BAB 2

TINJAUAN PUSTAKA

2.1 Pengertian Hoaks

Hoaks atau berita bohong merupakan informasi palsu yang disebarluaskan dengan tujuan tertentu, seperti menyesatkan, memengaruhi opini publik, atau sekadar menciptakan kehebohan. Istilah hoaks berasal dari bahasa Inggris hoax, yang berarti tipuan atau kebohongan yang dibuat untuk memperdaya orang lain. Menurut Tandoc, Lim, dan Ling [14], hoaks dapat dikategorikan sebagai bagian dari fake news, yaitu berita yang menyerupai berita sungguhan namun tidak berdasar pada fakta.

Wardle dan Derakhshan [15] membedakan hoaks ke dalam tiga bentuk utama: misinformasi (informasi salah yang disebar tanpa niat jahat), disinformasi (informasi salah yang disebar dengan sengaja untuk menipu), dan malinformasi (informasi benar yang digunakan untuk merugikan pihak tertentu). Dalam praktiknya, hoaks seringkali memanipulasi emosi pembaca, seperti rasa takut, marah, atau simpati, agar lebih mudah tersebar secara viral di media sosial.

Di Indonesia, hoaks menjadi isu serius seiring dengan meningkatnya penggunaan internet dan media sosial. Berdasarkan laporan Kementerian Komunikasi dan Informatika (Kominfo), ribuan konten hoaks ditemukan setiap tahun.

Sementara itu, di Malaysia, penyebaran berita palsu juga menjadi perhatian utama. Pemerintah Malaysia bahkan pernah mengesahkan Akta Anti Berita Tidak Benar 2018 untuk menangani penyebaran maklumat palsu, meskipun kemudian undang-undang tersebut dibatalkan dan digantikan dengan pendekatan hukum lainnya. Laporan dari Malaysian Communications and Multimedia Commission (MCMC) menunjukkan peningkatan kasus penyebaran hoaks terutama di platform WhatsApp dan Facebook, khususnya selama masa pemilu dan pandemi.

2.1.1 Sumber Dataset

Proses kompilasi dataset merupakan fase kritis untuk menjamin keandalan dan kemampuan generalisasi model deteksi berita hoaks. Untuk memastikan keragaman linguistik dan kekayaan kontekstual, dataset dalam bahasa Melayu dan Indonesia dikumpulkan dari repositori yang dapat diakses publik dan dirujuk

secara akademis. Seluruh dataset diintegrasikan ke dalam satu korpus terpadu guna memfasilitasi pembelajaran multibahasa dan generalisasi lintas bahasa.

Dataset bahasa Melayu diperoleh dari repositori publik GitHub [16] dan telah diproses ulang menggunakan tokenisasi dan padding standar untuk memastikan konsistensi format input.

Sementara itu, dataset bahasa Indonesia diambil dari dua sumber terpisah. Pertama adalah *Indonesian Fact and Hoax Political News Dataset* yang tersedia di platform Kaggle [17], berisi artikel berita politik berlabel fakta atau hoaks yang dikumpulkan melalui *web scraping* dari sumber terpercaya seperti CNN Indonesia, Kompas, Tempo, dan Turnbackhoax. Kedua adalah *Indonesia False News (Hoax) Dataset* [18] yang terdiri dari file *data_latih.csv* (pelatihan) dan *data_uji.csv* (pengujian) dengan total lebih dari 4.000 artikel berita (2018–2020). Dataset ini mencakup beragam topik seperti politik, kesehatan masyarakat, dan isu sosial terkait pandemi COVID-19, serta dilengkapi label hoax/fakta, tanggal publikasi, judul, teks narasi, dan nama file gambar.

2.2 Text Preprocessing

Proses *text preprocessing* merupakan tahap krusial dalam meningkatkan performa model Bi-LSTM, terutama dalam tugas klasifikasi teks. Tahapan ini bertujuan untuk mengubah data mentah menjadi bentuk yang lebih bersih, terstruktur, dan bermakna. Langkah-langkah seperti penghapusan URL, tag media sosial, simbol, serta karakter tidak relevan lainnya diperlukan untuk mengurangi gangguan atau *noise* pada data. Dengan demikian, proses ini membantu mengurangi dimensi fitur yang tidak penting, sekaligus mempertahankan informasi semantik yang esensial. Hal ini memungkinkan model Bi-LSTM untuk lebih fokus dalam mempelajari pola-pola linguistik yang relevan dalam proses klasifikasi [19].

2.2.1 Pembersihan Teks

Pembersihan teks (*text cleaning*) merupakan langkah penting dalam pengolahan bahasa alami untuk mengurangi noise dan inkonsistensi dalam data teks. Proses ini meliputi penghapusan karakter tidak relevan seperti simbol media sosial, URL, dan tanda baca yang tidak diperlukan, serta standardisasi format teks melalui lowercasing dan normalisasi karakter. Tujuannya adalah meningkatkan kualitas data masukan untuk proses analisis selanjutnya [20].

2.2.2 Tokenisasi

Tokenisasi adalah proses memecah teks menjadi unit-unit terkecil (token) seperti kata atau sub-kata. Dalam bahasa Indonesia yang kaya afiksasi, tokenisasi yang akurat penting untuk menangkap makna kata dasar. Proses ini sering dilanjutkan dengan penghapusan stopword (kata penghubung yang tidak bermakna) dan stemming (reduksi kata ke bentuk dasarnya) untuk menyederhanakan representasi teks [21].

2.2.3 Tokenisasi dan Padding

Tokenisasi numerik dan padding merupakan tahapan penting dalam praproses data teks untuk model *deep learning*. Proses ini bertujuan untuk mengubah teks menjadi representasi numerik berdimensi tetap yang dapat diterima oleh jaringan saraf tiruan.

A Tokenisasi Numerik

Tokenisasi numerik adalah proses konversi kata menjadi indeks integer berdasarkan frekuensi kemunculan dalam korpus. Tahapan yang dilakukan meliputi:

1. Pembentukan Kosakata: Menyusun daftar kata paling sering muncul hingga sejumlah V kata (ukuran kosakata), di mana V ditentukan sebagai hiperparameter.
2. Penambahan Token Khusus: Menambahkan token $\langle \text{OOV} \rangle$ (Out-of-Vocabulary) untuk kata yang tidak termasuk dalam kosakata, dan token $\langle \text{PAD} \rangle$ untuk kebutuhan padding.
3. Konversi Teks ke Angka: Setiap dokumen diubah menjadi sekuens angka (indeks) berdasarkan pemetaan kosakata yang telah dibuat.

Secara matematis, tokenisasi dinyatakan sebagai fungsi:

$$\mathcal{T} : \mathbb{W} \rightarrow \{1, 2, \dots, V\} \quad (2.1)$$

dengan $\mathbb{W}$ adalah himpunan kata dan V adalah ukuran kosakata.

B Padding

Karena panjang dokumen berbeda-beda, sementara model *deep learning* membutuhkan input berdimensi tetap, maka panjang sekuens diseragamkan dengan dua teknik:

1. Truncating: Sekuens yang melebihi panjang maksimum $\ell_{\max}$ akan dipotong.
2. Padding: Sekuens yang lebih pendek dari $\ell_{\max}$ akan ditambahkan nilai nol hingga mencapai panjang tersebut.

Strategi implementasi umum adalah:

1. Pre-truncating: Pemotongan dilakukan dari awal sekuens.
2. Post-padding: Padding ditambahkan di akhir sekuens.

Representasi ini memungkinkan pemrosesan *batch* secara efisien dan menjadi input standar bagi lapisan *embedding* dalam arsitektur Bi-LSTM [22, 23].

2.3 Long Short-Term Memory (LSTM)

LSTM merupakan jenis Recurrent Neural Network (RNN) yang dirancang untuk menangani masalah *vanishing gradient* pada RNN konvensional. Arsitektur ini diperkenalkan oleh Hochreiter & Schmidhuber (1997) [24] dan menjadi fondasi untuk berbagai model pemrosesan sekuensial.

2.3.1 Arsitektur LSTM

Inti LSTM terletak pada *memory cell* (C_t) yang mampu mempertahankan informasi dalam jangka panjang. Terdapat tiga gerbang (*gates*) yang mengatur aliran informasi:

1. *Forget Gate* (f_t): Menentukan informasi yang akan dibuang
2. *Input Gate* (i_t): Mengontrol informasi baru yang akan disimpan
3. *Output Gate* (o_t): Mengatur informasi yang akan dikeluarkan

2.3.2 Formulasi Matematis

Operasi LSTM pada waktu t dirumuskan sebagai berikut [25]:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.2)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.3)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (2.4)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (2.5)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.6)$$

$$h_t = o_t \odot \tanh(C_t) \quad (2.7)$$

dengan σ = fungsi sigmoid, $\odot$ = operasi Hadamard (perkalian elemen-per-elemen), x_t = input pada waktu t , dan h_t = *hidden state*.

2.4 Bidirectional LSTM (Bi-LSTM)

Bi-LSTM merupakan pengembangan dari LSTM yang mampu memproses informasi secara dua arah (maju dan mundur). Arsitektur ini efektif untuk menangkap konteks dalam data sekuensial seperti teks karena dapat memahami ketergantungan jangka panjang dan hubungan antar elemen secara lebih komprehensif [26].

2.4.1 Arsitektur Bi-LSTM

Bi-LSTM terdiri dari dua lapisan LSTM:

1. Lapisan *forward* memproses sekuens dari $t = 1$ ke $t = T$
2. Lapisan *backward* memproses sekuens dari $t = T$ ke $t = 1$

Output kedua lapisan digabungkan pada setiap langkah waktu t :

$$h_t = [\vec{h}_t \oplus \overleftarrow{h}_t] \quad (2.8)$$

dengan $\oplus$ menyatakan operasi konkatenasi.

2.4.2 Embedding

Lapisan embedding dalam Bi-LSTM berfungsi mengonversi token menjadi vektor padat (dense vector) yang merepresentasikan hubungan semantik antar kata. Representasi vektor ini memungkinkan model memahami kemiripan makna berdasarkan konteks penggunaannya [27].

2.5 Evaluasi Metrik

Evaluasi metrik merupakan alat ukur kuantitatif yang digunakan untuk menilai kinerja model klasifikasi [28]. Dalam penelitian deteksi berita hoaks, metrik ini mengevaluasi seberapa efektif model membedakan antara konten berita asli dan berita palsu. Metrik inti yang digunakan dalam penelitian ini diturunkan dari matriks konfusi (confusion matrix), yang memberikan dasar pemahaman tentang perilaku model [29].

2.5.1 Confusion Matrix

Matriks konfusi adalah representasi tabular yang membandingkan kelas prediksi dengan label aktual [30]. Untuk klasifikasi biner (hoaks vs berita asli), matriks ini terdiri dari empat komponen:

Tabel 2.1. Struktur Matriks Konfusi

	Prediksi Hoaks	Prediksi Asli
Aktual Hoaks	True Positive (TP)	False Negative (FN)
Aktual Asli	False Positive (FP)	True Negative (TN)

Keterangan:

- TP: Berita hoaks yang teridentifikasi benar
- TN: Berita asli yang teridentifikasi benar
- FP: Berita asli yang salah diklasifikasi sebagai hoaks (Kesalahan Tipe I)
- FN: Berita hoaks yang salah diklasifikasi sebagai asli (Kesalahan Tipe II)

Matriks ini memungkinkan analisis granular terhadap kesalahan klasifikasi dan menjadi dasar perhitungan metrik kinerja [31].

2.5.2 Metrik Klasifikasi Utama

Berdasarkan matriks konfusi, metrik berikut umum dilaporkan dalam evaluasi model:

1. Akurasi

Mengukur kebenaran klasifikasi secara keseluruhan [28]:

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.9)$$

2. Presisi

Mengukur keandalan prediksi positif [30]:

$$\text{Presisi} = \frac{TP}{TP + FP} \quad (2.10)$$

3. Recall

Mengukur kemampuan mendeteksi positif aktual [29]:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2.11)$$

4. F1-Score

Rata-rata harmonik yang menyeimbangkan presisi dan recall [32]:

$$\text{F1-Score} = 2 \times \frac{\text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} \quad (2.12)$$

2.5.3 Classification Report

Laporan klasifikasi [31] mengagregasi metrik-metrik tersebut per-kelas beserta:

1. Support: Jumlah instance aktual per kelas
2. Rata-rata Makro: Rata-rata metrik sederhana antar kelas [32]
3. Rata-rata Tertimbang: Rata-rata metrik yang ditimbang oleh support
4. Rata-rata Mikro: Dihitung dari total TP, FP, FN