.21%
47	rmsprop	4	1	no_dropout	0	1.867972	66.11%
48	adam	2	8	naive	0.2	2.416692	65.79%
49	nadam	3	1	no_dropout	0	2.161252	65.37%
50	rmsprop	2	8	variational	0.5	2.140892	64.74%
51	nadam	3	1	naive	0.2	2.191513	64.32%
52	nadam	3	1	variational	0.2	2.174867	64.00%
53	nadam	1	8	variational	0.2	2.384904	63.89%
54	adam	1	16	variational	0.5	2.485695	63.79%
55	nadam	1	8	variational	0.5	2.469128	63.47%
56	adam	2	16	no_dropout	0	2.595231	63.26%
57	rmsprop	2	8	naive	0.5	2.341336	63.05%
58	adam	2	8	variational	0.5	2.440281	63.05%
59	adam	2	1	variational	0.5	2.362024	62.84%
60	adam	2	8	no_dropout	0	2.530141	62.74%
61	adam	1	16	naive	0.5	2.472628	62.63%
62	adam	1	32	naive	0.5	2.537179	62.63%
63	nadam	1	16	naive	0.2	2.633418	62.63%
64	nadam	1	8	no_dropout	0	2.459873	62.63%
65	adam	1	16	variational	0.2	2.471888	62.63%
66	nadam	2	8	naive	0.5	2.594181	62.53%
67	adam	1	16	no_dropout	0	2.458639	62.53%
68	adam	2	16	variational	0.2	2.645699	62.53%
69	rmsprop	2	8	variational	0.2	2.377522	62.53%
70	nadam	1	16	variational	0.2	2.502084	62.53%
71	adam	2	16	naive	0.2	2.579638	62.42%
72	adam	3	16	no_dropout	0	2.63898	62.42%
73	nadam	1	16	variational	0.5	2.502169	62.42%
74	adam	2	16	naive	0.5	2.511913	62.42%
75	nadam	1	32	variational	0.2	2.532564	62.42%
76	adam	3	1	naive	0.5	2.497401	62.32%
77	adam	3	8	no_dropout	0	2.526024	62.32%
78	nadam	1	32	variational	0.5	2.462288	62.32%
79	nadam	3	16	naive	0.5	2.554773	62.32%

No.	*optimizer*	*num_layers*	*batch_size*	*dropout_type*	*dropout_rate*	Loss	*Accuracy*
80	rmsprop	4	1	naive	0.2	2.287521	62.32%
81	nadam	1	32	no_dropout	0	2.487551	62.32%
82	nadam	2	8	variational	0.2	2.48505	62.32%
83	nadam	4	1	no_dropout	0	2.510416	62.21%
84	adam	3	1	no_dropout	0	2.408577	62.21%
85	nadam	2	32	naive	0.5	2.593005	62.21%
86	nadam	1	8	naive	0.2	2.693054	62.21%
87	nadam	2	8	naive	0.2	2.571297	62.21%
88	nadam	2	16	no_dropout	0	2.567508	62.21%
89	nadam	4	8	no_dropout	0	2.616103	62.21%
90	rmsprop	2	8	no_dropout	0	2.403027	62.21%
91	nadam	2	8	variational	0.5	2.492713	62.21%
92	nadam	2	16	variational	0.5	2.566024	62.21%
93	nadam	4	16	variational	0.5	2.597714	62.21%
94	adam	1	32	variational	0.5	2.462224	62.21%
95	adam	3	8	variational	0.5	2.518792	62.21%
96	nadam	4	1	variational	0.2	2.516062	62.11%
97	nadam	1	16	naive	0.5	2.567308	62.11%
98	adam	3	32	naive	0.2	2.462754	62.11%
99	rmsprop	4	1	variational	0.2	2.323257	62.11%
100	nadam	1	8	naive	0.5	2.571954	62.11%
101	adam	2	8	naive	0.5	2.596477	62.11%
102	nadam	3	8	naive	0.2	2.547179	62.11%
103	adam	3	32	no_dropout	0	2.552089	62.11%
104	rmsprop	4	8	no_dropout	0	2.410195	62.11%
105	adam	2	32	variational	0.5	2.53826	62.11%
106	nadam	2	32	variational	0.2	2.650294	62.11%
107	adam	3	32	variational	0.2	2.555011	62.11%
108	rmsprop	3	8	no_dropout	0	2.353629	62.00%
109	nadam	4	1	variational	0.5	2.470955	62.00%
110	adam	3	1	variational	0.5	2.401381	62.00%
111	adam	1	32	variational	0.2	2.490998	62.00%
112	rmsprop	1	32	variational	0.5	2.39409	62.00%
113	adam	3	16	naive	0.2	2.693634	62.00%
114	adam	3	1	naive	0.2	2.445759	61.89%
115	adam	4	8	naive	0.5	2.526386	61.89%
116	rmsprop	1	16	naive	0.2	2.356132	61.89%
117	nadam	2	8	no_dropout	0	2.571889	61.89%
118	adam	4	32	no_dropout	0	2.624289	61.89%
119	rmsprop	2	32	no_dropout	0	2.458838	61.89%
120	adam	3	16	variational	0.5	2.528002	61.89%
121	adam	2	32	variational	0.2	2.686587	61.89%

No.	*optimizer*	*num_layers*	*batch_size*	*dropout_type*	*dropout_rate*	Loss	*Accuracy*
122	adam	3	16	variational	0.2	2.641593	61.89%
123	rmsprop	3	8	variational	0.2	2.366688	61.89%
124	rmsprop	4	1	variational	0.5	2.3023	61.89%
125	adam	2	32	naive	0.2	2.653783	61.89%
126	nadam	1	16	no_dropout	0	2.461667	61.89%
127	nadam	3	32	no_dropout	0	2.583977	61.89%
128	adam	1	32	no_dropout	0	2.507282	61.89%
129	rmsprop	1	16	no_dropout	0	2.399734	61.89%
130	nadam	3	1	variational	0.5	2.427877	61.79%
131	adam	3	1	variational	0.2	2.410651	61.79%
132	nadam	2	16	naive	0.5	2.590897	61.79%
133	nadam	1	32	naive	0.2	2.437074	61.79%
134	nadam	2	32	naive	0.2	2.602375	61.79%
135	rmsprop	3	8	naive	0.5	2.390154	61.79%
136	adam	4	1	no_dropout	0	2.419359	61.79%
137	adam	4	16	no_dropout	0	2.593234	61.79%
138	nadam	3	16	variational	0.5	2.717593	61.79%
139	adam	2	16	variational	0.5	2.523146	61.79%
140	adam	3	32	variational	0.5	2.593361	61.79%
141	nadam	3	8	variational	0.2	2.764401	61.79%
142	nadam	2	16	naive	0.2	2.691493	61.68%
143	rmsprop	1	32	naive	0.5	2.408773	61.68%
144	nadam	3	16	no_dropout	0	2.818657	61.68%
145	adam	2	32	no_dropout	0	2.587397	61.68%
146	nadam	3	16	variational	0.2	2.556082	61.68%
147	rmsprop	3	16	variational	0.2	2.489287	61.68%
148	nadam	2	32	variational	0.5	2.580677	61.58%
149	nadam	4	32	variational	0.5	2.65062	61.58%
150	rmsprop	4	8	variational	0.5	2.464917	61.58%
151	adam	4	8	variational	0.5	2.52215	61.58%
152	rmsprop	3	8	variational	0.5	2.383479	61.58%
153	rmsprop	1	16	naive	0.5	2.368356	61.58%
154	adam	2	32	naive	0.5	2.680014	61.47%
155	adam	4	1	variational	0.5	2.491288	61.47%
156	nadam	2	16	variational	0.2	2.591554	61.47%
157	adam	4	16	naive	0.2	2.725829	61.37%
158	rmsprop	2	32	naive	0.5	2.534356	61.37%
159	adam	3	8	variational	0.2	2.545067	61.37%
160	adam	4	1	variational	0.2	2.63404	61.37%
161	rmsprop	2	16	variational	0.5	2.413314	61.37%
162	rmsprop	2	32	variational	0.5	2.461067	61.37%
163	rmsprop	2	16	naive	0.2	2.458004	61.26%

No.	*optimizer*	*num_layers*	*batch_size*	*dropout_type*	*dropout_rate*	Loss	*Accuracy*
164	rmsprop	2	16	no_dropout	0	2.387129	61.26%
165	rmsprop	1	8	variational	0.5	2.341688	61.26%
166	rmsprop	1	32	naive	0.2	2.449719	61.26%
167	rmsprop	3	1	naive	0.5	2.599431	61.16%
168	adam	3	8	naive	0.2	2.662968	61.16%
169	rmsprop	3	16	naive	0.5	2.461143	61.16%
170	nadam	1	32	naive	0.5	2.495084	61.16%
171	rmsprop	2	32	variational	0.2	2.456096	61.16%
172	rmsprop	1	16	variational	0.5	2.362951	61.16%
173	adam	3	8	naive	0.5	2.841872	61.05%
174	rmsprop	1	32	variational	0.2	2.569478	61.05%
175	nadam	2	32	no_dropout	0	2.598432	60.95%
176	adam	4	32	variational	0.5	2.941041	60.95%
177	nadam	3	1	naive	0.5	2.857132	60.84%
178	nadam	3	32	naive	0.5	2.872651	60.84%
179	adam	4	8	variational	0.2	3.104322	60.84%
180	nadam	3	8	naive	0.5	2.923928	60.74%
181	rmsprop	4	8	naive	0.2	2.887697	60.74%
182	rmsprop	1	16	variational	0.2	2.396788	60.74%
183	nadam	3	32	variational	0.2	2.884948	60.74%
184	nadam	4	8	variational	0.2	2.907758	60.74%
185	nadam	4	32	variational	0.2	2.966883	60.74%
186	nadam	3	32	naive	0.2	2.866433	60.63%
187	rmsprop	3	32	no_dropout	0	2.858274	60.63%
188	adam	4	16	variational	0.2	2.990439	60.63%
189	nadam	4	8	naive	0.5	2.877853	60.63%
190	rmsprop	4	32	naive	0.5	2.578861	60.63%
191	rmsprop	4	1	naive	0.5	2.724541	60.53%
192	adam	3	32	naive	0.5	3.17015	60.53%
193	nadam	4	1	naive	0.2	2.891389	60.53%
194	adam	4	8	naive	0.2	2.941415	60.53%
195	rmsprop	3	8	naive	0.2	2.778354	60.53%
196	rmsprop	3	32	naive	0.5	3.772414	60.53%
197	rmsprop	3	32	variational	0.5	3.057322	60.53%
198	rmsprop	2	16	naive	0.5	2.712473	60.53%
199	rmsprop	1	32	no_dropout	0	2.492682	60.53%
200	adam	3	16	naive	0.5	2.994412	60.42%
201	adam	4	32	naive	0.5	3.205667	60.42%
202	nadam	3	16	naive	0.2	2.953252	60.42%
203	adam	4	32	naive	0.2	3.394743	60.42%
204	nadam	4	16	no_dropout	0	3.083358	60.42%
205	adam	4	8	no_dropout	0	2.866661	60.42%

No.	*optimizer*	*num_layers*	*batch_size*	*dropout_type*	*dropout_rate*	Loss	*Accuracy*
206	rmsprop	4	32	no_dropout	0	2.974523	60.42%
207	rmsprop	2	16	variational	0.2	2.50809	60.42%
208	rmsprop	4	32	variational	0.2	3.306109	60.42%
209	rmsprop	3	16	variational	0.5	2.961857	60.42%
210	adam	4	1	naive	0.5	2.630443	60.32%
211	adam	4	16	naive	0.5	4.029974	60.32%
212	rmsprop	2	8	naive	0.2	3.114218	60.32%
213	rmsprop	2	32	naive	0.2	3.083008	60.32%
214	rmsprop	3	16	naive	0.2	3.115518	60.32%
215	rmsprop	3	32	naive	0.2	3.15345	60.32%
216	rmsprop	4	8	naive	0.5	2.846509	60.32%
217	rmsprop	3	16	no_dropout	0	3.049287	60.32%
218	nadam	3	32	variational	0.5	2.889378	60.32%
219	nadam	4	16	variational	0.2	3.156633	60.32%
220	adam	4	32	variational	0.2	3.146433	60.32%
221	rmsprop	4	8	variational	0.2	3.016263	60.32%
222	rmsprop	4	16	variational	0.2	2.946331	60.32%
223	nadam	4	32	naive	0.5	2.856368	60.32%
224	nadam	4	16	naive	0.2	2.892906	60.32%
225	rmsprop	4	16	no_dropout	0	2.915859	60.32%
226	nadam	4	8	variational	0.5	2.880082	60.21%
227	rmsprop	4	16	variational	0.5	2.810408	60.21%
228	rmsprop	4	16	naive	0.5	2.821587	60.11%
229	adam	4	16	variational	0.5	2.984926	60.11%
230	nadam	4	8	naive	0.2	2.89781	60.00%
231	nadam	4	32	naive	0.2	2.832213	60.00%
232	rmsprop	4	32	naive	0.2	2.884353	59.68%
233	nadam	3	8	no_dropout	0	2.827358	59.68%
234	nadam	4	1	naive	0.5	3.200153	59.47%
235	rmsprop	4	32	variational	0.5	2.721359	58.53%
236	rmsprop	4	16	naive	0.2	3.606459	57.58%
237	adam	4	1	naive	0.2	2.995351	57.05%
238	nadam	4	16	naive	0.5	3.74073	55.58%
239	nadam	4	32	no_dropout	0	3.278875	54.21%
240	rmsprop	3	32	variational	0.2	3.896512	48.11%

LAMPIRAN 6

Surat penelitian dari fakultas

KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jalan Gajayana 50 Malang 65144 Telepon/Faksimile (0341) 558933
Website: http://saintek.uin-malang.ac.id, email: saintek@uin-malang.ac.id

Nomor : B-119.O/FST.01/TL.00/10/2024
Lampiran : -
Hal : Permohonan Penelitian

Yth. Pimpinan Pondok Pesantren Sabilurrosyad Gasek Malang
Jl. Raya Candi VI C No.303, Karangbesuki, Kec. Sukun, Kota Malang, Jawa Timur 65146

Dengan hormat,
Sehubungan dengan penelitian mahasiswa Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang atas nama:

Nama	: Sayyed Aamir Hassan
NIM	: 210605110098
Judul Penelitian	: IMPLEMENTASI METODE BI-LSTM UNTUK TEXT SUMMARIZATION UMPAN BALIK PENGGUNA DALAM PENGUJIAN PERANGKAT LUNAK SIBAYAR PONDOK PESANTREN SABILURROSYAD
Dosen Pembimbing	: SUPRIYONO,M.Kom

Maka kami mohon Bapak/Ibu berkenan memberikan izin pada mahasiswa tersebut untuk melakukan penelitian di Pondok Pesantren Sabilurrosyad Gasek Malang dengan waktu pelaksanaan pada tanggal 07 Oktober 2024 sampai dengan 02 Desember 2024.

Demikian permohonan ini, atas perhatian dan kerjasamanya disampaikan terimakasih.

Malang, 07 Oktober 2024
a.n Dekan

Scan QRCode ini

untuk verifikasi surat

Wakil Dekan Bidang Akademik,

Dr. Anton Prasetyo, M.Si
NIP. 19770925 200604 1 003

LAMPIRAN 7

Surat balasan persetujuan penelitian dari Pondok Pesantren Sabilurrosyad Gasek

معهد سبيل الرشاد الإسلامي السلفي
PONDOK PESANTREN SABILURROSYAD
GASEK KARANGBESUKI SUKUN MALANG
Sekretariat Jl. Candi Blok VI C Karangbesuki Sukun Malang
Telp. (0341) 564446 NSPP : 510035730051 website : ponpesgasek.id

Nomor : 88.11/SB/PPSR/XI/2024 Malang, 13 November 2024
Lampiran : -
Perihal : Peretujuan Penelitian

Kepada Yth.
Wakil Dekan Bidang Akademik
Di – Malang

Assalamu'alaikum Wr. Wb.

Menunjuk Surat dari Universitas Islam Negeri Malang Fakultas Sains dan Teknologi tanggal 7 Oktober 2024 Nomor : B-119.O/FST.01/TL.00/10/2024 tentang perizinan melakukan penelitian terhitung mulai dari Oktober 2024 hingga Februari 2025 maka dengan ini disampaikan secara hormat bahwa kami Pondok Pesantren Sabilurrosyad tidak keberatan memberi kesempatan untuk penelitian mahasiswa Fakultas Sains dan Teknologi UIN Maulana Malik Malang. Berikut ini mahasiswa yang akan melaksanakan penelitian di Pondok Pesantren Sabilurrosyad Gasek :

Nama	:	Sayyed Aamir Hassan
Judul Penelitian	:	Implementasi Metode BI-LSTM untuk text summarization umpan balik pengguna dalam pengujian perangkat lunak Sibayar Pondok Pesantren Sabilurrosyad

Demikian surat ini kami sampaikan atas kerja samaya kami ucapkan terima kasih.
Wassalamu'alaikum Wr. Wb.

Malang, 13 November 2023
Ketua Pondok Pesantren Sabilurrosyad

Muhammad Fikril Hakim