ANALISIS PHISHING PADA PESAN WHATSAPP MENGGUNAKAN LSTM

Alison de Jesus[1*], Petrus Katemba[2], Erna Rosani Nubatonis[3]
[1,2,3] Teknik Informatika, Universitas Uyelindo
[1*]koro290501@gmail.com, [2]petruskatemba@gmail.com, [3]ernarosaninubatonis@gmail.com
DOI: 10.46880/mtk.v12i2.6268

ABSTRACT

Phishing messages on instant messaging applications such as WhatsApp pose a cybersecurity threat, particularly through social engineering techniques, making automated detection of Indonesian-language phishing messages important. This study aims to develop a phishing detection system for Indonesian-language WhatsApp messages using the Long Short-Term Memory (LSTM) model. The dataset consisted of 1,162 messages, comprising 817 non-phishing and 345 phishing messages, which were processed through cleaning, case folding, normalization, tokenization, stopword removal, stemming using Sastrawi, sequencing, and padding before model training. The LSTM model was trained using 5 epochs, a batch size of 32, a learning rate of 0.001, and the Adam optimizer, and evaluation using a confusion matrix achieved an accuracy of 88.0%, precision of 72.50%, recall of 96.67%, and F1-score of 82.86%. These results indicate that the model can support the initial identification of potentially phishing messages, particularly due to its high recall in detecting phishing messages, although the imbalanced class distribution remains a limitation that may affect classification performance.

Keywords: WhatsApp Phishing, LSTM, NLP Indonesia, Deteksi Phishing, Text Preprocessing.

I. PENDAHULUAN

Perkembangan teknologi komunikasi telah mendorong penggunaan aplikasi pesan instan seperti WhatsApp sebagai media komunikasi digital yang cepat dan efisien [1]. Tingginya penggunaan WhatsApp juga meningkatkan peluang penyalahgunaan platform tersebut sebagai media penyebaran phishing. Kondisi ini menjadi perhatian di Indonesia. Berdasarkan laporan State of Scams in Indonesia 2025 yang diterbitkan oleh Global Anti-Scam Alliance (GASA), sebanyak 35% responden dewasa di Indonesia mengalami penipuan dalam 12 bulan terakhir, sedangkan WhatsApp menjadi platform yang paling banyak dilaporkan terkait aktivitas penipuan dengan persentase 89% [2]. Phishing merupakan bentuk serangan rekayasa sosial yang menyamarkan pelaku sebagai entitas tepercaya untuk mengelabui pengguna agar memberikan informasi pribadi atau kredensial [3]. Penyebaran pesan manipulatif dan tautan palsu menjadikan deteksi phishing pada komunikasi digital sebagai permasalahan yang perlu diperhatikan.

Deteksi phishing pada pesan memiliki tantangan karena karakteristik teks dapat berupa singkatan, variasi penulisan, kata manipulatif, dan kombinasi bahasa. Kondisi tersebut menyebabkan pendekatan konvensional berbasis aturan (rule-based) memiliki keterbatasan dalam mengenali pola teks yang kompleks [4]. Pendekatan deep learning seperti Long Short-Term Memory (LSTM) dapat digunakan untuk menangani data teks yang bersifat sekuensial karena mampu mempertahankan informasi dari urutan sebelumnya dan mempelajari hubungan antarkata [5]. Penerapan LSTM pada pengolahan teks juga menunjukkan bahwa metode tersebut dapat digunakan untuk memproses data berbasis teks secara otomatis [6]. Kemampuan tersebut menjadi pertimbangan dalam pemilihan LSTM karena pesan WhatsApp berbahasa Indonesia memiliki variasi kata, singkatan, dan susunan kalimat yang dapat membentuk pola berdasarkan urutan dan hubungan antarkata. Penelitian Manurung, Munawir, dan Pradeka (2025) telah menerapkan TF-IDF dan metode machine learning untuk melakukan penyaringan pesan spam dan phishing pada WhatsApp [1]. Chougule, Oza, Patil, dan Diwane (2025) menerapkan LSTM, GRU, dan GloVe untuk mendeteksi phishing dan smishing [3].

Sementara itu, Mahesh dan Duvvuri (2026) membandingkan LSTM dan BiLSTM untuk deteksi spam pada SMS [7]. Hasil dari penelitian-penelitian tersebut menunjukkan bahwa pendekatan machine learning dan deep learning dapat digunakan dalam klasifikasi pesan, tetapi terdapat perbedaan pada metode dan karakteristik sumber data yang digunakan. Berdasarkan penelitian terdahulu tersebut, masih terdapat celah penelitian pada penerapan LSTM secara khusus untuk klasifikasi pesan phishing pada WhatsApp berbahasa Indonesia. Penelitian pada WhatsApp sebelumnya menggunakan TF-IDF dan machine learning, sedangkan penelitian yang menggunakan LSTM lebih banyak diterapkan pada konteks phishing dan smishing secara umum atau pada SMS. Sementara itu, pesan WhatsApp berbahasa Indonesia memiliki karakteristik berupa variasi kata, singkatan, dan susunan kalimat yang dapat membentuk pola berdasarkan urutan antarkata. Oleh karena itu, diperlukan penerapan LSTM pada dataset pesan WhatsApp berbahasa Indonesia untuk mengklasifikasikan pesan ke dalam kategori Phishing dan Non Phishing berdasarkan karakteristik teks.

Berdasarkan permasalahan dan celah penelitian tersebut, penelitian ini bertujuan untuk: (1) menerapkan tahapan preprocessing pada pesan WhatsApp berbahasa Indonesia agar dapat digunakan sebagai masukan model LSTM; (2) membangun model LSTM untuk mengklasifikasikan pesan WhatsApp ke dalam kategori Phishing dan Non Phishing; dan (3) mengevaluasi kinerja model berdasarkan nilai Accuracy, Precision, Recall, dan F1-Score. Secara praktis, penelitian ini menghasilkan sistem berbasis web yang dapat digunakan sebagai alat bantu untuk melakukan

This work is licensed under a **Creative Commons Attribution 4.0 International License.**

identifikasi awal terhadap pesan WhatsApp yang terindikasi phishing.

II. KAJIAN TERKAIT

Deep Learning

Deep Learning merupakan metode pembelajaran mesin yang mampu mengekstraksi representasi data kompleks secara otomatis dan banyak digunakan dalam pemrosesan teks maupun data [8]. Dibandingkan metode konvensional seperti Naïve Bayes, SVM, dan HMM, Deep Learning memiliki keunggulan dalam mengolah data kompleks, khususnya pada pemrosesan bahasa dan data berbasis teks [9]. Beberapa arsitektur Deep Learning yang umum digunakan meliputi CNN, RNN, LSTM, GAN, DBN, dan DRL [8].

Long Short Term Memory

Long Short-Term Memory (LSTM) merupakan varian Recurrent Neural Network (RNN) yang dirancang untuk mengatasi vanishing gradient pada data sekuensial. LSTM memiliki Memory Cell, Input Gate, Forget Gate, dan Output Gate yang berfungsi menyimpan informasi penting, memasukkan informasi baru, mengatur keluaran, dan menghapus informasi yang tidak diperlukan [10]. Secara matematis, mekanisme cell state diatur melalui fungsi linear dan non-linear sebagai berikut:

1. Forget Gate (ft) : Forget Gate berfungsi untuk menentukan informasi dari cell state sebelumnya yang akan dipertahankan atau dibuang.

$$ft = \sigma(Wf \cdot [ht - 1, xt] + bf) \quad (1)$$

Keterangan:
ft = nilai Forget Gate pada waktu ke-t.
σ = fungsi aktivasi Sigmoid, menghasilkan nilai antara 0–1.
Wf = matriks bobot (weight) pada Forget Gate.
$ht - 1$ = hidden state dari waktu sebelumnya.
xt = data masukan (input) pada waktu ke-t.
$[ht - 1, xt]$ = proses penggabungan (concatenation) hidden state sebelumnya dengan input saat ini.
bf = nilai bias pada Forget Gate.

2. Input Gate (it) : Input Gate menentukan seberapa banyak informasi baru yang akan dimasukkan ke dalam cell state.

$$it = \sigma(Wi \cdot [ht - 1, xt] + bi) \quad (2)$$

Keterangan:
it = nilai Input Gate pada waktu ke-t.
σ = fungsi aktivasi Sigmoid.
Wi = matriks bobot pada Input Gate.
$ht - 1$ = hidden state sebelumnya.
xt = input saat ini.
bi = nilai bias pada Input Gate.

3. Candidate Cell $(C{\sim}t)$: Candidate Cell menghasilkan kandidat informasi baru yang akan dipertimbangkan untuk disimpan pada cell state.

$$C{\sim}t = tanh(WC \cdot [ht - 1, xt] + bC) \quad (3)$$

Keterangan:
$C{\sim}t$ = kandidat nilai cell state baru.
$tanh$ = fungsi aktivasi Hyperbolic Tangent, menghasilkan nilai antara -1 hingga 1.
WC = matriks bobot kandidat cell.
$ht - 1$ = hidden state sebelumnya.
xt = input saat ini.
bC = nilai bias kandidat cell.

4. Cell State Update (Ct) : Cell State Update menggabungkan informasi lama yang dipertahankan oleh Forget Gate dengan informasi baru yang dipilih oleh Input Gate.

$$Ct = ft \times Ct - 1 + it \times C{\sim}t \quad (4)$$

Keterangan:
Ct = cell state pada waktu ke-t.
$Ct - 1$ = cell state sebelumnya.
ft = output Forget Gate.
it = output Input Gate.
$C{\sim}t$ = kandidat cell state baru.
$\times$ = operasi perkalian elemen (element-wise multiplication).

5. Output Gate (ot) : Output Gate menentukan bagian dari cell state yang akan dikeluarkan sebagai hidden state**.**

$$ot = \sigma(Wo \cdot [ht - 1, xt] + bo) \quad (5)$$

Keterangan**:**
ot = nilai Output Gate pada waktu ke-t.
σ = fungsi aktivasi Sigmoid.
Wo = matriks bobot pada Output Gate.
$ht - 1$ = hidden state sebelumnya.
xt = input saat ini.
bo = nilai bias pada Output Gate.

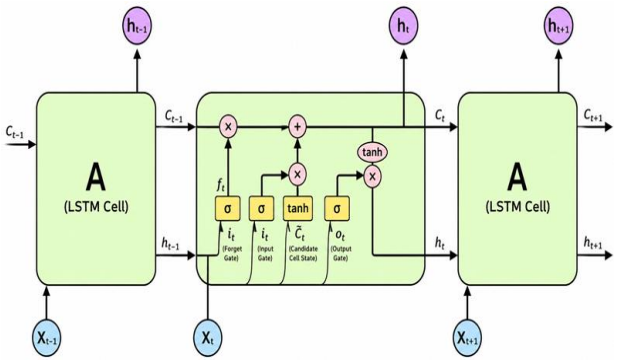
6. Hidden State (ht) : Hidden State merupakan keluaran akhir dari satu unit LSTM yang diteruskan ke langkah waktu berikutnya maupun ke lapisan output.

$$ht = ot \times tanh(Ct) \quad (6)$$

Keterangan:
ht = hidden state pada waktu ke-t.
ot = output dari Output Gate.
Ct = cell state yang telah diperbarui.
$tanh(Ct)$ = hasil normalisasi cell state menggunakan fungsi Hyperbolic Tangent.
$\times$ = operasi perkalian elemen (element-wise multiplication).

Kemampuan kalkulasi hidden state dalam menyimpan memori jangka panjang inilah yang menjadikan algoritma LSTM ideal untuk diterapkan dalam pemrosesan bahasa alami (NLP) dan pemodelan rentang waktu linguistik. Hal ini tervalidasi oleh pengujian model klasifikasi yang mampu mencatatkan akurasi maksimal sebesar 98,4%, terbukti melampaui algoritma prediktif lainnya seperti CNN, Naive Bayes, dan SVM [11]. Pada riset ini, mekanisme penyaringan dan perhitungan matematis LSTM dieksploitasi agar model dapat menangkap konteks linguistik ambigu, sehingga akurat dalam membedah mana yang tergolong pesan phishing dan mana pesan yang aman.

This work is licensed under a **Creative Commons Attribution 4.0 International License.**



Gambar 1. Arsitektur LSTM

Gambar 1. Arsitektur LSTM menunjukkan arsitektur Long Short-Term Memory (LSTM) yang merupakan pengembangan dari Recurrent Neural Network (RNN) untuk mengolah data berurutan (sequence) seperti teks pesan WhatsApp. Arsitektur ini terdiri dari beberapa LSTM Cell yang saling terhubung dan memproses data secara berurutan dari waktu sebelumnya (*t-1*), waktu saat ini (*t*), hingga waktu berikutnya (*t+1*). Setiap sel menerima input (x□)**,** hidden state (h□_ ₁)**,** dan cell state (C□_ ₁) dari langkah sebelumnya. Di dalam sel terdapat tiga gerbang utama yaitu Forget Gate yang menentukan informasi mana yang harus dilupakan, Input Gate yang menentukan informasi baru yang akan disimpan, dan Output Gate yang menentukan informasi yang akan diteruskan ke langkah berikutnya.

Melalui mekanisme ini, LSTM mampu menyimpan informasi penting dalam jangka panjang dan mengabaikan informasi yang tidak relevan. Pada penelitian Analisis Phishing Pada Pesan WhatsApp Menggunakan LSTM, arsitektur ini digunakan untuk memahami pola bahasa dan konteks pesan sehingga dapat membedakan pesan phishing dan non phishing dengan tingkat akurasi yang tinggi.

Phishing

Phishing merupakan kejahatan siber yang menipu korban agar memberikan informasi pribadi seperti kata sandi, data perbankan, atau kredensial akun melalui pesan yang menyamar sebagai sumber tepercaya. Pada WhatsApp, serangan phishing dapat berupa tautan palsu, malware, atau narasi penipuan yang memanfaatkan rekayasa sosial untuk mendorong korban melakukan tindakan tertentu, seperti mengklik tautan dengan alasan verifikasi akun, tawaran kerja, atau pengiriman paket [12]. Rendahnya kesadaran keamanan pengguna turut meningkatkan risiko pencurian data melalui tautan atau pesan yang tampak meyakinkan[13]. Selain itu, penyebaran pesan instan yang cepat dan masif turut meningkatkan keberhasilan serangan phishing pada platform komunikasi digital [14]

WhatsApp

WhatsApp merupakan salah satu aplikasi pesan instan yang banyak digunakan untuk bertukar pesan, gambar, video, dokumen, serta melakukan komunikasi secara real-time. Popularitas dan kemudahan berbagi tautan maupun lampiran menjadikan WhatsApp turut dimanfaatkan sebagai media penyebaran phishing. Pelaku dapat menyebarkan tautan palsu, berkas berbahaya, atau narasi penipuan yang memanfaatkan rekayasa sosial untuk mendorong pengguna melakukan tindakan tertentu. Kondisi ini berisiko karena rendahnya kesadaran sebagian pengguna terhadap keamanan digital dan ancaman pesan atau berkas palsu[12]

Visual Studio Code (VS Code) dan Python

Visual Studio Code (VS Code) merupakan perangkat lunak penyunting kode sumber atau Integrated Development Environment (IDE) yang mendukung integrasi Python, Jupyter Notebook, terminal, debugging, dan syntax highlighting. Dalam penelitian ini, VS Code digunakan untuk membangun dan mengembangkan sistem deteksi phishing berbasis NLP [15]. Python merupakan bahasa pemrograman yang banyak digunakan dalam pengembangan Machine Learning dan Deep Learning karena didukung pustaka seperti TensorFlow, Keras, Scikit-learn, dan NLTK. Python digunakan dalam proses preprocessing, seperti tokenizing, stopword removal, dan stemming. Penggunaan Python dalam penelitian terdahulu mencakup deteksi spam [16]. dan pemrosesan NLP [14].

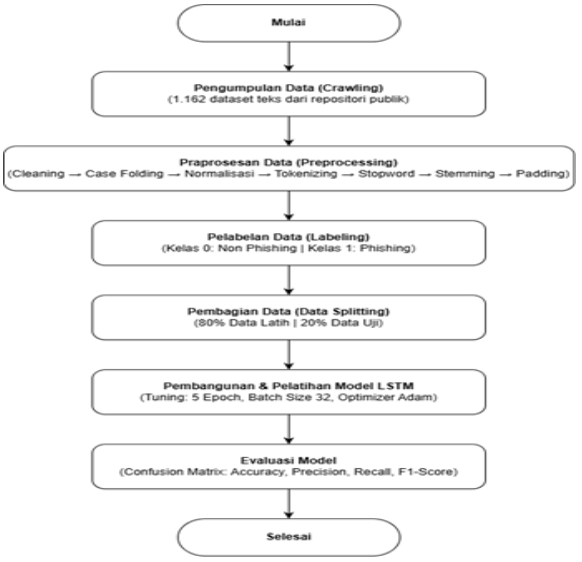
III. METODE

Bahan dan Alat Penelitian

Penelitian ini menggunakan perangkat keras dan perangkat lunak untuk mendukung analisis phishing pada pesan WhatsApp menggunakan algoritma Long Short-Term Memory (LSTM). Perangkat keras yang digunakan berupa laptop HP dengan prosesor AMD Ryzen 5, RAM 8 GB, dan penyimpanan 4 TB, serta smartphone Huawei Mate 80 Pro dengan RAM 16 GB dan penyimpanan 512 GB. Perangkat lunak yang digunakan meliputi Windows 11, Microsoft Office 2021, Mendeley, WhatsApp, Google Chrome, Visual Studio Code, Google Drive, dan Python untuk implementasi serta pelatihan model LSTM.

Prosedur Penelitian

Prosedur penelitian dilakukan secara sistematis, dimulai dari pengumpulan data (crawling), dilanjutkan dengan prapemrosesan teks, pelabelan data yang diverifikasi oleh pakar, pembagian data menjadi data latih dan data uji, pelatihan model Long Short-Term Memory (LSTM), serta diakhiri dengan evaluasi model untuk mengukur kinerja dan tingkat akurasi sistem.



Gambar 2. Prosedur Penelitian

This work is licensed under a Creative Commons Attribution 4.0 International License.

Prosedur penelitian ini dilakukan secara sistematis melalui serangkaian tahapan yang terstruktur, mulai dari pengumpulan 1.162 data teks melalui crawling, prapemrosesan (cleaning, case folding, normalisasi, tokenizing, stopword, stemming, padding), hingga pelabelan data menjadi kelas non-phishing dan phishing. Setelah data dibagi menjadi 80% data latih dan 20% data uji, model LSTM dibangun dan dilatih dengan konfigurasi 5 epoch, batch size 32, serta optimizer Adam, yang kemudian diakhiri dengan evaluasi performa model menggunakan confusion matrix berdasarkan metrik accuracy, precision, recall, dan f1-score.

Pengumpulan Data dan Prapemrosesan

Pengumpulan data dilakukan dari dataset publik, dokumentasi percakapan yang telah dianonimkan, dan penelitian terdahulu, kemudian diberi label 0 untuk non-phishing dan 1 untuk phishing. Data selanjutnya melalui tahap preprocessing meliputi case folding, pembersihan teks, stopword removal, tokenization, stemming, dan padding untuk menghasilkan data numerik yang seragam sebelum digunakan dalam pelatihan model LSTM.

Perhitungan Manual

1. Dataset

Dataset yang digunakan merupakan dataset sekunder hasil crawling dari repositori publik yang berisi pesan WhatsApp berbahasa Indonesia. Dataset terdiri atas 1.162 pesan, yaitu 817 pesan Non Phishing (kelas 0) dan 345 pesan Phishing (kelas 1). Data tersebut digunakan untuk proses pelatihan dan pengujian model Long Short-Term Memory (LSTM). Sebelum digunakan dalam proses pelatihan, seluruh dataset melalui tahapan prapemrosesan teks yang meliputi cleaning, case folding, normalisasi, tokenisasi, stopword removal, stemming menggunakan Sastrawi, sequencing, dan padding. Tahapan tersebut dilakukan untuk mengubah data teks mentah menjadi representasi numerik yang sesuai sebagai masukan model LSTM. Hasil pengolahan dataset menghasilkan 20.532 token dengan 4.215 kata unik, kemudian dilakukan padding dengan panjang maksimum (maxlen) 10. Setelah seluruh tahapan prapemrosesan selesai, dataset dibagi menjadi 80% data latih dan 20% data uji. Data latih digunakan untuk mempelajari pola teks pada kategori Phishing dan Non Phishing, sedangkan data uji digunakan sebagai unseen data untuk mengevaluasi kemampuan model dalam mengklasifikasikan data yang tidak digunakan selama proses pelatihan. Penelitian ini menggunakan pembagian data latih dan data uji tanpa menerapkan cross-validation.

Tabel 1. Dataset

No	Text (Pesan Asli)	Label
1.	Akun WhatsApp Anda Akan Diblokir!!! Klik Link Berikut Sekarang...	1
2.	Besok kita rapat jam 10 pagi :)	0
3.	Hadiah Rp50 juta!!! Klaim sekarang!!!	1
4.	Tolong kirim laporan hari ini.	0
5.	WhatsApp anda bermasalah!!!	1
6.	Jangan lupa tugas kuliah ya :)	0
7.	Selamat!!! Anda menang undian.	1
8.	Datang ke kantor jam 8.	0
9.	Klik link ini sekarang!!!	1
10.	Kita ketemu di kampus yah	0
11.	Pembayaran tertunda!!!	1
12.	Besok acara dibatalkan.	0
13.	Konfirmasi data anda sekarang!!!	1
14.	Cek email yang saya kirim.	0
15.	Akun bank diblokir!!!	1
16.	Kumpul di aula jam 6.	0
17.	Aktivitas mencurigakan!!!	1
18.	Makan malam di luar :)	0
19.	WhatsApp akan dinonaktifkan!!!	1
20.	File laporan sudah saya upload.	0

Tabel 1 menunjukkan contoh data mentah (*raw data*) sebelum melalui proses prapemrosesan. Teks masih mempertahankan bentuk aslinya, termasuk huruf kapital, tanda baca, simbol, dan variasi penulisan. Kolom label digunakan sebagai *ground truth*, dengan label 1 menunjukkan pesan Phishing dan label 0 menunjukkan pesan Non Phishing yang selanjutnya menjadi target klasifikasi model LSTM.

2. Preprocessing Text

Pada tahapan ini dilakukan tahapan proses membersihkan data dari kesalahan (cleaning), yaitu penghapusan angka (remove number), penghapusan tanda baca (remove punctuation) , penghapusan spasi yang berlebihan (remove multiple whitespace), dan penghapusan karakter tunggal (remove single char).

Tahap cleaning bertujuan membersihkan data dari elemen noise yang tidak memiliki nilai semantik bagi model, seperti tanda baca, simbol, emotikon, dan URL. Proses ini dilakukan untuk mengurangi variasi bentuk teks sehingga kata yang sama tidak dianggap sebagai token yang berbeda, serta membantu mengurangi kompleksitas vocabulary sebelum data memasuki tahap prapemrosesan berikutnya. Sebagai contoh, teks "Akun WhatsApp Anda Akan Diblokir!!! Klik Link Berikut Sekarang..." setelah proses cleaning menjadi "Akun WhatsApp Anda Akan Diblokir Klik Link Berikut Sekarang", karena tanda seru dan tanda titik yang tidak diperlukan telah dihapus. Setelah tahap cleaning, data diproses secara berurutan melalui case folding, normalisasi, tokenisasi, stopword removal, stemming, sequencing, dan padding. Case folding digunakan untuk mengubah seluruh huruf menjadi huruf kecil, sedangkan normalisasi digunakan untuk mengubah kata tidak baku atau singkatan ke bentuk yang sesuai. Selanjutnya, tokenisasi memecah teks menjadi unit kata, kemudian stopword removal menghapus kata-kata yang kurang memberikan informasi. Tahap stemming dilakukan menggunakan Sastrawi untuk memperoleh bentuk dasar kata. Hasil token kemudian diubah menjadi indeks berdasarkan vocabulary yang terbentuk. Berdasarkan hasil pengolahan dataset, diperoleh 20.532 token dengan 4.215 kata unik. Selanjutnya, dilakukan padding dengan maxlen = 10 untuk menyamakan panjang setiap urutan data sehingga dapat digunakan sebagai masukan model LSTM.

a. *Case folding* merupakan tahap prapemrosesan teks yang bertujuan mengubah seluruh huruf menjadi huruf kecil (lowercase) agar representasi teks menjadi konsisten. Proses ini dilakukan untuk menghindari perbedaan representasi antara huruf kapital dan huruf kecil ketika teks diproses pada tahap berikutnya.

This work is licensed under a **Creative Commons Attribution 4.0 International License.**

Sebagai contoh, teks "Akun WhatsApp Anda Akan Diblokir Klik Link Berikut Sekarang" setelah proses case folding menjadi "akun whatsapp anda akan diblokir klik link berikut sekarang".

b. Normalisasi, merupakan tahap prapemrosesan yang bertujuan mengubah kata atau frasa tidak baku, seperti bahasa slang, singkatan, dan kesalahan penulisan (typo), menjadi bentuk yang lebih standar menggunakan kamus normalisasi. Proses ini dilakukan untuk menghasilkan representasi kata yang konsisten sebelum memasuki tahap pemrosesan berikutnya. Sebagai contoh, teks "Besok Kita Rapat Jam 10 Pagi yah" dinormalisasi menjadi "Besok Kita Rapat Jam 10 Pagi ya", dengan mengubah kata tidak baku "yah" menjadi "ya".

c. Tokenizing, pada tahap ini dilakukan proses (tokenizing) yaitu memecahkan teks menjadi unit-unit kecil yang disebut token seperti kata, frasa, atau simbol untuk memudahkan analisis dan pengolahan teks.

Tabel 2. Proses Tokenizing

Sebelum Tokenizing	Sesudah Tokenizing (Array Kata)	Sequencing (Angka Unik)
akun whatsapp anda akan diblokir...	['akun', 'whatsapp', 'anda', 'akan', 'diblokir', 'klik', 'link', 'berikut', 'sekarang']	[1, 2, 3, 4, 5, 6, 7, 8, 9]
besok kita rapat jam 10 pagi	['besok', 'kita', 'rapat', 'jam', '10', 'pagi']	[10, 11, 12, 13, 14, 15]
hadiah rp50 juta klaim sekarang	['hadiah', 'rp50', 'juta', 'klaim', 'sekarang']	[16, 17, 18, 19, 9]
tolong kirim laporan hari ini	['tolong', 'kirim', 'laporan', 'hari', 'ini']	[20, 21, 22, 23, 24]

Tokenisasi merupakan tahap prapemrosesan yang memecah setiap kalimat menjadi token atau unit kata. Dari 1.162 data pesan, proses tokenisasi menghasilkan 20.532 token dengan 4.215 kata unik yang kemudian membentuk kosakata (vocabulary) sebagai dasar representasi numerik (word embedding) sehingga dapat diproses oleh model Long Short-Term Memory (LSTM).

d. Padding Sequence adalah tahapan prapemrosesan yang berfungsi untuk menyamakan panjang seluruh deret matriks angka (sequence) dari setiap pesan teks.

Tabel 3. Proses Padding Sequence (Maxlen = 10)

Hasil Sequence (Angka Unik)	Hasil Padding (Penambahan Angka 0)
[1, 2, 3, 4, 5, 6, 7, 8, 9]	[1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
[10, 11, 12, 13, 14, 15]	[10, 11, 12, 13, 14, 15, 0, 0, 0, 0]
[16, 17, 18, 19, 9]	[16, 17, 18, 19, 9, 0, 0, 0, 0, 0]
[20, 21, 22, 23, 24]	[20, 21, 22, 23, 24, 0, 0, 0, 0, 0]

Padding merupakan tahap prapemrosesan yang menyamakan panjang setiap urutan token dengan menambahkan nilai 0 pada pesan yang lebih pendek hingga mencapai panjang maksimum (maxlen = 10). Proses ini bertujuan menghasilkan dimensi input yang seragam agar dapat diproses oleh model Long Short-Term Memory (LSTM).

e. Stopword Removal merupakan tahap prapemrosesan yang bertujuan menghapus kata-kata umum yang kurang memiliki informasi penting dalam teks. Proses ini dilakukan untuk mempertahankan kata-kata yang lebih relevan sehingga data menjadi lebih ringkas dan dapat diproses secara lebih efektif oleh model Long Short-Term Memory (LSTM). Sebagai contoh, teks "akun whatsapp anda akan diblokir klik link berikut sekarang" setelah proses stopword removal menjadi "akun whatsapp diblokir klik link berikut sekarang", dengan menghilangkan kata umum seperti "anda" dan "akan".

f. *Stemming* merupakan tahap prapemrosesan yang bertujuan mengubah kata berimbuhan menjadi kata dasar menggunakan Sastrawi. Proses ini dilakukan untuk menyatukan variasi bentuk kata sehingga kata yang memiliki makna dasar sama dapat direpresentasikan secara konsisten sebelum diproses oleh model Long Short-Term Memory (LSTM). Sebagai contoh, teks "akun whatsapp diblokir klik link sekarang" setelah proses stemming menjadi "akun whatsapp blokir klik link sekarang", karena kata "diblokir" diubah menjadi bentuk dasarnya, yaitu "blokir".

3. Pembagian Data

Setelah seluruh tahapan preprocessing selesai, dataset dibagi menggunakan metode *split* menjadi 80% data latih dan 20% data uji. Data latih digunakan untuk mempelajari pola teks pada kategori Phishing dan Non Phishing, sedangkan data uji digunakan sebagai unseen data untuk mengevaluasi kemampuan model dalam mengklasifikasikan data yang tidak digunakan selama proses pelatihan. Penelitian ini menggunakan pembagian data latih dan data uji tanpa menerapkan cross-validation.

Tabel 4. Data Training

Teks Hasil Preprocessing	Label
akun whatsapp blokir klik link ikut sekarang	1
besok rapat jam 10 pagi	0
hadiah rp50 juta klaim sekarang	1
tolong kirim lapor hari	0
whatsapp masalah	1
jangan lupa tugas kuliah ya	0
selamat menang undi	1
datang kantor jam 8	0
klik link sekarang	1
temu kampus ya	0
bayar tunda	1
besok acara batal	0
cek email kirim	0
kumpul aula jam 6	0
makan malam luar	0
file lapor sudah upload	0

Tabel 4 (Data Training) menggunakan 80% dari total data (16 data) sebagai bahan pelatihan LSTM untuk menemukan pola pesan phishing dan non-phishing.

Tabel 5. Data testing

Teks Hasil Preprocessing	Label
konfirmasi data sekarang	1
akun bank blokir	1
aktivitas curiga	1
whatsapp nonaktif	1

This work is licensed under a Creative Commons Attribution 4.0 International License.

Tabel 5 (Data Testing) menggunakan 20% sisanya (4 data) sebagai unseen data untuk mengevaluasi akurasi model dalam mengklasifikasikan pesan baru.

4. Arsitektur LSTM dan Hyperparameter

Model yang digunakan terdiri atas Embedding Layer, LSTM Layer, dan Dense Layer dengan fungsi aktivasi sigmoid. Embedding Layer digunakan untuk mengubah indeks kata menjadi representasi vektor, sedangkan LSTM Layer memproses data berdasarkan urutan kata dalam pesan. Dense Layer menghasilkan klasifikasi biner antara Phishing dan Non Phishing. Model menggunakan optimizer Adam dengan learning rate 0,001 dan fungsi loss binary cross-entropy. Proses pelatihan dilakukan selama 5 epoch dengan batch size 32. Early Stopping dan Model Checkpoint diterapkan untuk memperoleh model dengan kinerja terbaik selama proses pelatihan. Pemilihan LSTM didasarkan pada kemampuannya dalam mempertahankan informasi dari urutan sebelumnya sehingga sesuai untuk mempelajari hubungan antarkata pada pesan WhatsApp. Konfigurasi learning rate 0,001, batch size 32, dan 5 epoch digunakan sesuai konfigurasi pelatihan yang diterapkan dalam penelitian, sedangkan Early Stopping dan Model Checkpoint digunakan untuk membantu memperoleh model dengan kinerja terbaik selama proses pelatihan.

5. Evaluasi Model

Evaluasi model dilakukan menggunakan confusion matrix dengan metrik Accuracy, Precision, Recall, dan F1-Score. Accuracy digunakan untuk mengukur proporsi keseluruhan prediksi yang benar, Precision mengukur ketepatan model dalam memprediksi kelas Phishing, Recall mengukur kemampuan model dalam menemukan pesan Phishing yang sebenarnya, sedangkan F1-Score mengukur keseimbangan antara Precision dan Recall. Tabel 6 menampilkan formula metrik evaluasi, sedangkan Tabel 7 menunjukkan struktur Confusion Matrix.

Evaluation Measure	Formula
Precision	$\frac{TP}{(TP + FP)}$
Recall	$\frac{TP}{(TP + FN)}$
F1-Score	$2 \times \frac{(Precision \times Recall)}{(Precision + Recall)}$
Accuracy	$\frac{(TP + TN)}{(TP + TN + FP + FN)}$

Tabel 6. Confusion Matrix

Actual / Predictions	Negative	Positive
Negative	TN	FP
Positive	FN	TP

Untuk menguji performa model LSTM, dilakukan simulasi perhitungan manual Confusion Matrix menggunakan 20 sampel dataset preprocessing dengan membandingkan Label Aktual dan Label Prediksi. Berikut detail hasil prediksi dari 20 data sampel:

Tabel 7. Simulasi Hasil Prediksi dan Confusion Matrix

Teks Hasil Preprocessing	Label Aktual	Label Prediksi	Keterangan Evaluasi
akun whatsapp blokir klik link ikut sekarang	1	1	TP (True Positive)
besok rapat jam 10 pagi	0	0	TN (True Negative)
hadiah rp50 juta klaim sekarang	1	1	TP (True Positive)
tolong kirim lapor hari	0	0	TN (True Negative)
whatsapp masalah	1	1	TP (True Positive)
jangan lupa tugas kuliah ya	0	0	TN (True Negative)
selamat menang undi	1	1	TP (True Positive)
datang kantor jam 8	0	0	TN (True Negative)
klik link sekarang	1	1	TP (True Positive)
temu kampus ya	0	0	TN (True Negative)
bayar tunda	1	1	TP (True Positive)
besok acara batal	0	1	FP (False Positive)*
konfirmasi data sekarang	1	1	TP (True Positive)
cek email kirim	0	0	TN (True Negative)
akun bank blokir	1	1	TP (True Positive)
kumpul aula jam 6	0	1	FP (False Positive)*
aktivitas curiga	1	1	TP (True Positive)
makan malam luar	0	0	TN (True Negative)
whatsapp nonaktif	1	1	TP (True Positive)
file lapor sudah upload	0	0	TN (True Negative)

Pada data No. 12 dan 16, model menebak 1 padahal aslinya 0 (False Positive). Hal ini terjadi karena kata "batal" dan "kumpul jam 6" mungkin dianggap sistem sebagai instruksi yang memiliki unsur urgensi/mendesak. Berdasarkan Tabel 8 di atas, dari 20 sampel data, kita dapat menyusun nilai Confusion Matrix sebagai berikut :

Tabel 8. Hasil Perhitungan Confusion Matrix

Komponen	Nilai	Keterangan
True Positive (TP)	10	Pesan phishing yang berhasil diklasifikasikan dengan benar.
True Negative (TN)	8	Pesan non-phishing yang berhasil diklasifikasikan dengan benar.
False Positive (FP)	2	Pesan non-phishing yang salah diklasifikasikan sebagai phishing.
False Negative (FN)	0	Tidak ada pesan phishing yang salah diklasifikasikan sebagai non-phishing.
Total Data	20	Jumlah keseluruhan data uji.

Selanjutnya, dilakukan perhitungan matematis manual untuk mendemonstrasikan pencarian nilai metrik performa model :

a. Perhitungan Akurasi (*Accuracy*)

$$\text{Accuracy} = \frac{(10 + 8)}{(10 + 8 + 2 + 0)}$$

$$\text{Accuracy} = \frac{18}{20}$$

$$\text{Accuracy} = 0.90\ (90\%)$$

b. Perhitungan Presisi (*Precision*)

$$\text{Precision} = \frac{10}{(10 + 2)}$$

This work is licensed under a Creative Commons Attribution 4.0 International License.

$$\text{Precision} = \frac{10}{12}$$
$$\text{Precision} = 0.833\ (83{,}3\%)$$

c. Perhitungan Recall (*Sensitivitas*)

$$\text{Recall} = \frac{10}{(10 + 0)}$$
$$\text{Recall} = \frac{10}{10}$$
$$\text{Recall} = 1.0\ (100\%)$$

d. Perhitungan *F1-Score*

$$\text{F1} - \text{Score} = 2 \times \frac{(0.833 \times 1.0)}{(0.833 + 1.0)}$$
$$\text{F1} - \text{Score} = 2 \times \frac{(0.833)}{(1.833)} = 2 \text{ x } 0.454$$
$$\text{F1} - \text{Score} = 0.908\ (90{,}8\%)$$

IV. HASIL DAN PEMBAHASAN

Implementasi Sistem

Implementasi sistem dilakukan dengan membangun aplikasi berbasis web yang mengintegrasikan model LSTM untuk mendeteksi pesan phishing pada WhatsApp. Sebanyak 1.162 pesan, terdiri atas 817 non-phishing dan 345 phishing, diproses menjadi 3.850 token unik sebagai vocabulary pada Embedding Layer. Sistem kemudian mengimplementasikan modul dashboard, crawling, preprocessing, pelabelan, pelatihan model, dan pengujian pesan.

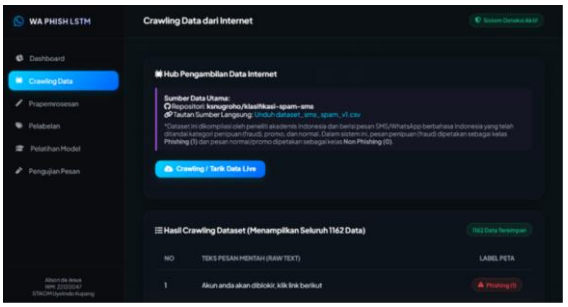
1. Antarmuka Dashboard

Halaman Dashboard berfungsi sebagai pusat kendali pemantauan (monitoring) yang menampilkan ringkasan metrik dari keseluruhan sistem. Pada implementasi ini, dasbor secara real-time menampilkan statistik dataset yang terdiri dari Total Pesan (1162), Pesan Non Phishing (817), dan Pesan Phishing (345). Halaman ini juga menyajikan hasil unjuk kerja (performa) model LSTM terbaik dengan metrik Accuracy sebesar 88.0%, Precision 72.50%, Recall 96.67%, dan F1-Score 82.86%. Selain itu, terdapat grafik distribusi kelas berjenis Donut Chart dan fitur untuk mengunduh hasil perhitungan manual ke format Excel.

Gambar 3. Antarmuka Dashboard Sistem WA PHISHING LSTM

2. Antarmuka Crawling

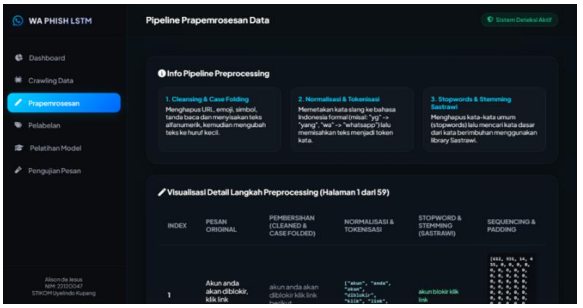
Halaman Crawling Data digunakan untuk mengambil dataset mentah dari repositori publik secara otomatis. Data yang diperoleh kemudian ditampilkan pada tabel hasil crawling yang memuat indeks, teks pesan, dan label.



Gambar 4. Antarmuka Crawling Data Pesan WhatsApp

3. Antarmuka Prapemrosesan Data

Halaman Pipeline Prapemrosesan Data mengimplementasikan tahapan cleansing, case folding, normalisasi, tokenisasi, stopword removal, dan stemming menggunakan Sastrawi. Proses dilanjutkan dengan sequencing dan padding hingga menghasilkan data numerik yang siap digunakan dalam model LSTM.



Gambar 5. Antarmuka Prapemrosesan Data Teks

4. Antarmuka Pelabelan Data

Halaman Manajemen Pelabelan Data berfungsi sebagai modul validasi label. Dari 1.162 data, sebanyak 1.161 data dilabeli secara otomatis dan 1 data diverifikasi secara manual. Tabel interaktif memungkinkan peneliti mengoreksi label Phishing (1) atau Non-Phishing (0) apabila terjadi kesalahan klasifikasi.

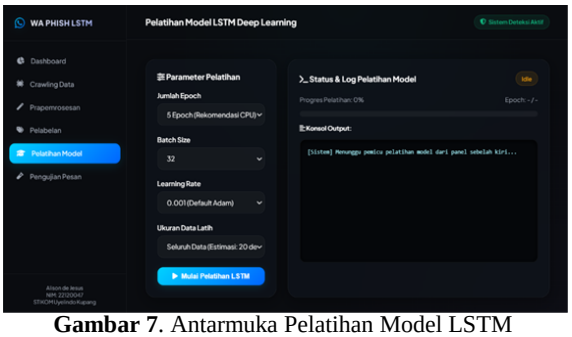
Gambar 6. Antarmuka Pelabelan Data Phishing dan Non Phishing

5. Antarmuka Pelatihan Model

Halaman Pelatihan Model LSTM Deep Learning merupakan modul inti untuk melakukan hyperparameter tuning. Antarmuka ini menyediakan form bagi pengguna untuk mengatur Jumlah Epoch (contoh: 5 Epoch), Batch Size (32), dan Learning Rate (0.001 - Default Adam). Setelah tombol "Mulai Pelatihan LSTM" dieksekusi, panel konsol hitam di

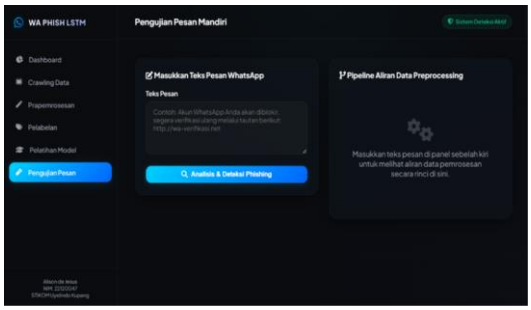
This work is licensed under a Creative Commons Attribution 4.0 International License.

sebelah kanan akan mencetak log sistem (Status & Log Pelatihan Model) untuk melacak persentase progres pelatihan secara langsung tanpa perlu berpindah aplikasi.



Gambar 7. Antarmuka Pelatihan Model LSTM

6. Antarmuka Pengujian Pesan
   Halaman Halaman Pengujian Pesan Mandiri digunakan untuk menguji model dengan memasukkan pesan WhatsApp dan menjalankan proses analisis untuk menghasilkan klasifikasi phishing atau non-phishing.



Gambar 8. Antarmuka Pengujian dan Klasifikasi Pesan WhatsApp

Hasil Penelitian

Berdasarkan implementasi dan komputasi yang telah dilakukan pada sistem "WA PHISHING LSTM", penelitian berhasil mengolah dataset sebanyak 1.162 pesan yang terdiri atas 817 pesan Non Phishing dan 345 pesan Phishing. Seluruh data melalui tahapan preprocessing yang meliputi cleaning, case folding, normalisasi, tokenizing, stopword removal, stemming, sequencing, dan padding. Tahapan tersebut bertujuan mengubah teks mentah menjadi bentuk yang lebih terstruktur dan dapat diproses oleh model LSTM.

Hasil proses tokenisasi menghasilkan 20.532 token dengan 4.215 kata unik yang kemudian digunakan sebagai vocabulary dalam representasi data. Pada proses pelatihan, model LSTM menggunakan konfigurasi hyperparameter berupa 5 Epoch, Batch Size 32, dan Learning Rate 0,001 dengan optimizer Adam. Konfigurasi tersebut digunakan untuk mempelajari pola urutan kata pada pesan Phishing dan Non Phishing. Hasil evaluasi menunjukkan bahwa model memperoleh Accuracy sebesar 88,0%, Precision sebesar 72,50%, Recall sebesar 96,67%, dan F1-Score sebesar 82,86%. Nilai Accuracy menunjukkan bahwa model mampu mengklasifikasikan sebagian besar data dengan benar. Sementara itu, Recall sebesar 96,67% menunjukkan bahwa model memiliki kemampuan yang tinggi dalam mengenali pesan Phishing dan menekan kemungkinan False Negative. Hasil evaluasi tersebut juga dapat dilihat melalui Confusion Matrix yang digunakan untuk mengetahui kesalahan klasifikasi berupa True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN). Nilai Precision sebesar 72,50% menunjukkan masih terdapat False Positive, yaitu pesan Non Phishing yang diprediksi sebagai Phishing.

Kondisi ini dapat dipengaruhi oleh kemiripan kosakata antara pesan Phishing dan Non Phishing, serta variasi bahasa pada WhatsApp seperti slang, singkatan, dan kesalahan ejaan. Selain itu, proses stemming dapat membuat beberapa kata dari konteks berbeda memiliki bentuk dasar yang sama sehingga informasi pembeda menjadi berkurang. Secara praktis, False Positive menyebabkan pesan aman perlu diperiksa kembali, sedangkan False Negative memiliki risiko lebih besar karena pesan Phishing dapat dianggap aman. Oleh karena itu, tingginya Recall menjadi penting dalam penelitian ini, meskipun berdampak pada Precision yang lebih rendah. Nilai F1-Score sebesar 82,86% menunjukkan keseimbangan yang cukup baik antara Precision dan Recall. Berdasarkan karakteristik pesan, pesan Phishing cenderung memiliki pola bahasa berupa ajakan atau tekanan untuk melakukan tindakan tertentu, seperti verifikasi, mengakses tautan, atau merespons informasi mengenai pemblokiran akun dan hadiah. Sementara itu, pesan Non Phishing lebih banyak menggunakan pola komunikasi sehari-hari seperti koordinasi, jadwal, dan penyampaian informasi.

Perbedaan tersebut menjadi pola yang dapat dipelajari oleh LSTM dalam proses klasifikasi. Hasil penelitian ini memiliki perbedaan pendekatan dengan penelitian Manurung, Munawir, dan Pradeka (2025) yang menggunakan TF-IDF dan metode machine learning untuk penyaringan pesan spam dan phishing pada WhatsApp. Penelitian ini menggunakan LSTM dengan memanfaatkan urutan kata dalam pesan. Namun, perbandingan nilai kinerja secara langsung tidak dilakukan karena terdapat perbedaan dataset, pembagian data, dan konfigurasi eksperimen. Secara keseluruhan, sistem mampu melakukan klasifikasi pesan WhatsApp dengan performa yang cukup baik, khususnya dalam mendeteksi pesan Phishing. Tingginya Recall menunjukkan bahwa model lebih mengutamakan kemampuan menangkap pesan Phishing, meskipun masih terdapat False Positive yang menyebabkan nilai Precision lebih rendah.

Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan fungsionalitas antarmuka berjalan sesuai rancangan menggunakan metode Black Box Testing. Skenario dan hasil pengujian ditampilkan pada Tabel 9.

Tabel 9. Hasil Pengujian BlackBox

Modul	Skenario Pengujian	Hasil yang Diharapkan	Status
Dashboard Sistem	Membuka dashboard.	Menampilkan 1.162 pesan, 817 *Non* Phishing, 345 Phishing, dan Accuracy 88,0%	Valid
Crawling Data	Menjalankan crawling	Dataset berhasil diambil dan ditampilkan	Valid
Prapemrosesan Data	Menampilkan preprocessing	Tahapan preprocessing	Valid

This work is licensed under a Creative Commons Attribution 4.0 International License.

	.	hingga sequencing dan padding ditampilkan	
Manajemen Pelabelan	Mengoreksi label	Label dan status verifikasi berhasil diperbarui	Valid
Pelatihan Model LSTM	Menjalankan pelatihan model dengan parameter yang ditentukan.	Sistem memproses pelatihan serta menampilkan log dan progres pelatihan secara real-time.	Valid
Pengujian Pesan	Memasukkan pesan WhatsApp	Pesan berhasil diproses dan diklasifikasikan.	Valid

Kelebihan dan Kekurangan

Kelebihan Sistem ini adalah model LSTM memiliki kinerja yang baik dengan Accuracy 88,0%, F1-Score 82,86%, dan Recall 96,67%, sehingga mampu mendeteksi sebagian besar pesan Phishing. Sistem juga transparan karena menampilkan setiap tahap prapemrosesan data, memiliki antarmuka dark mode yang intuitif, serta memungkinkan pengguna mengubah parameter Epoch dan Batch Size tanpa mengubah source code.

Kekurangan Sistem ini memiliki beberapa keterbatasan. Pertama, dataset yang digunakan berjumlah 1.162 pesan, sehingga variasi karakteristik pesan Phishing dan Non Phishing yang dapat dipelajari oleh model masih terbatas. Kedua, distribusi kelas tidak seimbang, yaitu 817 data Non Phishing dan 345 data Phishing, yang dapat memengaruhi hasil klasifikasi. Ketiga, penggunaan 5 Epoch dapat menyebabkan proses pembelajaran belum optimal, sehingga diperlukan pengujian lebih lanjut dengan variasi Epoch dan parameter pelatihan untuk mengetahui kinerja model secara lebih luas. Selain itu, kemampuan generalisasi model terhadap gaya bahasa, pola penipuan, atau bentuk pesan Phishing lain yang tidak terdapat dalam dataset masih perlu diuji. Sistem juga memiliki keterbatasan pada proses stemming menggunakan Sastrawi yang kurang optimal untuk slang atau bahasa daerah. Semakin besar jumlah dataset, semakin tinggi pula kebutuhan waktu komputasi dalam proses pelatihan model.

V. KESIMPULAN

Berdasarkan hasil penelitian, sistem WA Phishing LSTM berhasil menerapkan metode Long Short-Term Memory (LSTM) untuk mengklasifikasikan pesan WhatsApp berbahasa Indonesia ke dalam kategori Phishing dan Non Phishing. Dengan dataset sebanyak 1.162 pesan, terdiri dari 817 Non Phishing dan 345 Phishing, model menghasilkan Accuracy 88,0%, Precision 72,50%, Recall 96,67%, dan F1-Score 82,86%. Nilai Recall yang tinggi menunjukkan kemampuan model yang baik dalam mengenali pesan Phishing pada dataset yang digunakan, meskipun masih terdapat False Positive yang menyebabkan nilai Precision lebih rendah.

Penelitian ini memiliki keterbatasan berupa ketidakseimbangan kelas dan keterbatasan stemming dalam menangani kata tidak baku, slang, dan variasi bahasa tertentu. Hasil penelitian juga terbatas pada karakteristik dataset yang digunakan sehingga belum membuktikan kemampuan generalisasi model terhadap berbagai bentuk pesan Phishing di luar dataset. Oleh karena itu, penelitian selanjutnya direkomendasikan untuk: (1) menerapkan teknik penyeimbangan data seperti SMOTE serta memperluas dataset agar representasi kelas lebih beragam; dan (2) meningkatkan preprocessing, khususnya kemampuan stemming dalam menangani slang dan kata tidak baku, serta mengeksplorasi penggunaan model Transformer seperti BERT atau IndoBERT untuk meningkatkan kemampuan klasifikasi. Secara praktis, sistem dapat digunakan sebagai alat bantu awal untuk identifikasi pesan Phishing, sedangkan pengembangan selanjutnya dapat diarahkan pada pengujian dataset yang lebih beragam dan deteksi secara real-time.

Meskipun sistem telah berhasil dibangun dengan tingkat akurasi yang memadai, masih terdapat beberapa ruang lingkup yang dapat dikembangkan untuk menyempurnakan penelitian di masa mendatang. Perlu melakukan penanganan Ketimpangan Data: Menerapkan teknik SMOTE untuk menyeimbangkan dataset dan mengurangi bias model, perluasan Kamus Data: Menambahkan kosakata slang dan bahasa daerah untuk meningkatkan hasil stemming dan pengembangan Real-Time: Mengintegrasikan API atau bot agar sistem dapat mendeteksi pesan secara langsung.

VI. REFERENSI

[1] F. A. Manurung, Munawir, and D. Pradeka, “Spam and Phishing Whatsapp Message Filtering Application Using TF - IDF and Machine Learning Methods,” Green Intell. Syst. Appl., vol. 5, no. 1, pp. 1–13, 2025, doi: 10.53623/gisa.v5i1.551.

[2] G. A.-S. Alliance, “State of Scams in Indonesia 2025,” Global Anti-Scam Alliance, 2025. [Online]. Available: https://gasa.org/knowledge-base/reports/state-of-scams-in-indonesia-2025

[3] A. M. Chougule, K. S. Oza, V. T. Patil, and R. B. Diwane, “Real-World Phishing and Smishing Detection Using Deep Learning: A Comparative Study of LSTM, GRU, and GloVe Embeddings,” Int. J. Adv. Res. Comput. Commun. Eng., vol. 14, no. 11, 2025, doi: 10.17148/IJARCCE.2025.1411149.

[4] M. Al Kautsar, G. Guntoro Setiaji, and A. Rifa, “Analisis Komparasi Kinerja Lstm dan CNN dalam Deteksi Spam Email Berbasis Deep learning,” Bull. Comput. Sci. Res., vol. 5, no. 4, pp. 584–593, 2025, doi: 10.47065/bulletincsr.v5i4.572.

[5] K. Rangaswamy, “Phishing Website Detection using Machine Learning and Deep Learning Techniques,” Power Syst. Technol., vol. 49, no. 1, pp. 947–959, 2025, doi: 10.52783/pst.1643.

[6] R. Lestari and D. Angela, “Pengembangan Aplikasi Chatbot Untuk Pemasaran Digital Perguruan Tinggi Menggunakan Long Short Term Memory (LSTM) (Studi Kasus: Marketing ITHB),” 2022, [Online]. Available: http://www.ithb.ac.id

[7] G. Mahesh and S. K. Duvvuri, “A deep learning framework for SMS spam detection: Comparative performance of LSTM and BiLSTM Models,” Glob. J. Eng. Technol. Adv., vol. 27, no. 01, pp. 50–63, 2026, doi: 10.30574/gjeta.2026.27.1.0086.

[8] A. Prayogi, N. Novriyenny, and I. G. Prahmana, “Dinamika Sentimen Komunikasi Mahasiswa dan Dosen dengan Pemanfaatan Analisis Pesan Whatsapp Akademis Menggunakan Machine Learning,” Repeater Publ. Tek. Inform. Dan Jar., vol. 3, no. 2, pp. 45–54,

This work is licensed under a **Creative Commons Attribution 4.0 International License.**

2025, doi: 10.62951/repeater.v3i2.404.

[9] E. Setiawan and W. Widji Pamungkas, "Pengujian Algoritma Deep Learning Pada Analisis Terhadap Layanan Gojek Menggunakan Data Media Sosial Twitter," J. Inf., vol. 10, no. 2, pp. 53–67, 2024, [Online]. Available: https://ojs.stmik-im.ac.id/index.php/Informasi/article/view/273

[10] R. Cahyadi, A. Damayanti, and D. Aryadani, "Recurrent Neural Network (Rnn) Dengan Long Short Term Memory (Lstm) Untuk Analisis Sentimen Data Instagram," J. Inform. Dan Komput., vol. 5, no. 1, pp. 1–9, 2020, doi: 10.22236/jutikom.v5i1.

[11] A. R. Gunawan and R. F. Alfa Aziza, "Sentiment Analysis Using LSTM Algorithm Regarding Grab Application Services in Indonesia," J. Appl. Informatics Comput., vol. 9, no. 2, pp. 322–332, 2025, doi: 10.30871/jaic.v9i2.8696.

[12] A. Trianurahmah, A. Fauzi, E. N. Tyas, M. A. Suryanto, M. Rizky, and P. Wibisono, "Analisis Ancaman Pishing Melalui Aplikasi WhatsApp: Studi Kasus Manajemen Sekuriti Waspadai Maraknya Kejahatan Phising Dengan Modus Berbasis Link," Orbit J. Ilmu Multidisiplin Nusant., vol. 1, no. 2, pp. 74–88, 2025, doi: 10.63217/orbit.v1i2.81.

[13] H. S. Wicaksana, U. Ependi, and A. Muzakir, "URL-Based Phishing Detection Using a BERT-LSTM Model," J. Inf. Syst. Informatics, vol. 8, no. 1, pp. 1344–1367, 2026, doi: 10.63158/journalisi.v8i1.1543.

[14] S. Verma, V. Ayala-Rivera, and A. O. Portillo-Dominguez, "Detection of Phishing in Mobile Instant Messaging Using Natural Language Processing and Machine Learning," in Proceedings - 2023 11th International Conference in Software Engineering Research and Innovation, CONISOFT 2023, 2023, pp. 159–168. doi: https://doi.org/10.1109/CONISOFT58849.2023.00029.

[15] D. Aditya, "Klasifikasi Teks Data Short Message Services (SMS) Berbahasa Indonesia Menggunakan Artificial Neural Network," 2024. [Online]. Available: https://repository.unsri.ac.id/147874/3/RAMA_55201_09021282025099_0008118205_0018048003_01_front_ref.pdf

[16] M. W. Sampurno Utomo, H. Wisnu Murti, A. Widya Indah Sujatmoko, and A. Puspita Sari, "Deteksi Spam Email Menggunakan Metode Lstm (Long Short Term Memory)," JATI (Jurnal Mhs. Tek. Inform., vol. 8, no. 6, pp. 11406–11411, 2024, doi: 10.36040/jati.v8i6.11474.

This work is licensed under a **Creative Commons Attribution 4.0 International License.**