

FST UIN JAKARTA

# UAS NLP 2025

RNN vs. LSTM with IMDB Dataset of 50K Movie Reviews

Presented By

**Kelompok 5**

Dosen Pengampu: Muhaza Liebelinto, M.SI



NLP 2025

# Meet the Team

## Kelompok 5



**Ahmad Izza**  
11220940000006



**Dani Hidayat**  
11220940000014





# Project Background



## Setimen Analysis

Analisis sentimen adalah tugas klasifikasi teks untuk menentukan polaritas (positif/negatif).

## RNN

Model Recurrent Neural Network (RNN) cocok untuk data sekuensial seperti teks, namun memiliki kelemahan pada long-term dependencies karena masalah vanishing gradient.

## LSTM

Long Short-Term Memory (LSTM) merupakan varian RNN yang dirancang untuk mengatasi masalah tersebut dengan adanya cell state dan gates.

# Project Goals

- ▶ **Membandingkan performa RNN dan LSTM untuk memprediksi sentimen dari review movie pada dataset IMDB**
- ▶ **Menganalisis fenomena vanishing gradient pada kedua model saat proses training.**
- ▶ **Mengevaluasi kemampuan kedua model dalam menangani long-term dependencies melalui analisis kesalahan.**



# Dataset IMDB

- Dataset: **IMDb Movie Reviews**
- Total: **50.000 review** (berlabel **positif** atau **negatif**)
- **Karakteristik:** Terdapat review yang panjang, kompleks, dan mengandung kata tidak baku serta simbol/tag HTML.



**80% Training**

**39.665 data**  
**Positif: 19.907**  
**Negatif: 19.758**

**10% Validation**

**4.958 data**  
**Positif: 2.488**  
**Negatif: 2.470**

**10% Testing**

**4.959 data**  
**Positif: 2.489**  
**Negatif: 2.470**

# Project Flow

➡ EDA & Preprocessing

➡ Pembagian Data

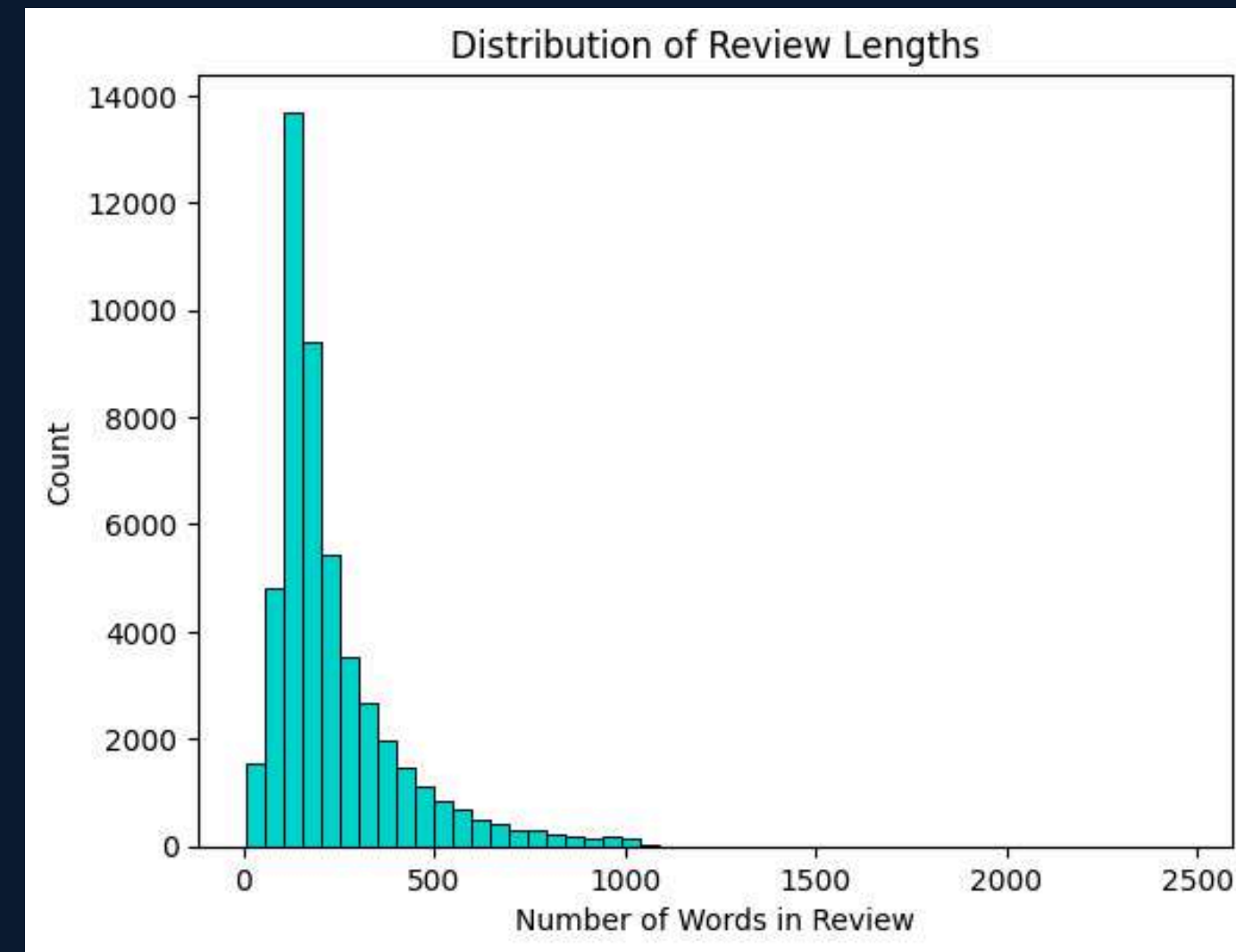
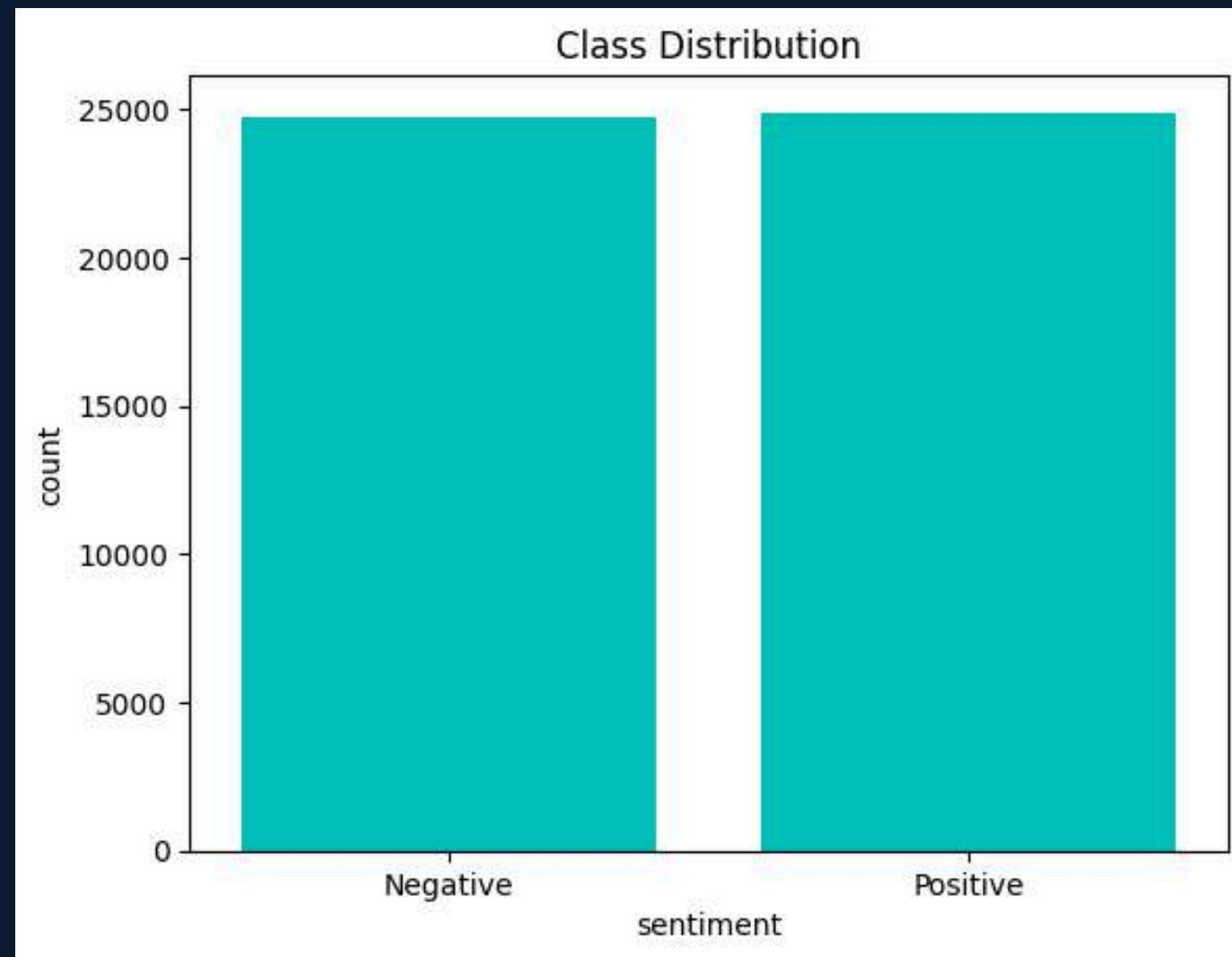
⬇ Tokenisasi & Padding

Analisis Lanjutan

⬅ Training & Evaluasi

⬅ Pemodelan

# Exploratory Data





# Preprocessing Text

## Simple Cleaning

Lowercase, Remove HTML Tags, Remove Punctuation & Digits, Tokenization, Remove Stopwords

## Encoding Sentiment

**Positif:** 1  
**Negatif:** 0

## Splitting Data

**Train:** 80%  
**Val:** 10%  
**Test:** 10%

## Tokenisasi Text

**Vocab size:** max 10.000  
**Padding:** 256 (Pendek ditambahkan angka 0 (padding) & panjang akan dipotong (truncating))  
Data teks diubah menjadi tensor numerik



# RNN & LSTM Architecture



## Komponen Utama



- **nn.Embedding:** Mengubah setiap kata (indeks) menjadi vektor makna berdimensi 64.
- **nn.RNN / nn.LSTM:** Lapisan inti yang memproses teks dari dua arah (bidirectional) untuk menangkap konteks.
- **nn.Linear:** Lapisan klasifikasi akhir untuk membuat keputusan.
- **nn.Sigmoid:** Fungsi aktivasi yang menghasilkan probabilitas (0-1).



## Parameter Utama



- **Embedding Dimension:** 64
- **Hidden Dimension:** 32
- **Bidirectional:** True
- **Dropout:** 0.2

# RNN Workflow

## Input & Embedding

Sekuens angka diubah menjadi vektor.

**Bentuk:**  $[B, T] \rightarrow [B, T, E]$

## Proses RNN & Penggabungan

Lapisan RNN memproses sekuens dari arah depan dan belakang, lalu menggabungkan vektor ringkasan (hidden state) terakhir dari kedua arah.

**Bentuk:**  $\rightarrow [B, H*2]$

## Klasifikasi & Output

Vektor ringkasan dimasukkan ke Linear Layer untuk menghasilkan skor prediksi, lalu Sigmoid.

**Bentuk:**  $\rightarrow [B, 1]$  kemudian menjadi  $[B]$

# LSTM Workflow

## Input & Embedding

Sekuens angka diubah menjadi vektor.

**Bentuk:**  $[B, T] \rightarrow [B, T, E]$

## Proses RNN & Penggabungan

Prosesnya bidirectional seperti RNN, namun lebih cerdas. LSTM menggunakan **forget gates** dan **cell state** (memori jangka panjang) untuk secara aktif "mengingat" informasi penting dan "melupakan" informasi tidak relevan dari awal kalimat. Vektor ringkasan terakhirnya tetap digabungkan.

**Bentuk:**  $\rightarrow [B, H*2]$

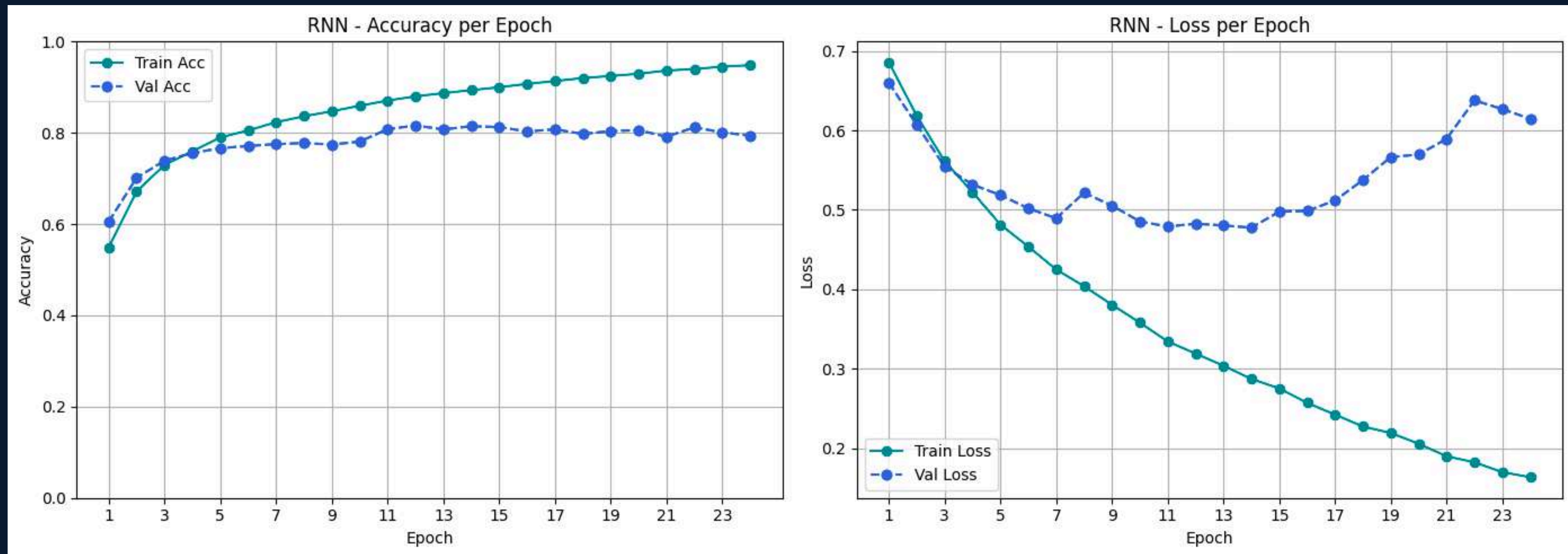
## Klasifikasi & Output

Vektor ringkasan dimasukkan ke Linear Layer untuk menghasilkan skor prediksi, lalu Sigmoid.

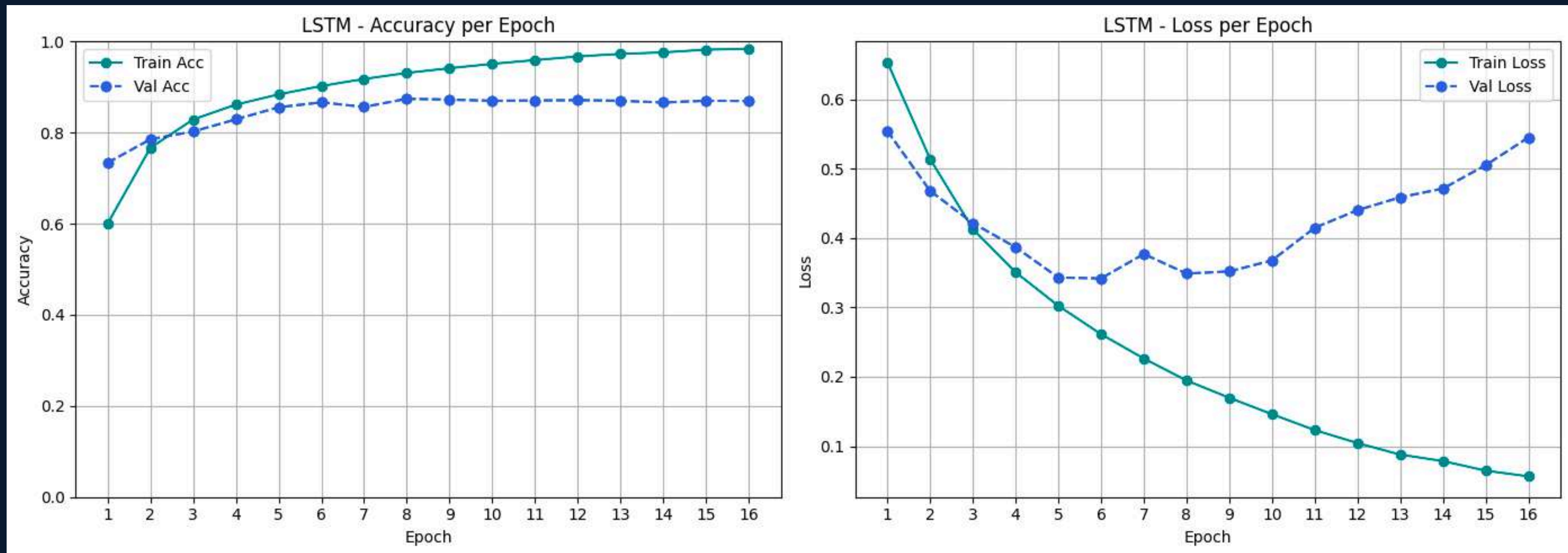
**Bentuk:**  $\rightarrow [B, 1]$  kemudian menjadi  $[B]$



# RNN Model Performance



# LSTM Model Performance



# RNN vs LSTM Accuracy



## Akurasi Training



Kedua model menunjukkan kemampuan belajar yang sangat baik pada data training. Akurasi **RNN** meningkat konsisten dari 55% hingga 94%, sementara **LSTM** meningkat lebih cepat dan mencapai puncak yang lebih tinggi di 98%.



## Akurasi Validasi



Pada data validasi, performa **LSTM** jelas lebih unggul dengan stabil setelah epoch 8 dan naik mencapai 87%. Sebaliknya, **RNN** mengalami stagnasi setelah epoch 6 dan hanya mampu bertahan stabil di bawah 82%.



# RNN vs LSTM Loss

## ▶ Loss Training



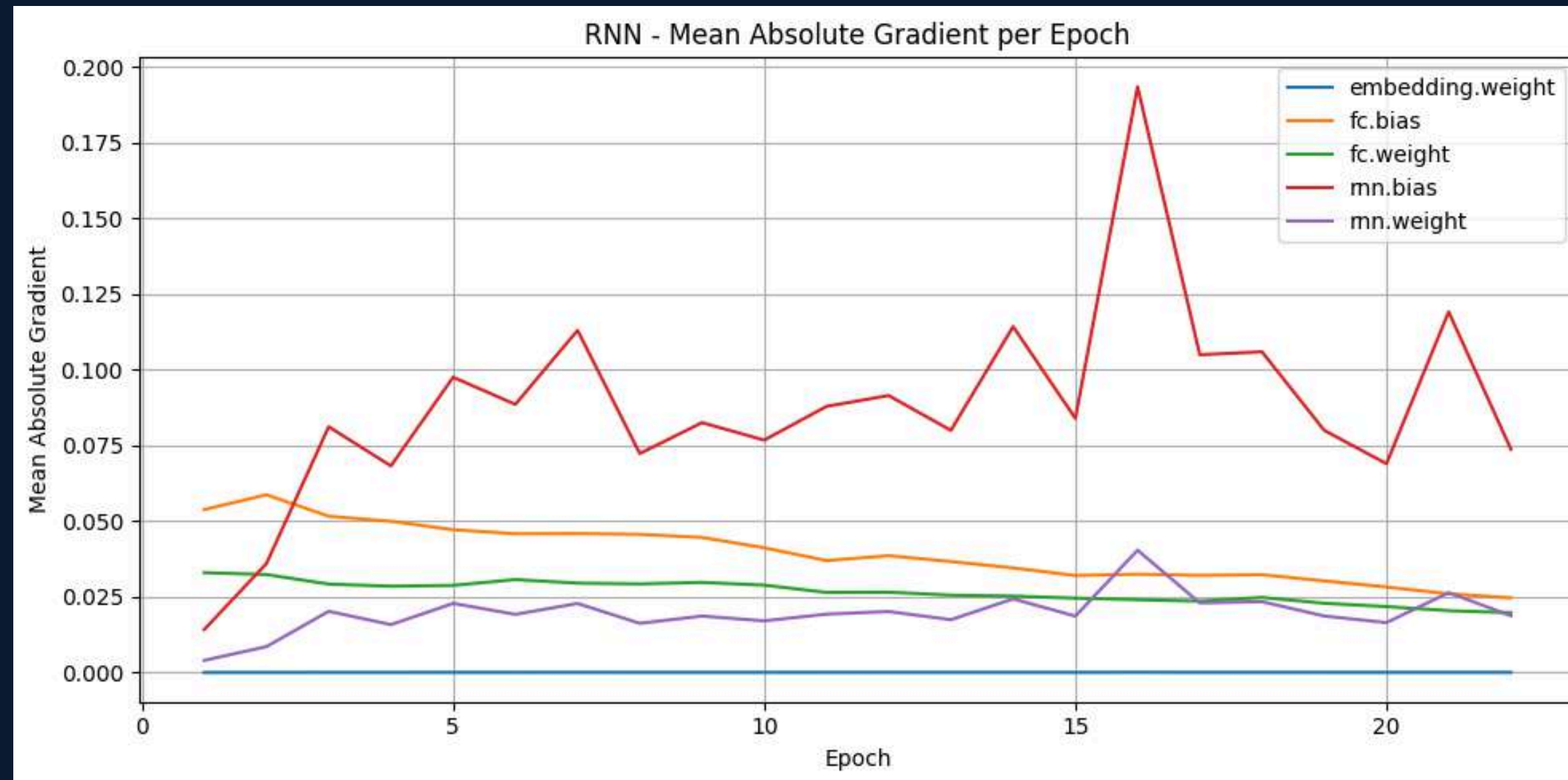
Loss pada data training untuk RNN terus menurun secara stabil, sedangkan LSTM menurun tajam dan konsisten, menandakan proses pembelajaran berjalan lancar.

## ▶ Validation Loss

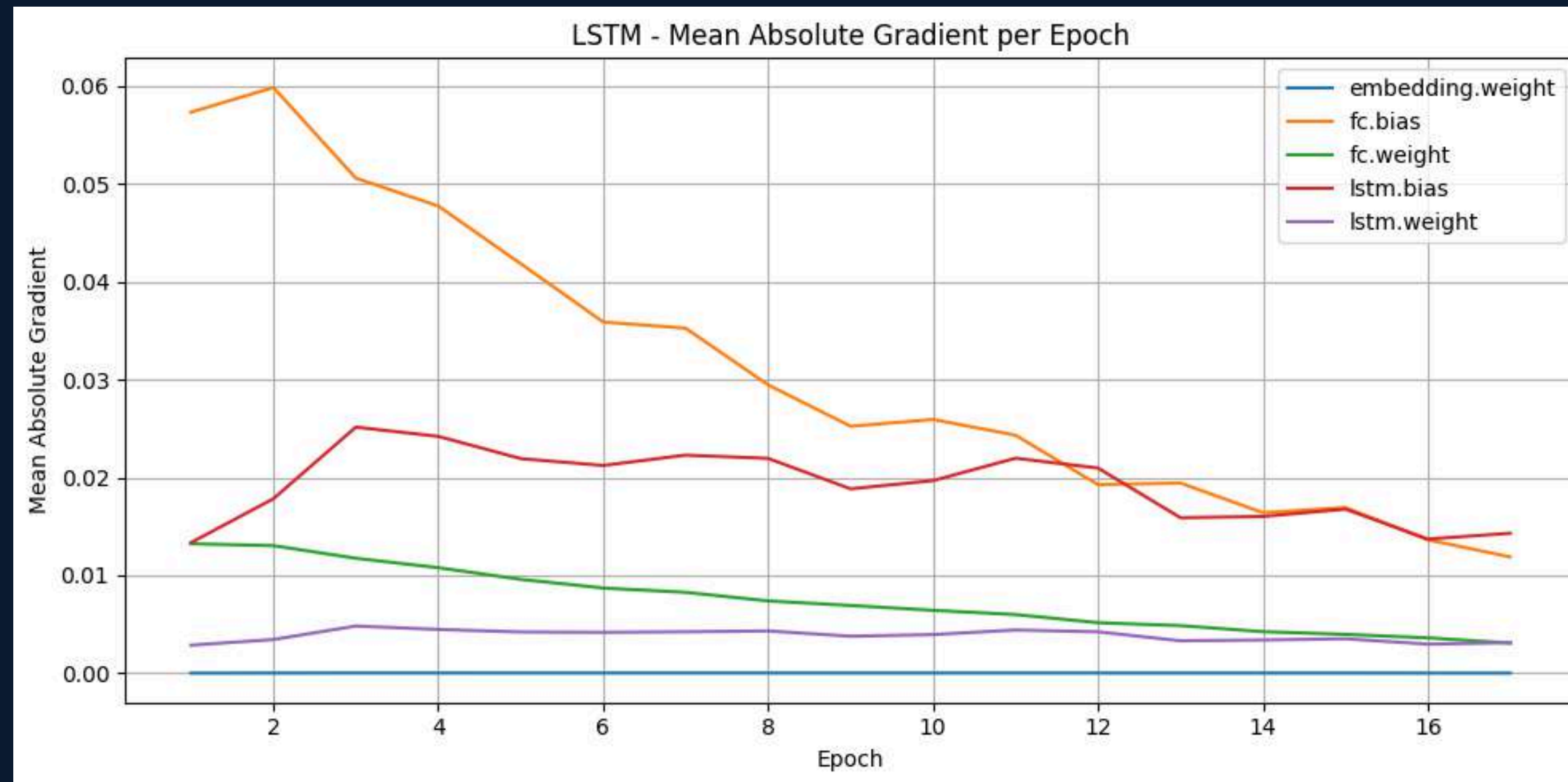


RNN menunjukkan overfitting yang lebih dini dan parah, yang terlihat jelas setelah epoch ke-8 atau ke-9. Meskipun LSTM juga mengalami overfitting, namun terjadi lebih lambat (sekitar epoch ke-13) dan tidak separah RNN

# RNN Gradient Analysis



# LSTM Gradient Analysis





# RNN vs LSTM Gradient

## ▶ Stabilitas & Skala Gradien



- **RNN:** Menunjukkan gradien yang tidak stabil dan sangat fluktuatif. Skala gradien sangat besar di beberapa lapisan namun sangat kecil (mendekati nol) di lapisan lainnya. Hal ini membuat RNN rentan terhadap masalah vanishing dan exploding gradients.
- **LSTM:** Memiliki gradien yang jauh lebih stabil dan terkontrol. Skalanya pun menengah dan konsisten di semua lapisan, tanpa ada indikasi vanishing gradient yang signifikan.

## ▶ Implikasi & Kesimpulan

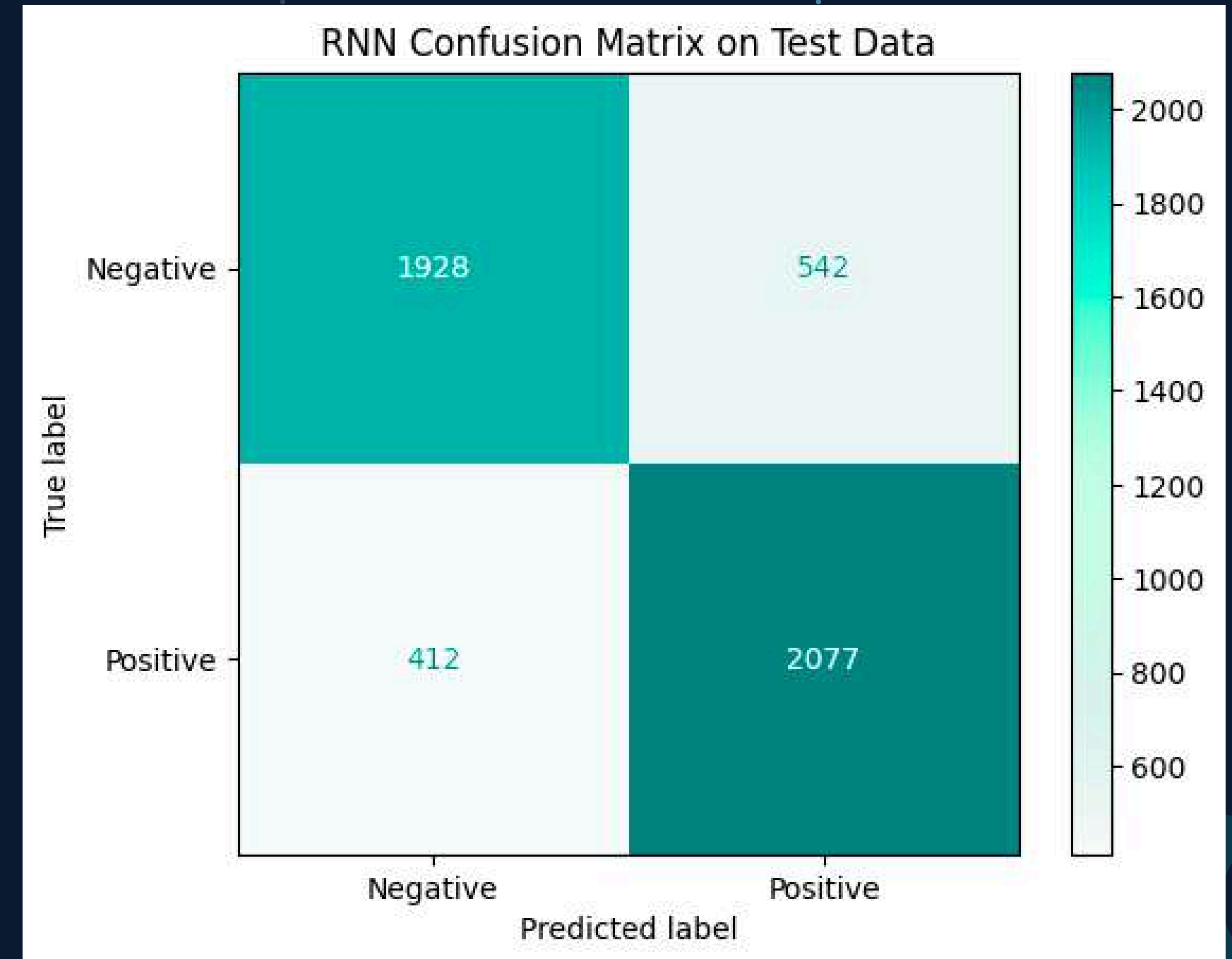


- **Implikasi Training:** Gradien RNN yang tidak stabil membuatnya sulit melakukan generalisasi dan rawan overfitting. Sebaliknya, LSTM belajar lebih konsisten dan efektif.
- **Kesimpulan:** Analisis ini membuktikan vanishing gradient nyata terjadi di RNN. Arsitektur LSTM berhasil mengatasi masalah ini, memungkinkannya belajar dari dependensi data jangka panjang.

# RNN Test Data Evaluation

## Classification Report:

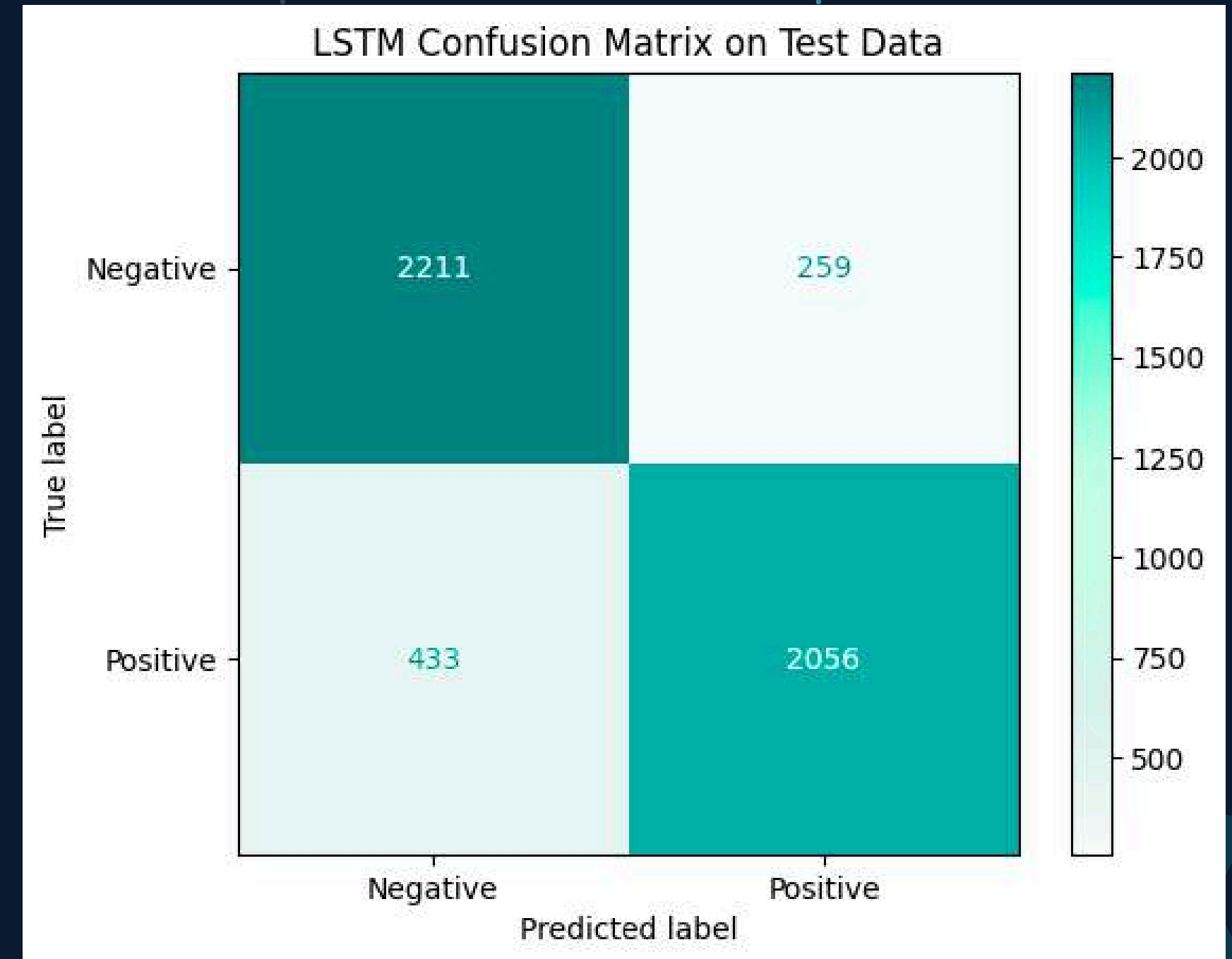
	precision	recall	f1-score	support
Negative	0.82	0.78	0.80	2470
Positive	0.79	0.83	0.81	2489
accuracy			0.81	4959
macro avg	0.81	0.81	0.81	4959
weighted avg	0.81	0.81	0.81	4959



# LSTM Test Data Evaluation

## Classification Report:

	precision	recall	f1-score	support
Negative	0.84	0.90	0.86	2470
Positive	0.89	0.83	0.86	2489
accuracy			0.86	4959
macro avg	0.86	0.86	0.86	4959
weighted avg	0.86	0.86	0.86	4959



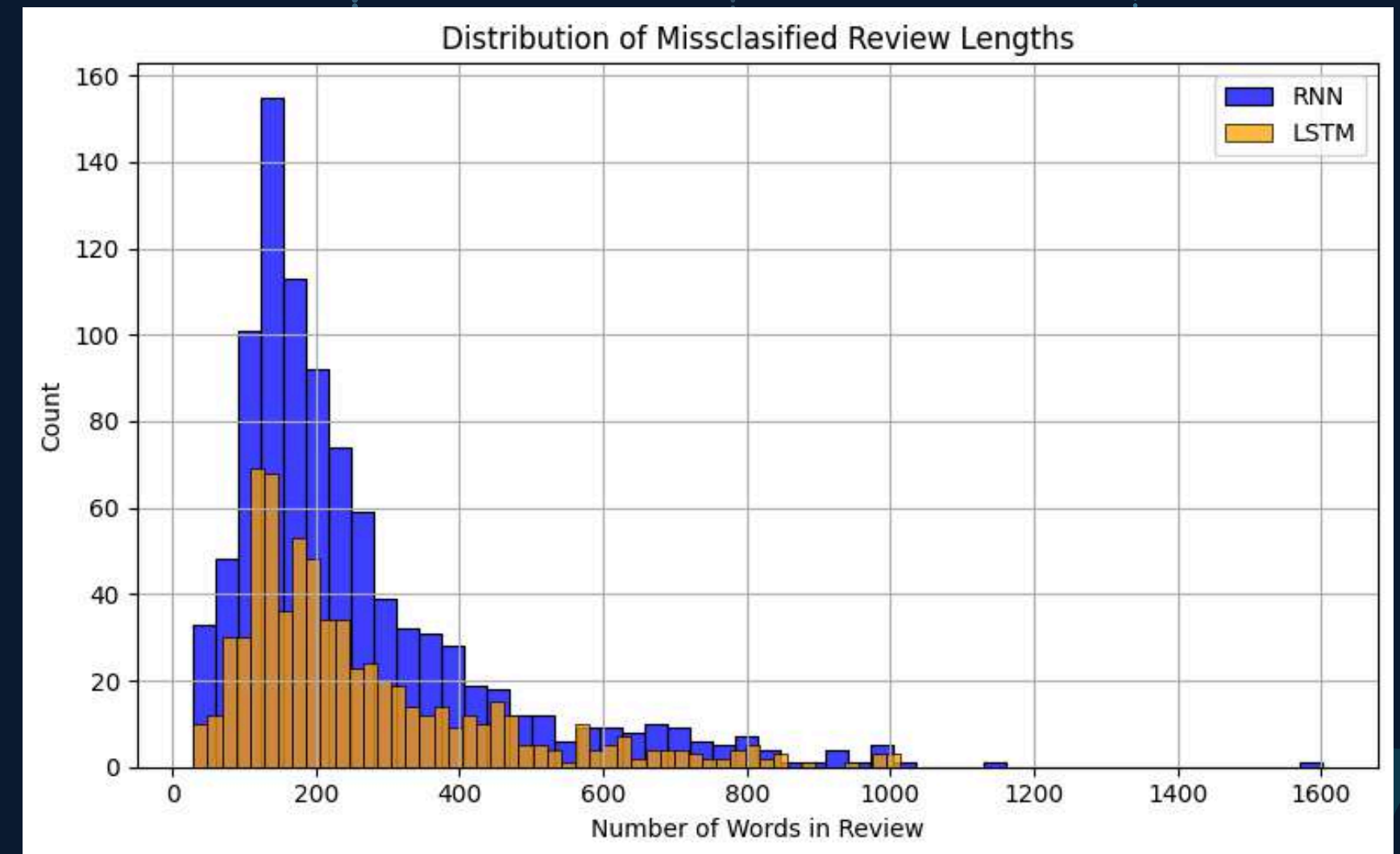
# Test Data Evalation

- ▶ **LSTM outperform RNN dalam hampir semua metrik evaluasi.**
- ▶ **Ini sejalan dengan teori bahwa LSTM lebih baik dalam mempertahankan informasi jangka panjang.**
- ▶ **Selain itu, hasil yang lebih seimbang antara precision dan recall menunjukkan LSTM lebih baik untuk digunakan.**



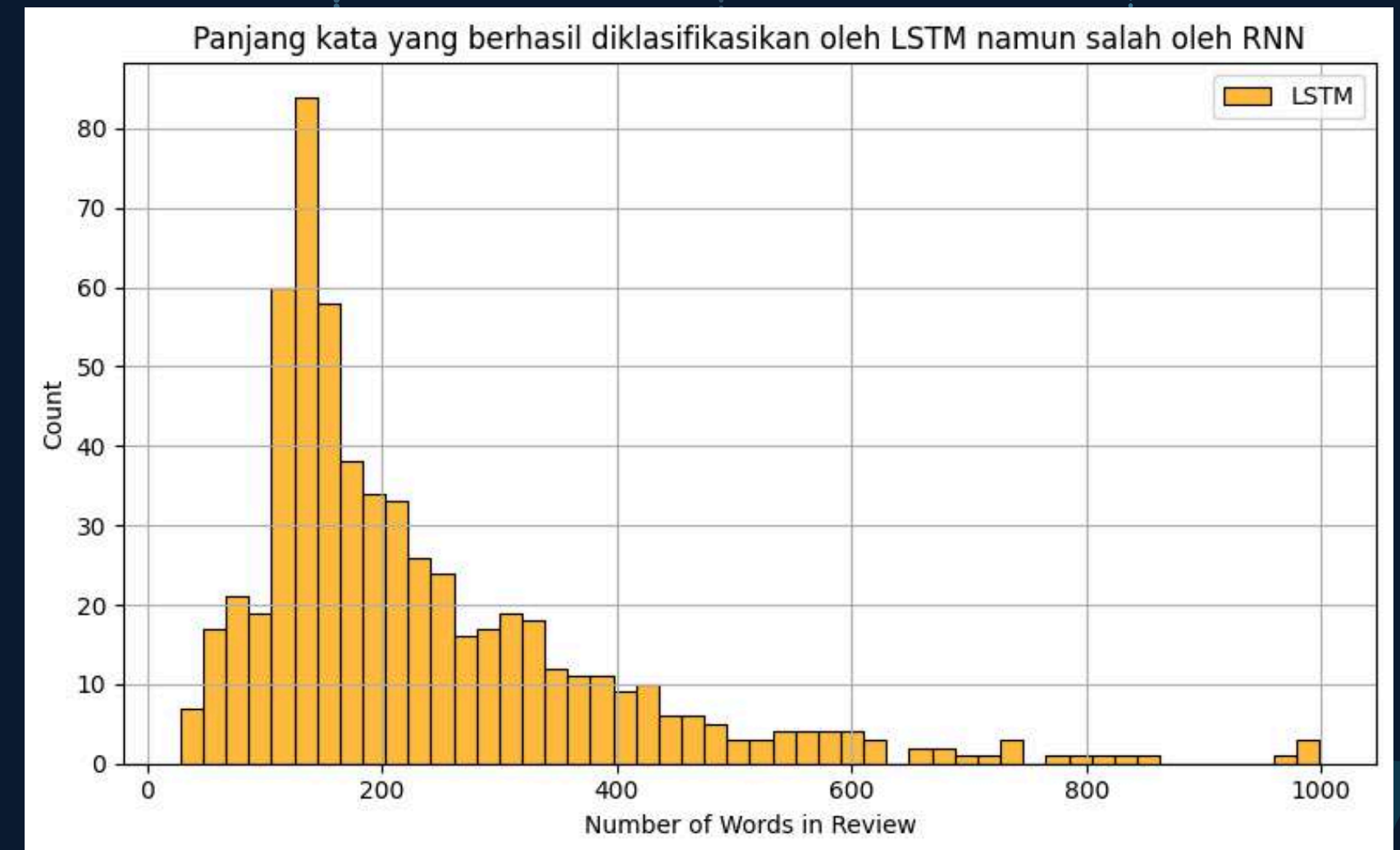
# At What Sequence Length Do They Struggle?

- RNN sering gagal pada review panjang (>400 kata), bahkan hingga >1000 kata.
- LSTM lebih stabil dan banyak keliru pada review pendek-menengah (100–300 kata).
- Review sangat panjang (>600 kata) hampir hanya salah oleh RNN.
- Kesimpulan: LSTM lebih mampu menangani long-term dependencies, sedangkan RNN kesulitan mempertahankan konteks pada teks panjang.



# When Is LSTM Stronger?

- Histogram menunjukkan review yang benar diklasifikasikan oleh LSTM tapi salah oleh RNN
- Mayoritas review tersebut memiliki panjang sekitar 250 kata
- LSTM tetap akurat bahkan untuk review dengan >800 kata
- Menunjukkan kemampuan LSTM dalam memahami konteks panjang
- RNN kesulitan mempertahankan konteks pada review panjang



# Project Summary



LSTM terbukti jauh lebih superior dibandingkan RNN untuk analisis sentimen pada dataset IMDB. Keunggulan utamanya adalah kemampuan LSTM dalam mengatasi vanishing gradient, masalah yang nyata terjadi pada RNN. Hal ini membuat LSTM lebih andal dalam memahami konteks pada review teks yang panjang, sehingga menghasilkan akurasi yang lebih tinggi dan seimbang.





FST UIN JKT

# Thank you

Thank you for taking the time to watch our UAS video.  
We hope it was clear and informative.  
Your feedback is always welcome.

Presented By

**Kelompok 5**

UAS NLP 2025  
RNN vs. LSTM with  
IMDB Dataset of 50K Movie Reviews

