

**LAPORAN TUGAS**  
**KLASIFIKASI TEKS MENGGUNAKAN ARSITEKTUR RNN: STUDI**  
**KASUS SPAM VS NON-SPAM SMS MENGGUNAKAN DEEP**  
**LEARNING BERBASIS RNN**

Dosen Pengampuh Matakuliah Pembelajaran Mesin 2:  
**Al-Ustadz Dr. Oddy Virgantara Putra, S.Kom., M.T.**



Fayshal Karan Athilla / 442023611088

**FAKULTAS SAINS DAN TEKNOLOGI**  
**PROGRAM STUDI TEKNIK INFORMATIKA**  
**UNIVERSITAS DARUSSALAM GONTOR, PONOROGO**  
**TAHUN AKADEMIK: 1446-1447 H / 2025-2026 M**

## 1. Latar Belakang

Klasifikasi teks merupakan salah satu cabang penting dalam pemrosesan bahasa alami (Natural Language Processing/NLP). Tugas ini bertujuan untuk mengelompokkan teks ke dalam kategori tertentu berdasarkan isi pesan. Salah satu contoh aplikasinya adalah klasifikasi pesan menjadi **spam** atau **non-spam**, yang memiliki nilai praktis dalam dunia nyata, seperti pada sistem penyaringan email atau pesan SMS.

Pada tugas ini, saya membangun model klasifikasi teks menggunakan metode **Recurrent Neural Network (RNN)** khususnya varian **LSTM (Long Short-Term Memory)**. Tujuan utama dari tugas ini adalah menerapkan deep learning untuk mengklasifikasikan teks SMS, sekaligus memahami alur preprocessing, tokenisasi, hingga evaluasi performa model.

## 2. Dataset

### Sumber data:

Dataset diambil dari [Kaggle – SMS Spam Collection Dataset](#), yang dikembangkan oleh Almeida dkk.

### Deskripsi data:

- Jumlah data: 5572 pesan
  - Spam: 747
  - Ham (non-spam): 4825
- Format kolom:
  - label: spam atau ham
  - text: isi pesan SMS
- Contoh:
  - Ham: "I'll call you later"
  - Spam: "Congratulations! You've won a free ticket. Call now!"

### Alasan pemilihan data:

Dataset ini dipilih karena:

- Sudah bersih dan siap digunakan
- Sangat umum dalam studi NLP pemula
- Relevan dengan kehidupan nyata (filtering spam)
- Berukuran kecil, cocok untuk eksperimen RNN

### 3. Implementasi Model

- **Arsitektur RNN**

Model yang digunakan adalah **LSTM** karena kemampuannya dalam menangani urutan panjang dan mengatasi permasalahan vanishing gradient. Arsitektur terdiri dari:

- Layer Embedding untuk representasi kata
- Layer LSTM dengan 64 unit
- Layer output sigmoid (karena biner)

- **Preprocessing**

Langkah-langkah preprocessing:

- Lowercasing seluruh teks
- Menghapus simbol dan angka
- Menghapus stopwords (menggunakan nltk)
- Tokenisasi dan padding
  - Max words: 5000
  - Max sequence length: 100

- **Pengaturan Eksperimen**

- Epoch: 5
- Batch size: 32
- Optimizer: Adam
- Loss function: Binary Crossentropy
- Validation split: 20%

- **Log Eksperimen**

Percobaan	Model	Dropout	Optimizer	Akurasi Validasi	Catatan
#1	LSTM(64)	0	Adam	97.1%	Overfitting sedikit
#2	LSTM(64)	0.2	Adam	97.6%	Lebih stabil setelah epoch 3

## 4. Evaluasi Hasil

### Evaluasi Matrix:

Berdasarkan hasil pengujian pada data test sebanyak 1.115 sampel, diperoleh hasil sebagai berikut:

- Akurasi: 98%
- F1-score Spam: 0.93
- F1-score Non-Spam: 0.99

Label	Precision	Recall	F1-score	Jumlah Data
Non-Spam	0.98	1.00	0.99	966
Spam	0.99	0.87	0.93	149
TOTAL	0.98	0.98	0.98	1115

### Learning Curve:

Berdasarkan hasil training:

Epoch	Accuracy	Val accuracy	Loss	Val loss
1	88.6%	94.8%	0.3918	0.1702
2	98.3%	96.9%	0.0811	0.0889
3	99.0%	97.6%	0.0393	0.0798
4	99.3%	98.2%	0.0246	0.0718
5	99.8%	97.8%	0.0125	0.0842

## 5. Refleksi Pribadi

- **Apa tantangan utama yang saya hadapi?**
  1. Pada awalnya saya mengalami kendala dalam menentukan tema yang sesuai. Saya berencana untuk mengangkat topik klasifikasi berdasarkan emosi dari lirik lagu. Namun, setelah menelusuri beberapa dataset, saya mendapati bahwa sebagian besar tidak menyediakan lirik secara lengkap, sehingga menyulitkan dalam proses analisis.
  2. Karena mengalami hambatan tersebut dan belum menemukan alternatif yang tepat, saya akhirnya memutuskan untuk menggunakan contoh studi kasus yang disarankan oleh dosen, yaitu klasifikasi teks antara pesan spam dan non-spam.

3. Beberapa dataset awal tidak cocok (tidak ada label, struktur tidak sesuai).
  4. Model sempat menunjukkan **overfitting ringan** (val\_loss naik di akhir epoch)
- **Apa Solusi yang anda coba dan bagaimana hasilnya?**
    1. Dalam upaya mengembangkan studi kasus klasifikasi emosi dari lirik lagu, saya sempat mencoba menggunakan bantuan AI untuk memproses data. Namun, sebagian besar dataset yang ditemukan tidak memiliki label emosi secara langsung, sehingga memerlukan pelabelan manual yang cukup memakan waktu dan tenaga.
    2. Sebaliknya, setelah beralih ke contoh studi kasus yang disarankan oleh ustadz, yaitu klasifikasi pesan spam dan non-spam, proses implementasi dapat berjalan dengan lebih lancar dan terstruktur, mulai dari preprocessing hingga pelatihan model.
    3. Menambahkan dropout di LSTM untuk mengurangi overfitting.
    4. Mengurangi epoch agar model tidak overtrain.
  - **Jika Anda menggunakan ChatGPT atau AI lain, bagian mana yang dibantu dan bagaimana Anda memverifikasinya?**
    1. Membantu merancang arsitektur LSTM / RNN
    2. Menyusun kode preprocessing dan tokenisasi
    3. Interpretasikan hasil evaluasi dan log training
  - **Apa pelajaran paling penting dari tugas ini?**
    1. Memahami data preprocessing sangat krusial dalam NLP.
    2. Metrik seperti recall untuk spam perlu jadi perhatian karena spam lebih jarang dan sensitif terhadap kesalahan klasifikasi.

## 6. Kesimpulan

Model LSTM berhasil mengklasifikasikan pesan spam dan non-spam dengan akurasi tinggi (98%+). Penggunaan embedding + LSTM sangat cocok untuk teks pendek seperti SMS. Saran pengembangan selanjutnya:

- Coba arsitektur BiLSTM atau GRU
- Tambahkan preprocessing seperti stemming/lemmatization
- Uji pada dataset lain seperti email atau komentar YouTube

## **7. Referensi**

Almeida, T. A., Hidalgo, J. M. G., & Yamakami, A. (2011). Contributions to the study of SMS spam filtering: new collection and results. DocEng.

Kaggle – SMS Spam Collection Dataset

Chollet, F. (2017). Deep Learning with Python. Manning Publications.

TensorFlow Documentation: <https://www.tensorflow.org>