

Sistem Penyaringan Spam Email Menggunakan Long Short Term Memory (LSTM)

Mohammad Irfan Wahyudi¹, Muhammad Fernanda Mulya Putra²

^{1,2}Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Nusantara PGRI Kediri

E-mail: ¹[*irfanea34@gmail.com](mailto:irfanea34@gmail.com), ²muhammadfernanda2711@gmail.com

Abstrak – Email merupakan media komunikasi yang banyak digunakan, namun meningkatnya jumlah spam email menimbulkan gangguan dan risiko keamanan informasi. Pendekatan penyaringan spam konvensional masih memiliki keterbatasan dalam mengenali pola spam yang terus berkembang. Penelitian ini bertujuan untuk mengembangkan sistem penyaringan spam email menggunakan algoritma Long Short-Term Memory (LSTM) yang mampu mengklasifikasikan email spam dan non-spam berdasarkan isi pesan. Data penelitian berupa dataset email berlabel spam dan non-spam yang diproses melalui tahapan preprocessing teks, tokenisasi, padding sekuens, serta pelatihan model LSTM dengan mekanisme early stopping. Evaluasi kinerja model dilakukan menggunakan confusion matrix dan metrik accuracy, precision, recall, serta F1-score. Hasil pengujian menunjukkan bahwa model LSTM mencapai nilai akurasi sebesar 0,99 dengan nilai precision, recall, dan F1-score yang sama-sama tinggi pada kedua kelas, yang menandakan kemampuan klasifikasi yang sangat baik dan stabil. Model yang telah dilatih kemudian diimplementasikan ke dalam sistem penyaringan spam email berbasis web menggunakan Streamlit dan protokol IMAP. Hasil penelitian ini menunjukkan bahwa LSTM efektif dan aplikatif untuk digunakan sebagai sistem penyaringan spam email otomatis.

Kata Kunci —Email, LSTM, Plugin, Spam

1. PENDAHULUAN

Perkembangan teknologi informasi telah menjadikan email sebagai media komunikasi digital yang dominan dan digunakan secara luas dalam lingkungan organisasi serta perusahaan untuk mendukung komunikasi internal dan eksternal secara efisien [1]. Peningkatan intensitas penggunaan email menyebabkan lonjakan volume pertukaran informasi yang signifikan, namun secara bersamaan juga memicu meningkatnya jumlah spam email yang masuk ke sistem pengguna [2]. Spam email merupakan pesan elektronik yang tidak diinginkan dan umumnya berisi iklan, promosi, atau konten berbahaya yang dikirim secara massal tanpa persetujuan penerima[3].

Keberadaan spam email tidak hanya menurunkan kenyamanan pengguna, tetapi juga menimbulkan ancaman serius terhadap keamanan informasi, seperti phishing, pencurian data, serta penyebaran malware melalui lampiran atau tautan berbahaya[4]. Selain itu, spam email dapat menghambat akses terhadap pesan penting karena tertimbun oleh pesan yang tidak relevan, sehingga berdampak pada menurunnya produktivitas pengguna dan efisiensi kerja organisasi [5].

Pendekatan penyaringan spam email yang umum digunakan masih didominasi oleh metode konvensional berbasis aturan, skor, serta daftar kata kunci seperti whitelist dan blacklist [6]. Metode tersebut memerlukan pengaturan ambang batas dan penyesuaian manual secara berkala agar tetap mampu mengenali spam dengan baik[7]. Namun, pendekatan konvensional cenderung kurang adaptif terhadap perubahan pola spam yang dinamis, variasi bahasa, serta teknik manipulasi teks yang digunakan oleh spammer[8].

Perkembangan machine learning dan deep learning mendorong pemanfaatan metode klasifikasi teks yang lebih adaptif dan berbasis pembelajaran data dalam mendeteksi spam email [9]. Salah satu algoritma deep learning yang banyak digunakan untuk pemrosesan teks sekuensial adalah Long Short-Term Memory (LSTM), yang dirancang untuk menangkap ketergantungan jangka panjang antar kata dalam suatu urutan teks [10]. Kemampuan LSTM dalam memahami konteks linguistik menjadikannya efektif dalam membedakan email spam dan non-spam berdasarkan isi pesan secara lebih akurat [4].

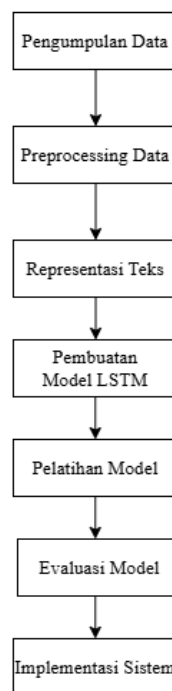
Sejumlah penelitian terdahulu menunjukkan bahwa algoritma LSTM mampu menghasilkan performa klasifikasi spam email yang tinggi berdasarkan metrik evaluasi seperti akurasi, precision, recall, dan F1-score [5]. Meskipun demikian, sebagian besar penelitian masih berfokus pada pengujian model secara eksperimental dan belum banyak yang mengimplementasikan model LSTM ke dalam sistem aplikasi penyaringan email yang bekerja secara otomatis dan real-time [7].

Berdasarkan permasalahan tersebut, penelitian ini mengembangkan sistem penyaringan spam email berbasis Long Short-Term Memory (LSTM) yang diintegrasikan ke dalam aplikasi berbasis web [1]. Sistem dirancang untuk melakukan deteksi dan klasifikasi email masuk secara otomatis melalui protokol IMAP sehingga mampu

mengidentifikasi *spam* secara langsung berdasarkan isi pesan [6]. Integrasi antara model LSTM dan sistem aplikasi diharapkan dapat meningkatkan akurasi penyaringan *spam* sekaligus menyediakan solusi yang praktis, adaptif, dan aplikatif dalam pengelolaan email modern [9].

2. METODE PENELITIAN

Tahapan penelitian dalam penelitian ini disusun secara sistematis sebagaimana ditunjukkan pada Gambar 1. Setiap tahapan saling berkaitan dan dirancang untuk menghasilkan model serta sistem penyaringan *spam* email menggunakan *Long Short-Term Memory* (LSTM) yang efektif.



Gambar 1. Diagram Alir Metode Penelitian

2.1 Pengumpulan Data

Tahap pengumpulan data dilakukan dengan memanfaatkan *dataset* email yang telah diberi label *spam* dan *non-spam*. Data terdiri atas subjek dan isi pesan email yang merepresentasikan konten komunikasi elektronik. *Dataset* ini digunakan sebagai data pembelajaran terawasi karena setiap data telah memiliki label kelas yang jelas, sehingga dapat dimanfaatkan secara langsung dalam proses pelatihan dan pengujian model klasifikasi.

2.2 Preprocessing Data

Data email yang telah dikumpulkan selanjutnya melalui tahap praproses untuk meningkatkan kualitas data dan mengurangi gangguan (*noise*) yang dapat mempengaruhi kinerja model. Proses praproses meliputi normalisasi huruf menjadi huruf kecil (*case folding*), penghapusan URL, angka, dan karakter khusus, serta pembersihan spasi berlebih. Selain itu, bagian subjek dan isi email digabungkan menjadi satu kesatuan teks agar konteks pesan dapat dipelajari secara utuh oleh model LSTM.

2.3 Representasi Teks

Pada tahap representasi teks dilakukan untuk mengubah email hasil praproses menjadi bentuk numerik agar dapat diproses oleh model LSTM. Pada tahap ini digunakan *Tokenizer* dari Keras untuk membentuk kamus kata (*vocabulary*) dari data latih, kemudian setiap kata pada teks dipetakan menjadi indeks bilangan bulat. Penelitian ini membatasi jumlah kata unik yang digunakan sebanyak 20000 kata teratas ($MAX_WORDS = 20000$) agar dimensi input tetap terkendali, serta menerapkan token khusus *out-of-vocabulary* untuk merepresentasikan kata yang tidak terdapat pada kamus saat proses inferensi, sehingga model tetap dapat memproses email dengan variasi kosakata yang lebih luas.

Selanjutnya, teks yang telah ditokenisasi diubah menjadi urutan indeks (*sequence*) menggunakan *texts_to_sequences*. Karena panjang email berbeda-beda, dilakukan *padding* untuk menyeragamkan panjang sekuens input menjadi 200 token ($MAX_LEN = 200$) menggunakan *pad_sequences* dengan *padding="post"*

dan `truncating="post"`, artinya nilai 0 ditambahkan di bagian akhir untuk sekuens yang lebih pendek, sedangkan sekuens yang lebih panjang dipotong dari bagian akhir. Hasil akhir tahap ini berupa metrik numerik berdimensi seragam yang kemudian menjadi masukan lapisan Embedding dan LSTM untuk mempelajari pola urutan kata dalam klasifikasi *spam* dan *non-spam*.

2.4 Pembuatan Model LSTM

Tahap pembangunan model dilakukan dengan merancang arsitektur *Long Short-Term Memory* (LSTM) sebagai metode utama dalam klasifikasi *spam* email. Model terdiri atas lapisan embedding untuk merepresentasikan kata dalam bentuk vektor berdimensi tetap, lapisan bidirectional LSTM untuk mempelajari hubungan sekuensial teks secara dua arah, serta lapisan dense terdiri dari hidden dense layer dengan fungsi aktivasi *ReLU* dan output layer dengan fungsi aktivasi sigmoid untuk klasifikasi biner. Untuk meningkatkan kemampuan generalisasi model dan mengurangi risiko *overfitting*, digunakan lapisan dropout.

2.5 Pelatihan Model

Model yang telah dibangun selanjutnya dilatih menggunakan data latih dengan parameter pelatihan tertentu, seperti jumlah epoch dan ukuran batch. Selama proses pelatihan, diterapkan mekanisme *early stopping* untuk menghentikan pembelajaran ketika performa model pada data validasi tidak menunjukkan peningkatan. Tahap ini bertujuan untuk memperoleh model dengan kinerja optimal dan stabil.

2.6 Evaluasi Model

Model *Long Short-Term Memory* (LSTM) yang telah dilatih selanjutnya dievaluasi menggunakan data uji untuk menilai kinerjanya dalam mengklasifikasikan email ke dalam dua kelas, yaitu *spam* dan *non-spam*. Evaluasi dilakukan untuk mengukur tingkat ketepatan prediksi model serta memastikan kemampuan generalisasi model terhadap data yang belum pernah dilatih sebelumnya.

Proses evaluasi dilakukan dengan menggunakan *confusion matrix*, yang merepresentasikan jumlah prediksi benar dan salah pada masing-masing kelas, meliputi *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Berdasarkan *confusion matrix* tersebut, kinerja model dianalisis menggunakan metrik evaluasi *accuracy*, *precision*, *recall*, dan *F1-score*, yang umum digunakan dalam permasalahan klasifikasi biner untuk menilai performa model secara komprehensif.

Secara matematis, perhitungan metrik evaluasi dirumuskan sebagai berikut.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \dots\dots\dots(1)$$

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots(2)$$

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots(3)$$

$$F1-Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \dots\dots\dots(4)$$

2.7 Implementasi Sistem

Tahap akhir penelitian adalah implementasi model ke dalam sistem deteksi *spam* email berbasis web. Model LSTM yang telah dilatih dan dievaluasi diintegrasikan ke dalam aplikasi menggunakan Streamlit dan dihubungkan dengan server email melalui protokol IMAP. Sistem ini dirancang untuk mengambil email masuk secara otomatis, melakukan klasifikasi *spam*, serta menampilkan hasil klasifikasi melalui antarmuka pengguna sehingga dapat digunakan secara langsung dalam lingkungan nyata.

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Data

Tahap pengumpulan data menghasilkan *dataset* email yang telah diberi label *spam* dan *non-spam*, yang terdiri dari subjek dan isi pesan email. *Dataset* ini digunakan sebagai data pembelajaran terawasi karena setiap email memiliki label kelas yang jelas, sehingga memungkinkan proses pelatihan dan evaluasi model dilakukan secara terarah. Komposisi data *spam* dan *non-spam* relatif seimbang, sehingga dapat meminimalkan bias kelas dalam proses pembelajaran model LSTM. Sumber *dataset* . <https://www.kaggle.com/datasets/willyard/spam-email-dataset?resource=download>.

3.2 Preprocessing Data

Hasil *preprocessing* menunjukkan bahwa data email menjadi lebih bersih dan seragam setelah melalui proses normalisasi huruf (*case folding*), penghapusan URL, angka, karakter khusus, serta pembersihan spasi berlebih. Penggabungan subjek dan isi email menjadi satu kesatuan teks memberikan konteks pesan yang lebih lengkap, sehingga informasi semantik dapat dipelajari secara optimal oleh model LSTM. Tahapan ini berperan penting dalam mengurangi noise dan meningkatkan kualitas representasi teks pada tahap selanjutnya.

3.3 Representasi Teks

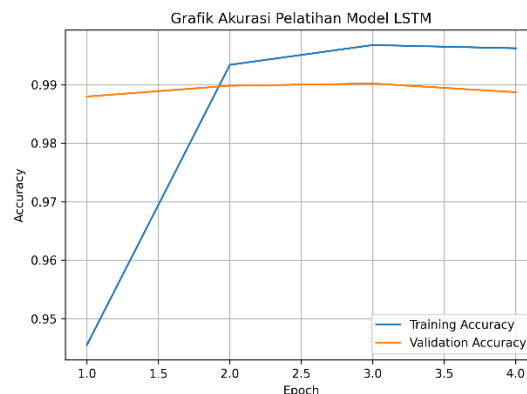
Pada tahap representasi teks, email hasil *preprocessing* berhasil dikonversi ke dalam bentuk numerik menggunakan *Tokenizer* dari Keras. Setiap email direpresentasikan sebagai urutan indeks kata dengan jumlah maksimum 200 token setelah proses *padding*. Hasil representasi ini menghasilkan metrik numerik berdimensi seragam yang menjadi input bagi lapisan embedding dan LSTM. Dengan pendekatan sekuensial ini, model mampu mempelajari urutan kata dan konteks linguistik dalam membedakan email *spam* dan *non-spam*.

3.4 Pembuatan Model LSTM

Model *Long Short-Term Memory* (LSTM) berhasil dibangun menggunakan arsitektur yang terdiri atas lapisan embedding, bidirectional LSTM, dropout, dan dense sebagai lapisan klasifikasi akhir. Lapisan embedding berfungsi untuk memetakan setiap kata ke dalam vektor berdimensi tetap, sedangkan lapisan bidirectional LSTM memungkinkan model mempelajari hubungan sekuensial teks dari dua arah, baik konteks sebelum maupun sesudah suatu kata. Penambahan lapisan dropout bertujuan untuk mengurangi risiko *overfitting*, sementara lapisan dense dengan fungsi aktivasi sigmoid digunakan untuk menghasilkan prediksi kelas *spam* dan *non-spam*. Arsitektur ini sesuai dengan karakteristik data teks yang bersifat sekuensial dan dinilai efektif untuk permasalahan klasifikasi *spam* email.

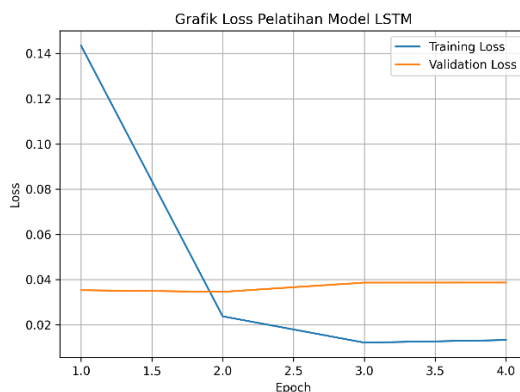
3.5 Pelatihan Model

Model *Long Short-Term Memory* (LSTM) berhasil dibangun menggunakan arsitektur embedding, bidirectional LSTM, dropout, dan dense sebagai lapisan klasifikasi akhir. Selama proses pelatihan, model menunjukkan peningkatan akurasi dan penurunan nilai loss secara bertahap hingga mencapai kondisi konvergen. Penerapan mekanisme *early stopping* membantu menghentikan pelatihan pada saat performa validasi tidak lagi meningkat, sehingga model yang dihasilkan memiliki kinerja optimal dan mampu menghindari *overfitting*.



Gambar 2. Grafik Akurasi Pelatihan Model

Selama proses pelatihan dan validasi pada setiap epoch. Pada epoch pertama, akurasi pelatihan masih berada di kisaran 0,94, kemudian meningkat secara signifikan pada epoch kedua hingga mencapai lebih dari 0,99. Peningkatan ini menunjukkan bahwa model LSTM mampu mempelajari pola sekuensial pada data email dengan cepat. Setelah epoch ketiga, akurasi pelatihan cenderung stabil dengan nilai mendekati 0,995, sedangkan akurasi validasi berada pada kisaran 0,98–0,99 dan relatif stabil. Perbedaan yang kecil antara kurva akurasi pelatihan dan validasi mengindikasikan bahwa model memiliki kemampuan generalisasi yang baik dan tidak mengalami *overfitting* secara signifikan.



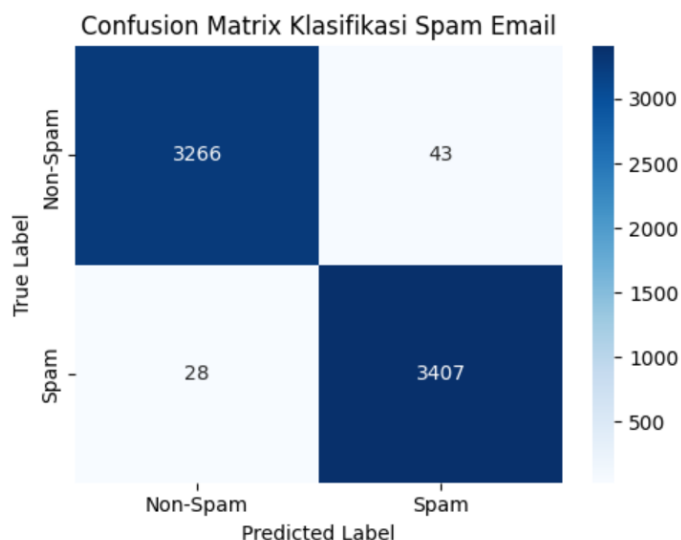
Gambar 3. Grafik Loss Pelatihan Model LSTM

Sementara itu, Gambar 3 memperlihatkan penurunan nilai loss pelatihan yang sangat tajam dari sekitar 0,14 pada epoch pertama menjadi di bawah 0,02 pada epoch ketiga. Penurunan ini menunjukkan bahwa kesalahan prediksi model semakin kecil seiring bertambahnya epoch pelatihan. Nilai loss validasi cenderung stabil di kisaran 0,03–0,04 dengan sedikit fluktuasi pada epoch selanjutnya. Kondisi ini menandakan bahwa model mampu mempertahankan performa yang konsisten pada data validasi dan tidak mengalami degradasi kinerja meskipun pelatihan berlanjut.

Secara keseluruhan, kedua grafik tersebut menunjukkan bahwa proses pelatihan model LSTM berjalan secara konvergen dan stabil. Penerapan mekanisme *early stopping* terbukti efektif dalam menghentikan pelatihan pada kondisi optimal, sehingga model yang dihasilkan memiliki kinerja yang baik, stabil, dan siap digunakan pada tahap evaluasi serta implementasi sistem penyaringan *spam* email.

3.6 Evaluasi Model

Evaluasi model dilakukan menggunakan data uji untuk menilai kemampuan model dalam mengklasifikasikan email *spam* dan *non-spam*. Proses evaluasi menghasilkan *confusion matrix* yang menggambarkan jumlah prediksi benar dan salah pada masing-masing kelas.



Gambar 4. *Confusion Matrix*

Berdasarkan Gambar 4 tersebut, model berhasil mengklasifikasikan 3266 email *non-spam* dan 3407 email *spam* secara benar, dengan jumlah kesalahan klasifikasi yang relatif kecil, yaitu 43 false positive dan 28 false negative. Jumlah kesalahan yang rendah pada kedua kelas menunjukkan bahwa model memiliki kemampuan yang baik dalam membedakan email *spam* dan *non-spam* secara seimbang.

Selanjutnya, kinerja model dianalisis menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score*.

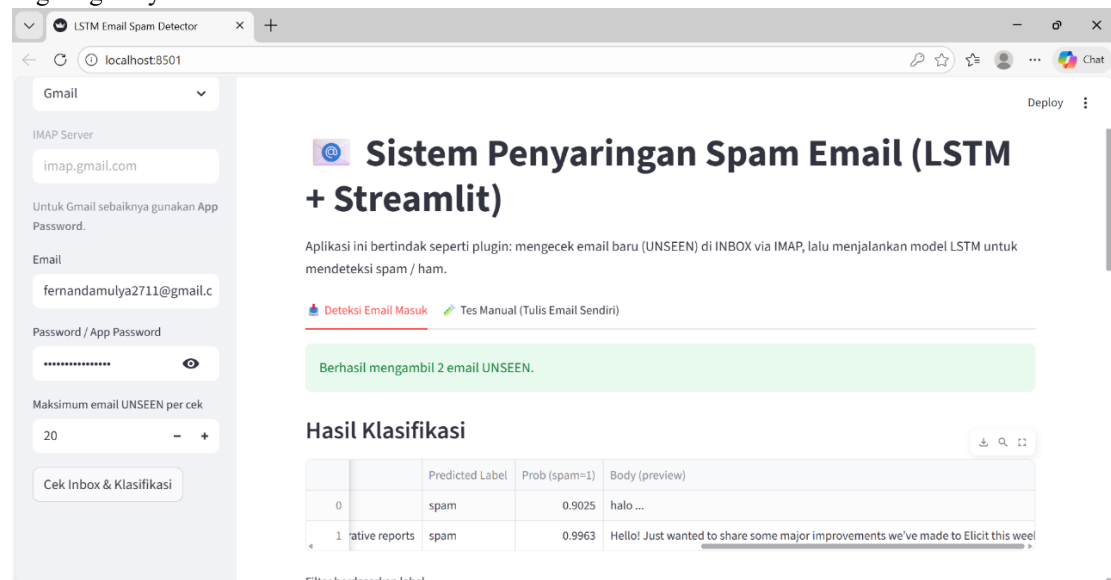
Tabel 1. Kinerja Model LSTM

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
<i>Ham</i>	0.99	0.99	0.99	3309
<i>Spam</i>	0.99	0.99	0.99	3435

Model memperoleh nilai *accuracy* sebesar 0,99, yang menunjukkan tingkat ketepatan klasifikasi yang sangat tinggi. Nilai *precision* dan *recall* yang sama-sama tinggi pada kedua kelas menunjukkan bahwa model tidak hanya mampu mendeteksi email *spam* dengan baik, tetapi juga konsisten dalam mengklasifikasikan email *non-spam*. Nilai *F1-score* yang tinggi menunjukkan keseimbangan yang baik antara *precision* dan *recall*, sehingga model memiliki performa yang stabil dan andal.

3.7 Implementasi Sistem

Model LSTM yang telah melalui proses pelatihan dan evaluasi selanjutnya diintegrasikan ke dalam sistem penyaringan *spam* email berbasis web menggunakan Streamlit dan protokol IMAP. Sistem mampu mengambil email masuk secara otomatis, melakukan klasifikasi *spam* dan *non-spam* secara *real-time*, serta menampilkan hasil klasifikasi kepada pengguna. Implementasi ini menunjukkan bahwa model yang dikembangkan tidak hanya valid secara eksperimen, tetapi juga aplikatif dan siap digunakan dalam lingkungan nyata.



Gambar 5. Tampilan UI Sistem

4. SIMPULAN

Penelitian ini berhasil mengembangkan sistem penyaringan *spam* email berbasis Long Short-Term Memory (LSTM) yang mampu mengklasifikasikan email *spam* dan *non-spam* secara efektif melalui tahapan pengumpulan data, *preprocessing* teks, representasi sekuensial, pembuatan dan pelatihan model, hingga evaluasi dan implementasi sistem. Hasil pengujian menunjukkan bahwa model LSTM memiliki performa yang sangat tinggi dengan nilai akurasi sebesar 0,99 serta nilai *precision*, *recall*, dan *F1-score* masing-masing mencapai 0,99 pada kedua kelas, yang mengindikasikan kemampuan generalisasi dan ketepatan klasifikasi yang sangat baik. Analisis *confusion matrix* menunjukkan jumlah kesalahan klasifikasi yang relatif kecil dan seimbang, sehingga model mampu mendeteksi *spam* secara akurat tanpa mengorbankan klasifikasi email *non-spam*. Proses pelatihan yang stabil, ditunjukkan oleh grafik akurasi dan loss serta penerapan *early stopping*, membuktikan bahwa model tidak mengalami *overfitting* dan mencapai kondisi pembelajaran yang optimal. Integrasi model ke dalam sistem berbasis web menggunakan Streamlit dan protokol IMAP menunjukkan bahwa solusi yang dikembangkan tidak hanya valid secara eksperimen, tetapi juga aplikatif dan siap digunakan dalam lingkungan nyata. Meskipun demikian, keterbatasan masih terdapat pada variasi dataset dan konteks bahasa email yang digunakan, sehingga diperlukan pengembangan lebih lanjut untuk meningkatkan adaptivitas model terhadap pola *spam* yang semakin beragam.

5. SARAN

Penelitian selanjutnya disarankan untuk memperluas dan memperkaya dataset dengan melibatkan variasi bahasa, konteks, dan pola *spam* yang lebih beragam agar kemampuan generalisasi model dapat semakin ditingkatkan. Selain itu, eksplorasi arsitektur lanjutan seperti LSTM dengan attention mechanism atau pendekatan berbasis transformer dapat dipertimbangkan untuk meningkatkan pemahaman konteks semantik teks. Optimalisasi parameter model serta penggunaan teknik word embedding yang lebih representatif juga berpotensi meningkatkan performa klasifikasi. Dari sisi sistem, pengembangan lebih lanjut dapat diarahkan pada integrasi dengan platform email yang lebih luas serta penerapan mekanisme retraining model secara berkala agar sistem tetap adaptif terhadap dinamika pola *spam* di masa mendatang.

DAFTAR PUSTAKA

- [1] T. A. Almeida, J. María, G. Hidalgo, and T. P. Silva, “Towards SMS Spam Filtering : Results under a New Dataset,” vol. 2, no. 1, pp. 1–18.
- [2] I. AbdulNabi and Q. Yaseen, “Spam email detection using deep learning techniques,” *Procedia Comput. Sci.*, vol. 184, pp. 853–858, 2021.
- [3] S. Wali, A. Alsudani, H. Ali, M. Nasrawi, M. Hasan, and A. Ghazikhani, “Enhancing Spam Detection : A Crow-Optimized FFNN with LSTM for Email Security,” pp. 28–39, 2024.
- [4] H. A. M. Bert, “ScienceDirect Spam Spam Email Email Detection Detection Using Using Deep Deep Learning Learning Techniques Techniques,” *Procedia Comput. Sci.*, vol. 184, no. 2019, pp. 853–858, 2021, doi: 10.1016/j.procs.2021.03.107.
- [5] M. Al Kautsar, G. G. Setiaji, and A. Rifa, “BULLETIN OF COMPUTER SCIENCE RESEARCH Analisis Komparasi Kinerja LSTM dan CNN dalam Deteksi Spam Email Berbasis Deep learning,” vol. 5, no. 4, pp. 584–593, 2025, doi: 10.47065/bulletincsr.v5i4.572.
- [6] D. Rizkiono, “Komparasi Deep Learning Dan Traditional Machine Learning Untuk Email Spam Filtering,” vol. 12, pp. 636–643, 2023.
- [7] S. A. Aklani, H. Haeruddin, and N. Putri, “Implementasi Mail Gateway Security Dalam Meningkatkan Keamanan Email,” *J. Inf. Syst. Manag.*, vol. 5, no. 2, pp. 150–155, 2024.
- [8] A. Mustofa and S. Pradana, “Perbandingan Pengujian Deteksi Phising menggunakan Metode SVM dengan Kernel RBF dan Linear Comparison of Phishing Detection Tests using the SVM Method with RBF and Linear Kernels,” vol. 12, no. September, pp. 754–759, 2023.
- [9] N. Adila, S. Khasanah, and T. Sutabri, “STRATEGI PERANCANGAN SISTEM AMAVIS DAN,” vol. 5, no. 2, pp. 154–166, 2023.
- [10] A. Harbani and A. Sidiyantoro, “Implementasi Simple Mail Transfer Protocol Relay Pada Mail Gateway Untuk Menentukan Konten Email Spam,” vol. 12, no. 1, pp. 57–66, 2022, doi: 10.36350/jbs.v12i1.130.