

# Ringkasan Buku 1

Nama Buku: Tensorflow in Actions

Team: Wilhelmina Arlene, Luthfiah Maulidya, Fadhilah Dwi Istiani

Chapter 9 — *Natural Language Processing with TensorFlow: Sentiment Analysis*

## 🎯 Tujuan Utama Chapter 9

Chapter 9 memperkenalkan **NLP end-to-end** menggunakan TensorFlow dengan studi kasus **sentiment analysis**.

Jika pada Computer Vision kita bertanya:

"*Apa yang terlihat?*"

maka pada NLP kita bertanya:

**"Apa yang dimaksud oleh teks?"**

Fokus utama chapter ini:

- bagaimana **teks mentah diproses**
- bagaimana **teks diubah menjadi angka**
- bagaimana membangun **pipeline NLP yang rapi**
- bagaimana **RNN (LSTM)** digunakan untuk memahami sentimen
- bagaimana **embedding** memberi makna semantik

## GAMBARAN BESAR SENTIMENT ANALYSIS

### Apa itu Sentiment Analysis?

Sentiment analysis adalah tugas:

- mengklasifikasikan opini dalam teks
- biasanya menjadi:
  - positive
  - negative
  - (kadang neutral)

Contoh:

"This movie was absolutely amazing" → **Positive**

"The plot was boring and predictable" → **Negative**

## STRUKTUR BESAR CHAPTER 9

Chapter 9 dibagi menjadi **6 bagian utama**:

1. Eksplorasi & preprocessing teks
2. Persiapan teks untuk model
3. End-to-end NLP pipeline dengan TensorFlow

4. Sentiment analysis dengan LSTM
5. Training & evaluasi model
6. Word embeddings & semantic injection

## **9.1 What the Text? Exploring and Processing Text**

### **Tantangan Utama NLP**

Teks:

- tidak terstruktur
- panjang bervariasi
- penuh noise (tanda baca, huruf besar, typo)

Berbeda dengan image:

- tidak ada ukuran baku
- tidak ada "pixel"

### **EDA teks sama pentingnya dengan EDA image**

Yang dianalisis:

- panjang review
- distribusi kata
- kata paling sering muncul
- perbedaan kata antara sentimen positif & negatif

Tujuan:

- menentukan preprocessing strategy
- menentukan panjang sequence optimal

## **9.2 Getting Text Ready for the Model**

### **9.2.1 Split Dataset**

Dataset dibagi menjadi:

- training
- validation
- testing

Prinsip:

- test set **tidak boleh disentuh** selama training

### **9.2.2 Vocabulary Analysis**

Masalah NLP:

- vocabulary bisa sangat besar
- banyak kata jarang muncul

Solusi:

- batasi ukuran vocabulary
- buang kata terlalu jarang

Trade-off:

- vocabulary kecil → cepat, tapi kehilangan nuance
- vocabulary besar → berat, tapi kaya makna

### **9.2.3 Sequence Length Analysis**

Karena model butuh input tetap:

- teks harus:
  - dipotong (truncate)
  - atau dipadding
- menganalisis distribusi panjang kalimat
- memilih panjang yang mencakup mayoritas data

### **9.2.4 Text → Numbers (Tokenization)**

Menggunakan **Keras TextVectorization**:

- lowercase
- tokenization
- mapping kata → integer

Hasil:

- setiap kalimat = sequence angka

Ini adalah **langkah paling krusial NLP**.

## **9.3 End-to-End NLP Pipeline with TensorFlow**

### **Filosofi End-to-End Pipeline**

Pipeline ideal:

- preprocessing
- model
- inference

### **semua berada dalam graph TensorFlow**

Manfaat:

- konsistensi training & inference
- mudah deploy
- mengurangi bug preprocessing

### **Pipeline Komponen**

1. TextVectorization layer
2. Embedding layer
3. Sequence model (LSTM)
4. Output layer

## **9.4 Happy Reviews Mean Happy Customers: Sentiment Analysis**

### **Kenapa LSTM?**

RNN biasa:

- sulit mengingat konteks panjang

LSTM:

- memiliki gate (input, forget, output)
- mengatasi vanishing gradient
- cocok untuk teks panjang

## Arsitektur Model

Struktur umum:

Input Text

- TextVectorization
- Embedding
- LSTM
- Dense
- Output (sigmoid / softmax)

Embedding:

- mengubah integer → vector bermakna
- membuat model memahami kesamaan kata

## 9.5 Training and Evaluating the Model

### Training Strategy

- optimizer: Adam
- loss: binary cross-entropy
- metric: accuracy

### Evaluasi Model

Yang dianalisis:

- training vs validation accuracy
- overfitting
- kesalahan prediksi

### Accuracy saja tidak cukup

Harus dicek:

- false positive
- false negative
- contoh teks yang salah diklasifikasi

## 9.6 Injecting Semantics with Word Vectors

### Apa itu Word Embeddings?

Embedding memetakan:

- kata → vektor

- kata mirip → vektor dekat

Contoh:

- $king - man + woman \approx queen$

## Trainable vs Pretrained Embeddings

### 1. Trainable embeddings

- dilatih dari nol
- cocok jika data besar

### 2. Pretrained embeddings

- Word2Vec, GloVe
- membawa pengetahuan bahasa umum

Menjelaskan:

- embedding meningkatkan performa
- terutama pada data terbatas