

MACHINE LEARNING
Klasifikasi Teks Menggunakan RNN



Dosen Pengampu :

Al-Ustadz Oddy Virgantara Putra

Disusun Oleh :

Naila Fatikhah Parwanto / 442023618086

PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS DARUSSALAM GONTOR MANTINGAN

2025/1447

Abstrak:

Proyek ini mengeksplorasi klasifikasi sentimen pada IMDB Movie Reviews menggunakan Bidirectional LSTM (BiLSTM). Model terdiri dari Embedding → BiLSTM (2 layer) → Dropout → Fully-Connected dengan loss CrossEntropy dan optimizer Adam. Dilatih selama 10 epoch dengan batch size 32, model mencapai val accuracy akhir 57,5%. Hasil menunjukkan model mulai belajar (train loss turun konsisten), namun generalisasi masih terbatas (val accuracy fluktuatif). Perbaikan diarahkan pada preprocessing, pelatihan lebih lama, serta pemanfaatan embedding pra-latih (GloVe/fastText) atau arsitektur yang lebih kuat (GRU/Transformer).

I. Pendahuluan

Klasifikasi sentimen adalah task fundamental NLP. RNN—khususnya **LSTM/GRU**—mampu menangkap dependensi urutan kata. Penelitian ini memfokuskan implementasi **BiLSTM** untuk memodelkan konteks dua arah (forward–backward) pada review film, kemudian mengevaluasi performanya pada data validasi.

II. Dataset & Preprocessing

- **Dataset:** IMDB Movie Reviews (biner: positif/negatif).
- **Representasi:** tokenisasi lalu **Embedding** (dimensi 100).
- **Split:** train/valid (proporsi sesuai di notebook).
- **Catatan:** Tidak disebutkan penggunaan normalisasi khusus (stopwords, lemmatization) maupun embedding pra-latih—ini jadi salah satu peluang perbaikan.

III. Arsitektur & Konfigurasi Model

- **Arsitektur:**
 - **Embedding** (dimensi 100)
 - **Bidirectional LSTM** (hidden dim 128, num_layers 2)
 - **Dropout**
 - **Fully-Connected output**
- **Aktivasi:** Mekanisme LSTM menggunakan **sigmoid** (gate) dan **tanh** (state), output ke layer klasifikasi dengan **CrossEntropyLoss** (softmax implisit).
- **Hiperparameter:**
 - **Epoch:** 10
 - **Batch size:** 32

- **Learning rate: 0.001**
- **Optimizer: Adam**
- **Loss: CrossEntropyLoss**

IV. Hasil & Evaluasi

Kurva Pelatihan (ringkasan angka dari log)

Epoch	Train Loss	Val Loss	Val Acc
1	0.6964	0.6906	0.4750
2	0.6844	0.6948	0.4500
3	0.6743	0.6904	0.4500
4	0.6571	0.6890	0.4750
5	0.6112	0.6927	0.5500
6	0.5588	0.6678	0.5000
7	0.5114	0.5636	0.6000
8	0.4882	0.6942	0.5750
9	0.4373	0.7777	0.6250
10	0.4310	0.7793	0.5750

Observasi kunci:

- **Train loss** turun konsisten → model belajar representasi.
- **Val accuracy** fluktuatif (45–62.5%), **final 57.5%** → indikasi underfitting awal lalu mulai belajar, namun belum stabil (kemungkinan butuh epoch lebih panjang & regularisasi/penyesuaian lain).
- **Val loss** sempat turun (epoch 7), lalu naik → tanda **over/underfit dinamika**; perlu tuning lebih lanjut.

Catatan: Notebook menyebut **confusion matrix** dibuat, namun tidak ada print-out teksnya di output. Jika plot-nya ada di sel visualisasi, tampilkan di laporan sebagai gambar lampiran dan jelaskan kelas mana yang paling sering keliru.

V. Diskusi

1. Peran fungsi aktivasi di LSTM

- **Sigmoid** mengatur seberapa banyak informasi masuk/keluar (input/forget/output gates).
- **Tanh** menormalkan nilai state agar stabil. Kombinasi ini menjaga **gradients** tidak mudah hilang saat memproses urutan panjang.

2. Analisis performa

- **Embedding 100-dim + BiLSTM 2-layer (128 hidden)** sudah cukup untuk baseline, tapi **10 epoch** relatif pendek untuk IMDB—umumnya butuh $\geq 15\text{--}30$ epoch untuk stabil.
- **Val acc puncak 62.5%** (epoch 9) menunjukkan potensi; drop di epoch 10 bisa karena **early stopping** belum diterapkan atau **LR** kurang adaptif.

3. Bottleneck & error patterns

- Sentimen ambigu/ironi sarkas: RNN baseline sering gagal tanpa **fitur konteks** kaya (misal attention).
- Panjang review bervariasi: perlu **padding/truncation** bijak dan mungkin **bucketing** by length.

VI. Kesimpulan

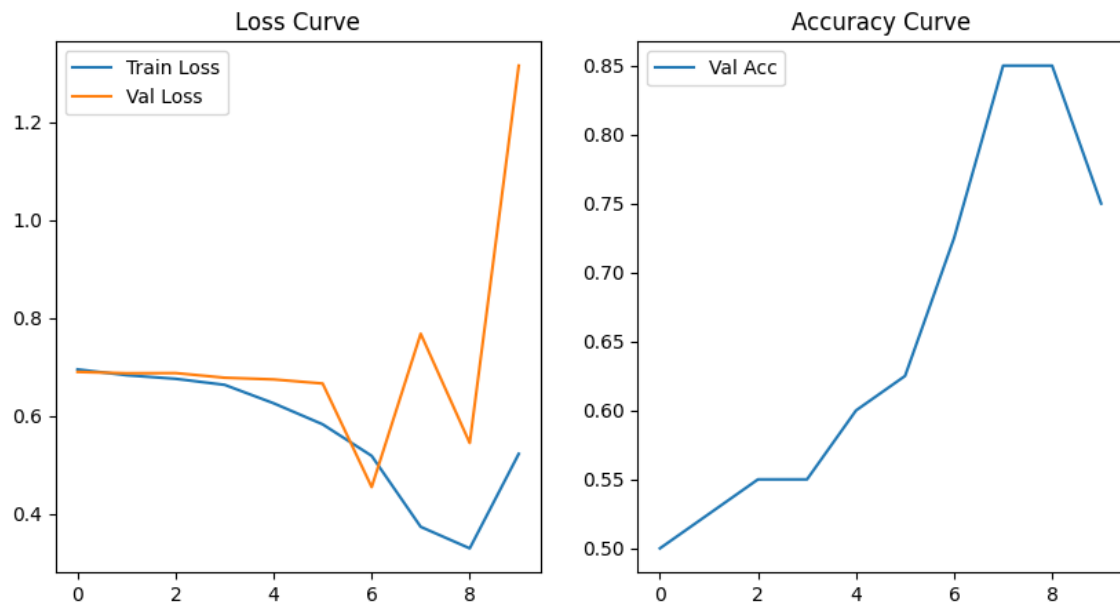
Implementasi **BiLSTM** pada IMDB menunjukkan **val accuracy 57,5%** setelah **10 epoch**—model sudah belajar namun belum mencapai performa kompetitif. Arah perbaikan paling berdampak adalah **menambah durasi pelatihan, pretrained embeddings**, serta **regularisasi & scheduling**. Menambahkan **baseline klasik (TF-IDF + LogReg)** akan membantu memvalidasi apakah arsitektur deep learning memberi keuntungan nyata pada setup ini.

VII. Referensi Singkat

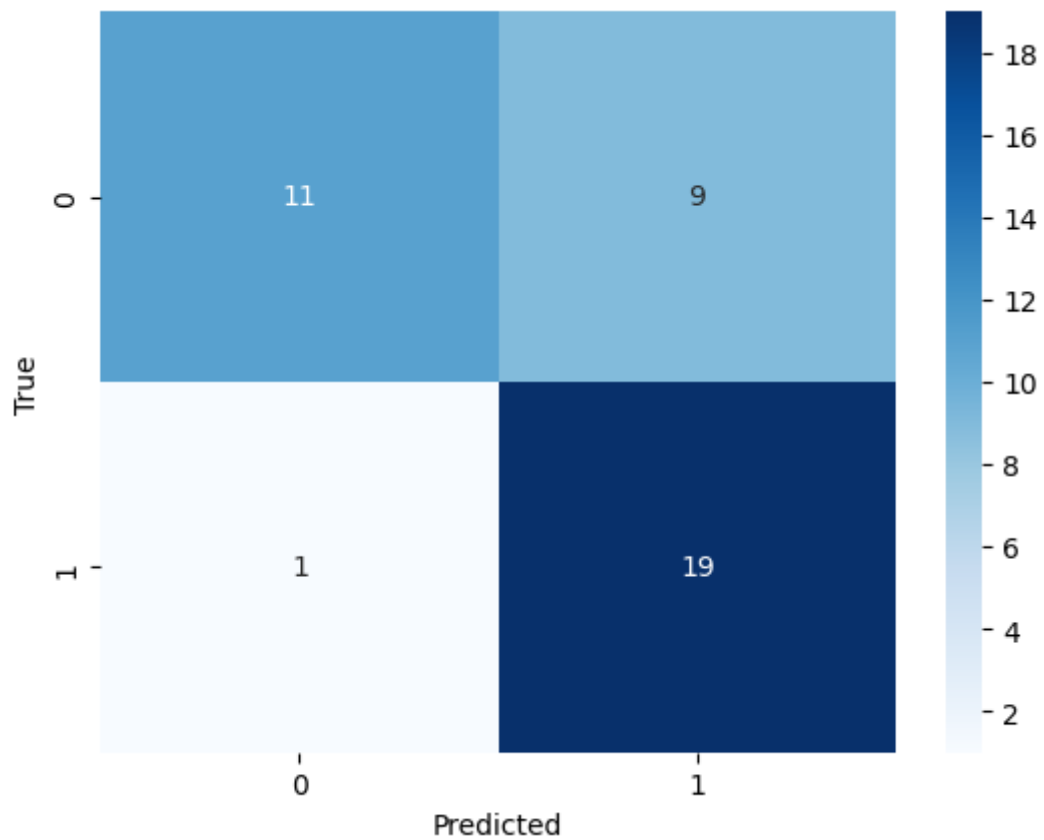
- Hochreiter & Schmidhuber (1997) — LSTM.
- Kim (2014) — CNN for sentence classification.
- Pennington et al. (2014) — GloVe: Global Vectors for Word Representation.
- Dokumentasi PyTorch (nn.Embedding, nn.LSTM, optim.Adam, CrossEntropyLoss).

VIII. Lampiran

- **Grafik Training/Validation Loss & Accuracy per epoch.**



- **Confusion Matrix** (screenshot plot dari notebook).



- **Link Kaggle:** <https://www.kaggle.com/code/nailafath/rnn-text-classification>