BAB 2
TINJAUAN PUSTAKA

2.1 Berita Hoaks Berbahasa Indonesia

Berita hoaks, atau berita palsu, merupakan ancaman serius di era digital, terutama dengan pesatnya penggunaan media sosial di Indonesia. Penyebarannya yang cepat tanpa verifikasi dapat menimbulkan kebingungan, disinformasi, bahkan konflik sosial di berbagai bidang seperti politik, kesehatan, dan keamanan. Menurut *FirstDraft* [5], hoaks dapat diklasifikasikan menjadi tujuh jenis, yaitu: *Satir/Parodi*, yang tidak berniat jahat namun dapat mengecoh; *False Connection*, di mana judul tidak sesuai dengan isi berita; *False Context*, yaitu konten yang disajikan dengan konteks yang salah; *Misleading Content*, yang bertujuan untuk memelintir konten agar menjelekkan; *Imposter Content*, yang mencatut nama tokoh publik; *Manipulated Content*, yaitu konten yang dimanipulasi untuk mengecoh; dan *Fabricated Content*, yang merupakan konten sepenuhnya palsu.

Pencegahan penyebaran berita hoaks memerlukan peningkatan literasi digital masyarakat serta penerapan teknologi, seperti *machine learning*, untuk mendeteksi dan mengklasifikasikan berita palsu secara efektif. Teknologi ini mampu mengidentifikasi pola pada teks dan memisahkan berita hoaks dari berita valid, sehingga dapat berkontribusi dalam mitigasi penyebaran hoaks.

2.1.1 Sumber Dataset

Dalam penelitian ini, data yang digunakan untuk membangun dan melatih model klasifikasi berita hoaks diambil dari 4 sumber dataset yang berasal dari situs-situs pengecekan fakta serta portal berita nasional. Adapun sumber dataset yang digunakan meliputi:

1. **Mafindo (Masyarakat Anti Fitnah Indonesia)**
   MAFINDO adalah komunitas anti-hoaks yang dibentuk dengan tujuan memerangi penyebaran berita hoaks di Indonesia. Didirikan pada tahun 2016, MAFINDO telah berkembang menjadi lembaga yang memiliki lebih dari 95.000 anggota daring, serta lebih dari 1.000 relawan yang tersebar di lebih dari 20 cabang di berbagai wilayah Indonesia [6]. Tim profesional di MAFINDO secara aktif melakukan pengecekan fakta terhadap berita-berita yang diduga palsu dan menyebarkannya melalui berbagai platform.

2. **CekFakta**

   CekFakta.com adalah proyek kolaboratif yang diinisiasi oleh MAFINDO, Aliansi Jurnalis Independen (AJI), dan Asosiasi Media Siber Indonesia (AMSI) untuk melakukan pengecekan fakta terhadap berita yang tersebar di Indonesia. Website ini bertujuan untuk menyediakan klarifikasi dan fakta atas berita-berita yang banyak beredar, terutama yang memiliki potensi untuk menyesatkan masyarakat. Data dari *CekFakta* mencakup informasi berita yang telah melalui proses verifikasi dan penilaian.

3. **Tempo dan Kompas**

   Tempo dan Kompas merupakan dua portal berita nasional yang telah lama dikenal sebagai sumber berita terpercaya di Indonesia. Kedua media ini memiliki standar jurnalistik yang tinggi dan secara rutin melakukan pengecekan fakta sebelum mempublikasikan berita. Dalam penelitian ini, berita dari Tempo dan Kompas yang telah terverifikasi juga digunakan sebagai dataset untuk membedakan berita yang valid dengan berita hoaks.

Pengumpulan data dari sumber-sumber di atas dilakukan melalui teknik *web scraping*, yang memungkinkan peneliti untuk mengambil data secara otomatis dari situs-situs tersebut. Data yang dikumpulkan kemudian akan diproses dan digunakan untuk melatih model *machine learning* dalam mengklasifikasikan berita hoaks. Dengan memanfaatkan dataset dari berbagai sumber terpercaya ini, diharapkan model yang dibangun dapat lebih akurat dalam mendeteksi berita hoaks berbahasa Indonesia.

2.2 Metode Preprocessing Teks

Pra-pemrosesan teks adalah langkah penting dalam klasifikasi teks menggunakan *machine learning*. Pada tahap ini, data mentah yang diperoleh dari berbagai sumber diubah menjadi format yang dapat dipahami oleh model *machine learning*. Proses ini sangat penting karena kualitas data yang diproses dengan baik akan meningkatkan akurasi dan performa model. Secara umum, pra-pemrosesan mencakup beberapa tahap, seperti pembersihan teks, tokenisasi, dan *stemming*. Keseluruhan proses pra-pemrosesan teks memegang peranan penting dalam menentukan performa model *machine learning*, karena data yang telah diproses dengan baik memungkinkan algoritma untuk mempelajari pola secara lebih efektif dan efisien. Berdasarkan beberapa penelitian, tahap pra-pemrosesan dapat menyita

waktu hingga 50-80% dari keseluruhan proses klasifikasi, menunjukkan betapa pentingnya langkah ini dalam membangun model yang akurat dan handal [7].

2.2.1 Pembersihan Teks

Pembersihan teks bertujuan untuk menghilangkan elemen-elemen yang tidak relevan atau tidak berguna dari data mentah, seperti tanda baca, angka, simbol, atau karakter khusus. Langkah ini penting untuk memastikan bahwa data yang akan digunakan hanya berisi informasi yang relevan. Pembersihan teks dapat mencakup penghapusan tautan, penanganan spasi berlebih, serta normalisasi teks seperti mengubah semua huruf menjadi huruf kecil (*lowercasing*) untuk mengurangi variasi kata yang sama. Selain itu, langkah pembersihan data juga melibatkan penghapusan *missing value* pada dataset. *Missing value* adalah data yang hilang atau tidak terisi dalam dataset, yang dapat mengganggu proses analisis jika tidak ditangani dengan baik. Penghapusan *missing value* dilakukan dengan menghapus baris atau kolom yang mengandung data kosong. Data yang tidak bersih, termasuk adanya *missing value*, dapat menyebabkan hasil yang tidak akurat atau tidak berguna [8]. Oleh karena itu, pembersihan data adalah langkah krusial dalam memastikan kualitas data dan hasil analisis yang akurat. Berikut contoh hasil dari pembersihan teks.

Sebelum dibersihkan:

”Presiden Joko Widodo mengunjungi daerah bencana di Sulawesi Tengah pada hari Rabu (10/10/2018). https://www.kompas.com #bencana #sulawesitengah”

Setelah pembersihan:

”presiden joko widodo mengunjungi daerah bencana di sulawesi tengah pada hari rabu”

2.2.2 Tokenisasi

Tokenisasi adalah proses memecah teks menjadi unit-unit kecil yang disebut token. Token ini biasanya berupa kata atau frasa yang akan dianalisis oleh algoritma *machine learning*. Tokenisasi penting untuk memudahkan model dalam memahami struktur dan makna teks, karena algoritma akan memproses data dalam bentuk token, bukan sebagai kalimat utuh. Dalam bahasa Indonesia, tokenisasi dapat menjadi tantangan tersendiri karena beberapa kata gabungan atau awalan-pasangan dapat mempengaruhi hasil pemecahan. Tokenisasi yang baik

akan menghasilkan representasi teks yang lebih akurat dan dapat meningkatkan performa model klasifikasi [9]. Berikut contoh hasil dari tokenisasi.

Sebelum Tokenisasi:
”Presiden Joko Widodo mengunjungi daerah bencana di Sulawesi Tengah”

Setelah Tokenisasi:
’Presiden’, ’Joko’, ’Widodo’, ’mengunjungi’, ’daerah’, ’bencana’, ’di’, ’Sulawesi’, ’Tengah’

2.2.3 Stemming

Stemming adalah proses mengubah kata menjadi bentuk dasar atau akar katanya. Dalam bahasa Indonesia, banyak kata yang memiliki imbuhan seperti awalan atau akhiran, sehingga proses *stemming* bertujuan untuk mengurangi kata ke bentuk dasarnya agar variasi dari kata yang sama dapat diidentifikasi dengan tepat oleh model. *Stemming* membantu dalam mengurangi kompleksitas data dan meningkatkan konsistensi representasi kata. Proses ini sangat penting dalam klasifikasi teks karena dapat mengurangi dimensi data tanpa kehilangan makna penting.

Dalam penelitian ini, digunakan *library Sastrawi* sebagai alat untuk melakukan *stemming* pada teks berbahasa Indonesia. Berdasarkan hasil penelitian sebelumnya, *Sastrawi* lebih akurat dalam menghasilkan *stem* kata, dengan tingkat akurasi mencapai 92% dibandingkan dengan 82% untuk *Porter Stemmer* [7]. Hal ini menunjukkan bahwa *Sastrawi* lebih efektif dalam menangani karakteristik khusus bahasa Indonesia, sehingga dapat meningkatkan performa dalam tugas klasifikasi teks. Berikut contoh implementasi dari *Sastrawi Stemmer*.

Sebelum Stemming:
”Presiden mengunjungi daerah bencana untuk melihat dampak kerusakan dan memberikan bantuan kepada para korban.”

Setelah Stemming:
”presiden unjung daerah bencana untuk lihat dampak rusak dan beri bantu kepada para korban.”

2.2.4 Feature Extraction

Feature extraction adalah tahapan penting dalam pemrosesan teks yang bertujuan untuk mengubah data teks menjadi representasi numerik yang dapat digunakan oleh algoritma *machine learning*. Salah satu teknik yang umum digunakan untuk *feature extraction* pada teks adalah *Count Vectorizer*. *Count Vectorizer* bekerja dengan cara mengubah koleksi dokumen teks menjadi matriks angka, di mana setiap baris mewakili satu dokumen dan setiap kolom mewakili satu fitur atau kata unik dalam kumpulan dokumen. Nilai pada setiap sel matriks menunjukkan frekuensi kemunculan kata tersebut dalam dokumen yang sesuai. Berikut contoh dari hasil *Count Vectorizer* pada 2.1. Matriks ini menunjukkan frekuensi kemunculan setiap kata dalam masing-masing dokumen. Misalnya, kata ”saya” muncul satu kali dalam dokumen 1 dan tidak muncul dalam dokumen 2, sedangkan kata ”nasi” dan ”goreng” muncul pada kedua dokumen.

- Dokumen 1: ”saya suka makan nasi goreng”
- Dokumen 2: ”nasi goreng itu enak sekali”

Setelah melalui proses *Count Vectorizer*, kita akan mendapatkan matriks seperti berikut:

	saya	**suka**	**makan**	**nasi**	**goreng**	**itu**	**enak**
Dokumen 1	1	1	1	1	1	0	0
Dokumen 2	0	0	0	1	1	1	1

Tabel 2.1. Matriks frekuensi kata dari dua dokumen menggunakan *Count Vectorizer*

2.3 Logistic Regression

Logistic Regression atau *Logistic Regression* adalah salah satu algoritma pembelajaran mesin yang populer digunakan untuk masalah klasifikasi. Berbeda dengan *Regresi Linear* yang digunakan untuk memprediksi nilai kontinu, *Logistic Regression* digunakan untuk memprediksi probabilitas kelas diskrit, seperti klasifikasi biner (dua kelas) atau klasifikasi multikelas.

Logistic Regression bekerja dengan menggunakan fungsi sigmoid untuk memetakan nilai input ke probabilitas kelas output yang sesuai. Fungsi sigmoid, juga dikenal sebagai fungsi logistik, memiliki bentuk:

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{2.1}$$

di mana z adalah kombinasi linear dari fitur-fitur input x_i dengan bobot w_i dan bias b:

$$z = w_1x_1 + w_2x_2 + ... + w_nx_n + b \tag{2.2}$$

Fungsi sigmoid menghasilkan nilai antara 0 dan 1, yang dapat diinterpretasikan sebagai probabilitas kelas positif (kelas 1) untuk masalah klasifikasi biner. Jika probabilitas lebih besar dari ambang batas tertentu (biasanya 0,5), sampel diklasifikasikan sebagai kelas positif, sedangkan jika probabilitas kurang dari ambang batas, sampel diklasifikasikan sebagai kelas negatif (kelas 0). Keterangan:

$\sigma(z)$	fungsi sigmoid untuk memetakan probabilitas kelas
z	kombinasi linear dari fitur-fitur input
w_i	bobot untuk fitur ke-i
x_i	nilai fitur ke-i
b	bias
n	jumlah fitur
e	bilangan eksponensial alami (sekitar 2.718)

2.4 Ridge Classifier

Ridge Classifier, atau *Regularized Least Squares Classification*, adalah algoritma klasifikasi yang memperluas *Linear Regression* dengan menambahkan regularisasi L2. Algoritma ini bertujuan menemukan *hyperplane* pemisah optimal antara dua kelas dengan meminimalkan fungsi kerugian berikut:

$$J(w,b) = \frac{1}{2}\sum_{i=1}^{n}(y_i - (w^T x_i + b))^2 + \lambda \sum_{j=1}^{m} w_j^2 \tag{2.3}$$

Keterangan:

- n jumlah sampel pelatihan
- y_i label sebenarnya untuk sampel ke-i
- w vektor bobot
- x_i vektor fitur untuk sampel ke-i
- b bias
- λ parameter regularisasi
- m jumlah fitur

Nilai λ menentukan keseimbangan antara akurasi model dan kesederhanaannya. Nilai λ yang lebih besar menghasilkan model yang lebih sederhana, sedangkan nilai λ yang lebih kecil memungkinkan model lebih fleksibel untuk menyesuaikan data.