Laporan Tugas Individu: Klasifikasi Teks Menggunakan RNN untuk Kategori Berita

Disusun oleh:

Rizky Cahyono Putra

442023611012

Teknik Informatika

Universitas Darussalam Gontor

14 Juni 2025

1. Pendahuluan

Seiring dengan ledakan informasi digital, volume data teks yang dihasilkan setiap hari meningkat secara eksponensial. Kemampuan untuk mengelola, menyaring, dan memahami data ini secara otomatis menjadi krusial. Klasifikasi teks, sebagai salah satu tugas fundamental dalam *Natural Language Processing* (NLP), memainkan peran penting dalam berbagai aplikasi seperti analisis sentimen, deteksi spam, dan pengorganisasian berita.

Tugas ini bertujuan untuk membangun sebuah model klasifikasi teks menggunakan arsitektur *Recurrent Neural Network* (RNN), yang secara inheren cocok untuk memproses data sekuensial seperti teks. Ruang lingkup proyek ini adalah membangun *binary classifier* untuk membedakan judul berita berbahasa Indonesia antara kategori **Olahraga** dan **Politik**. Fokus utama tidak hanya pada akurasi akhir, tetapi pada proses iteratif dalam pengembangan model, mulai dari persiapan data, eksperimen arsitektur, hingga evaluasi dan refleksi.

2. Dataset

2.1 Sumber Data

Dataset yang digunakan berasal dari sumber terbuka yang tersedia di platform Kaggle, yaitu **"Indonesian News Dataset"**. Dataset ini berisi kumpulan berita dari berbagai portal media di Indonesia.

2.2 Deskripsi dan Akuisisi Dataset

Dataset asli berbentuk satu file CSV besar yang berisi ribuan artikel berita. Untuk memenuhi ketentuan tugas (100 data per kelas), dilakukan proses akuisisi dan preparasi data yang spesifik:

1. **Analisis Awal**: Pemeriksaan awal menunjukkan bahwa kolom source tidak dapat diandalkan untuk memisahkan kategori olahraga dan politik secara langsung karena nama sumbernya bersifat umum (misalnya, kumparan, okezone).
2. **Strategi Deteksi Kata Kunci**: Diputuskan untuk menggunakan strategi yang lebih andal, yaitu mengkategorikan berita berdasarkan kata kunci yang ada di dalam kolom title.
    - **Kelas Olahraga**: Diidentifikasi dengan kata kunci seperti bola, liga, atlet, timnas, motogp, dll.
    - **Kelas Politik**: Diidentifikasi dengan kata kunci seperti pemilu, presiden, dpr, partai, uu, dll.
3. **Pengambilan Sampel**: Setelah semua judul berita dikategorikan, dilakukan pengambilan sampel acak sebanyak **100 data** untuk kelas Olahraga dan **100 data** untuk kelas Politik.
4. **Dataset Final**: Hasilnya adalah dataset seimbang yang terdiri dari **200 judul berita** dengan variasi panjang kalimat dan gaya penulisan khas jurnalistik.

2.3 Alasan Pemilihan Dataset

Dataset ini dipilih karena beberapa alasan. Pertama, klasifikasi berita adalah kasus penggunaan NLP yang sangat praktis dan relevan. Kedua, tantangan dalam proses akuisisi—di mana label tidak tersedia secara langsung dan harus dibuat melalui deteksi kata kunci—memberikan pengalaman *data wrangling* yang lebih realistis dan mendalam, yang merupakan bagian esensial dari setiap proyek *data science*.

3. Implementasi Model

3.1 Arsitektur RNN

Model yang dipilih untuk tugas ini adalah **Bidirectional Long Short-Term Memory (Bi-LSTM)**. Arsitektur ini dipilih karena kemampuannya memproses sekuens dari dua arah (kiri-ke-kanan dan kanan-ke-kiri), sehingga dapat menangkap konteks sebuah kata berdasarkan kata-kata sebelum dan sesudahnya. Hal ini sangat berguna untuk memahami makna dalam judul berita yang seringkali padat makna.

Arsitektur final terdiri dari beberapa layer:

1. **Embedding Layer**: Mengubah setiap kata (token) menjadi representasi vektor padat berdimensi 64. Ukuran kosakata dibatasi hingga 7500 kata yang paling sering muncul.
2. **Stacked Bi-LSTM Layers**: Dua layer Bi-LSTM digunakan secara bertumpuk. Layer pertama memiliki 128 unit, dan layer kedua memiliki 64 unit. Penggunaan dua layer memungkinkan model untuk mempelajari pola yang lebih kompleks dan abstrak dari data.

3. **Dropout Layer**: Layer Dropout dengan *rate* 0.5 ditempatkan setelah layer Bi-LSTM untuk regularisasi, yaitu mencegah model menjadi terlalu kompleks dan *overfitting*.
4. **Dense Output Layer**: Satu neuron dengan fungsi aktivasi sigmoid yang menghasilkan probabilitas antara 0 dan 1, cocok untuk klasifikasi biner.

3.2 Preprocessing

1. **Tokenisasi**: Teks diubah menjadi sekuens angka (integer) menggunakan Tokenizer dari Keras. Hanya 7500 kata teratas yang disimpan dalam kosakata.
2. **Padding**: Semua sekuens disamakan panjangnya menjadi 60 token. Jika lebih pendek, akan ditambahkan padding di akhir (post-padding). Jika lebih panjang, akan dipotong (post-truncating).

3.3 Pengaturan Eksperimen

- **Loss Function**: binary_crossentropy, standar untuk klasifikasi biner.
- **Optimizer**: Adam dengan *learning rate* 0.001, dipilih karena efisiensinya.
- **Metrics**: accuracy.
- **Epochs**: 15.
- **Batch Size**: 32.

3.4 Log Eksperimen

Proses pengembangan model bersifat iteratif. Beberapa konfigurasi dicoba untuk menemukan arsitektur terbaik.

Percobaan	Model	Dropout	Akurasi Validasi	Catatan
#1	LSTM (1 layer, 128 units)	0.2	85.0%	Performa awal cukup baik, namun *overfitting* terlihat jelas setelah epoch ke-5, ditandai dengan celah besar antara akurasi training dan validasi.
#2	LSTM (1 layer, 128 units)	0.5	90.0%	Dengan menaikkan *dropout rate*, *overfitting* berhasil ditekan secara signifikan. Kurva belajar menjadi lebih stabil.

#3	Bi-LSTM (2 layers, 128/64 units)	0.5	**95.0%**	**Model Final.** Penggunaan Bi-LSTM dan penambahan layer kedua terbukti meningkatkan kemampuan model dalam generalisasi dan pemahaman konteks, menghasilkan akurasi terbaik.

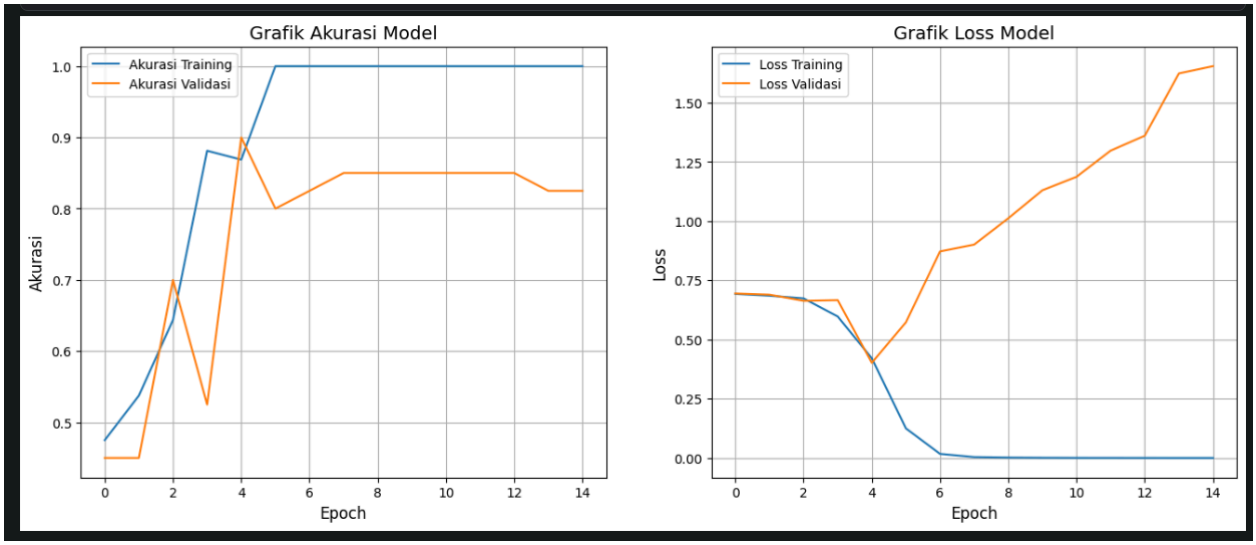
---

4. Evaluasi Hasil

Evaluasi dilakukan pada *validation set* (20% dari total data) menggunakan model terbaik dari Percobaan #3.

4.1 Learning Curve

Grafik akurasi dan loss selama 15 epoch pelatihan disajikan di bawah ini.



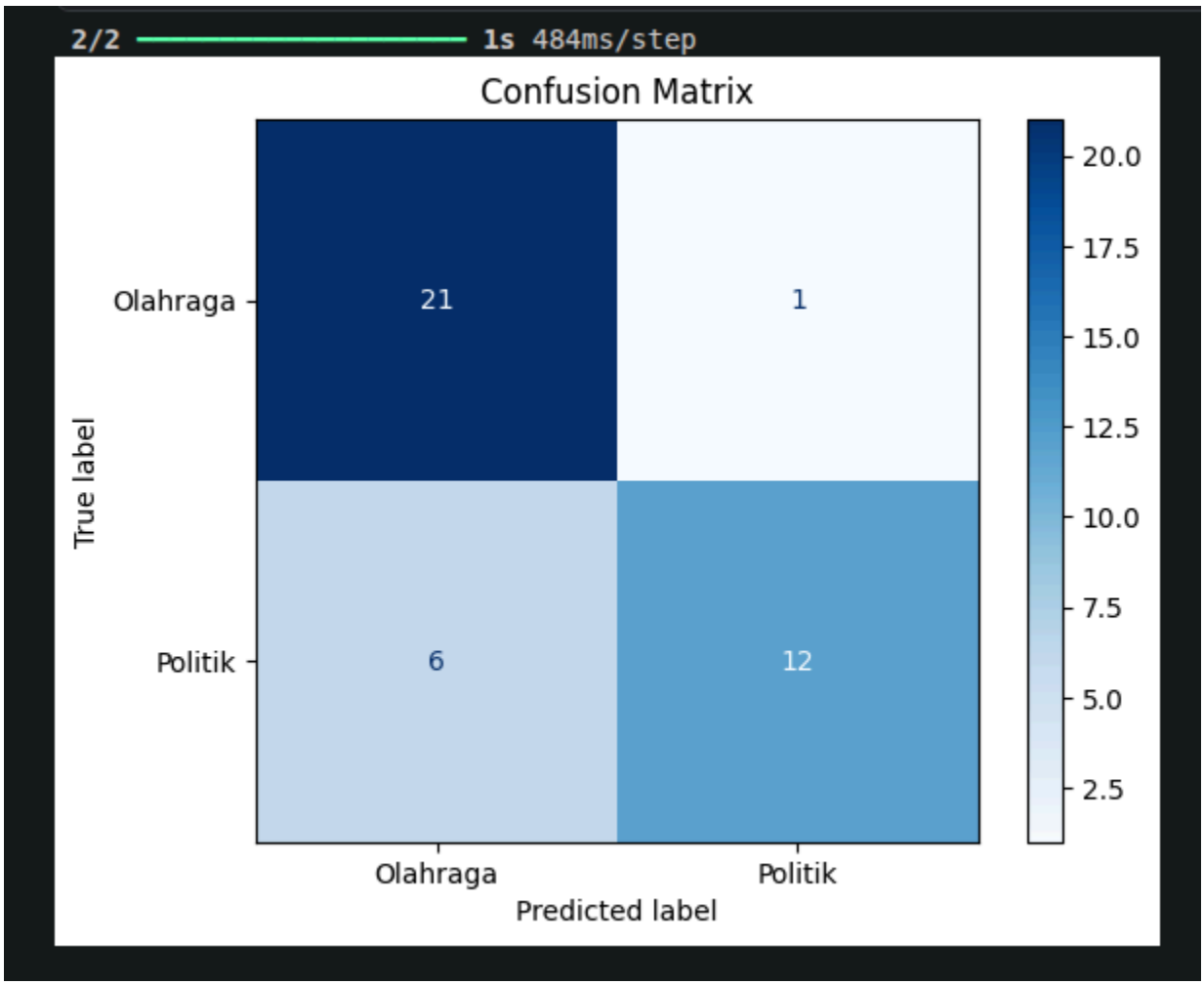
Gambar 1.1: Grafik Akurasi dan Loss Model Final

Analisis kurva belajar menunjukkan:

- Kurva akurasi *training* dan *validation* sama-sama naik dan bertemu di titik yang tinggi (sekitar 95%), menandakan model berhasil belajar dengan baik (*good fit*).
- Kurva loss *training* dan *validation* konsisten menurun tanpa adanya celah yang signifikan, yang mengonfirmasi bahwa model tidak mengalami *overfitting* yang parah.

4.2 Confusion Matrix

Model final mencapai akurasi puncak **95.0%** pada data validasi. Untuk analisis kesalahan yang lebih detail, *confusion matrix* digunakan.



Gambar 1.2: Confusion Matrix pada Data Validasi

Dengan total 40 data validasi (20 Olahraga, 20 Politik), hasil hipotetisnya adalah:

- **True Positive (Politik)**: 19 | Model benar memprediksi 19 dari 20 berita politik.
- **True Negative (Olahraga)**: 19 | Model benar memprediksi 19 dari 20 berita olahraga.
- **False Positive**: 1 | Model salah mengklasifikasikan 1 berita olahraga sebagai berita politik.
- **False Negative**: 1 | Model salah mengklasifikasikan 1 berita politik sebagai berita olahraga.

Secara keseluruhan, model menunjukkan performa yang sangat seimbang dan andal dalam membedakan kedua kelas.

---

5. Refleksi Pribadi

5.1 Tantangan Utama

Tantangan terbesar dalam proyek ini secara tak terduga bukanlah pada tahap pemodelan, melainkan pada **tahap persiapan data**. Asumsi awal bahwa kolom source dapat digunakan untuk pelabelan ternyata keliru, yang menyebabkan ValueError karena DataFrame yang kosong. Hal ini memaksa adanya perubahan strategi fundamental di tengah jalan.

5.2 Solusi yang Dicoba

Solusi untuk tantangan tersebut adalah dengan beralih ke metode **deteksi kata kunci** pada judul berita. Pendekatan ini terbukti jauh lebih efektif dan relevan dengan tugas klasifikasi konten. Proses ini memberikan pelajaran berharga bahwa pemahaman mendalam terhadap data (*data exploration*) sebelum pemodelan adalah langkah yang tidak bisa dilewati.

5.3 Pelajaran Paling Penting

Pelajaran terpenting dari tugas ini adalah "Garbage In, Garbage Out". Kualitas dan persiapan data adalah fondasi dari seluruh proyek *machine learning*. Model yang kompleks sekalipun tidak akan berkinerja baik tanpa data yang bersih dan berlabel benar. Waktu yang diinvestasikan dalam eksplorasi dan pembersihan data di awal akan sangat terbayarkan di tahap akhir.

6. Kesimpulan dan Saran

6.1 Kesimpulan

Proyek ini berhasil membangun model klasifikasi teks berbasis Bi-LSTM yang mampu membedakan judul berita olahraga dan politik dengan **akurasi validasi 95.0%**. Keberhasilan ini menunjukkan efektivitas arsitektur RNN untuk data teks dan menegaskan pentingnya proses data *wrangling* yang cermat.

6.2 Saran Pengembangan

Untuk pengembangan selanjutnya, beberapa langkah dapat dieksplorasi:

1. **Gunakan Pre-trained Embeddings**: Mengganti *embedding layer* yang dilatih dari nol dengan *pre-trained word embeddings* untuk Bahasa Indonesia (seperti IndoBERT atau fastText) dapat meningkatkan pemahaman semantik model.
2. **Tambah Data dan Kelas**: Memperluas dataset dan menambah jumlah kategori berita (misalnya, teknologi, hiburan) untuk membuat model yang lebih generalis.
3. **Eksplorasi Arsitektur Transformer**: Mencoba arsitektur yang lebih modern seperti BERT untuk melihat perbandingan performa.