ISSN: 2598-8565 (media cetak)
ISSN: 2620-4339 (media online)

MODEL BIDIRECTIONAL LSTM UNTUK PEMROSESAN SEKUENSIAL DATA TEKS SPAM

Rimbun Siringoringo✉, Jamaluddin, Resianta Perangin-angin, Eva Julia G. Harianja, Gortap Lumbantoruan, Eviyanti Novita Purba
Universitas Methodist Indonesia, Medan, Indonesia
Email: rimbun.ringo@gmail.com

DOI: https://doi.org/10.46880/jmika.Vol7No2.pp265-271

ABSTRACT

This study examines the LSTM-based model for processing spam in text data. Spam poses several dangers and risks, both for individuals and organizations. Spam can be a nuisance that hampers both individual and organizational productivity. Much spam contains fraudulent or phishing attempts to obtain sensitive information. Spam detection using deep learning involves the utilization of algorithms and deep neural network models to accurately classify messages as either spam or not spam. Typically, spam detection systems use a combination of these methods to improve the accuracy of identifying spam messages. This study applies the Bi-LSTM deep learning model to sequentially process text (sequencing). The performance of the model is determined based on the loss and accuracy. The data used are the Spam SMS and Spam Email datasets. The test results show that the Bi-LSTM model demonstrates better performance on all tested datasets. Bi-LSTM is able to capture textual patterns from both the context and the text itself, as it can combine information from both directions. The test results prove that the Bi-LSTM model is more effective in text comprehension.

Keyword: Spam Data, Recurrent Neural Networks, Long Short-Term Memory, Bidirectional Long Short-Term Memory.

ABSTRAK

Penelitian ini menguji model berbasis LSTM dalam melakukan perosesan teks data spam. Spam memiliki beberapa bahaya dan risiko, baik bagi individu maupun organisasi. Spam dapat menjadi gangguan yang mengganggu produktivitas individu dan organisasi. Banyak spam mengandung upaya penipuan atau phishing untuk mendapatkan informasi yang sensitive. Deteksi spam menggunakan deep learning melibatkan penggunaan algoritma dan model neural networks yang dalam untuk mengklasifikasikan pesan sebagai spam atau bukan spam. Biasanya, sistem deteksi spam menggunakan kombinasi dari beberapa metode ini untuk meningkatkan akurasi dalam mengenali pesan spam. Penelitian ini menerapkan model deep learning Bi-LSTM dalam melakukan pemrosesan teks berurut (sequencing). Kinerja model ditentukan berdasarkan nilai loss dan akurasi. Data yang digunakan adalah dataset SMS Spam dan Email spam. Hasil pengujian menunjukkan bahwa model Bi-LSTM dapat menunjukkan kinerja yang lebih baik pada semua dataset yang diuji. Bi-LSTM mampu menangkap pola teks dari sisi konteks dan teks, karena informasi dari kedua arah dapat digabungkan. Hasil pengujian membuktikan bahwwa model Bi-LSTM lebih efektif dalam pemahaman teks.

Kata Kunci: Data Spam, Recurrent Neural Networks, Long Short-Term Memory, Bidirectional Long Short-Term Memory.

PENDAHULUAN

Deteksi spam adalah proses mengidentifikasi dan memisahkan pesan yang tidak diinginkan atau mengganggu, yang sering kali berupa pesan elektronik seperti email, pesan teks, atau komentar online. Spam dapat berupa iklan yang tidak diinginkan, tautan berbahaya, pesan penipuan, atau konten yang tidak relevan. Untuk mendeteksi spam, berbagai metode dan algoritma dapat digunakan, termasuk yang berbasis aturan dan yang berbasis pembelajaran mesin.

Spam memiliki beberapa bahaya dan risiko, baik bagi individu maupun organisasi. Spam dapat menjadi gangguan yang mengganggu produktivitas individu dan organisasi. Banyak spam mengandung upaya penipuan atau phishing untuk mendapatkan informasi yang sensitif (Sulthana et al., 2023). Spammer dapat mengirim e-mail atau pesan yang meniru lembaga keuangan, situs web populer, atau layanan online terkenal untuk mencuri informasi pribadi seperti kata sandi, nomor kartu kredit, atau

informasi keuangan lainnya. Beberapa spam mengandung lampiran atau tautan yang menyebabkan unduhan malware atau virus ke perangkat yang menerima spam tersebut (Barushka & Hajek, 2020). Malware ini dapat merusak sistem, mencuri data sensitif, atau mengambil kendali atas perangkat yang terinfeksi. Spam sering kali mengandung konten yang tidak pantas, termasuk materi pornografi, kekerasan, atau promosi ilegal. Menerima spam semacam ini dapat mengekspos individu, terutama anak-anak, pada konten yang tidak pantas dan berpotensi merugikan. Spam yang dikirim dalam jumlah besar dapat membebani infrastruktur jaringan. Penting untuk melindungi diri dari spam dengan menggunakan filter spam yang andal, tidak memberikan informasi pribadi kepada sumber yang tidak tepercaya, dan tidak mengklik tautan atau membuka lampiran yang mencurigakan. Organisasi juga harus menerapkan langkah-langkah keamanan seperti firewall dan perangkat lunak antivirus yang terbaru untuk melindungi sistem mereka dari spam dan ancaman lainnya.

Sejauh ini terdapat berbagai pendekatan yang dilakukan pada pendeteksian spam. Diantaranya adalah Pertama, filter berbasis aturan (Xia, 2020), (Jain & Gupta, 2018). Metode ini melibatkan penggunaan aturan atau pola tertentu yang ditentukan sebelumnya untuk mengidentifikasi spam. Aturan ini mungkin melibatkan kata-kata kunci, frasa, atau pola tertentu yang sering digunakan dalam pesan spam. Kedua, Analisis Bayesian (Peng et al., 2018). Metode ini menggunakan teori statistik Bayesian untuk mengklasifikasikan pesan sebagai spam atau bukan spam. Model Bayesian didasarkan pada probabilitas dan memperbarui perkiraan mereka berdasarkan informasi baru yang diberikan. Ketiga, Pembelajaran mesin (Navaney et al., 2018). Metode ini melibatkan penggunaan algoritma pembelajaran mesin untuk mempelajari pola dari data pelatihan yang ada dan mengklasifikasikan pesan baru sebagai spam atau bukan spam. Algoritma pembelajaran mesin yang umum digunakan termasuk *Naive Bayes, Support Vector Machines* (SVM), dan *Decision Tree*. Ke empat, Analisis heuristik (Pirozmand et al., 2023) Metode ini melibatkan penggunaan aturan berbasis heuristik yang didasarkan pada karakteristik umum spam, seperti penggunaan huruf kapital berlebihan, penekanan pada tautan, atau penggunaan frasa yang menipu. Kelima, deeplearning (Kaddoura et al., 2020; Roy et al., 2020). Deteksi spam menggunakan *deep learning* melibatkan penggunaan algoritma dan model neural networks yang dalam untuk mengklasifikasikan pesan sebagai spam atau bukan spam. Biasanya, sistem deteksi spam menggunakan kombinasi dari beberapa metode ini untuk meningkatkan akurasi dalam mengenali pesan spam. Perkembangan teknologi juga berperan penting dalam pengembangan metode baru untuk mendeteksi dan memfilter spam yang semakin canggih. Deteksi spam menggunakan *deep learning* melibatkan penggunaan algoritma dan model *neural networks* yang dalam untuk mengklasifikasikan pesan sebagai spam atau bukan spam. *Deep learning* memungkinkan sistem untuk secara otomatis mengekstraksi fitur dan pola kompleks dari data tanpa perlu menentukan aturan atau pola secara eksplisit.

Deteksi spam menggunakan LSTM (Long Short-Term Memory) adalah salah satu pendekatan populer dalam penggunaan *deep learning* untuk masalah klasifikasi teks seperti deteksi spam. LSTM adalah jenis arsitektur jaringan saraf rekurensi yang dapat memahami konteks jangka panjang dalam urutan data, seperti teks. Penggunaan LSTM dalam deteksi spam memungkinkan model untuk mempelajari pola-pola yang kompleks dan hubungan kontekstual dalam teks. Namun, penting untuk memperhatikan jumlah data latih yang cukup, serta melakukan pemrosesan data dan penyetelan model yang tepat agar model dapat memberikan hasil yang baik.

LITERATURE REVIEW

Terdapat beragam penelitian yang menerapkan LSTM pada deteksi dan penanganan spam. Penelitian yang dilakukan oleh (Raj et al., 2018), menyatakan bahwa kinerja *machine learning* konvensional seperti Naive Bayes (NB), Random Forest (RF), dan Support Vector Machine (SVM) sesungguhnya tidak memuaskan pada pengklasifikasian spam. Oleh karena itu, penelitian tersebut mengeksplorasi keunggulan *deep recurrent neural network* (DRNN) dengan model LSTM dan teknik *embedding* Word2Vec untuk mendeteksi spam pada teks pesan singkat (SMS). Hasil klasifikasi dengan model tersebut dapat menghasilkan performa yang sangat baik dari sisi akurasi.

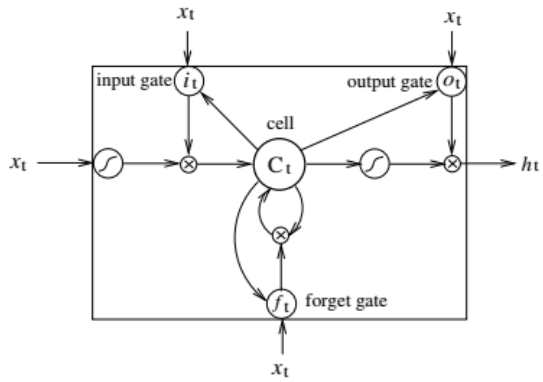
Penelitian yang dilakukan oleh (Salunkhe, 2021) menyatakan bahwa ulasan yang diberikan oleh netizen pada online commerce seringkali mengandung ulasan yang curang atau *fraudulent*. Hal ini dapat berakibat buruk pada reputasi produk tertentu. Oleh karenanya filter spam dapat diperluas pada ulasan komentar di aplikasi e-commerce. Penelitian ini menerapkan model *Attention-based* Bi-LSTM. Penelitian yang dilakukan oleh (Pushpalatha & Vijaya, 2022) menghasilkan model berbasis deep learnng *web socio-spider feature selection* (WS2 FS) dalam penanganan spam khususnya *malicious website.*

Penelitian yang dilakukan (Mustagfirin et al., 2022) terhadap penanganan SMS spam berbahasa Indonesia dengan menerapkan model LSTM. Penelitian yang dilakukan oleh Shaik et al., 2023, menggaungkan Bi-LSTM dan algoritma pengklasifikasi konvesional untuk filterisasi Email Spam. Pada bagian ini, dijabarkan model model yang diuji pada penelitian ini adalah LSTM dan BI-LSTM.

LSTM Networks

LSTM (Long Short-Term Memory) adalah jenis arsitektur jaringan saraf yang dapat digunakan untuk memfilter spam. LSTM merupakan salah satu jenis RNN (Recurrent Neural Network) yang dirancang khusus untuk mengatasi masalah mengenai memori jangka panjang dan jangka pendek dalam rangkaian data yang berkesinambungan. Pada penerapannya dalam filter spam, LSTM dapat digunakan untuk mempelajari pola dan konteks dari email atau pesan yang masuk. Ini memungkinkan LSTM untuk mengenali ciri-ciri yang umumnya terkait dengan spam, seperti kata-kata yang sering muncul dalam spam, penggunaan bahasa yang tidak lazim, tautan mencurigakan, atau frase tertentu yang sering digunakan dalam pesan spam. LSTM memproses teks dalam urutan kata per kata, dan setiap kata direpresentasikan sebagai vektor dalam ruang fitur. Selama proses pelatihan, LSTM belajar untuk mengidentifikasi pola-pola yang terkait dengan spam dari sejumlah besar data latihan yang telah dikategorikan sebagai spam atau bukan spam. Penggunaan LSTM dalam filter spam telah terbukti efektif dalam mengurangi jumlah email spam yang mencapai kotak masuk pengguna. Dengan memanfaatkan kemampuan LSTM dalam memahami dan mengenali pola-pola yang terkait dengan spam, filter spam yang menggunakan LSTM dapat meningkatkan pengalaman pengguna dengan mengurangi gangguan dari pesan-pesan yang tidak diinginkan. Gambar 1 menampilkan struktur dasar LSTM Networks

Recurrent neural networks (RNN) merupakan sebuah metode yang beroperasi untuk data sequence. Metode ini mengambil input dari vektor sequence (x1, x2, ..., xn) dan menjadi sequence yang lain (h1, h2, ..., hn). RNN tidak dapat diterapkan dalam *long-term dependency* yang menimbulkan masalah *vanishing gradient*. Oleh sebab itu *long short term memory* (LSTM) dikembangkan untuk mengatasi permasalahan *vanishing gradient*. LSTM mengganti hidden unit pada arsitektur RNN dengan unit yang bisa disebut memory block yang terdiri dari empat komponen yaitu : input gate, output gate, forget gate, dan memory cell. Rumus dari ke-empat komponen tersebut adalah sebagai berikut :



Gambar 1. Long Short-Term Memory Cell

Pada persamaan (1), input gate (it) digunakan untuk mengubah nilai pada cell state (c_t) dengan W_{xi} dan W_{hi} adalah bobot matriks yang dikalikan dengan vektor X_t dan h_{t-1}

$$i_t = \sigma\,(W_{xi}\,X_t + W_{hi}\,h_{t-1} + W_{ci}\,c_{t-1} + b_i) \quad (1)$$

Pada persamaan (2), forget gate (f_t) digunakan untuk menghapus informasi dari cell state (c_t). Informasi dari hidden state sebelum (h_{t-1}) dan inputan (X_t) dihitung dengan fungsi sigmoid (σ) dengan nilai keluaran antara 0 dan 1. Jika nilai keluaran mendekati 0 maka informasi akan di hapus, dan jika nilai keluaran mendekati 1 maka informasi akan disimpan

$$f_t = \sigma\,(W_{xf}\,X_t + W_{hf}\,h_{t-1} + W_{cf}\,) \quad (2)$$

Pada persamaan (3), cell state sebelumnya(c_{t-1}) dikali ($\odot$) dengan forget gate (f_t), terdapat kemungkinan nilai dari hasil perkalian tersebut akan menurun jika dikalikan dengan nilai (f_t) yang mendekati 0. Lalu nilai dari i_t dikali ($\odot$) dengan fungsi cell state saat ini untuk mendapatkan nilai cell state saat ini dengan nilai antara 1 sampai -1.

$$c_t = f_t\,c_{t-1} + i_t \tanh(W_{xc}\,X_t + W_{hc}\,h_{t-1} + b_c) \quad (3)$$

Pada persamaan (4) dan (5), sama halnya dengan input gate (i_t), output gate (o_t) digunakan untuk menentukan nilai dari hidden state baru (h_t).

$$o_t = \sigma\,(W_{xo}\,X_t + W_{ho}\,h_{t-1} + W_{co}\,c_t + b_o) \quad (4)$$

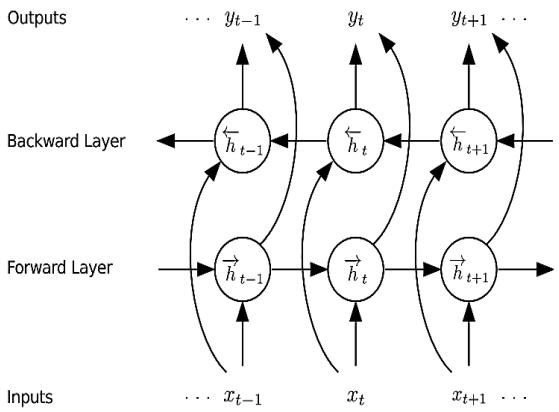
$$h_t = o_t \tanh(c_t) \quad (5)$$

Nilai dari output gate (o_t) dikalikan dengan hasil dari fungsi tanh untuk cell state (c_t). Yang mana hidden state baru dan cell state baru akan digunakan untuk perhitungan pada langkah selanjutnya (t).

Bidirectional LSTM Networks

Perbedaan utama dari LSTM biasa dengan Bidirectional LSTM adalah bahwa Bidirectional LSTM menggunakan dua lapisan LSTM yang saling terhubung. Lapisan pertama memproses urutan data dari awal hingga akhir, seperti pada LSTM biasa. Namun, lapisan kedua memproses urutan data dari akhir ke awal. Dengan demikian, jaringan dapat "melihat" informasi kontekstual dari kedua arah, baik sebelum dan setelah token yang sedang diproses.

Kelebihan dari Bidirectional LSTM adalah kemampuannya untuk menangkap konteks lebih kaya dari teks, karena informasi dari kedua arah dapat digabungkan. Hal ini membuatnya lebih efektif dalam pemahaman teks dan tugas-tugas seperti analisis sentimen di mana konteks sebelum dan sesudah sebuah kata atau frasa dapat mempengaruhi maknanya.



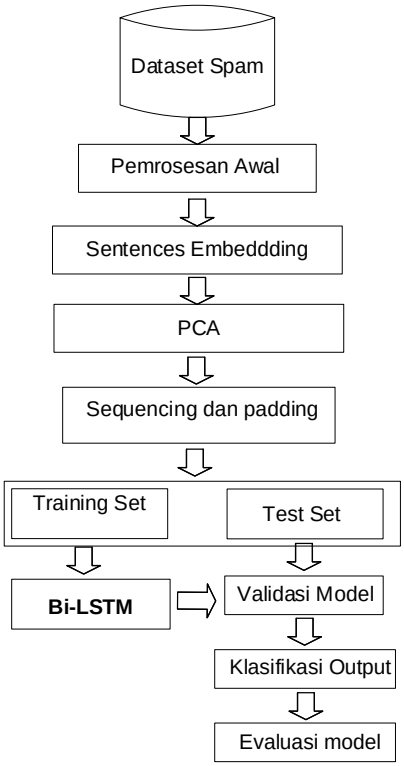
Gambar 2. Struktur Bi-LSTM

Bidirectional LSTM memanfaatkan konteks sebelumnya dan konteks setelahnya dengan memproses data dari dua arah dengan hidden layer terpisah (Graves et al., 2013). Forward layer untuk merepresentasikan konteks sebelumnya, dan backward layer untuk merepresentasikan konteks setelahnya. Keluaran dari kombinasi dua arah hidden layer $\overrightarrow{h_t}$ dan $\overleftarrow{h_t}$ adalah :

$$y_t = W_{\vec{h}\,y}\vec{h}_t + W_{\overleftarrow{h}y}\vec{h}t \qquad (6)$$

METODOLOGI

Metodologi dan tahapan-tahapan penyelesaian masalah dapat diuraikan melalui gambar 3



Gambar 3. Alur Penyelesaian Masalah

Tahap 1: *Data Pre-Processing*

Pemrosesan awal bertujuan untuk menyiapkan data yang layak untuk diolah oleh algoritma Bi-LSTM. Ada beberapa tahap pemrosesan pada tahap ini, yaitu konversi teks ke bentuk *lowercase*, penghapusan tanda baca, penanganan stopwords.

Tahap 2: *Sentence Embedding*

Sentence embedding adalah representasi numerik dari kalimat teks yang bertujuan untuk menangkap makna atau informasi yang terkandung dalam kalimat tersebut. Representasi ini biasanya diperoleh melalui teknik-teknik pemrosesan bahasa alami dan pembelajaran mesin. Dalam sentence embedding, setiap kata dalam kalimat diwakili oleh vektor angka, dan vektor-vektor ini dikombinasikan untuk menghasilkan vektor yang merepresentasikan keseluruhan kalimat. Tujuan dari sentence embedding adalah untuk menyederhanakan dan mengkompakkan informasi yang terkandung dalam kalimat sehingga dapat lebih mudah diolah oleh algoritma pembelajaran mesin atau digunakan dalam berbagai tugas pemrosesan bahasa alami seperti klasifikasi teks, pencarian informasi, dan pemahaman bahasa alam.

Pada penelitian ini digunakan model word2vec dengan metode Continuous Skip-gram Model (Mikolov et al., 2013)

Tahap 3: Tokenisasi

Tokenisasi adalah proses menguraikan teks atau kalimat menjadi unit-unit yang lebih kecil yang disebut dengan token. Setiap token bisa berupa kata, frasa, atau bahkan karakter. Pada tingkat yang paling sederhana, tokenisasi teks melibatkan pemisahan kata dari kalimat menggunakan spasi sebagai pemisah, tetapi dapat melibatkan langkah-langkah lebih kompleks seperti pemisahan tanda baca, penghapusan karakter khusus, atau pengelompokan kata-kata yang terkait menjadi satu token. Tokenizer (*jumlah kata*, *oov_**token***). Parameter *oov_token* adalah parameter yang berfungsi untuk mengganti kata-kata yang tidak ditokenisasi menjadi karakter tertentu

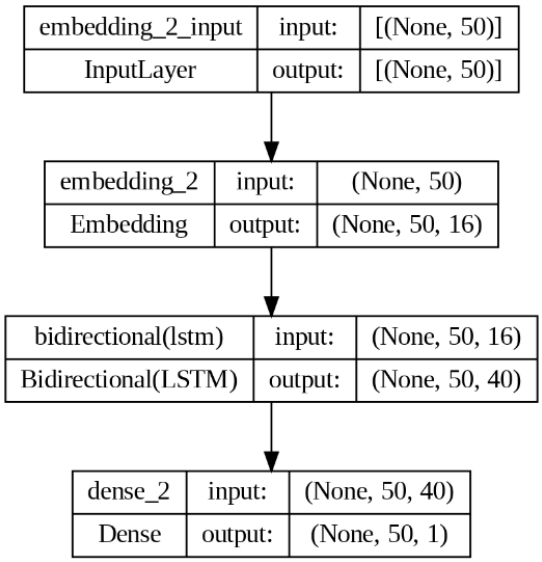
Tahap 4: *Sequencing* dan *padding*

Setelah melakukan tokenisasi pada teks, hal selanjutnya adalah mengubah setiap kalimat dalam teks ke dalam *sequence*. Sebuah sekuens adalah sebuah larik yang berisi kumpulan token sesuai dengan setiap kata pada sebuah kalimat dalam teks. Setelah setiap kalimat pada teks dikonversi menjadi sequence, harus memastikan agar setiap sequence sama panjang agar bisa dilatih pada model. Proses untuk mengubah setiap sequence agar memiliki panjang yang sama adalah padding. Pada padding, setiap sequence dibuat sama panjang dengan menambahkan nilai 0 secara sufiks atau prefiks hingga mencapai panjang maksimum sequence. Selain itu padding juga dapat memotong sequence hingga panjangnya sesuai dengan panjang maksimum sequence.

Bi-LSTM

Arsitektur Bi-LSTM ditapilkan pada gambar 4. Terdapat 4 layer pada Bi-LSTM yaitu layer input, layer embedding, layer bidirectional, dan layer output. Dimensi embedding sebesar 16, serta dimensi dari input sebesar nilai jumlah kata pada objek tokenizer sebesar 500. Layer Bi-LSTM menggunakan 40 layer LSTM. Layer terakhir adalah layer Dense dengan output =1. Fungsi aktifasi yang diterapkan adalah fungsi *sigmoid*.

Model Bi-LSTM selanjutnya dikompilasi dengan menggunakan fungsi loss binary_crossentropy, dengan optimizer adam optimizer, serta kriteria metrik adalah akurasi



Gambar 4. Model Bi-LSTM

Tabel 1. Hyperparameter Bi-LSTM

No	Parameter	Nilai Rujukan	Keterangan
1	vocab_size	500	Ukuran Jumlah kata
2	embeding_dim	16	Dimensi embedding
3	max_len	50	Ukuran input
4	n_lstm	40	Jumlah layer LSTM
5	drop_lstm	0.2	Drop rate

HASIL DAN DISKUSI

Data

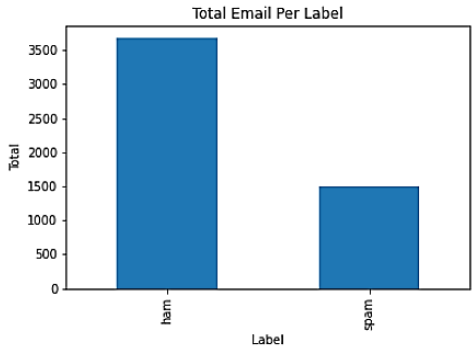
Penelitian ini menggunakan dua jenis dataset spam yaitu *SMS Spam Collection Dataset (https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset)*. Sample data spam pada dataset dapat ditampilkan pada tabel 2 berikut.

Tabel 2. Sample Data Spam

Teks Spam	Kategori
Go until jurong point, crazy.. Available only ...	ham
Ok lar... Joking wif u oni...	ham
Free entry in 2 a wkly comp to win FA Cup fina...	spam
U dun say so early hor... U c already then say...	ham
Nah I don't think he goes to usf, he lives aro...	ham

ISSN: 2598-8565 (media cetak)
ISSN: 2620-4339 (media online)

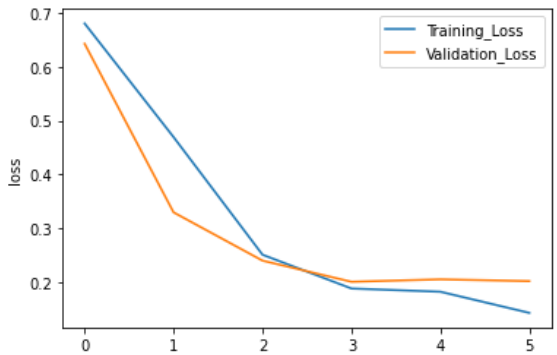
Terdapat 5170 data spam, yang terdiri atas 3672 ham, dan 1499 data spam sebagaimana di tampilkan pada gambar 5.



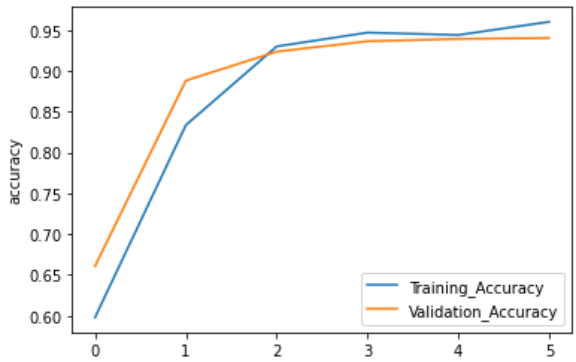
Gambar 6. Gambaran Dataset

Pembahasan

Pada gambar 6 dan 7 ditampilkan hasil pengujian performa model. Selama proses training dan validasi, terlihat bahwa nilai *loss* menurun. Demikian juga pada gambar 7 terdapat peningkatan akurasi dari epoch awal sampai epoch akhir. Dari kedua gambar tersebut, model Bi-LSTM tidak mengalami *overfitting*.



Gambar 6. Grafik loss model Bi-LSTM



Gambar 7. Grafik *accuracy* model Bi-LSTM

Pada tabel 3 ditampilkan perbadingan model LSTM dan Bi-LSTM. Model Bi-LSTM memperoleh akurasi yang lebih baik dibading LSTM. Selain menggunakan SMS spam dataset, model juga diuji dengan dataset email spam dataset. Pada dataset Pertama, model Bi-LSTM memiliki akurasi sebesar 94,00% dan pada dataset ke 2, model Bi-LSTM memiliki performa akurasi 95, 75%

Tabel 3. Perbadingan Performa Model

Corpus	Model	loss	accuracy
SMS	LSTM	0.2482	0.9281
	Bi-LSTM	0.2023	**0.9400**
email	LSTM	0.1833	0.9283
	Bi-LSTM	0.1070	**0.9573**

KESIMPULAN

Penelitian ini menguji model berbasis LSTM dalam melakukan perosesan teks data spam. Bi-LSTM mampu menangkap pola teks dari sisi konteks dan teks, karena informasi dari kedua arah dapat digabungkan. Hasil pengujian membuktikan bahwwa model Bi-LSTM lebih efektif dalam pemahaman teks.

DAFTAR PUSTAKA

Barushka, A., & Hajek, P. (2020). Spam detection on social networks using cost-sensitive feature selection and ensemble-based regularized deep neural networks. *Neural Computing and Applications*, *32*(9), 4239–4257. https://doi.org/10.1007/s00521-019-04331-5

Graves, A., Mohamed, A. R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, *3*, 6645–6649. https://doi.org/10.1109/ICASSP.2013.6638947

Jain, A. K., & Gupta, B. B. (2018). Rule-Based Framework for Detection of Smishing Messages in Mobile Environment. *Procedia Computer Science*, *125*, 617–623. https://doi.org/https://doi.org/10.1016/j.procs.2017.12.079

Kaddoura, S., Alfandi, O., & Dahmani, N. (2020). A Spam Email Detection Mechanism for English Language Text Emails Using Deep Learning Approach. *2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, 193–198. https://doi.org/10.1109/WETICE49692.2020.00045

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, 1–12.

Mustagfirin, M. L., Wiriasto, G. W., Suksmadana, I. M. B., & Kinasih, I. P. (2022). Android-Based

Short Message Service Filtering using Long Short-Term Memory Classification Model. *Khazanah Informatika : Jurnal Ilmu Komputer Dan Informatika*, *8*(2), 163–171. https://doi.org/10.23917/khif.v8i2.17995

Navaney, P., Dubey, G., & Rana, A. (2018). SMS Spam Filtering Using Supervised Machine Learning Algorithms. *2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 43–48. https://doi.org/10.1109/CONFLUENCE.2018.8442564

Peng, W., Huang, L., Jia, J., & Ingram, E. (2018). Enhancing the Naive Bayes Spam Filter Through Intelligent Text Modification Detection. *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 849–854. https://doi.org/10.1109/TrustCom/BigDataSE.2018.00122

Pirozmand, P., Sadeghilalimi, M., Hosseinabadi, A. A. R., Sadeghilalimi, F., Mirkamali, S., & Slowik, A. (2023). A feature selection approach for spam detection in social networks using gravitational force-based heuristic algorithm. *Journal of Ambient Intelligence and Humanized Computing*, *14*(3), 1633–1646. https://doi.org/10.1007/s12652-021-03385-5

Pushpalatha, M., & Vijaya, A. (2022). An Optimized WS2FS: Web Socio-Spider Feature Ranking Based Malicious Website Detection Using Deep Spectral Soft-Max Recurrent Neural Network. *Computer Integrated Manufacturing Systems*, *28*(11 SE-Articles), 1507–1525.

Raj, H., Weihong, Y., Banbhrani, S. K., & Dino, S. P. (2018). LSTM Based Short Message Service (SMS) Modeling for Spam Classification. *Proceedings of the 2018 International Conference on Machine Learning Technologies*, 76–80. https://doi.org/10.1145/3231884.3231895

Roy, P. K., Singh, J. P., & Banerjee, S. (2020). Deep learning to filter SMS Spam. *Future Generation Computer Systems*, *102*, 524–533. https://doi.org/https://doi.org/10.1016/j.future.2019.09.001

Salunkhe, A. (2021). *Attention-based Bidirectional LSTM for Deceptive Opinion Spam Classification*.

Shaik, C. M., Penumaka, N. M., Abbireddy, S. K., Kumar, V., & Aravinth, S. S. (2023). Bi-LSTM and Conventional Classifiers for Email Spam Filtering. *2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, 1350–1355. https://doi.org/10.1109/ICAIS56108.2023.10073776

Sulthana, R., Verma, A., & Jaithunbi, A. K. (2023). *A Detailed Analysis on Spam Emails and Detection Using Machine Learning Algorithms BT - Inventive Systems and Control* (V. Suma, P. Lorenz, & Z. Baig (eds.); pp. 65–76). Springer Nature Singapore.

Xia, T. (2020). A Constant Time Complexity Spam Detection Algorithm for Boosting Throughput on Rule-Based Filtering Systems. *IEEE Access*, *8*, 82653–82661. https://doi.org/10.1109/ACCESS.2020.2991328