

## Klasifikasi Spam Email Mahasiswa ITERA Menggunakan Metode *Hybrid Attention LSTM*

Natasya Ega Lina Marbun<sup>1,\*</sup>, Eksanty F Sugma Islamiaty<sup>2</sup>, Muhammad Regi Abdi Putra Amanta<sup>3</sup>

<sup>1</sup>Institut Teknologi Sumatera, Lampung Selatan, Indonesia

Corresponding Author. Email: natasya.122450024@student.itera.ac.id

### Abstrak

Email merupakan media komunikasi utama di lingkungan akademik, namun peningkatan penggunaannya juga diikuti oleh bertambahnya email spam yang dapat mengganggu efektivitas komunikasi dan berpotensi menimbulkan risiko keamanan digital. Penelitian ini bertujuan untuk mengklasifikasikan spam pada email mahasiswa Institut Teknologi Sumatera (ITERA) menggunakan metode *Hybrid Attention Long Short-Term Memory* (LSTM). Model yang diusulkan mengombinasikan pemrosesan teks email berbasis LSTM dengan *temporal attention mechanism* untuk memperhatikan kata-kata penting, serta pemanfaatan fitur numerik email melalui jalur jaringan saraf terpisah. Kedua jenis fitur tersebut digabungkan dalam satu arsitektur model untuk meningkatkan kinerja klasifikasi. Evaluasi dilakukan menggunakan *confusion matrix*, dan ROC-AUC. Hasil pengujian menunjukkan bahwa model *Hybrid Attention* LSTM mencapai performa yang sangat baik dengan nilai AUC sebesar 0.9574. Visualisasi *attention heatmap* digunakan untuk mengidentifikasi kata-kata kunci yang relevan pada masing-masing kelas. Dengan demikian, pendekatan yang diusulkan efektif untuk mendukung penyaringan spam email di lingkungan akademik.

**Kata kunci:** *Dense, Deep Learning, Long Short-Term Memory*

## Pendahuluan

Surat elektronik (email) merupakan media komunikasi utama dalam lingkungan akademik, termasuk di Institut Teknologi Sumatera (ITERA). Seiring meningkatnya intensitas penggunaan email, mahasiswa juga semakin sering menerima email spam yang tidak relevan dengan aktivitas akademik dan berpotensi mengganggu efektivitas komunikasi serta keamanan digital pengguna [1], [2]. Oleh karena itu, diperlukan sistem klasifikasi spam yang mampu bekerja secara akurat dan adaptif.

Berbagai metode machine learning, seperti Naïve Bayes, *Support Vector Machine* (SVM), dan *Decision Tree*, telah digunakan untuk mendeteksi email spam. Meskipun menunjukkan kinerja yang cukup baik, metode tersebut masih memiliki keterbatasan dalam menangani kompleksitas data teks, perubahan pola spam yang dinamis, serta risiko kesalahan klasifikasi yang relatif tinggi [3]. Selain itu, metode klasifikasi umumnya hanya memanfaatkan fitur teks atau fitur statistik secara terpisah, sehingga belum sepenuhnya merepresentasikan karakteristik email spam secara menyeluruh.

Perkembangan *deep learning*, khususnya *Long Short-Term Memory* (LSTM), memiliki kemampuan yang lebih baik dalam memahami urutan kata pada teks email [4]. Namun, LSTM berbasis teks saja masih memiliki keterbatasan karena belum mampu membedakan tingkat prioritas setiap kata secara optimal serta belum mempertimbangkan fitur numerik email, seperti panjang pesan, jumlah tautan, dan karakter khusus, yang sering menjadi indikator penting spam.

Untuk mengatasi keterbatasan tersebut, *attention mechanism* digunakan agar model dapat memperhatikan bagian teks yang paling penting [5]. Selain itu, integrasi fitur numerik melalui jaringan saraf *dense* digunakan untuk mendapatkan informasi email lainnya yang tidak ada dalam representasi teks. Oleh karena itu, penelitian ini mengombinasikan pemrosesan teks berbasis LSTM dan *attention mechanism* dengan analisis fitur numerik melalui *dense layer* untuk klasifikasi spam email mahasiswa ITERA. Pendekatan ini diharapkan mampu meningkatkan akurasi dan kemampuan generalisasi model dalam membedakan email spam dan ham secara lebih efektif.

## Metode

### A. Deskripsi Data

Data yang digunakan dalam penelitian ini merupakan data primer email mahasiswa ITERA yang telah diberi label sebagai ham (0) atau spam (1) untuk klasifikasi biner menggunakan *Hybrid LSTM*. Data terdiri dari sembilan atribut yaitu ‘id’, ‘label’, ‘subject’, ‘body’, ‘from\_domain’, ‘num\_urls’, ‘num\_exclaim’, ‘has\_attachment’, dan ‘body\_len’ yang terlihat pada Tabel 1 berikut.

Tabel 1 Data yang Digunakan

<b>id</b>	<b>label</b>	<b>subject</b>	<b>body</b>	<b>from domain</b>	<b>num urls</b>	<b>num exclaim</b>	<b>has attachment</b>	<b>body_len</b>
1	ham	itera - amd tech gen: innovate, learn, lead!	amd tech gen: innovate, learn...	itera.id	1	2	0	567

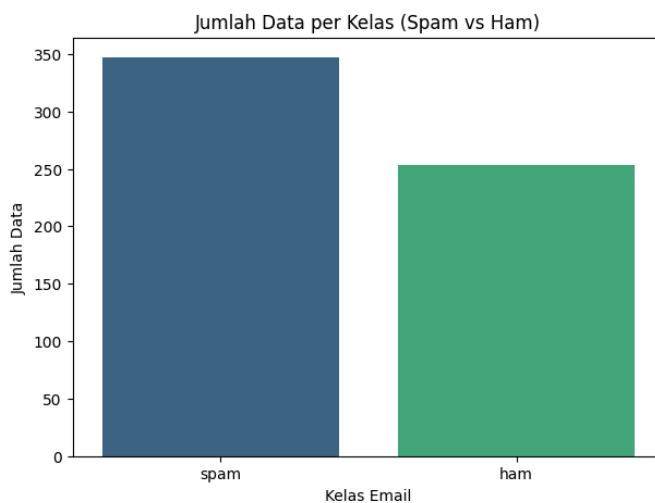
2	ham	keep presentation s on brand with themes	Hi <NAME>, your brand is...	gamma.app	5	2	0	767
:	:	:	:	:	:	:	:	:
599	spam	you have upcoming activities due	hi <NAME>, you have upcoming...	itera.id	3	0	0	174
600	spam	andrew ng is speaking at nodes	hey happy syahrul, we're thrilled to...	neo4j.com	0	0	0	1137

---

Tabel 1 menyajikan data yang diperoleh dari responden. Dalam penelitian ini, peneliti menambahkan dua kolom baru sebagai atribut tambahan, yaitu ‘*num\_special\_chars*’ dan ‘*avg\_word\_len*’. Atribut tersebut digunakan untuk merepresentasikan jumlah karakter khusus dan rata-rata panjang kata dalam email, yang selanjutnya dimanfaatkan sebagai fitur numerik guna mendukung proses analisis dan meningkatkan akurasi klasifikasi.

## B. Eksplorasi Data

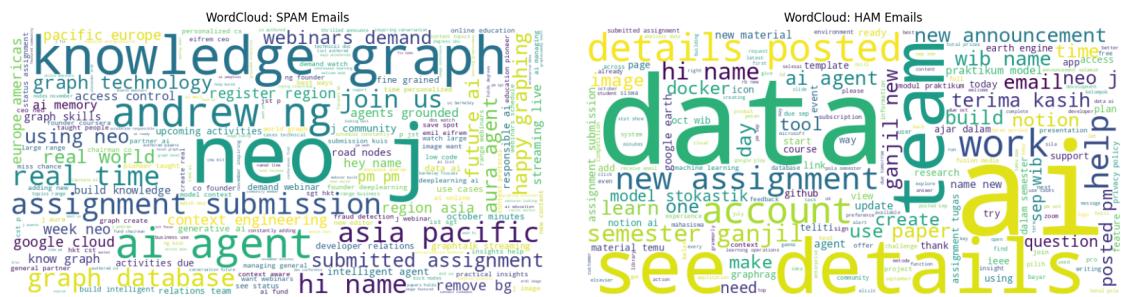
Analisis data eksploratori (EDA) dilakukan terlebih dahulu pada dataset email untuk mengenali pola dan karakteristik data sebelum model dilatih. Jumlah email pada masing-masing kelas ditunjukkan pada Gambar 1 berikut.



Gambar 1 Distribusi Jumlah Email Berdasarkan Kelas

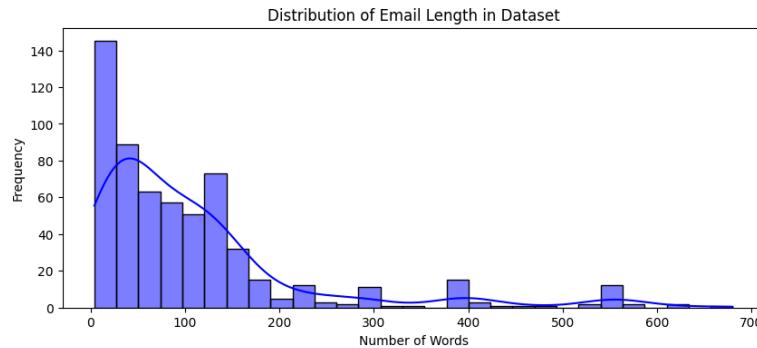
Gambar 1 menunjukkan distribusi jumlah data pada setiap kelas yang berbeda. Kelas spam memiliki sebanyak 347 data, sedangkan kelas ham berjumlah 253 data. Dengan demikian, total email yang digunakan dalam penelitian ini adalah 600 data. Meskipun distribusi data antar kelas tidak sepenuhnya seimbang, perbedaannya masih tergolong ringan dan tidak signifikan. Oleh karena itu, dataset ini tetap layak digunakan tanpa

teknik penyeimbangan tambahan. Gambar 2 menunjukkan kata terbanyak yang muncul pada email spam dan ham.



Gambar 2 Kata Terbanyak Muncul pada Email

Berdasarkan Gambar 2 kata-kata yang paling sering muncul pada email spam didominasi oleh ‘ai’, ‘neo’, ‘graph’, dan ‘data’, artinya email spam cenderung berisi konten promosi tertentu. Sementara itu, email ham lebih banyak memuat kata-kata terkait kegiatan akademik seperti ‘new’, ‘assignment’, ‘email’, ‘google’, dan ‘tugas’. Gambar 3 berikut menampilkan distribusi email berdasarkan panjang isi email.



Gambar 3 Distribusi Panjang Email

Gambar 3 menunjukkan distribusi panjang email sebagian besar email memiliki jumlah kata yang relatif pendek, terutama pada rentang di bawah 100 kata, dengan frekuensi yang menurun seiring bertambahnya panjang email. Hanya sedikit email yang mencapai panjang ratusan kata, sehingga pola ini mencerminkan kecenderungan komunikasi email yang umumnya ringkas dan tidak terlalu panjang.

## C. Prapemrosesan Data

Prapemrosesan data merupakan tahap penting untuk mengubah email mentah menjadi data yang siap digunakan dalam pelatihan model *deep learning*. Data yang digunakan dalam penelitian ini mencakup teks dan numerik maka diperlukan alur prapemrosesan yang bisa menangani kedua jenis informasi tersebut secara tepat. Tahap ini meliputi membersihkan dan menormalisasi teks, serta pengolahan fitur numerik agar sesuai dengan kebutuhan model. Tahapan prapemrosesan yang pertama adalah pemberian label pada tiap baris data, lalu dilanjutkan dengan prapemrosesan pada data teks sebagai berikut [6].

- 1) Gabungkan kolom ‘*subject*’ dan ‘*body*’ agar model dapat memproses isi dan konteks email secara lengkap untuk klasifikasi.

- 2) Membersihkan teks pada hasil gabungan kolom dengan cara menormalisasi huruf menjadi kecil serta menghapus elemen-elemen yang tidak relevan seperti URL, angka, dan karakter khusus.
- 3) Penggunaan *Stopword Removal* gabungan, dan *stemming* untuk bahasa Indonesia yaitu ‘Sastrawi’. Hal ini digunakan untuk menghilangkan kata tidak penting dan menormalkan bentuk kata seperti yang, dan, *the*, *is*, mengirimkan menjadi kirim.
- 4) Lakukan tokenisasi menggunakan *Keras Tokenizer*, yang mengubah kata menjadi indeks bilangan bulat berdasarkan frekuensi kemunculannya, dengan ukuran kosakata dibatasi pada 5.000 kata paling sering muncul.
- 5) Setiap urutan teks hasil tokenisasi disesuaikan menjadi panjang tetap 100 token dengan menambahkan padding berupa nilai nol.

Sementara itu, pemrosesan data numerik pada kolom jumlah url, jumlah tanda seru, jumlah lampiran, dan panjang teks bagian ‘body’ email berdasarkan Tabel 1 dilakukan melalui langkah-langkah berikut [1].

- 1) Encoding untuk mengubah data kategori berbentuk teks, seperti domain email pengirim, menjadi representasi numerik.
- 2) Normalisasi fitur menggunakan *MinMaxScaler* agar semua variabel berada pada rentang 0 sampai 1. Dihitung dengan Persamaan 1 berikut dengan  $X$  adalah nilai fitur.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

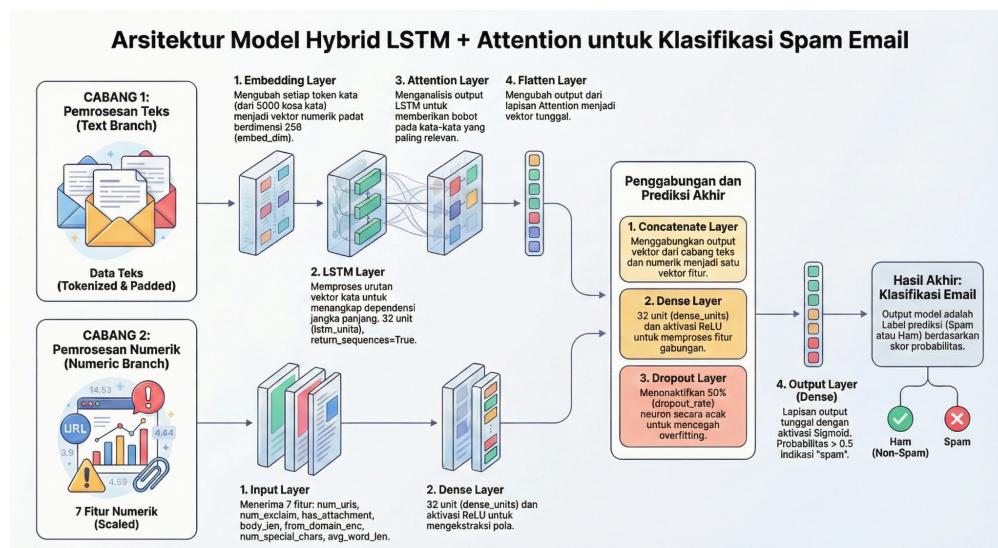
Setelah melalui tahap prapemrosesan, fitur teks dan numerik digabungkan menjadi satu vektor input, kemudian data dibagi menjadi 80% data pelatihan dan 20% data pengujian. Data pelatihan digunakan untuk membangun model, sedangkan data pengujian digunakan untuk mengevaluasi kinerjanya. Tabel 2 berikut menampilkan deskripsi variabel input yang digunakan.

Tabel 2 Deskripsi Variabel Input

Variabel Input	Tujuan	Bentuk Input ( <i>Shape</i> )
text_input	Menerima sekuen token teks yang sudah di-padding.	(None, 100) (Menggunakan maxlen = 100)
num_input	Menerima fitur numerik/metadata email yang telah di-scaling.	(None, 7) (Sesuai dengan jumlah fitur numerik yang digunakan: 7)

## D. Arsitektur Model

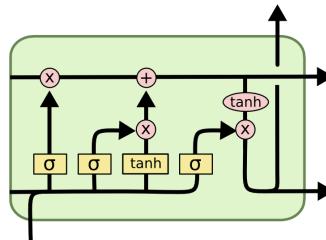
Secara garis besar, Gambar 4 berikut adalah arsitektur model yang digunakan dalam penelitian ini.



Gambar 4 Arsitektur yang Digunakan

### 1) Long Short-Term Memory (LSTM)

LSTM adalah salah satu jenis *Recurrent Neural Network* (RNN) yang digunakan untuk memproses data berurutan, seperti teks atau suara, dengan kemampuan menyimpan informasi penting dalam jangka panjang sehingga mengurangi masalah hilangnya gradien (*vanishing gradient*) yang sering terjadi pada RNN biasa [6]. Struktur LSTM dapat dilihat pada Gambar 5 berikut.



Gambar 5 Arsitektur LSTM

Berdasarkan Gambar 5 struktur LSTM terdiri atas unit memori dengan tiga gerbang utama yaitu gerbang input untuk memasukkan informasi baru, gerbang lupa (*forget gate*) untuk membuang informasi yang tidak penting, dan gerbang output untuk menghasilkan output sesuai kondisi memori. Perhitungan tiap gerbang dapat dihitung menggunakan persamaan berikut ini [2].

#### Gerbang Lupa

$$f_t = \sigma(W_f \times [h_{t-1}, x_t] + b_f) \quad (2)$$

#### Gerbang Input

$$i_t = \sigma(W_i \times [h_{t-1}, x_t] + b_i) \quad (3)$$

$$\tilde{c}_t = \tanh(W_c \times [h_{t-1}, x_t] + b_c) \quad (4)$$

### Perbarui Cell State

$$C_t = f_t \odot C_{t-1} + i_t \times \tilde{C}_t \quad (5)$$

### Gerbang Output

$$o_t = \sigma(W_o \times [h_{t-1}, x_t] + b_o) \quad (6)$$

$$h_t = o_t \odot \tanh(C_t) \quad (7)$$

dengan  $x_t$  adalah data input,  $h_t$  dan  $C_t$  adalah *hidden* dan *output state*,  $W$  adalah bobot, dan  $b$  adalah bias

## 2) Dense Neural Network (DNN)

Lapisan *fully-connected* pada DNN digunakan untuk setiap neuron menerima masukan dari seluruh neuron pada lapisan sebelumnya. Pada arsitektur ini, *output LSTM* dari data teks dan *output Dense* dari fitur numerik digabungkan (*Concatenate*) sehingga  $i_{concat} = [i_{teks}, i_{num}]$ , kemudian hasil gabungan tersebut diproses melalui lapisan DNN dengan fungsi aktivasi seperti ReLU dengan  $i_{fc1} = \text{ReLU}(W_1 i_{concat} + b_1)$  dan  $i_{fc2} = \text{ReLU}(W_2 i_{fc1} + b_2)$ . Pada tahap akhir, lapisan output dengan aktivasi sigmoid dengan  $\hat{y} = \sigma(W_o i_{fc2} + b_o)$ ,  $\hat{y} \in [0, 1]$  digunakan untuk menghasilkan keputusan klasifikasi biner. Berikut adalah persamaan pada DNN [1].

$$z^{(l)} = W^{(l)} a^{(l-1)} + b^{(l)} \quad (8)$$

$$a^{(l)} = f(z^{(l)}) \quad (9)$$

dengan  $f$  adalah fungsi aktivasi berupa ReLU atau Sigmoid sedangkan  $a^{(0)}$  adalah input vektor fitur.

## E. Attention Mechanism

*Attention mechanism* digunakan dalam penelitian ini untuk meningkatkan kemampuan model dalam mengekstraksi informasi penting dari urutan teks email. Pada data sekuensial, tidak semua kata atau bagian teks memiliki kontribusi yang sama terhadap penentuan kelas spam atau non-spam. Oleh karena itu, *attention mechanism* diterapkan untuk memberikan bobot yang berbeda pada setiap output LSTM di sepanjang urutan teks [7]. Keluaran LSTM tersebut dinotasikan sebagai  $h_t$  dengan  $t = 1, 2, \dots, T$  dengan  $T$  merupakan panjang urutan teks.

*Attention mechanism* kemudian menghitung skor kepentingan untuk setiap keluaran LSTM untuk menentukan tingkat kontribusinya terhadap representasi akhir teks. Skor attention dihitung menggunakan fungsi berikut:

$$e_t = g(h_t) = v^T \tanh(Wh_t + b) \quad (10)$$

dengan  $W$ ,  $v$ , dan  $b$  adalah parameter yang dipelajari selama proses pelatihan, serta  $e_t$  merepresentasikan skor kepentingan pada *timestep* ke- $t$ . Selanjutnya, skor *attention* dinormalisasi menggunakan fungsi *softmax* untuk memperoleh bobot *attention*.

$$\alpha_t = \frac{\exp(e_t)}{\sum_{i=1}^T \exp(e_i)} \quad (11)$$

bobot *attention*  $\alpha_t$  menunjukkan tingkat kontribusi masing-masing *timestep* terhadap pembentukan representasi akhir, dengan nilai total bobot bernilai satu.

Representasi akhir teks, yang disebut sebagai *context vector*, diperoleh dengan menghitung kombinasi dari seluruh output LSTM sebagai berikut:

$$c = \sum_{t=1}^T \alpha_t h_t \quad (12)$$

*context vector* merepresentasikan informasi penting dari teks email yang telah difokuskan oleh *attention mechanism*. Selanjutnya, *context vector* digabungkan dengan fitur numerik email menggunakan layer *concatenate* sebelum diproses oleh layer *fully connected* untuk menghasilkan keluaran klasifikasi.

*Attention mechanism* yang digunakan dalam penelitian ini merupakan *temporal attention*, karena bobot *attention* diberikan pada setiap keluaran LSTM di sepanjang urutan teks.

#### F. **Binary Cross Entropy Loss (BCELoss)**

BCELoss digunakan sebagai alat ukur ketidakmiripan antara dua kelas (membedakan antara spam dan ham). Fungsi ini memberikan hukuman (*loss value*) yang sangat besar jika model sangat yakin dengan prediksi yang salah. Sebaliknya, nilai *loss* akan mendekati nol jika prediksi model sangat dekat dengan prediksi benar. Secara matematis, BCELoss dirumuskan sebagai berikut [1].

$$\mathcal{L}_{BCE} = -\frac{1}{N} \sum_{i=1}^N [y_i \log \hat{y}_1 + (1 - y_i) \log (1 - \hat{y}_1)] \quad (13)$$

dengan  $N$  adalah jumlah sampel pada *batch*,  $y$  adalah label kelas tersebut.

#### G. Optimasi *Adaptive Moment Estimation* (ADAM)

Optimasi Adam dipilih karena sifatnya yang adaptif dan efisien. Metode ini memanfaatkan keunggulan Momentum dan RMSProp untuk memperbarui bobot, sehingga mampu menyesuaikan laju pembelajaran secara dinamis selama proses pelatihan. Pembaruan bobot pada optimasi Adam dilakukan dengan persamaan berikut [8].

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (14)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (15)$$

$$\widehat{m}_t = \frac{m_t}{1-\beta_1^t}; \quad \widehat{v}_t = \frac{v_t}{1-\beta_2^t} \theta_t + 1 = \theta_t - \alpha \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t + \epsilon}} \quad (16)$$

Dengan  $\theta_t$  adalah bobot yang diperbarui,  $g_t$  adalah gradien dari fungsi *loss*,  $\beta$  adalah tingkat peluruhan,  $\alpha$  adalah *learning rate*, dan  $\epsilon$  adalah epsilon.

## H. Regularisasi

*Dropout* diterapkan setelah lapisan *dense* dengan probabilitas  $p = 0.5$  untuk menekan ketergantungan antar neuron. Dilakukan secara acak menonaktifkan setengah dari neuron pada setiap langkah pelatihan, dengan  $i_{drop} = i \odot Bernoulli(p)$ .

Penghentian Awal (*Early Stopping*) digunakan sebagai mekanisme regularisasi waktu pelatihan. Proses pelatihan akan dihentikan secara otomatis apabila nilai *loss* validasi (*validation loss*) tidak menunjukkan perbaikan selama tiga epoch berturut-turut.

Normalisasi batch bersifat opsional dan diaplikasikan setelah fungsi aktivasi pada beberapa lapisan tertentu. Tujuannya adalah untuk menstabilkan *learning rate*.

## I. Arsitektur dan *Hyperparameter*

Tabel 3 berikut menampilkan detail arsitektur layer dan jumlah parameter pada penelitian *Hybrid Attention LTSM*.

Tabel 3 Arsitektur Layer dan Jumlah Parameter yang Digunakan

Layer (type)	Output Shape	Param #	Connected to
text_input (InputLayer)	(None, 100)	0	-
embedding_1 (Embedding)	(None, 100, 256)	1,280,000	text_input[0][0]
lstm (LSTM)	(None, 100, 32)	36,992	embedding[0][0]
attention (Attention)	(None, 100, 32)	0	lstm[0][0], lstm[0][0]
num_input (InputLayer)	(None, 7)	0	-
flatten (Flatten)	(None, 3200)	0	attention[0][0]
dense (Dense)	(None, 32)	256	num_input[0][0]
concatenate (Concatenate)	(None, 3232)	0	flatten[0][0], dense[0][0]
dense_1 (Dense)	(None, 32)	103,456	concatenate[0][0]
dropout (Dropout)	(None, 32)	0	dense_1[0][0]
dense_2 (Dense)	(None, 1)	33	dropout[0][0]

Berdasarkan Tabel 3 cabang teks menerima urutan 100 token yang dipetakan melalui lapisan *Embedding*, kemudian diproses oleh LSTM 32-unit dan *Attention Mechanism* untuk melihat kata-kata penting, sebelum diubah menjadi vektor fitur. Cabang numerik memproses fitur numerik menggunakan lapisan Dense 32-unit. Kedua representasi digabungkan melalui *Concatenate*, dilanjutkan dengan lapisan Dense dan Dropout, serta

diakhiri lapisan output dengan fungsi aktivasi sigmoid untuk menghasilkan probabilitas klasifikasi. Model ini mengombinasikan pemodelan sekuensial dan informasi struktural dengan total 1.420.737 parameter yang dapat dilatih. *Hyperparameter* yang digunakan dalam penelitian ini dapat dilihat pada Tabel 4 berdasarkan hasil terbaik.

Tabel 4 Nilai hyperparameter yang digunakan

<i>Hyperparameter</i>	Nilai
Dimensi <i>Embedding</i>	256
Jumlah Unit LSTM	32
Jumlah Unit Dense	32
<i>Dropout Rate</i>	0.5
<i>Learning Rate</i>	0.0005

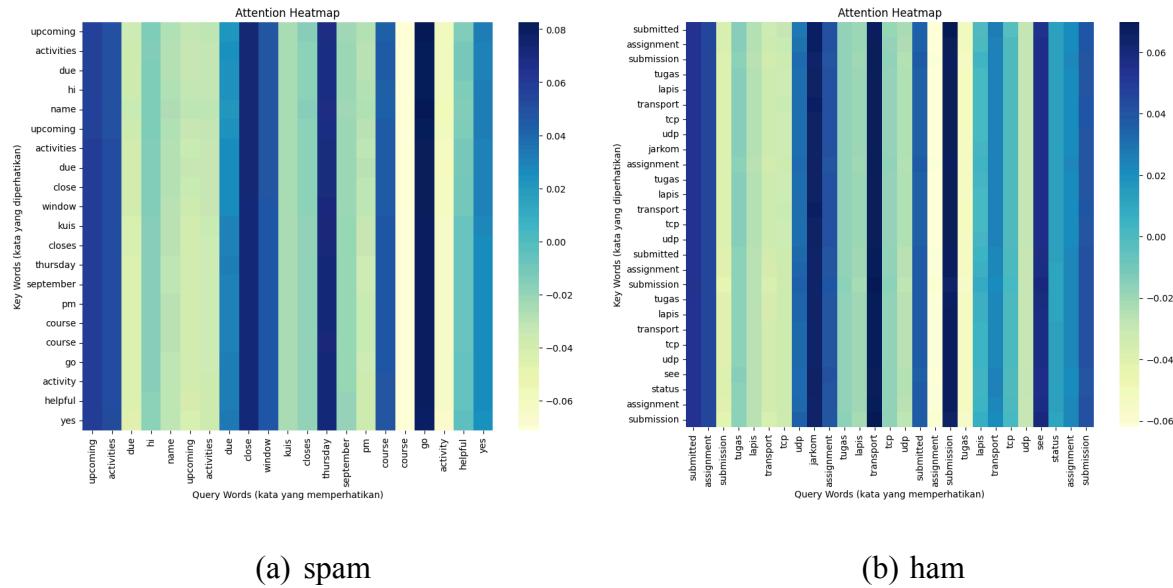
## Hasil

Model ini menggunakan *hyperparameter* pada Tabel 3. Gambar 6 menunjukkan konvergensi model, ditandai dengan akurasi meningkat dan *loss* berkurang secara konsisten sebelum kondisi *Early Stopping* terpenuhi.

```
Epoch 1/100
12/12 5s 134ms/step - accuracy: 0.6239 - loss: 0.6804 - val_accuracy: 0.7188 - val_loss: 0.5988
Epoch 2/100
12/12 2s 133ms/step - accuracy: 0.7134 - loss: 0.5456 - val_accuracy: 0.6562 - val_loss: 0.5653
Epoch 3/100
12/12 2s 82ms/step - accuracy: 0.7714 - loss: 0.4506 - val_accuracy: 0.8750 - val_loss: 0.4577
Epoch 4/100
12/12 1s 84ms/step - accuracy: 0.8710 - loss: 0.3519 - val_accuracy: 0.8542 - val_loss: 0.3033
Epoch 5/100
12/12 1s 81ms/step - accuracy: 0.9218 - loss: 0.2120 - val_accuracy: 0.8229 - val_loss: 0.3666
Epoch 6/100
12/12 1s 82ms/step - accuracy: 0.9264 - loss: 0.1784 - val_accuracy: 0.8125 - val_loss: 0.5571
Epoch 7/100
12/12 1s 81ms/step - accuracy: 0.9410 - loss: 0.1655 - val_accuracy: 0.8229 - val_loss: 0.5236
Epoch 8/100
12/12 1s 81ms/step - accuracy: 0.9245 - loss: 0.1973 - val_accuracy: 0.7812 - val_loss: 0.4103
Epoch 9/100
12/12 1s 80ms/step - accuracy: 0.9413 - loss: 0.1411 - val_accuracy: 0.8750 - val_loss: 0.3718
```

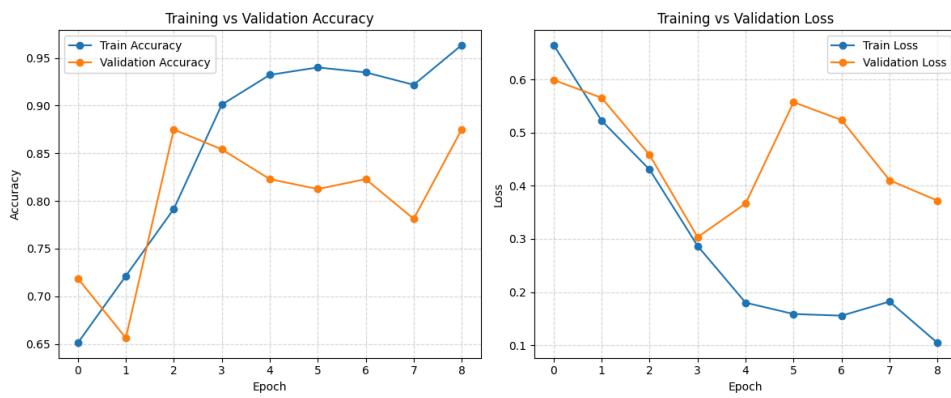
Gambar 6 Log Perkembangan Kinerja Model Berdasarkan Setiap *Epoch* Pelatihan

Hasil *attention mechanism* ditunjukkan pada Gambar 7, di mana subgambar (a) merepresentasikan kelas spam dan subgambar (b) merepresentasikan kelas ham.

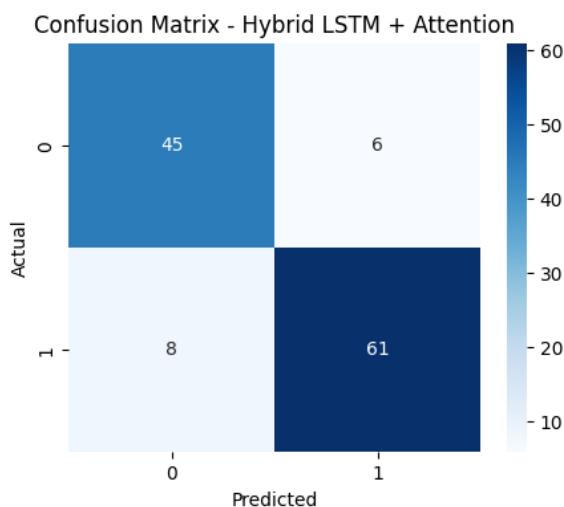
Gambar 7 Visualisasi *Attention Heatmap* pada Model Hybrid Attention LSTM

Gambar 7 menunjukkan visualisasi *attention heatmap* yang menggambarkan bobot perhatian (*attention weights*) berdasarkan model *Hybrid Attention LSTM* terhadap kata-kata kunci dalam teks email. Sumbu horizontal merepresentasikan *query words* atau kata yang menjadi fokus perhatian model, sedangkan sumbu vertikal menunjukkan *key words* yang dipertimbangkan dalam proses klasifikasi. Intensitas warna pada *heatmap* menandakan tingkat kepentingan setiap kata, di mana warna yang lebih gelap menunjukkan bobot attention yang lebih tinggi.

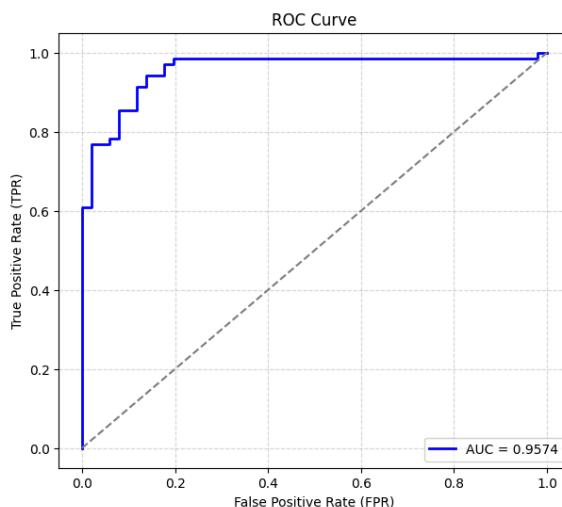
Untuk menilai kemampuan generalisasi model dan mendeteksi indikasi *overfitting*, Gambar 8 memvisualisasikan perbandingan antara Akurasi dan Fungsi *Loss* pada set data pelatihan (*Training*) dan validasi (*Validation*) di setiap *epoch*.

Gambar 8 Kurva Nilai Akurasi dan *Loss* pada Data Training dan Validasi

Gambar 9 menunjukkan *Confusion Matrix* berdasarkan model *Hybrid Attention LSTM*. Matriks ini merupakan dasar perhitungan Akurasi, Presisi, Recall, dan F1-Score.

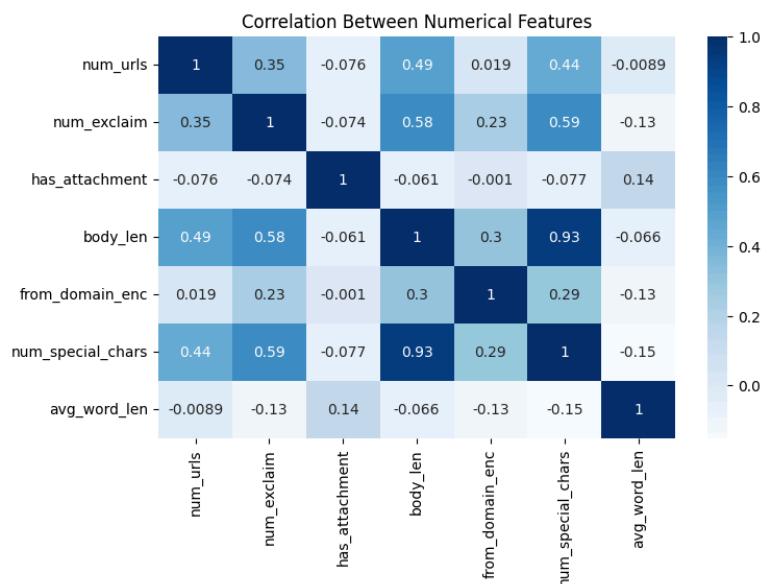
Gambar 9 *Confusion Matrix*

Gambar 10 menunjukkan kurva ROC (Receiver Operating Characteristic) yang menggambarkan kinerja model dalam membedakan kelas spam dan ham berdasarkan hubungan antara True Positive Rate dan False Positive Rate. Nilai AUC yang dihasilkan menunjukkan tingkat kemampuan model dalam melakukan klasifikasi secara keseluruhan.



Gambar 10 Kurva ROC-AUC

Gambar 11 menampilkan matriks yang menunjukkan tingkat hubungan antar fitur, korelasi yang mendekati 1 atau  $-1$  mengindikasikan hubungan yang kuat, sedangkan nilai mendekati 0 menunjukkan hubungan yang lemah.



Gambar 11 Matriks Korelasi Fitur Numerik

## Pembahasan

Model *Hybrid Attention LSTM* dilatih menggunakan *batch size* sebesar 32 dengan jumlah maksimum 100 *epoch*, dengan *Early Stopping* menggunakan *patience* 5 untuk mencegah terjadinya *overfitting*. Perkembangan kinerja model ditunjukkan melalui kurva akurasi dan *loss* pada Gambar 8. Berdasarkan kurva tersebut, akurasi pelatihan meningkat secara konsisten dan mencapai nilai sekitar 0.95 sebelum *epoch* ke-9, sedangkan, *train loss* mengalami penurunan signifikan hingga mendekati 0.1. Meskipun *validation loss* menunjukkan fluktuasi, namun *validation accuracy* mengikuti pola peningkatan pada fase awal pelatihan. Kondisi ini menunjukkan bahwa model mampu mempelajari pola data secara efektif tanpa mengalami *overfitting* yang signifikan hingga titik *early stopping* tercapai, serta memiliki kemampuan generalisasi yang memadai terhadap data yang belum pernah dilihat.

Berdasarkan Gambar 7, pada kelas spam (Gambar 7a), model memberikan *weight attention* yang lebih tinggi pada kata-kata yang sering muncul dalam konteks email akademik yang bersifat informatif atau mendesak, seperti “*upcoming*”, “*activities*”, “*due*”, “*assignment*”, “*class*”, dan “*transport*”. Sementara itu, pada kelas ham (Gambar 7b), *weight attention* lebih tinggi pada kata-kata yang berkaitan dengan pengumpulan dan penugasan, seperti “*submitted*”, “*assignment*”, “*tugas*”, dan “*submission*”. Perbedaan pola distribusi *attention weight* ini menunjukkan bahwa *attention mechanism* mampu mengidentifikasi dan memperhatikan kata-kata kunci yang tepat untuk masing-masing kelas.

Evaluasi kinerja model secara kuantitatif dilakukan menggunakan *Confusion Matrix* (Gambar 9) dan kurva ROC-AUC (Gambar 10). Berdasarkan *Confusion Matrix*, diperoleh nilai TP sebesar 61 dan TN sebesar 45, yang menunjukkan tingkat ketepatan prediksi yang tinggi pada kedua kelas. Selanjutnya, hasil evaluasi menggunakan kurva ROC menunjukkan kemampuan diskriminatif model yang sangat baik, dengan nilai AUC sebesar 0.9574. Nilai AUC yang mendekati 1 artinya model *Hybrid Attention LSTM* memiliki performa klasifikasi yang sangat baik dalam membedakan email spam dan ham.

Analisis tambahan dilakukan untuk mengevaluasi korelasi antar fitur numerik pada Gambar 11. Hasil analisis menunjukkan adanya korelasi positif yang kuat antara fitur ‘*body\_len*’ (panjang badan email) dan ‘*num\_special\_chars*’ (jumlah karakter khusus) dengan nilai korelasi sebesar 0.93. Korelasi kuat lainnya juga ditemukan antara ‘*num\_urls*’ (jumlah URL) dan ‘*num\_exclaim*’ (jumlah tanda seru) dengan nilai mendekati 0.95, yang mencerminkan karakteristik umum email spam yang cenderung memiliki konten panjang, banyak tautan, serta penggunaan tanda seru sebagai bentuk penekanan. Sebaliknya, fitur ‘*has\_attachment*’ (keberadaan lampiran) menunjukkan korelasi yang sangat lemah dengan fitur lainnya, sehingga fitur tersebut cenderung memberikan informasi tersendiri dan tidak banyak dipengaruhi oleh fitur numerik lainnya dalam model.

## Kesimpulan

Penelitian ini berhasil menerapkan model *Hybrid Attention LSTM* untuk mengklasifikasikan spam pada email mahasiswa ITERA dengan kinerja yang sangat baik. Hasil pelatihan dan evaluasi menunjukkan bahwa gabungan metode *attention mechanism* dengan pemrosesan teks berbasis LSTM serta fitur numerik email mampu meningkatkan kemampuan model dalam mengenali pola penting pada email spam dan ham. Nilai akurasi dan AUC yang tinggi menunjukkan bahwa model memiliki kemampuan generalisasi yang baik terhadap data yang belum pernah dilihat. Selain itu, visualisasi *attention weight* membuktikan bahwa model dapat menyoroti kata-kata kunci yang relevan dalam proses klasifikasi. Dengan demikian, pendekatan yang diusulkan efektif untuk mendukung penyaringan email dan meningkatkan keamanan komunikasi digital di lingkungan akademik ITERA.

## Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada Institut Teknologi Sumatera (ITERA) atas dukungan fasilitas dan lingkungan akademik yang menunjang pelaksanaan penelitian ini. Ucapan terima kasih juga disampaikan kepada pihak-pihak yang telah membantu dalam proses pengumpulan data serta memberikan masukan dan saran yang berharga selama penyusunan penelitian ini. Kontribusi dan dukungan tersebut sangat berarti dalam meningkatkan kualitas penelitian yang dilakukan.

## Konflik Kepentingan

Penulis menyatakan bahwa tidak terdapat kepentingan pribadi maupun finansial yang berpotensi menimbulkan konflik kepentingan terkait dengan topik yang dibahas dalam naskah ini.

## Kontribusi Penulis

- Perumusan konsep dan tujuan penelitian: NELM

- Perancangan metode dan model penelitian: NELM, MRAPA
- Pengumpulan, pengelolaan, dan pengolahan data: NELM, EFSI, MRAPA
- Pelaksanaan eksperimen dan analisis data: NELM, EFSI, MRAPA
- Penyusunan naskah Awal: EFSI
- Penelaahan dan penyunting naskah: NELM, EFSI, MRAPA

## Daftar Pustaka

- [1] K. S. Ubale and K. A. Shirasath, “SPAMNET: A Hybrid Deep Learning Framework for Robust Spam Email Detection Using Multi-Modal Features,” International Journal of Applied Mathematics, vol. 38, no. 6s, Oct. 15, 2025. doi: 10.12732/ijam.v38i6s.425
- [2] Maugy Al Kautsar, Galet Guntoro Setiaji, and Ahmad Rifa'i, “Analisis Komparasi Kinerja LSTM dan CNN dalam Deteksi Spam Email Berbasis Deep learning”, bulletincsr, vol. 5, no. 4, pp. 584-593, Jun. 2025.
- [3] S. Aleem, Z. U. Islam, S. S. U. Hasan, H. Akbar, M. F. Khan, and S. A. Ibrar, “Spam Email Detection Using Long Short-Term Memory and Gated Recurrent Unit,” Applied Sciences, vol. 15, no. 13, p. 7407, 2025, doi: 10.3390/app15137407.
- [4] A. Tholib, N. K. Agusmawati, and F. Khoiriyah, “Prediksi Harga Emas Menggunakan Metode LSTM dan GRU,” Jurnal Informatika dan Teknik Elektro Terapan, vol. 11, no. 3, pp. 194–203, 2023, doi: 10.23960/jitet.v11i3.3250.
- [5] S. A. Anggara, W. Witanti, dan Melina, “Peramalan Harga Cabai Rawit Merah Menggunakan Attention Mechanism Berbasis Long Short-Term Memory,” Journal of Applied Computer Science and Technology (JACOST), vol. 5, no. 2, pp. 128–135, 2024, doi: 10.52158/jacost.v5i2.875.
- [6] 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.
- [7] T. Zhang, K. Zhang, dan J. Wu, “Multi-modal attention mechanisms in LSTM and its application to acoustic scene classification,” in Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 2018, pp. 530–534.
- [8] 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.