

Abstractive Text Summarization Umpan Balik Pengguna dengan Metode Bi-LSTM

Sayyed Aamir Hassan, Supriyono, Zainal Abidin

Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang, Jl. Gajayana No.50, Dinoyo, Kec. Lowokwaru, Kota Malang, Jawa Timur 65144, Indonesia

Info Artikel

Riwayat Artikel:

Received 2025-05-26

Revised 2025-09-10

Accepted 2025-09-16

Abstract – The SiBayar website was developed to support administration and payment management at Sabilurrosyad Islamic Boarding School. To optimize its functionality, developers require user feedback. However, the large volume of feedback makes manual analysis inefficient. Therefore, an automated approach such as text summarization is required to condense and extract essential information. This study aims to generate an automatic summary of user feedback on the SiBayar website using the Bidirectional Long Short-Term Memory (Bi-LSTM) method. The dataset was collected through questionnaires distributed in February 2025 to 120 student respondents, covering five questions grouped into three categories: registration, monthly billing, and suggestions. Manual summaries created by language experts served as the reference. The model was trained with a hyperparameter tuning scenario involving 240 parameter configurations, including optimizer, number of layers, batch size, and dropout type. Experimental results show that the best configuration was obtained using the Nadam optimizer, one layer, batch size of 1, and variational dropout of 0.5. The quality of the generated summaries was evaluated using ROUGE metrics, achieving ROUGE-1 of 0.6221, ROUGE-2 of 0.5462, and ROUGE-L of 0.660. These results indicate that Bi-LSTM is effective for text summarization tasks on small-scale, Indonesian-language corpora, although limitations remain in maintaining word order and coherence. This research contributes to enriching NLP studies in the pesantren domain and provides practical implications for improving the efficiency of user feedback management.

Keywords: Bidirectional Long Short Term Memory; Hyperparameter Tuning; Text Summarization; User Feedback.

Corresponding Author:

Supriyono

Email: priyono@ti.uin-malang.ac.id



This is an open access article under the [CC BY 4.0](https://creativecommons.org/licenses/by/4.0/) license.

Abstrak – Website SiBayar dikembangkan untuk mendukung pengelolaan administrasi dan pembayaran di Pondok Pesantren Sabilurrosyad. Agar layanan semakin optimal, pengembang membutuhkan umpan balik dari pengguna. Namun, jumlah umpan balik yang banyak membuat analisis manual menjadi tidak efisien. Oleh karena itu, diperlukan pendekatan otomatis seperti text summarization untuk menyarikan masukan secara ringkas. Penelitian ini bertujuan menghasilkan ringkasan otomatis dari umpan balik pengguna website SiBayar menggunakan metode Bidirectional Long Short-Term Memory (Bi-LSTM). Data diperoleh melalui kuesioner yang disebarkan pada Februari 2025 kepada 120 responden santri, dengan lima pertanyaan yang dikelompokkan ke dalam tiga kategori (registrasi, bulanan, dan saran). Ringkasan manual dari ahli bahasa digunakan sebagai pembandingan. Model dilatih dengan skenario hyperparameter tuning menggunakan kombinasi 240 konfigurasi parameter (optimizer, jumlah lapisan, batch size, dan dropout). Hasil pengujian menunjukkan konfigurasi terbaik diperoleh dengan optimizer Nadam, satu lapisan, batch size 1, dan variational dropout 0,5. Evaluasi kualitas ringkasan menggunakan metrik ROUGE menghasilkan skor ROUGE-1 sebesar 0,6221, ROUGE-2 sebesar 0,5462, dan ROUGE-L sebesar 0,660. Hasil ini menunjukkan bahwa Bi-LSTM efektif untuk tugas peringkasan teks dalam korpus kecil dan berbahasa Indonesia, meskipun masih ada keterbatasan dalam koherensi urutan kata. Penelitian ini berkontribusi pada eksplorasi bidang studi NLP di dunia pesantren serta memiliki implikasi praktis dalam meningkatkan efisiensi pengelolaan umpan balik pengguna.

Kata Kunci: Bidirectional Long Short Term Memory, Hyperparameter Tuning, Text Summarization, Umpan Balik.

I. PENDAHULUAN

Transformasi digital telah menjadi bagian penting dalam berbagai sektor, termasuk lembaga pendidikan keagamaan seperti pesantren. Banyak pesantren mulai memanfaatkan teknologi untuk mendukung kegiatan belajar, administrasi, dan pengelolaan santri [1]. Pemanfaatan aplikasi mobile dan sistem informasi internal terbukti membantu efektivitas manajemen lembaga [2]. Beberapa penelitian menunjukkan bahwa penggunaan teknologi dapat memperluas akses pendidikan bagi santri. Selain itu, teknologi juga berperan dalam memperkaya sumber belajar dan memperbaiki metode evaluasi hasil belajar. Manfaat lain yang dirasakan adalah peningkatan transparansi dalam pengelolaan data dan layanan akademik [3]. Meskipun demikian, tidak semua pesantren

mengadopsi sistem informasi secara penuh. Sebagian masih mengandalkan cara manual atau perangkat dasar seperti *Microsoft Office* dan *Google* [4].

Kebutuhan akan sistem informasi terintegrasi di pesantren semakin terasa mendesak. Lembaga pendidikan keagamaan ini memiliki aktivitas administrasi dan layanan yang kompleks. Pengelolaan data keuangan, absensi, hingga informasi santri membutuhkan sistem yang efisien. Salah satu tantangan yang muncul adalah bagaimana menampung dan mengolah umpan balik pengguna sistem. Umpan balik dalam jumlah besar seringkali sulit dipahami jika hanya dianalisis secara manual. Proses manual tersebut memerlukan tenaga, waktu, dan konsistensi yang tinggi. Untuk menjawab persoalan ini, teknologi berbasis *Natural Language Processing* (NLP) dapat dimanfaatkan. Salah satu implementasi yang relevan adalah *text summarization* untuk meringkas data teks dalam jumlah besar [5], [6].

Penelitian dalam bidang *text summarization* telah menunjukkan perkembangan pesat dalam beberapa tahun terakhir. Metode berbasis *Bidirectional Long Short-Term Memory* (Bi-LSTM) dinilai mampu menangkap konteks teks lebih baik dibanding metode searah [7]. Model ini memungkinkan informasi diproses dari dua arah, sehingga ringkasan yang dihasilkan dapat lebih representatif. Hasil penelitian sebelumnya membuktikan bahwa Bi-LSTM memberikan performa yang cukup baik untuk ringkasan teks [6]. Akan tetapi, penelitian khusus yang mengkaji ringkasan abstraktif berbahasa Indonesia masih terbatas. Terlebih lagi, penerapan Bi-LSTM dalam *domain* pesantren belum banyak dieksplorasi. Hal ini menimbulkan celah penelitian yang penting untuk diisi. Celah tersebut terutama terkait optimasi model melalui *hyperparameter tuning* agar kualitas ringkasan lebih baik [8].

Penelitian ini dirancang untuk mengisi kekosongan kajian tersebut. Fokus utama penelitian adalah penerapan Bi-LSTM untuk menghasilkan ringkasan otomatis dari umpan balik pengguna. Data penelitian diperoleh dari responden santri melalui kuesioner yang berisi pengalaman, pendapat, dan saran terkait sistem informasi pesantren. Pendekatan ini tidak hanya memberikan solusi praktis bagi pengelolaan umpan balik, tetapi juga menegaskan kontribusi ilmiah dalam bidang NLP. Penelitian ini bertujuan mengidentifikasi kombinasi parameter terbaik model Bi-LSTM melalui *hyperparameter tuning*. Evaluasi performa dilakukan menggunakan metrik ROUGE untuk mengukur kualitas ringkasan. Hasil penelitian diharapkan menunjukkan efektivitas Bi-LSTM dalam konteks teks berbahasa Indonesia. Selain itu, penelitian ini juga memberikan implikasi praktis terhadap efisiensi manajemen umpan balik di pesantren.

II. METODE

Penelitian ini menggunakan metode *Bidirectional Long Short Term Memory* (Bi-LSTM) dalam melakukan tugas peringkasan teks dengan pendekatan abstraktif. Hasil ringkasan yang dihasilkan dari pendekatan abstraktif berupa kalimat yang memiliki struktur yang berbeda dari kalimat aslinya sebagai hasil dari cara kerjanya yang menangkap ide pokok dari teks [9]. Penggunaan metode Bi-LSTM memungkinkan input yang berupa data sekuensial, diproses dari dua arah sekaligus (*forward* dan *backward pass*) [10].

Data penelitian diperoleh melalui kuesioner seputar *website Sibayar* yang disebarakan pada 1 Februari 2025 hingga 1 Maret 2025 kepada 38 responden santri Pondok Pesantren Sabilurrosyad Gasek Malang. Kuesioner menghasilkan total 114 teks umpan balik.

Pertanyaan kuesioner meliputi lima aspek utama:

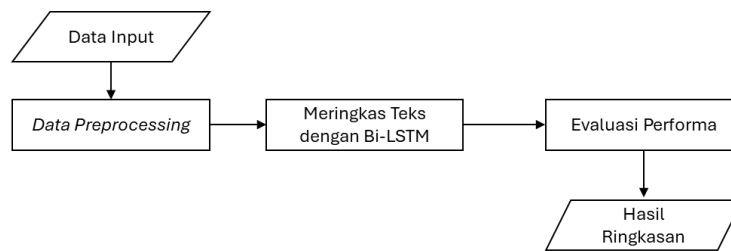
- 1) Pengalaman responden terhadap dalam mengakses form daftar ulang (santri baru) atau form daftar santri baru (untuk santri baru)
- 2) Pendapat responden terhadap kelengkapan dan relevansi data yang diminta dalam form
- 3) Kendala teknis saat mengisi form registrasi maupun form pendaftaran santri baru
- 4) Pengalaman Pengguna dalam mengakses SiBayar untuk melihat tagihan bulanan dan akses Wi-Fi
- 5) Saran responden terhadap fitur-fitur yang ada dalam SiBayar

Masing-masing dari pertanyaan digabungkan menjadi tiga kategori dengan pertanyaan (a, b, c) dalam kategori registrasi, pertanyaan dalam kategori bulanan, dan pertanyaan ke dalam kategori saran. Data yang telah dikelompokkan tersebut kemudian dilakukan validasi oleh ahli bahasa untuk dibuat ringkasan sebagai validasi dalam evaluasi hasil ringkasan oleh system. Selanjutnya data dapat dilakukan pemrosesan lebih lanjut oleh sistem.

Analisis deskriptif terhadap panjang teks menunjukkan bahwa jumlah kata pada umpan balik pengguna bervariasi antara 4 hingga 123 kata, dengan median 34,5 kata dan rata-rata sekitar 42,08 kata. Sementara itu, ringkasan manual yang dibuat oleh ahli bahasa memiliki panjang antara 4 hingga 52 kata, dengan median 19 kata dan rata-rata 21,02 kata.

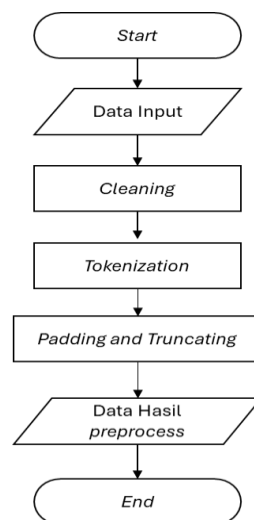
A. Desain Sistem

Berikut adalah desain sistem dari penelitian ini



Gambar 1. Alur desain sistem

Berdasarkan Gambar 1. alur sistem pada penelitian ini meliputi tahap *data preprocessing*, peringkasan dengan Bi-LSTM, evaluasi performa hingga menghasilkan ringkasan teks. Tahap pertama yang dilakukan adalah melakukan *preprocessing* terhadap data. Tahap pra-pemrosesan meliputi *cleaning* teks, tokenisasi, serta *padding* dan *truncating* agar setiap data memiliki panjang yang seragam.[11]. Untuk alur *preprocessing* adalah sebagai berikut.

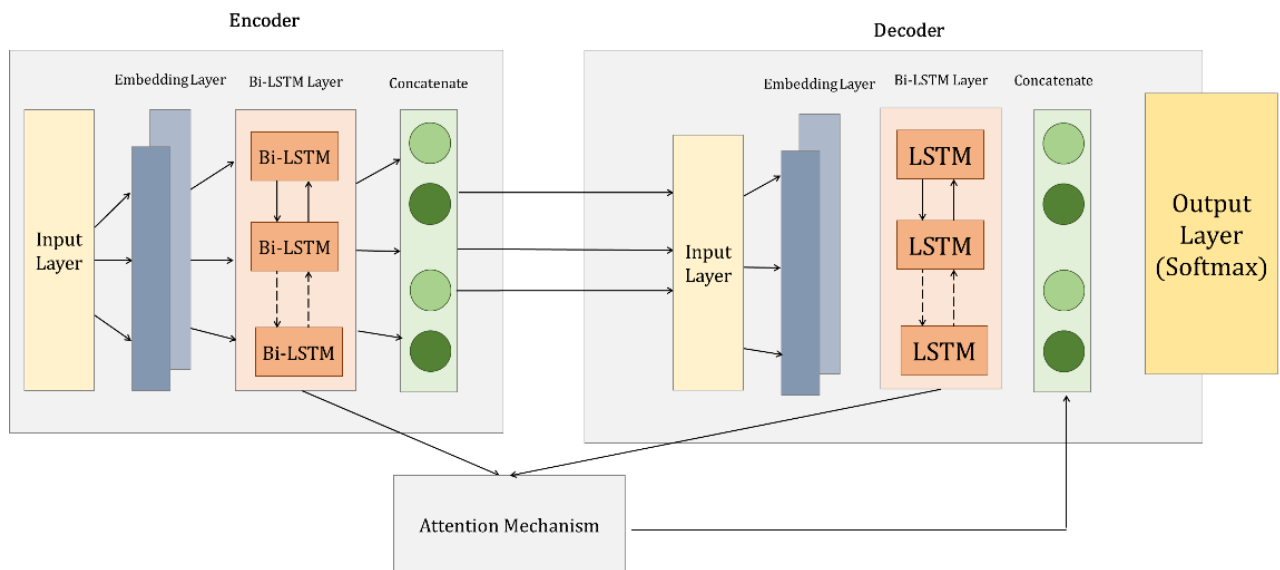


Gambar 2. Tahap *preprocessing*

Tahap *preprocessing* berdasarkan Gambar 2. melakukan pemrosesan awal terhadap data input dengan melakukan *cleaning* untuk menghilangkan karakter non-huruf pada data. Kemudian dilakukan *tokenization* untuk mengubah data menjadi bentuk *token* yang diperoleh dari setiap kata yang unik. Selanjutnya proses pemberian *padding* atau *truncating* berfungsi untuk memastikan data memiliki panjang yang sama [12]. Apabila data telah melalui proses tersebut maka, data telah dianggap siap untuk masuk ke dalam model peringkasan teks.

B. Arsitektur Model

Penelitian ini menggunakan arsitektur model Bi-LSTM *Sequence-to-Sequence* yang terdiri dari lapisan *encoder* dan *decoder* guna memproses input sekuensial berupa teks [13]. Dalam menerapkan arsitektur tersebut, penelitian ini menggunakan library dari *Keras* khususnya pada *class Bidirectional* dengan parameter LSTM untuk menerapkan metode Bi-LSTM. Berikut adalah arsitektur lengkap dari model.

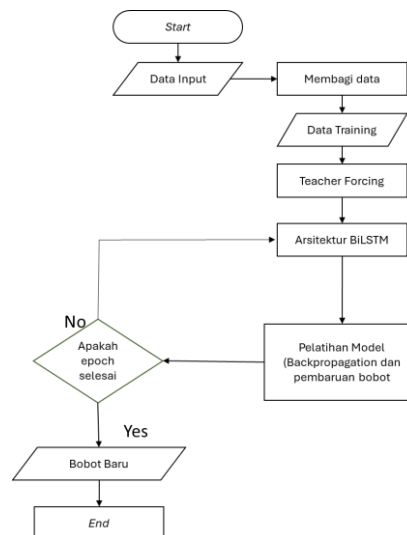


Gambar 3. Arsitektur Bi-LSTM

Dapat diamati pada Gambar 3. Pada tahap awal, input dari data mentah dimasukkan ke dalam *Embedding Layer* yang berfungsi untuk mengubah kata-kata atau token menjadi representasi vektor berdimensi tetap [14]. Penelitian ini menggunakan *Embedding* pada Keras dengan dimensi vektor sebesar 100. Selanjutnya, data tersebut diteruskan ke lapisan Bi-LSTM, yang memungkinkan model menangkap konteks informasi dari dua arah, baik dari masa lalu maupun masa depan dalam urutan data. Hasil dari lapisan Bi-LSTM kemudian dikombinasikan atau digabungkan dalam tahap *Concatenate*.

Setelah proses pertama selesai, output dari tahap sebelumnya bersama dengan input awal diproses kembali oleh lapisan *Embedding* dan LSTM, yang berfungsi untuk menangkap informasi temporal secara searah. *Output LSTM* ini kemudian kembali digabungkan dalam tahap *Concatenate* dan disalurkan ke *Output Layer* yang menggunakan fungsi aktivasi *Softmax* untuk menghasilkan prediksi akhir berupa distribusi probabilitas kelas. Arsitektur ini juga dilengkapi dengan mekanisme perhatian (*Attention Mechanism*) yang berfungsi menyoroti bagian-bagian penting dari data input, meningkatkan kemampuan model dalam menangkap hubungan relevan dalam urutan dan meningkatkan performa prediksi secara keseluruhan [15], [16].

C. Alur Pelatihan



Gambar 4. Alur Pelatihan

Alur pelatihan model Bi-LSTM pada Gambar 4. dimulai dengan memasukkan data hasil *preprocessing* yang kemudian dibagi menjadi 80% data latih dan 20% data uji [16], [17]. Pada data latih, dilakukan *teacher forcing* agar output dari langkah sebelumnya dapat digunakan sebagai input pada langkah berikutnya [17]. Data

latih kemudian diproses dalam arsitektur Bi-LSTM untuk menghasilkan *output* sekuensial. Selanjutnya, dilakukan perhitungan nilai *loss*, *backpropagation*, dan pembaruan bobot menggunakan fungsi pelatihan untuk memperbarui parameter model [18].

Selama pelatihan, performa model diukur pada setiap *epoch* berdasarkan nilai *loss* dan akurasi prediksi dari data latih dan data uji. Proses pelatihan diulang sesuai jumlah *epoch* yang ditentukan atau berhenti jika kondisi *early stopping* terpenuhi [19]. Agar hasil pelatihan *reproducible*, digunakan *random seed* tetap pada proses pelatihan. Mekanisme *early stopping* juga diterapkan untuk menghentikan pelatihan apabila validasi akurasi tidak mengalami peningkatan dalam lima *epoch* berturut-turut.

D. Skenario Pengujian

Penelitian ini melakukan pengujian hyperparameter untuk model Bi-LSTM dengan yang bertujuan untuk melakukan tuning atau optimasi algoritma dengan menemukan setiap kombinasi hyperparameter terbaik. [20]. Pengujian *hyperparameter* ini melibatkan pemilihan konfigurasi parameter optimizer dan jumlah lapisan, *batch size*, dan *dropout* menggunakan *Grid Search* yang dirincikan pada Tabel 1. Parameter-parameter tersebut dipilih karena memiliki pengaruh yang signifikan terhadap performa model [21].

TABEL 1
KONFIGURASI PARAMETER

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

III. HASIL DAN PEMBAHASAN

Berdasarkan rincian konfigurasi parameter pada Tabel 1, 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*. Kriteria *early stopping* pada penelitian ini ditetapkan dengan memantau nilai *val_loss* dan menghentikan pelatihan jika tidak terjadi perbaikan setelah 3 *epoch* berturut-turut (*patience=3*), serta mengembalikan bobot terbaik yang diperoleh selama proses pelatihan (*restore_best_weights = True*).

Pada saat proses pengujian parameter, 20% data latih diambil sebagai data validasi. Dalam *grid search*, kombinasi parameter terbaik ditentukan dari nilai *val_loss* yang rendah dan stabil serta *val_accuracy* yang tinggi, karena keduanya mencerminkan keseimbangan antara akurasi prediksi dan kemampuan generalisasi model. Dalam *grid search*, kombinasi parameter terbaik dipilih berdasarkan *val_loss* yakni ukuran kesalahan model pada data validasi yang rendah, serta *val_accuracy* yakni tingkat ketepatan prediksi *token* pada data validasi yang tinggi, karena keduanya mencerminkan keseimbangan antara akurasi dan kemampuan generalisasi model [19]. Berikut adalah hasil dari pencarian konfigurasi terbaik *dengan grid search*.

TABEL 2
10 KOMBINASI PARAMETER 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%

TABEL 3
 10 KOMBINASI PARAMETER 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%
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%

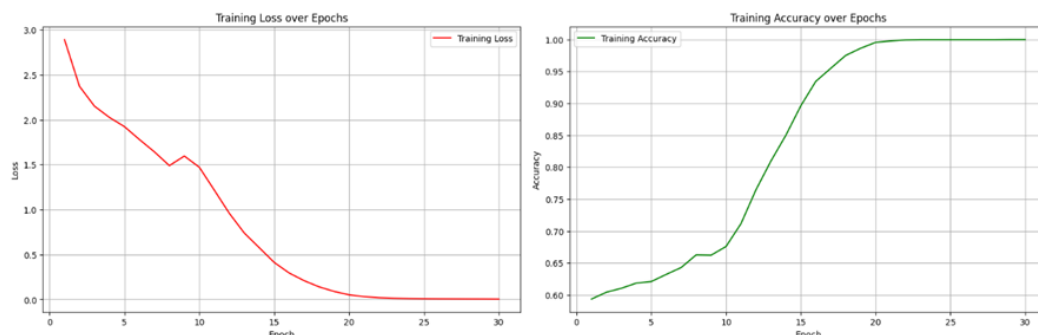
Hasil pengujian terlihat bahwa pemilihan kombinasi parameter memiliki pengaruh signifikan terhadap performa model Bi-LSTM. *Optimizer* Adam dan Nadam secara konsisten menunjukkan performa terbaik daripada RMSProp (Tabel 3) yang dibuktikan pada hasil 10 kombinasi terbaik di Tabel 2. Kombinasi terbaik 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 pengujian *hyperparameter*, diketahui parameter terbaik untuk tugas peringkasan teks pada penelitian ini menggunakan kombinasi konfigurasi parameter pada Tabel 4.5 berikut

TABEL 4
 KOMBINASI KONFIGURASI TERBAIK

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

Konfigurasi Bi-LSTM paling sederhana dengan satu lapisan, *batch size* kecil, dan optimizer Nadam terbukti memberikan hasil terbaik karena sesuai dengan karakter dataset yang terbatas, yakni 114 teks dengan panjang rata-rata 40 kata. Ukuran data yang kecil membuat arsitektur kompleks lebih rentan *overfitting*, sehingga model sederhana dengan regularisasi *dropout* lebih mampu menghasilkan ringkasan yang stabil dan representatif. Selanjutnya, model dilatih kembali berdasarkan 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*.



Gambar 5. Grafik Loss dan Accuracy

Hasil pelatihan menunjukkan training *loss* menurun tajam dari 3,0 hingga mendekati nol, sementara *training accuracy* meningkat dari 0,60 hingga hampir 1,00 pada *epoch* ke-30. Pola ini menandakan model sangat menyesuaikan diri dengan data latih, namun juga mengindikasikan potensi *overfitting*, terlebih karena dataset

yang digunakan relatif kecil, yaitu 114 teks dengan rata-rata panjang 40 kata. Ukuran data yang terbatas memungkinkan model menghafal pola spesifik teks, sehingga meskipun skor ROUGE tinggi pada data latih, performa pada data baru berpotensi menurun. 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 contoh hasil ringkasan sistem.

TABEL 5
 CONTOH HASIL RINGKASAN

No	Ringkasan Referensi	Ringkasan Sistem
1	fitur bisa ditambah dengan sistem pembayaran syariah yang otomatis mendeteksi rekening saat membayar tagihan bulanan jadi tidak perlu konfirmasi ke bendahara	sistem bisa ditambah dengan sistem pembayaran syariah yang otomatis mendeteksi rekening saat membayar tagihan bulanan jadi
2	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
3	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

Evaluasi hasil ringkasan dilakukan dengan membandingkan ringkasan yang dihasilkan oleh sistem dengan ringkasan referensi dari ahli. Metrik yang digunakan untuk evaluasi pada penelitian ini menggunakan ROUGE, yang menilai kesesuaian *n-gram* dan urutan kata [22]. Berikut adalah nilai skor ROUGE untuk model Bi-LSTM secara keseluruhan.

TABEL 6
 EVALUASI MODEL

No.	Statistik	ROUGE-1	ROUGE-2	ROUGE-L
1	Rata-rata	0.6221	0.5462	0.660
2	Min	0.0714	0	0.0606
3	Max	1	1	1

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

IV. SIMPULAN

Penelitian ini bertujuan untuk mengidentifikasi konfigurasi parameter terbaik Bi-LSTM agar menghasilkan model peringkasan teks yang optimal. Evaluasi menunjukkan bahwa Bi-LSTM mampu menghasilkan ringkasan dengan skor ROUGE-1 sebesar 0.6221, ROUGE-2 sebesar 0.5462, dan ROUGE-L sebesar 0.660, sehingga terbukti efektif dalam mempertahankan inti informasi dari teks umpan balik pengguna. Konfigurasi terbaik yang diperoleh cenderung sederhana, menyesuaikan dengan karakteristik dataset yang relatif kecil, sehingga menunjukkan pentingnya pemilihan parameter sesuai kondisi data. Kontribusi ilmiah penelitian ini terletak pada penerapan Bi-LSTM untuk peringkasan teks dalam konteks sistem informasi pesantren, yang masih jarang diteliti dalam bidang NLP berbahasa Indonesia. Dari sisi praktis, hasil ringkasan dapat mendukung efisiensi pengelolaan umpan balik pengguna pada website SiBayar, sehingga membantu pengambilan keputusan lebih cepat dan tepat. Ke depan, penelitian dapat diperluas dengan menambah variasi metode, memperbesar jumlah data, serta mengeksplorasi arsitektur dan parameter lain untuk meningkatkan performa model.

DAFTAR PUSTAKA

- [1] M. Marsum and Abd. W. Syahroni, "EFEKTIFITAS PENGGUNAAN TEKNOLOGI PADA PESANTREN MODERN DALAM MENGHADAPI REVOLUSI INDUSTRI 4.0," *Jurnal Kariman*, vol. 8, no. 02, pp. 233–242, Dec. 2020, doi: 10.52185/kariman.v8i02.155.
- [2] D. A. Nawangnugraeni, M. Z. Abdillah, and M. Al'Amin, "Implementasi Aplikasi Android untuk Sistem Penjadwalan Kegiatan Pondok Pesantren PDF Walindo," *Jurnal Pengabdian Masyarakat Progresif Humanis Brainstorming*, vol. 6, no. 1, pp. 1–8, Jan. 2023, doi: 10.30591/japhb.v6i1.3728.
- [3] A. Muchasan, Nur Syam, and Anis Humaidi, "Pemanfaatan Teknologi di Pesantren (Dampak dan Solusi dalam Konteks Pendidikan)," *INOVATIF: Jurnal Penelitian Pendidikan, Agama, dan Kebudayaan*, vol. 10, no. 1, pp. 16–33, Feb. 2024, doi: 10.55148/inovatif.v10i1.849.
- [4] M. Maimunah and J. Junadi, "IMPLEMENTASI SISTEM INFORMASI AKADEMIK DI PONDOK PESANTREN," *Al-Afkar : Manajemen pendidikan Islam*, vol. 11, no. 01, pp. 56–70, Apr. 2023, doi: 10.32520/al-afkar.v11i01.594.
- [5] Z. Munawar, "Meningkatkan Kinerja Individu melalui Kritik/Saran menggunakan Recommender System," *TEMATIK*, vol. 6, no. 1, pp. 20–38, Jun. 2019, doi: 10.38204/tematik.v6i1.185.
- [6] K. Kothari, A. Shah, S. Khara, and H. Prajapati, "A novel approach in user reviews analysis using text summarization and sentiment analysis: Survey," *arXiv preprint arXiv:2001.XXXX*, 2020. [Online]. Available: <https://api.semanticscholar.org/CorpusID:214738395>
- [7] Y. Yuliska and K. U. Syaliman, "Peringkasan Dokumen Teks Otomatis Berdasarkan Sebuah Kueri Menggunakan Bidirectional Long Short Term Memory Network," *INTECOMS: Journal of Information Technology and Computer Science*, vol. 5, no. 2, pp. 65–71, Dec. 2022, doi: 10.31539/intecom.v5i2.4729.
- [8] P. Liashchynskiy and P. Liashchynskiy, "Grid search, random search, genetic algorithm: a big comparison for NAS," *arXiv preprint arXiv:1912.06059*, 2019.
- [9] S. Varade, E. Sayyed, V. Nagtode, and S. Shinde, "Text Summarization using Extractive and Abstractive Methods," *ITM Web of Conferences*, vol. 40, p. 03023, Aug. 2021, doi: 10.1051/itmconf/20214003023.
- [10] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *arXiv preprint arXiv:1409.3215*, 2014.
- [11] N. Singh and B. Gaur, "Data Preprocessing: A Step-by-Step Guide for Clean and Usable Data," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 10, no. 2, pp. 1148–1153, Sep. 2019, doi: 10.61841/turcomat.v10i2.14384.
- [12] H. Ding, S. et al., "Fewer truncations improve language modeling," *arXiv preprint arXiv:2404.10830*, 2024. [Online]. Available: <https://api.semanticscholar.org/CorpusID:269187631>
- [13] Y. Wu and Y. Xing, "Efficient machine translation with a BiLSTM-attention approach," *arXiv preprint arXiv:2410.22335*, 2024.
- [14] Satvika, V. Thada, and J. Singh, "A primer on word embedding," in *Handbook of Research on Natural Language Processing and Smart Service Systems*, Singapore: Springer, 2021, pp. 525–541, doi: 10.1007/978-981-15-8530-2_42.
- [15] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.
- [16] V. R. Joseph, "Optimal ratio for data splitting," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 15, no. 4, pp. 531–538, Aug. 2022, doi: 10.1002/sam.11583.
- [17] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, "Sequence level training with recurrent neural networks," *arXiv preprint arXiv:1511.06732*, 2015.
- [18] J. Terven, D. M. Cordova-Esparza, A. Ramirez-Pedraza, E. A. Chavez-Urbiola, and J. A. Romero-Gonzalez, "Loss functions and metrics in deep learning," *arXiv preprint arXiv:2307.02694*, 2023.
- [19] B. M. Hussein and S. M. Shareef, "An Empirical Study on the Correlation between Early Stopping Patience and Epochs in Deep Learning," *ITM Web of Conferences*, vol. 64, p. 01003, Jul. 2024, doi: 10.1051/itmconf/20246401003.
- [20] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, Nov. 2020, doi: 10.1016/j.neucom.2020.07.061.
- [21] N. Reimers and I. Gurevych, "Optimal hyperparameters for deep lstm-networks for sequence labeling tasks," *arXiv preprint arXiv:1707.06799*, 2017.
- [22] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," in *Annual Meeting of the Association for Computational Linguistics*, 2004. [Online]. Available: <https://api.semanticscholar.org/CorpusID:964287>