BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 5, No 4, June 2025 | Hal 584-593
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.572

Analisis Komparasi Kinerja LSTM dan CNN dalam Deteksi Spam Email Berbasis Deep learning

Maugy Al Kautsar[1], Galet Guntoro Setiaji[1], Ahmad Rifa'i[2,*]

[1] Fakultas Teknologi Informasi dan Komunikasi, Prodi Teknik Informatika, Universitas Semarang, Semarang, Indonesia
[2] Fakultas Teknologi Informasi dan Komunikasi, Prodi Sistem Informasi, Universitas Semarang, Semarang, Indonesia
Email: [1]maugyalkautsar@gmail.com, [2]gallet@usm.ac.id, [3,*]rifai@usm.ac.id
Email Penulis Korespondensi: rifai@usm.ac.id

Abstrak−*Spam Email* merupakan salah satu tantangan utama dalam komunikasi digital karena dapat dimanfaatkan untuk penyebaran informasi palsu serta aktivitas penipuan daring. Penelitian ini bertujuan untuk menganalisis serta membandingkan kinerja dua model *deep learning*, yaitu *Convolutional Neural Network* (CNN) dan *Long Short-Term Memory* (LSTM), dalam klasifikasi *email spam* berbasis teks. Dataset diperoleh dari platform Kaggle dan terdiri dari 5.572 data yang telah dilabeli sebagai *spam* dan non-*spam*. Proses praproses data meliputi tahapan pelabelan, pembersihan data, konversi huruf menjadi kecil, tokenisasi, penghapusan kata umum (*stopword*), dan stemming. Setelah diproses, data dibagi menjadi data latih dan data uji dengan rasio 70:30. Kedua model dilatih menggunakan parameter yang sama dan dievaluasi menggunakan metrik akurasi, *loss*, *confusion matrix*, dan *F1-score*. Hasil evaluasi menunjukkan bahwa model LSTM mampu mencapai akurasi sebesar 98,72% dan *loss* sebesar 0,0377. Sementara itu, model CNN menghasilkan akurasi 87,78% dengan *loss* 0,3659. Berdasarkan hasil tersebut, LSTM menunjukkan performa yang lebih unggul dalam mendeteksi *email spam* berbasis teks. Penelitian ini diharapkan dapat menjadi referensi dalam pengembangan sistem deteksi *spam* berbasis teks yang lebih efektif di masa mendatang.

Kata Kunci: Spam Email; Deep Learning; CNN; LSTM; Klasifikasi Teks

Abstract−*Spam email* remains a critical issue in digital communication due to its potential misuse in spreading false information and online fraud. This study aims to evaluate and compare the performance of two deep learning models Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) for text-based *spam email* classification. The dataset used in this study was obtained from Kaggle and contains 5,572 labeled *email* entries categorized as *spam* and non-*spam*. The preprocessing stage included labeling, cleaning, lowercasing (casefolding), tokenization, stopword removal, and stemming. The data was split into training and testing sets with a 70:30 ratio. Both models were trained using the same configuration and evaluated using accuracy, loss, confusion matrix, and F1-score metrics. The results indicate that the LSTM model achieved the highest accuracy of 98.72% with a loss value of 0.0377, outperforming the CNN model, which achieved 87.78% accuracy and a loss of 0.3659. Based on these findings, LSTM demonstrated superior performance in detecting *spam email*s using text-based input. This research is expected to serve as a reference for developing more accurate and effective *spam* detection systems in the future.

Keywords: Spam Email; Deep Learning; CNN; LSTM; Text Classification

1. PENDAHULUAN

Surat elektronik (*email*) merupakan salah satu sarana komunikasi utama dalam kehidupan modern, baik di bidang profesional maupun personal. Dengan semakin majunya teknologi digital, penggunaan *email* meningkat secara signifikan. Namun, meningkatnya penggunaan *email* juga diiringi dengan lonjakan distribusi *spam* yang dapat mengganggu kenyamanan pengguna dan menimbulkan ancaman terhadap keamanan data. Oleh karena itu, deteksi *spam* menjadi topik riset yang semakin penting, khususnya dalam bidang kecerdasan buatan dan pembelajaran mesin [1][2][3].

Penggunaan *deep learning* telah menunjukkan kemampuan yang menjanjikan untuk meningkatkan ketepatan dalam deteksi *spam email*. Oleh karena itu, diperlukan sistem yang efektif dan tepat guna untuk mendeteksi *spam* guna menjaga keamanan, serta kelancaran komunikasi digital. Metode *deep learning* menjadi penerapan yang menarik dalam mengatasi permasalahan ini, terutama melalui penerapan model seperti *Convolutional Neural Network* (CNN) dan *Long Short-Term Memory* (LSTM). CNN dikenal unggul dalam mengekstraksi fitur penting dari teks, sedangkan LSTM memiliki keunggulan dalam memahami urutan konteks dalam kalimat [4][5][6].

Walaupun sejumlah studi sebelumnya telah membandingkan algoritma pembelajaran mesin untuk analisis kinerja, studi pertama diawali tentang Perbandingan Kinerja RNN dan CNN dalam Klasifikasi Sentimen Ulasan Aplikasi Playstore melakukan klasifikasi dengan menggunakan kedua algoritma pada lima kategori data yang telah melalui tahap praproses. Hasil pengujian menunjukkan bahwa CNN secara konsisten memberikan akurasi lebih tinggi dibandingkan RNN, dengan performa terbaik ditemukan pada kategori desain CNN mencapai akurasi 85% dan loss 0,41%. Sebaliknya, kategori *streaming* menunjukkan kinerja terendah pada kedua model, di mana CNN memperoleh akurasi 69% dan RNN 67% [7].

Penelitian selanjutnya membahas Perbandingan Metode *Deep* dan *Traditional Learning* Untuk Penyaringan *Spam Email*. Berdasarkan algoritma *Machine Learning* dan *Deep learning*, hasil studi tersebut menunjukkan bahwa dengan mengumpulkan berbagai dataset dan menerapkan model *Deep learning*, tingkat akurasi deteksi meningkat secara signifikan—dari 85,46% menjadi hampir 97,52% setelah seluruh dataset, termasuk data *Email*, dimasukkan [8].

Studi berikutnya membahas perbandingan antara algoritma *Machine* dan *Deep learning* Dalam Menganalisis Sentimen Dari Teks Umpan Balik Terkait Evaluasi Pengajaran Dosen. Hasil penelitian mengungkapkan bahwa model BiLSTM terbukti paling unggul dalam hal akurasi, sementara LSTM menempati posisi selanjutnya dengan akurasi saat proses pelatihan dan pengujian masing-masing sebesar 95,64% dan 70,81% [9].

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 5, No 4, June 2025 | Hal 584-593
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.572

Kajian berikutnya membahas Model Deteksi *Spam* Berbasis Platform TensorFlow Dengan Pendekatan Algoritma *Deep learning*. Dalam penelitian tersebut, model RNN menunjukkan performa unggul dengan akurasi mencapai 98,36%. Sementara itu, hasil perbandingan menunjukkan bahwa model CNN mencatat akurasi 97,10%, dan model LSTM memperoleh akurasi sebesar 92,85%. Temuan ini mengindikasikan bahwa RNN memiliki kemampuan lebih tinggi dalam mendeteksi *spam* secara efektif menggunakan TensorFlow [10].

Studi berikutnya membahas Perbandingan Performa Algoritma CNN, LSTM, dan FNN Dalam Proses *diagnosis fibrosis* Hati Berbasis Citra Medis. Hasil analisis menunjukkan bahwa CNN memberikan performa terbaik dengan tingkat akurasi mencapai 98%, disusul oleh LSTM dengan 97%, dan FNN dengan 80%. Keunggulan CNN terletak pada kemampuannya mengekstraksi fitur spasial dari citra medis secara otomatis dan berjenjang, menjadikannya sangat efektif dalam tugas klasifikasi gambar. Di sisi lain, meskipun LSTM handal dalam mengolah data berurutan, algoritma ini kurang optimal dalam menangani data citra yang bersifat spasial. Sementara itu, FNN menunjukkan hasil yang kurang memuaskan pada dataset yang digunakan [11].

Berdasarkan tinjauan tersebut, dapat disimpulkan walaupun CNN dan LSTM telah diterapkan dalam banyak penelitian sebelumnya, masih terdapat celah riset khususnya kurangnya studi yang secara eksplisit membandingkan performa keduanya dalam konteks deteksi *spam email*. Hal ini terutama menyangkut penggunaan data terbaru dan pengujian faktor-faktor seperti kecepatan proses pelatihan, ketepatan prediksi, serta kemampuan model dalam melakukan generalisasi. Selain itu, penelitian yang secara konsisten mengevaluasi kedua arsitektur ini dengan pendekatan eksperimental dalam domain klasifikasi *spam* berbasis *email* masih relatif sedikit.

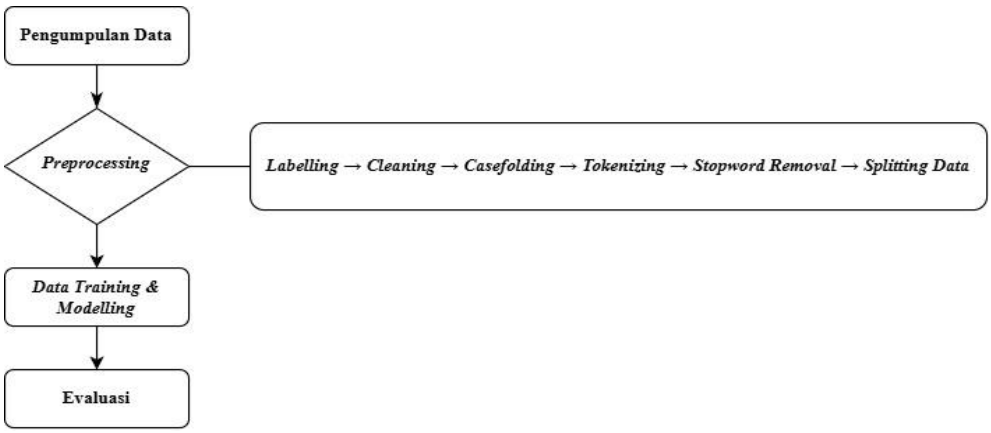
Tujuan penelitian ini adalah mengevaluasi dan membandingkan performa model *Convolutional Neural Network* (CNN) dan *Long Short-Term Memory* (LSTM) dalam mengklasifikasikan *spam email* berbasis teks. Selama penelitian, digunakan berbagai metode *Natural Language Processing* (NLP) untuk menyiapkan data dengan cara yang paling optimal. Melalui pendekatan ini, diharapkan dapat diperoleh gambaran yang lebih jelas mengenai kinerja masing-masing model *deep learning* dalam melakukan deteksi *spam* secara otomatis [12][13].

Hasil dari penelitian ini diharapkan dapat menjadi salah satu referensi dalam merancang sistem deteksi spam yang lebih optimal dan akurat, sekaligus sebagai acuan bagi peneliti lain dalam memilih model yang sesuai untuk pengolahan data *email* berbasis teks. Dengan cara khusus, studi perbandingan mengenai kinerja model CNN dan LSTM, serta penentuan faktor-faktor yang berpengaruh terhadap efisiensi tiap arsitektur, diharapkan bisa menjadi referensi berharga bagi para peneliti dan pengembang sistem keamanan digital. Penelitian ini juga diharapkan dapat menawarkan petunjuk praktis dalam memilih, mengoptimalkan, dan menerapkan algoritma yang sesuai dalam sistem klasifikasi *spam*, baik untuk ukuran kecil maupun besar, serta untuk berbagai macam konten *email* yang berkembang.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Dari penelitian ini digunakan pendekatan eksperimental dengan melakukan pengujian terhadap beberapa model *deep learning*. Tujuan pengujian adalah untuk membandingkan hasil performa masing-masing model berdasarkan literatur yang telah dikaji sebelumnya. Untuk memberikan gambaran umum mengenai tahapan yang dilakukan dalam penelitian ini, berikut disajikan alur proses penelitian pada Gambar 1.



Gambar 1. Alur Penelitian

Susunan diagram proses penelitian ditunjukkan pada Gambar 1. yang mencakup beberapa tahapan, yaitu :

2.1.1 Pengumpulan Data

Dataset yang digunakan dalam penelitian ini diambil dari platform Kaggle, yang berisi kumpulan data *email* yang telah dikelompokkan ke dalam dua kategori, yaitu *spam* dan *non-spam* (ham). Data ini digunakan untuk menganalisis kinerja model LSTM dan CNN dalam melakukan klasifikasi *email spam* dengan pendekatan *deep learning*. Dataset terdiri atas

dua kolom utama: kolom pertama berisi teks dari *email* yang dianalisis, sedangkan kolom kedua menunjukkan label klasifikasinya, yaitu *spam* (label 1) dan *ham* atau *non-spam* (label 0). Total data yang dipakai dalam penelitian ini sebanyak 5.572 baris, dengan konten *email* yang bervariasi, mulai dari promosi, undangan, hingga komunikasi sehari-hari.

2.1.2 *Preprocessing*

Langkah *preprocessing* bertujuan untuk menyiapkan data teks sebelum digunakan dalam proses pelatihan model *deep learning*. Dalam penelitian ini, proses tersebut dilakukan dengan menerapkan berbagai teknik dari *Natural Language Processing* (NLP) untuk membersihkan dan menormalisasikan teks *email* [14]. Tahapan yang dilakukan meliputi:

a. *Labeling*
   Langkah pertama adalah *labeling*, yaitu proses menambahkan label pada data untuk mengklasifikasikan *email* ke dalam kategori *spam* atau *non-spam*. Pemberian label ini penting agar model dapat mempelajari pola secara terarah berdasarkan data yang telah diberi kategori [15].
b. *Cleaning*
   Pada tahap ini, teks dibersihkan dari karakter atau elemen yang tidak relevan dengan analisis seperti angka, simbol, atau tanda baca, dan URL [16].
c. *Casefolding*
   Langkah ini mengganti semua huruf dalam teks menjadi format huruf kecil untuk mencegah penggadaan data akibat perbedaan kapitalisasi [17] .
d. *Tokenizing*
   Tahap *tokenizing* adalah proses pemisahan kalimat atau paragraf menjadi per kata atau token, sehingga setiap kata dapat dianalisis sebagai satuan data tersendiri [18].
e. *Stopword Removal*
   Pada tahap ini, kata-kata umum yang sering muncul namun memiliki makna lemah seperti “dan”, “yang”, atau “adalah” akan dihapus agar fokus model tertuju pada kata-kata yang lebih bermakna [19].
f. *Stemming*
   Stemming adalah proses mengubah kata ke bentuk dasarnya. Misalnya, kata “mengirimkan” dan “dikirim” diubah menjadi bentuk dasar “kirim”, sehingga model lebih mudah mengenali kata yang memiliki makna serupa [20].
g. *Splitting Data*
   Tahap akhir adalah memisahkan dataset menjadi dua bagian, yaitu data latih (*training data*) dan data uji (*testing data*). Pemisahan ini bertujuan untuk melatih model pada sebagian data, lalu menguji performanya pada data yang belum dilihat sebelumnya [21].

2.1.3 *Data Training*

Setelah melewati tahap *preprocessing*, data yang telah dibersihkan dan dinormalisasi dibagi menjadi dua bagian: data latih dan data uji. Selanjutnya, data latih digunakan dalam proses pelatihan (*training*) model *deep learning*. Proses ini bertujuan agar model dapat memahami pola serta karakteristik teks yang tersedia.

2.1.4 *Modelling*

Modelling adalah tahap inti dalam pengembangan sistem klasifikasi berbasis *deep learning*. Setelah pelatihan selesai, proses ini dilanjutkan dengan merancang dan menerapkan arsitektur model yang sesuai untuk tugas klasifikasi yang diinginkan.

2.1.5 Evaluasi

Tahapan evaluasi bertujuan untuk mengkaji dan membandingkan performa model, seperti CNN dan LSTM, dalam mengklasifikasikan *spam email*. Evaluasi dilakukan menggunakan metrik seperti tingkat akurasi dan nilai *loss* untuk menilai sejauh mana model mampu membedakan antara *spam* dan *non-spam*. Hasil evaluasi ini digunakan untuk mengidentifikasi model mana yang paling efektif, dan dapat menjadi acuan dalam memilih arsitektur terbaik untuk sistem deteksi *spam* berbasis *deep learning*. Dengan pendekatan ini, peneliti dapat memperoleh pemahaman menyeluruh tentang kemampuan masing-masing model dalam mengelola data *email* secara otomatis.

2.2 Metode Modelling

Dalam proses pemodelan sistem, digunakan dua algoritma utama, yaitu CNN (*Convolutional Neural Network*) dan LSTM (*Long Short-Term Memory*) yang sering digunakan dalam pemrosesan data berbasis teks.

a. *Convolutional Neural Network* (CNN)
   CNN merupakan tipe jaringan saraf buatan yang umumnya diterapkan untuk pengolahan data berdimensi, seperti gambar/foto. Akan tetapi, dalam penerapan pemrosesan teks, CNN bisa digunakan untuk mengidentifikasi fitur-fitur signifikan dari data teks melalui metode konvolusi dan *pooling*. CNN memiliki kemampuan untuk menangkap pola-pola lokal dalam teks seperti urutan dari kata-kata atau ungkapan yang sering muncul, yang sangat berguna untuk tugas klasifikasi atau analisis sentimen [22]. Dalam kasus teks, setiap kata diubah menjadi vektor melalui *embedding layer*, lalu CNN mengekstrak fitur *n-gram* dari vektor tersebut untuk menangkap pola-pola penting dalam teks.

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 5, No 4, June 2025 | Hal 584-593
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.572

Rumus dasar:
Operasi *Convolution*

$$h_i = f(w \cdot x_{i:i+k} - 1 + b) \quad (1)$$

$x_{i:i+k}$ = potongan urutan embedding dari kata ke-i sampai ke-(i+k-1)
w = bobot filter
b = fungsi aktivasi (umumnya ReLU)
h_i = hasil fitur pada posisi ke-i
Max Pooling

$$\tilde{h} = max\ (h_1, h_2, \dots, h_n) \quad (2)$$

Digunakan untuk memilih fitur paling dominan dari hasil *convolution*

b. *Long Short-Term Memory* (LSTM)
LSTM merupakan pengembangan dari arsitektur *Recurrent Neural Network* (RNN) yang dirancang untuk mengolah data berurutan seperti teks atau suara. Keunggulan utama LSTM adalah kemampuannya dalam mempertahankan informasi penting dalam jangka waktu yang panjang, sehingga dapat menghindari masalah hilangnya informasi (*vanishing gradient*) yang umum terjadi pada RNN konvensional. Ketika digunakan untuk memproses teks, LSTM mampu menangkap hubungan antar kata dalam kalimat secara lebih mendalam karena memperhatikan urutan kemunculan kata dalam konteks [23]. LSTM memiliki struktur unit memori dengan tiga gerbang utama yaitu *input gate* untuk mengatur informasi baru yang akan disimpan, *forget gate* untuk menentukan informasi mana yang harus dihapus dari memori, dan *output gate* untuk menghasilkan keluaran berdasarkan kondisi memori saat ini. Rumus-rumus dasar dari masing-masing gerbang ditunjukkan sebagai berikut:
Forget Gate

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (3)$$

Input Gate

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (4)$$

$$\tilde{C}_t = \tan \mathrm{h}(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (5)$$

Update Cell State

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (6)$$

Output Gate

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o \quad (7)$$

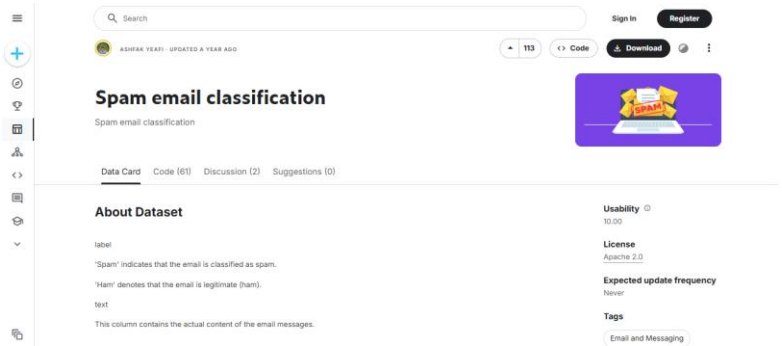
$$h_t = o_t * \tanh(C_t) \quad (8)$$

Keterangan : x_t : input pada waktu ke-t, h_t : *hiddent state* saat ini, C_t: *cell state* saat ini, σ : *sigmoid function*, dan tanda *: perkalian elemen (*element wise*)

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Data

Dataset yang digunakan dalam penelitian ini diperoleh dari Kaggle dengan pencarian menggunakan kata kunci "*Spam Email*". Pengambilan serta pemrosesan awal data dilakukan dengan menggunakan bahasa pemrograman *Python*. Dataset pada Gambar 2. terdiri dari 5.572 entri yang terbagi dalam dua kolom utama, yaitu *Text* yang berisi isi pesan dalam bentuk teks bebas, dan *Label* yang menunjukkan kategori *email*, apakah termasuk *spam* atau *non-spam* (ham).



Gambar 2. Dataset *Spam Email Classification*

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 5, No 4, June 2025 | Hal 584-593
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.572

Dataset dipilih karena memiliki struktur yang sesuai untuk klasifikasi teks, dan telah banyak digunakan dalam studi sebelumnya. Data mencakup berbagai tipe *email* seperti iklan, promosi, penawaran, hingga pesan personal. Sebelum data digunakan dalam proses pelatihan model, dilakukan eksplorasi awal dan pembersihan data untuk memastikan formatnya konsisten dan layak untuk analisis. Lalu data tersebut disimpan dalam format .csv, dengan tiga kolom: ID, Text, dan Label, yang kemudian digunakan dalam pelatihan dan pengujian model CNN dan LSTM. Ringkasan isi data dapat dilihat pada Tabel 1. berikut.

Tabel 1. Dataset *Email Classification*

No	Category	Message
0	ham	*Go until jurong point, crazy.. Available only ...*
1	ham	*Ok lar... Joking wif u oni..*
2	*spam*	*Free entry in 2 a wkly comp to win FA Cup fina...*
….	…..	…..
5569	ham	*Pity, * was in mood for that. So...any other s...*
5570	ham	*The guy did some bitching but I acted like i'd...*
5571	ham	*Rofl. Its true to its name*

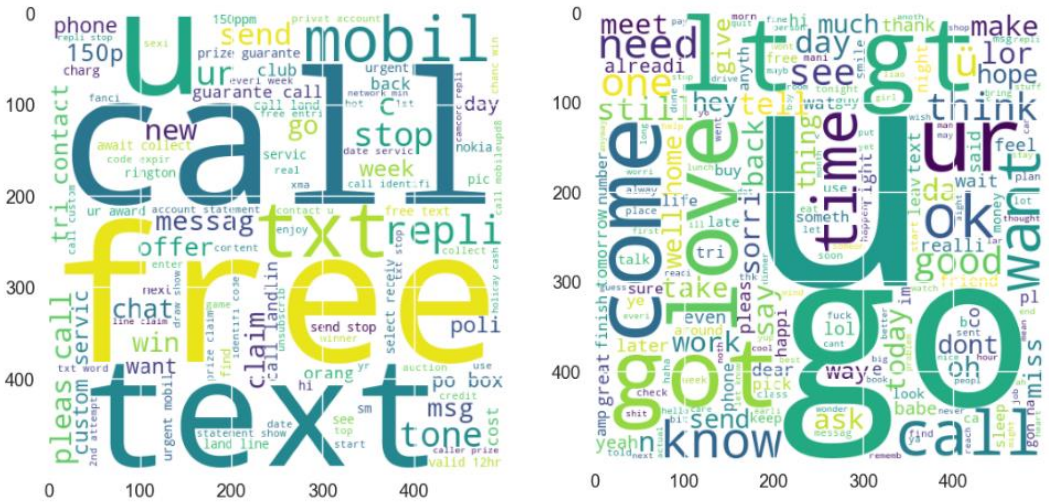
3.2 *Preprocessing*

Pada tahap *preprocessing*, teks melalui proses persiapan sebelum dilatih dalam model *deep learning*. Langkah-langkah ini meliputi pembersihan data, transformasi teks, serta penandaan label, dengan tujuan memastikan bahwa data yang digunakan sudah bersih dan terstruktur dengan baik. Salah satu proses penting adalah labeling, di mana setiap *email* diberi kategori: *spam* (label 1) atau non-*spam*/ham (label 0). Pelabelan ini menjadi dasar untuk proses pelatihan dalam sistem klasifikasi berbasis supervised learning. Contoh data awal dan hasil pelabelan ditunjukkan pada Tabel 2.

Tabel 2. *Labeling*

Text	Label	Target
Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C'sapply 08452810075over18's	*spam*	1
"Nah I don't think he goes to usf, he lives around here though"	*ham*	0

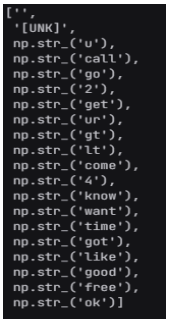
Langkah berikutnya adalah *cleaning*, yaitu membersihkan teks dari elemen yang tidak relevan, seperti simbol, angka, tanda baca, dan URL. Proses ini mengubah seluruh kata atau teks menjadi huruf kecil guna menyamakan format dan menghindari duplikasi. Setelah teks dibersihkan, dilakukan visualisasi dengan *WordCloud* untuk melihat kata per kata yang sering muncul. Visualisasi ini membantu memahami konten umum dari dataset dan mengenali kata kunci yang dominan dalam *email spam* maupun *non-spam* (lihat Gambar 3).



Gambar 3. *WordCloud* Hasil Teks *spam email*

Setelah tahap pembersihan, dilakukan *casefolding* untuk mengonversi semua huruf menjadi huruf kecil. Hal ini menghindari perbedaan interpretasi kata akibat kapitalisasi. Selanjutnya dilakukan tokenisasi, yaitu memecah teks menjadi unit-unit kata (*tokens*) yang dapat diproses lebih lanjut oleh model. Proses ini dilakukan menggunakan pustaka NLTK, dan hasil tokenisasi nantinya akan digunakan dalam tahap klasifikasi. Setiap kalimat dalam data teks diubah menjadi daftar kata (*tokens*) seperti pada Gambar 3. hasil dari bagian Tokenisasi

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 5, No 4, June 2025 | Hal 584-593
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.572



Gambar 4. Tokenisasi Kata

Setelah teks dipecah menjadi token, langkah berikutnya adalah *stopword removal*, yakni menghapus kata-kata umum yang tidak memberi makna signifikan terhadap klasifikasi, seperti "dan", "yang", atau "adalah". Dengan menghilangkan *stopword*, sistem dapat fokus pada kata-kata bermakna. Langkah selanjutnya adalah *stemming*, yaitu mengubah kata ke bentuk dasarnya, misalnya "mengirimkan" menjadi "kirim". Tabel 3 menunjukkan contoh perubahan teks setelah proses *cleaning*, *casefolding*, *stopword removal*, dan *stemming*.

Tabel 3. Data setelah proses *Cleaning, Casefolding, Stopword Removal, Stemming*

Text	Transformed_text
Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C'sapply 08452810075over18's	free entri 2 wkli comp win fa cup final tkt 21...
"Nah I don't think he goes to usf, he lives around here though"	nah think goe usf live around though
I HAVE A DATE ON SUNDAY WITH WILL!!	date sunday

Tahap akhir dari pra-pemrosesan adalah membagi data menjadi dua bagian: data latih dan data uji. Seperti pada Tabel 4. pembagian ini bertujuan agar model dilatih dengan sebagian data, lalu diuji performanya pada data yang belum pernah dilihat sebelumnya. Rasio pembagian umumnya adalah 80% untuk pelatihan dan 20% untuk pengujian. Proses ini membantu mengukur kemampuan model dalam menggeneralisasi data baru yang belum dikenali sebelumnya.

Tabel 4. *Splitting Data*

Dataset	Jumlah Data
Training Set	3.610
Testing Set	1.548

3.3 Data Training dan Modelling

Tahap ini melibatkan proses pelatihan dan pemodelan data. Data *email* yang telah dibersihkan kemudian dibagi menjadi dua bagian, yaitu data pelatihan (*training data*) dan data pengujian (*testing data*). Data pelatihan digunakan untuk melatih model *deep learning* agar dapat mengenali pola yang membedakan *email spam* dan *non-spam* berdasarkan isi teks. Penelitian ini menggunakan dua jenis model *deep learning*, yaitu *Convolutional Neural Network* (CNN) dan *Long Short-Term Memory* (LSTM), yang dibandingkan berdasarkan efektivitasnya dalam mengklasifikasikan *email spam*. Model LSTM dirancang khusus untuk menangani data berurutan seperti teks. Arsitekturnya dimulai dengan lapisan *Embedding* untuk mengubah setiap kata menjadi vektor numerik berdimensi 32. Kemudian, data tersebut diproses oleh lapisan *Bidirectional LSTM* yang memungkinkan pemahaman konteks dari dua arah (depan dan belakang). Setelah itu, data diteruskan ke lapisan *Dense* berisi 32 unit *neuron* untuk menyusun representasi fitur yang lebih kompleks, dan dilanjutkan dengan *Dropout* untuk mencegah *overfitting* dengan menonaktifkan sebagian *neuron* selama pelatihan. Bagian akhir dari model ini adalah lapisan *Dense output* dengan satu *neuron* yang menggunakan fungsi aktivasi *sigmoid* untuk menghasilkan output berupa nilai klasifikasi *biner* (1 untuk *spam*, 0 untuk non-*spam*). Model dikompilasi dengan fungsi *loss*, *optimizer*, dan metrik akurasi sebagai acuan evaluasi kinerjanya. Struktur lengkap model LSTM ditunjukkan pada Tabel 5.

Tabel 5. Hasil Pembuatan Model LSTM

Layer (Type)	Output Shape	Param
embedding (*Enbedding)*	(*None, 100*, 32)	16000
bidirectional (Bidirectional)	(*None,* 128)	49664
dense (*Dense*)	(*None*, 32)	4128
dropout (*Dropout*)	(*None*, 32)	0
*Dense*_1 (*Dense*)	(*None*, 1)	33
Total params: 69, 825		

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 5, No 4, June 2025 | Hal 584-593
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.572

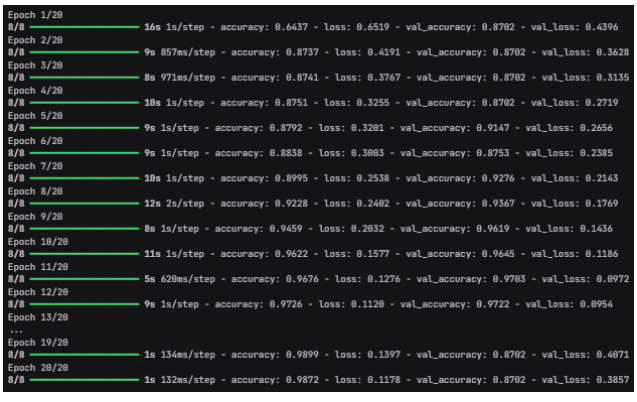
Layer (Type)	Output Shape	Param
Trainable params: 69, 825		
Non-trainable params: 0		

Model *Convolutional Neural Network* (CNN) yang digunakan dalam penelitian ini terdiri atas beberapa lapisan inti yang dirancang untuk mengekstraksi fitur penting dari teks dan mengelompokkan *email* ke dalam kategori *spam* atau *non-spam*. Proses diawali dengan lapisan *Embedding*, yang mengubah setiap kata dalam teks menjadi vektor angka berdimensi 32. Vektor ini kemudian diproses melalui lapisan *Convolutional 1D (Conv1D)* dengan 32 *filter*, yang berfungsi menangkap pola lokal dari urutan kata dalam teks, terutama pola yang sering muncul dalam *email spam*. Setelah itu, hasil konvolusi dikirim ke lapisan *MaxPooling1D* untuk menyaring informasi yang paling relevan. Untuk mencegah terjadinya *overfitting*, digunakan lapisan *Dropout*, yang diikuti oleh *Batch Normalization* guna menstabilkan proses pelatihan dan mempercepat proses pelatihan model. Selanjutnya, data dilewatkan melalui lapisan *Flatten* agar dapat diubah menjadi bentuk vektor satu dimensi sebelum diteruskan ke lapisan-lapisan *Dense*. Model CNN ini memiliki dua lapisan *Dense* berukuran 32 *neuron*, yang salah satunya disertai dengan lapisan *Dropout* tambahan sebagai langkah pencegahan *overfitting*. Lapisan terakhir adalah *Dense output* dengan satu *neuron* yang dilengkapi fungsi aktivasi *sigmoid*, yang menghasilkan *output* berupa klasifikasi *biner* (1 untuk *spam*, 0 untuk *non-spam*). Total parameter yang digunakan dalam model ini adalah 71.521, dan sekitar 71.45% di antaranya dapat dilatih (*trainable*). Model ini kemudian dikompilasi dengan fungsi *loss*, *optimizer*, dan metrik akurasi untuk mengevaluasi performanya selama pelatihan. Rincian lengkap arsitektur model dapat dilihat pada Tabel 6.

Tabel 6. Hasil Pembuatan Model CNN

Layer (Type)	Output Shape	Param
embedding_1 (Embedding)	(None, 100, 32)	16,000
conv1d (Conv1D)	(None, 100, 32)	4,128
max_pooling1d (MaxPooling1D)	(None, 50, 32)	0
dropout_1 (Dropout)	(None, 50, 32)	0
batch_normalization (BatchNormalization)	(None, 50, 32)	128
flatten (Flatten)	(None, 1600)	0
dense_2 (Dense)	(None, 32)	51,232
dropout_2 (Dropout)	(None, 32)	0
dense_3 (Dense)	(None, 1)	33
Total params: 71,521		
Trainable params: 71,457		
Non-trainable params: 64		

Pelatihan model merupakan tahap di mana arsitektur jaringan saraf yang telah dirancang mulai dilatih dengan data untuk mengenali pola yang terdapat dalam dataset. Tahapan ini mencakup beberapa langkah krusial, seperti membagi data menjadi data pelatihan dan validasi, menyusun arsitektur model, menjalankan proses pelatihan dalam beberapa iterasi (*epoch*), serta mengevaluasi kinerjanya selama proses tersebut berlangsung. Dalam penelitian ini, model dilatih selama 20 *epoch* dengan *batch size* sebesar 512, menggunakan data yang telah melalui tahapan *preprocessing*. Sepanjang proses pelatihan, model secara bertahap menyesuaikan bobot-bobotnya untuk mengurangi nilai *loss* dan meningkatkan nilai *accuracy*. Evaluasi dilakukan secara berkala di akhir setiap *epoch* menggunakan data validasi, dengan tujuan untuk mengukur kemampuan model dalam menghadapi data baru yang belum pernah dilihat sebelumnya (*generalization*). Beberapa metrik evaluasi yang digunakan meliputi akurasi pada data pelatihan (*accuracy*), akurasi pada data validasi (*val_accuracy*), serta nilai *loss* untuk kedua data tersebut (*loss* dan *val_loss*). Hasil pelatihan menunjukkan bahwa akurasi, baik pada data latih maupun validasi, mengalami peningkatan yang cukup stabil sepanjang proses, sementara nilai *loss* cenderung menurun. Visualisasi hasil pelatihan dan evaluasi model secara menyeluruh ditampilkan pada Gambar 7.



Gambar 5. Proses melatih model

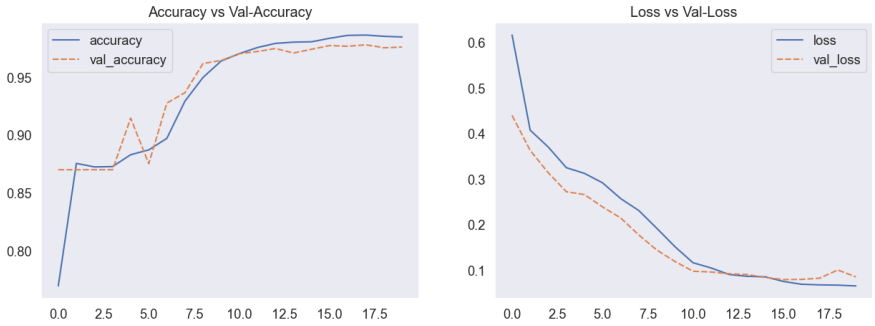
BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 5, No 4, June 2025 | Hal 584-593
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.572

3.4 Evaluasi

Setelah tahap pelatihan selesai, kedua model yaitu *Long Short-Term Memory* (LSTM) dan *Convolutional Neural Network* (CNN) diuji menggunakan dataset pengujian (*X_test* dan *y_test*). Tujuan pengujian ini adalah untuk menilai sejauh mana model dapat mengenali pola pada data yang sama sekali baru dan tidak termasuk dalam data pelatihan. Hasil pengujian menunjukkan bahwa model LSTM mencapai akurasi 98,72% dengan nilai *loss* 0,0377, mengindikasikan kinerja yang sangat optimal dalam mengidentifikasi *email spam*. Di sisi lain, model CNN memperoleh akurasi 87,78% disertai *loss* 0,3659.

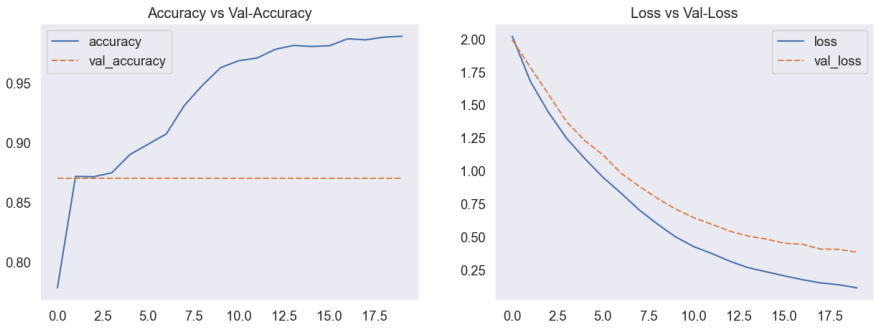
Tabel 7. Hasil Akurasi kedua Model

Model	Akurasi (%)	*Loss*
LSTM	98,72	0,0377
CNN	87,78	0,3659



Gambar 6. Grafik Akurasi dan *Loss* pada Model LSTM Selama Proses Pelatihan

Selain menyajikan hasil evaluasi akhir berupa angka akurasi dan nilai *loss*, performa model juga divisualisasikan melalui grafik. Gambar 8 memperlihatkan perbandingan antara akurasi pelatihan dan validasi, serta nilai *loss* pada model LSTM selama 20 *epoch* pelatihan. Pada grafik di sisi kiri, terlihat bahwa akurasi pada data latih dan validasi meningkat secara stabil seiring bertambahnya *epoch*, dengan akurasi tertinggi mencapai 0,98. Di sisi lain, grafik *loss* di sebelah kanan menunjukkan tren penurunan tajam, yang mengindikasikan bahwa kesalahan prediksi model berkurang secara konsisten selama pelatihan.

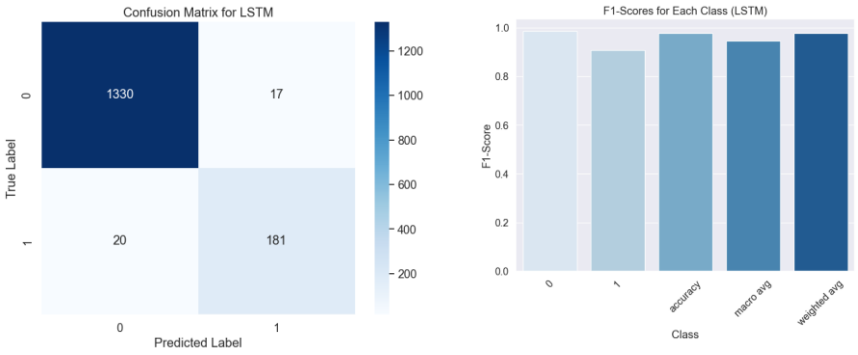


Gambar 7. Grafik Akurasi dan *Loss* pada Model CNN Selama Proses Pelatihan

Sementara itu, Gambar 9 menampilkan performa model CNN selama 20 *epoch* pelatihan. Grafik kiri memperlihatkan perkembangan akurasi pada data pelatihan dan validasi, sementara grafik kanan menunjukkan perubahan nilai *loss* untuk keduanya. Dari grafik kiri, dapat dilihat bahwa akurasi data latih meningkat signifikan sejak awal pelatihan hingga mencapai sekitar 0,98. Namun, akurasi pada data validasi hanya mencapai kisaran 0,87 dan tidak menunjukkan tren peningkatan yang berarti. Hal ini menunjukkan bahwa meskipun model cukup berhasil meningkatkan akurasi pada data pelatihan, performanya pada data validasi tidak mengalami perbaikan yang serupa, yang dapat mengindikasikan potensi *overfitting*. Pada grafik *loss*, penurunan nilai terlihat cukup stabil baik pada data latih maupun validasi, tetapi tetap menunjukkan bahwa model tidak mampu menjaga kinerja yang konsisten terhadap data baru. Kondisi ini mengisyaratkan bahwa meskipun model dapat menyesuaikan diri dengan baik terhadap data pelatihan, kemampuannya dalam menggeneralisasi belum optimal saat diterapkan pada data validasi. Selain menggunakan metrik akurasi dan *loss*, evaluasi juga dilakukan dengan *confusion matrix* untuk mengenali jumlah prediksi benar dan salah pada masing-masing kelas. Di

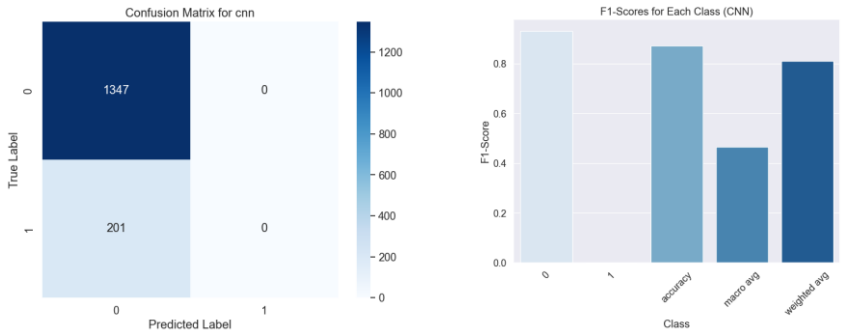
BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 5, No 4, June 2025 | Hal 584-593
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.572

samping itu, digunakan juga metrik *F1-Score* untuk menilai keseimbangan antara nilai presisi dan *recall* dalam hasil klasifikasi.



Gambar 8. *Confusion Matrix* Model LSTM pada Data Uji dan Grafik F1-Score per Kelas

Gambar 10 memperlihatkan hasil *confusion matrix* dari model LSTM. Dari total 1.548 data uji, model mampu mengklasifikasikan 1.330 data sebagai *non-spam* (label 0) dan 181 data sebagai *spam* (label 1) secara benar. Namun, terdapat kesalahan klasifikasi sebanyak 17 data *non-spam* yang diklasifikasikan sebagai *spam*, serta 20 data *spam* yang diklasifikasikan sebagai *non-spam*. Hasil ini mengindikasikan bahwa LSTM masih memiliki kelemahan dalam membedakan kedua kategori dengan tepat. Evaluasi lebih lanjut menggunakan metrik *F1-Score* dilakukan untuk masing-masing kelas, termasuk perhitungan nilai *precision*, *recall*, *macro average*, dan *weighted average*. Nilai *F1-Score* yang diperoleh, baik untuk kategori *spam* maupun *non-spam*, menunjukkan bahwa model tidak hanya memiliki tingkat akurasi yang baik, tetapi juga seimbang dalam menangani klasifikasi kedua kelas tersebut. Tingginya nilai *F1-Score* pada kedua kelas tersebut menjadi indikasi bahwa model LSTM mampu menangani tantangan klasifikasi biner secara efektif, terutama untuk data teks dengan struktur berurutan seperti *email*.



Gambar 9. Confusion Matrix Model CNN pada Data Uji dan Grafik F1-Score per Kelas

Sebagai perbandingan, Gambar 11. memperlihatkan confusion matrix dari model CNN. Berdasarkan hasil evaluasi, model CNN berhasil mengklasifikasikan seluruh data *non-spam* (sebanyak 1.347) dengan benar, namun gagal mengklasifikasikan semua data *spam*, terlihat dari 201 data *spam* yang diklasifikasikan sebagai *non-spam*. Hal ini menunjukkan bahwa model hanya mengenali satu kelas (*non-spam*), dan tidak berhasil mendeteksi kelas *spam* sama sekali, yang merupakan indikasi underfitting atau bias kelas. Evaluasi lebih lanjut ditunjukkan menggambarkan F1-Score pada setiap kelas. Terlihat bahwa F1-Score untuk kelas *non-spam* (label 0) cukup tinggi, namun F1-Score untuk kelas *spam* (label 1) bernilai 0 karena tidak ada prediksi benar terhadap kelas tersebut. Akibatnya, nilai rata-rata F1 (macro average) jauh lebih rendah dibandingkan model LSTM, meskipun nilai akurasi keseluruhan tampak cukup tinggi (sekitar 87%). Hal ini menjadi bukti bahwa akurasi tidak selalu mencerminkan performa model yang baik, terutama pada data yang tidak seimbang. Dengan tidak berhasilnya CNN mengenali *spam*, model menjadi tidak efektif untuk diterapkan dalam skenario deteksi *spam* yang nyata.

4. KESIMPULAN

Berdasarkan hasil pengujian dan evaluasi yang telah dilakukan, dapat disimpulkan bahwa model *Long Short-Term Memory* (LSTM) memiliki performa yang lebih baik dibanding *Convolutional Neural Network* (CNN) dalam mengklasifikasikan *email spam* berbasis teks. Hasil pengujian menunjukkan bahwa LSTM berhasil mencapai akurasi tertinggi sebesar 98,72% dan nilai *loss* yang paling rendah sebesar 0,0377. Sementara itu, CNN hanya mampu memperoleh akurasi sebesar 87,78% dengan nilai *loss* 0,3659. Perbedaan ini menunjukkan bahwa LSTM lebih efisien dalam mengenali pola urutan kata yang sering muncul dalam *email spam*, sehingga lebih adaptif terhadap struktur kalimat dalam bahasa alami. Evaluasi menggunakan *confusion matrix* dan *F1-score* juga mendukung keunggulan LSTM, karena model ini menunjukkan distribusi prediksi yang lebih seimbang antara kelas *spam* dan non-*spam*. Sebaliknya, CNN

cenderung hanya mampu mengenali *email* non-*spam* dengan baik dan kurang akurat dalam mendeteksi *email spam*. Hal ini menandakan bahwa CNN kurang mampu menangkap konteks sekuensial yang penting dalam teks *email*. Dengan demikian, LSTM dinilai lebih cocok untuk digunakan sebagai dasar dalam pengembangan sistem klasifikasi *email spam* berbasis teks. Hasil Penelitian ini diharapkan menjadi kontribusi yang bermanfaat bagi pengembangan sistem deteksi *spam* yang lebih efisien dan akurat di masa depan, serta dapat dijadikan pedoman dalam memilih model klasifikasi yang tepat sesuai dengan karakteristik data teks yang digunakan.

REFERENCES

[1] A. Muhaimin, I. A. Taufik, and D. D. Daniswara, "Pendeteksian *Spam* pada E-mail menggunakan Pendekatan Natural Language Processing," *Pros. Semin. Nas. Sains Data*, vol. 3, no. 1, pp. 116–121, 2023, doi: 10.33005/senada.v3i1.90.

[2] A. N. Salim, A. Adryani, and T. Sutabri, "Deteksi *Email Spam* dan *Non-spam* Berdasarkan Isi Konten Menggunakan Metode K-Nearest Neighbor dan Support Vector Machine," *Syntax Idea*, vol. 6, no. 2, pp. 991–1001, 2024, doi: 10.46799/syntax-idea.v6i2.3052.

[3] C. M. Bachri and W. Gunawan, "Deteksi *Email Spam* menggunakan Algoritma Convolutional Neural Network (CNN)," *Edukasi dan Penelit. Inform.*, vol. 10, no. 1, pp. 88–94, 2024.

[4] I. AbdulNabi and Q. Yaseen, "*Spam email* detection using *deep learning* techniques," *Procedia Computer Science*, vol. 184, no. January 2021, pp. 853–858, 2021, doi: 10.1016/j.procs.2021.03.107.

[5] E. al. Ratnam Dodda, "Precision in Classification: A Comparative Study of Logistic Regression, Naive Bayes, LSTM, and CNN for *Spam Email* Detection," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 11, no. 9, pp. 2276–2280, 2023, doi: 10.17762/ijritcc.v11i9.9233.

[6] R. Approaches and G. Airlangga, "A Comparative Analysis of *Deep learning* Models for SMS *Spam* Detection : CNN-LSTM, CNN-GRU, and ResNet Approaches," *Journal of Computer Networks , Architecture and High Performance Computing*, vol. 6, no. 4, pp. 1952–1960, 2024.

[7] S. N. Saputra, G. G. Setiaji, and M. T. A. C. Widiyanto, "Perbandingan Kinerja RNN dan CNN Dalam Klasifikasi Sentimen Ulasan Pengguna Aplikasi di Play Store," *Journal of Computer System and Informatics (JoSYC)*, vol. 6, no. 1, pp. 349–362, 2024, doi: 10.47065/josyc.v6i1.6408.

[8] A. Sheneamer, "Comparison of *Deep learning* and Traditional Methods for *Email Spam* Filtering," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 1, pp. 560–565, 2021, doi: 10.14569/IJACSA.2021.0120164.

[9] H. Setiawan and D. Ariatmanto, "Analisis Perbandingan Algoritma Machine Learning Dan *Deep learning* Untuk Sentimen Analisis Teks Umpan Balik Tentang Evaluasi Pengajaran Dosen," *JSAI (Journal Sci. Appl. Informatics)*, vol. 7, no. 2, pp. 379–385, 2024, doi: 10.36085/jsai.v7i2.6572.

[10] S. Govindan, A. F. A. Abidin, M. A. Mohamed, S. D. Mohd Satar, M. F. Abdul Kadir, and N. Abd Hamid, "*Spam* Detection Model Using Tensorflow and *Deep learning* Algorithm," *Malaysian J. Comput. Appl. Math.*, vol. 6, no. 2, pp. 11–21, 2023, doi: 10.37231/myjcam.2023.6.2.84.

[11] B. A. Febryanto and I. Tahyudin, "Perbandingan Algoritma CNN, LSTM, FNN untuk Diagnosa Fibrosis Hati dengan Citra Medis," *Techno.Com*, vol. 24, no. 1, pp. 41–55, 2025.

[12] K. Thakur, M. L. Ali, M. A. Obaidat, and A. Kamruzzaman, "A Systematic Review on Deep-Learning-Based Phishing *Email* Detection," *Electron.*, vol. 12, no. 21, pp. 1–26, 2023, doi: 10.3390/electronics12214545.

[13] S. Atawneh and H. Aljehani, "Phishing *Email* Detection Model Using *Deep learning*," *Electron.*, vol. 12, no. 20, 2023, doi: 10.3390/electronics12204261.

[14] MR Adepu Rajesh and Dr Tryambak Hiwarkar, "Exploring Preprocessing Techniques for Natural LanguageText: A Comprehensive Study Using Python Code," *Int. J. Eng. Technol. Manag. Sci.*, vol. 7, no. 5, pp. 390–399, 2023, doi: 10.46647/ijetms.2023.v07i05.047.

[15] F. Jáñez-Martino, R. Alaiz-Rodríguez, V. González-Castro, E. Fidalgo, and E. Alegre, "Classifying *spam email*s using agglomerative hierarchical clustering and a topic-based approach," *Appl. Soft Comput.*, vol. 139, p. 110226, 2023, doi: 10.1016/j.asoc.2023.110226.

[16] N. C. Dewi and A. Qoiriah, "Implementasi Algoritma Jaro-Winkler Distance dan N-Gram untuk Deteksi dan Prediksi Perbaikan Kesalahan Penulisan Kata Bahasa Indonesia pada Karya Tulis Ilmiah Mahasiswa," *J. Informatics Comput. Sci.*, vol. 2, no. 03, pp. 169–177, 2021, doi: 10.26740/jinacs.v2n03.p169-177.

[17] Y. A. Risqi and I. Susilawati, "Pemanfaatan Artificial Intelligence dan Cognitive Behavioral Therapy Untuk Pengembangan Chatbot Pembelajaran Matematika Sekolah Menengah," *Journal of Information System Research (JOSH)*, vol. 6, no. 2, 2025, doi: 10.47065/josh.v6i2.6523.

[18] N. Nurwanda, N. Suarna, and W. Prihartono, "Penerapan NLP (Natural Language Processing) Dalam Analisis Sentimen Pengguna Telegram Di Playstore," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 2, pp. 1841–1846, 2024, doi: 10.36040/jati.v8i2.8469.

[19] M. S. Anwar, I. Much, I. Subroto, and S. Mulyono, "Sistem Pencarian E-Journal Menggunakan Metode Stopword Removal dan Stemming," *Prosiding Konstelasi Ilmiah Mahasiswa Unissula (KIMU),* pp. 58–70, 2019.

[20] J. Pardede, and D. Darmawan, "Perbandingan Algoritma Stemming Porter, Sastrawi, Idris, Arifin dan Setiono pada Dokumen Teks Bahasa Indonesia", *JTIK (Jurnal Teknologi Informasi dan Ilmu Komputer)*, vol. 12, no. 1, 2025, doi: 10.25126/jtiik.2025128860.

[21] E. Muningsih, "Kombinasi Metode K-Means Dan Decision Tree Dengan Perbandingan Kriteria Dan Split Data," *J. Teknoinfo*, vol. 16, no. 1, p. 113, 2022, doi: 10.33365/jti.v16i1.1561.

[22] E. Rasywir, R. Sinaga, and Y. Pratama, "Analisis dan Implementasi Diagnosis Penyakit Sawit dengan Metode Convolutional Neural Network (CNN)," *Paradig. - J. Komput. dan Inform.*, vol. 22, no. 2, pp. 117–123, 2020, doi: 10.31294/p.v22i2.8907.

[23] A. Hanifa, S. A. Fauzan, M. Hikal, and M. B. Ashfiya, "Perbandingan Metode LSTM dan GRU (RNN) untuk Klasifikasi Berita Palsu Berbahasa Indonesia," *Din. Rekayasa*, vol. 17, no. 1, p. 33, 2021, doi: 10.20884/1.dr.2021.17.1.436.