

**tbaPENGEMBANGAN SISTEM IDENTIFIKASI BERITA HOAX BAHASA
INDONESIA MENGGUNAKAN MULTINOMIAL NAIVE BAYES**

SKRIPSI

Disusun sebagai syarat untuk memperoleh gelar Sarjana Komputer (S.kom)



Disusun Oleh:

Muhamad Rizal Rifaldi

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS SULTAN AGENG TIRTAYASA

2025

LEMBAR PERSETUJUAN

SKRIPSI

PENGEMBANGAN SISTEM IDENTIFIKASI BERITA HOAX BAHASA INDONESIA MENGGUNAKAN MULTINOMIAL NAIVE BAYES

Disusun dan diajukan oleh

Muhamad Rizal Rifaldi

3337210044

**Telah disetujui oleh pembimbing untuk diajukan sebagai Proposal Penelitian
pada tanggal bulan tahun 2026**

Dosen Pembimbing I

Dosen Pembimbing II

Prof. Dr. Ir.Supriyanto, S.T., M.Sc.

NIP 197605082003121002

Mohamad Hilman, S.Kom., M.T.I

NIP 202102012156

Pembangan Sistem Identifikasi Berita Hoax Bahasa Indonesia menggunakan Multinomial Naive Bayes

Muhamad Rizal Rifaldi

ABSTRAK

Penelitian ini bertujuan untuk membangun sistem deteksi berita hoax dengan menerapkan algoritma *Multinomial Naive Bayes* (MNB) menggunakan metode pembobotan *Term Frequency–Inverse Document Frequency* (TF-IDF). Penelitian menggunakan dataset berita daring dan media sosial yang telah melalui proses pra-pemrosesan teks meliputi *cleaning*, *lowercasing*, *stopword removal*, dan *stemming*. Model MNB dilatih secara terpisah untuk masing-masing domain guna menyesuaikan karakteristik bahasa yang berbeda, kemudian dievaluasi menggunakan metrik akurasi. Hasil pengujian menunjukkan bahwa model mampu mengklasifikasikan berita hoax dan real dengan performa yang sangat baik, dengan akurasi sebesar 98,1% pada berita daring dan 81,4% pada media sosial. Pengujian berdasarkan variasi panjang teks juga menunjukkan bahwa penggunaan *fulltext* menghasilkan tingkat keyakinan prediksi yang lebih tinggi dibandingkan *headline*. Sistem diimplementasikan dalam bentuk aplikasi web berbasis *Streamlit* dengan antarmuka yang sederhana dan mudah digunakan, sehingga dapat dimanfaatkan sebagai alat bantu verifikasi awal terhadap informasi yang berpotensi hoax.

Kata Kunci: *Multinomial Naive Bayes*, TF-IDF, Deteksi Hoax, *Machine Learning*, Berita Daring

Development of an Indonesian Hoax News Identification System Using Multinomial Naive Bayes

Muhamad Rizal Rifaldi

ABSTRACT

This study aims to develop a hoax news detection system by applying the Multinomial Naive Bayes (MNB) algorithm with the Term Frequency–Inverse Document Frequency (TF-IDF) weighting method. The research utilizes online news and social media datasets that have undergone text preprocessing stages including cleaning, lowercasing, stopword removal, and stemming. The MNB model is trained separately for each domain to accommodate different language characteristics and evaluated using accuracy metrics. The results show that the proposed model achieves excellent classification performance, with an accuracy of 98.1% for online news data and 91.4% for social media data. Further testing based on text length variations indicates that full-text inputs produce higher prediction confidence compared to headlines. The system is implemented as a Streamlit-based web application with a simple and user-friendly interface, making it suitable as an initial verification tool for identifying potential hoax information.

Keyword: Multinomial Naive Bayes, TF-IDF, Hoax Detection, Machine Learning, Online News

DAFTAR ISI

ABSTRAK	iii
DAFTAR ISI	v
DAFTAR GAMBAR	vii
DAFTAR TABEL	viii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian	3
1.6 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	5
2.1 Penelitian Terdahulu.....	5
2.2 Ketertarikan Penelitian.....	6
2.3 Identifikasi Berita Hoax	6
2.4 Berita Hoax	7
2.5 Prapemrosesan Data	8
2.6 <i>Naive Bayes Multinomial</i>	9
2.7 <i>Term Frequency-Inverse Document Frequency</i>	11
2.8 <i>N-Gram</i>	12
2.9 <i>Confusion Matrix</i>	13
2.10 Perbandingan <i>Metode Machine Learning</i>	15
BAB III METODOLOGI	17
3.1 Alur Penelitian.....	17
3.2 Pengumpulan Dataset.....	18
3.3 <i>Preprocessing</i>	19
3.4 Pembobotan TF-IDF	21

3.5 Pelatihan Model <i>Multinomial Naive Bayes</i>	23
3.6 Evaluasi dan Analisis Hasil Pemodelan	25
3.7 Implementasi Website	25
3.8 Perangkat dan Sumber Daya Penelitian	26
BAB IV HASIL DAN PEMBAHASAN.....	28
4.1 Gambaran Umum Sistem	28
4.1.1 Input Data Teks	28
4.1.2 <i>Preprocessing</i> Teks	29
4.1.3 Penerapan Pembobotan TF-IDF pada Data Teks	30
4.1.4 Klasifikasi Menggunakan Model <i>Multinomial Naive Bayes</i>	32
4.1.5 Output Prediksi	33
4.2 Implementasi Sistem	33
4.2.1 Implementasi Load dan Standarisasi Dataset	34
4.2.2 Implementasi Pra-pemrosesan Teks.....	35
4.2.3 Implementasi Ekstraksi Fitur dan Split Data	35
4.2.4 Implementasi dan Pelatihan Model <i>Multinomial Naive Bayes</i>	36
4.2.5 Implementasi Fungsi Prediksi dan Penyimpanan Model.....	37
4.3 Hasil Pengujian	39
4.3.1 Hasil dan Contoh Dataset yang Dipakai	39
4.3.2 Hasil Pra-Pemrosesan Teks.....	40
4.3.3 Hasil Vektorisasi TF-IDF	42
4.3.4 Hasil Evaluasi Model Multinomial Naive Bayes	43
4.4 Uji Coba pada Data baru	45
4.5 Uji Coba pada Data penelitian Sebelumnya.....	47
4.6 Pengaruh Panjang Teks terhadap Tingkat Keyakinan Prediksi Model	48
BAB V PENUTUP.....	50
5.1 Simpulan	50
5.2 Saran.....	50
DAFTAR PUSTAKA.....	51

DAFTAR GAMBAR

Gambar 1. Diagram Alur Penelitian.....	17
Gambar 2. Alur Pengumpulan dan Standarisasi Dataset.....	18
Gambar 3. Alur Preprocessing Data.....	19
Gambar 4. Alur Pembobotan TF-IDF	22
Gambar 5. Flowchart Proses Pelatihan Model Multinomial Naive Bayes.....	24
Gambar 6. Flowchart Implementasi Sistem pada Aplikasi Web.....	26
Gambar 7. Diagram Alur Deteksi Hoax Menggunakan Algoritma MNB.....	28
Gambar 8. Tampilan Awal Website	29
Gambar 9. Interface Halaman Hasil Prediksi.....	33
Gambar 10. Implementasi Kode Load Dataset	34
Gambar 11. Kode Implementasi Fungsi <code>clean_text</code> pada Tahap Pra-pemrosesan Teks	35
Gambar 12. Kode Implementasi Ekstraksi Fitur TF-IDF dan Pembagian Data ...	36
Gambar 13. Kode Implementasi Pelatihan dan Evaluasi Model MNB	37
Gambar 14. Implementasi Fungsi Prediksi	37
Gambar 15. Implementasi Penyimpanan Model.....	38
Gambar 16. Implementasi Load Model	38

DAFTAR TABEL

Tabel 1 Penelitian Terkait.....	5
Tabel 2 Hasil Proses Tokenisasi Berdasarkan Jenis N-Gram.....	13
Tabel 3 Confusion Matrix untuk Klasifikasi Hoax	14
Tabel 4 Perbandingan Beberapa Algoritma Machine Learning.	16
Tabel 5 Contoh Hasil Prapemrosesan Teks	21
Tabel 6 Contoh Hasil Pembobotan TF-IDF	21
Tabel 7 Perangkat dan Sumber Daya Penelitian	27
Tabel 8 Hasil Pembobotan Kata Menggunakan Ekstraksi Fitur TF-IDF	31
Tabel 9 Probabilitas Kata terhadap Kelas Real dan Hoax pada Model MNB	32
Tabel 10 Dataset Berita yang digunakan.....	39
Tabel 11 Dataset Sosmed yang digunakan	40
Tabel 12 Contoh Hasil Pra-pemrosesan Data Berita.....	41
Tabel 13 Contoh Hasil Pra-Pemrosesan Data Sosmed.....	41
Tabel 14 Hasil Vektorisasi dan Pembagian Dataset Berita	42
Tabel 15 Hasil Vektorisasi dan Pembagian Dataset Sosmed	43
Tabel 16 Hasil Evaluasi Model Multinomial Naive Bayes Data Berita Daring ..	43
Tabel 17 Confusion Matrix Model Berita Daring	43
Tabel 18 Hasil Evaluasi Model Multinomial Naive Bayes Data Media Sosial ...	44
Tabel 19 Confusion Matrix Model Media Sosial	44
Tabel 20 Hasil Uji Coba Prediksi Model Berita pada Data Baru.....	45
Tabel 21 Hasil Uji Coba Prediksi Model Sosmed pada Data Baru.....	46
Tabel 22 Perbandingan Hasil Prediksi Model Penelitian Terdahulu dan Model Peneliti	47
Tabel 23 Perbandingan Rata-rata Probabilitas Prediksi pada Berita Real	48

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi berkembang cukup pesat belakangan ini, di mana hal ini sangat membantu setiap orang[1], teknologi yang berkembang di hampir setiap aspek sangat membantu pekerjaan manusia[2]. Penyebaran informasi lewat internet dan media sosial semakin cepat. Sayangnya, hal ini tidak sepenuhnya bergerak ke arah hal yang positif. Salah satu permasalahan terbesar diera digital ini adalah banyak sekali berita hoax atau berita palsu yang disebarluaskan melalui berbagai macam platform sehingga persebaran berita hoax ini menjadi sangat cepat[3].

Hoax biasnya diartikan sebagai berita bohong atau palsu yang dibuat secara sengaja, biasanya berisi ujaran kebencian, hasutan, judul yang provokatif, konten yang berisi sara, dan sumber media yang tidak jelas. Pengaruh berita hoax ini sangat buruk karena dapat menyesatkan opini publik, memicu perpecahan, serta membuat kepercayaan masyarakat terhadap informasi yang beredar. Selain itu, dampak dari berita hoax yang lain adalah dapat mempengaruhi stabilitas politik, membuat kepanikan yang tidak perlu, dan memiliki potensi menimbulkan konflik di berbagai sektor, seperti ekonomi, kesehatan, dan keamanan[3][4]. Dari situ peran teknologi kecerdasan buatan (AI) dan *machine learning* sangat dibutuhkan untuk mengidentifikasi pola-pola dalam berita yang mengindikasikan hoax.

Berdasarkan data dari Dewan Pers, terdapat sekitar 61.800 situs web di Indonesia yang mengklaim sebagai portal berita, namun hanya sekitar 1.700 di antaranya yang telah memiliki verifikasi sebagai situs berita resmi[3]. Hal ini menunjukkan bahwa setidaknya ada puluhan ribu situs lainnya yang berpotensi menjadi sumber penyebaran berita hoax yang perlu diwaspadai[6]. Oleh karena itu, diperlukan metode yang mampu mengidentifikasi dan mengklasifikasikan berita hoax. Salah satu pendekatan yang dapat digunakan adalah teknologi *machine learning*, yang merupakan bagian dari kecerdasan buatan.

Meachine learning memiliki kemampuan untuk mengenali pola dan tren dalam data sehingga dapat digunakan dalam berbagai bidang, termasuk deteksi

berita hoax. Dengan memanfaatkan beberapa algoritma seperti *Naive Bayes*, *Support Vector Machine* (SVM), atau *Deep Learning*, sistem menganalisis teks berita dan membuat keputusan apakah suatu informasi memiliki karakteristik hoax atau tidak[2][7]. Selain itu, dengan terus menerima data baru, model dapat terus meningkatkan akurasi seiring berjalananya waktu. Integrasi dengan teknik Natural *Language Processing* atau NLP juga memungkinkan model untuk memahami struktur bahasa, konteks kalimat, serta pola-pola tertentu yang sering muncul dalam berita hoax. Dengan demikian, *machine learning* menjadi salah satu opsi yang efektif untuk solusi dari maraknya berita hoax yang ada[8].

Seiring dengan meningkatnya penggunaan internet dan media sosial, tantangan dalam menyaring informasi yang valid menjadi semakin kompleks. Banyak pengguna internet yang membagikan berita tanpa melakukan verifikasi terlebih dahulu, sehingga penyebaran berita hoax menjadi semakin masif[9]. Faktor seperti rendahnya literasi digital, kecenderungan masyarakat untuk percaya pada informasi yang sesuai dengan pandangan mereka (*confirmation bias*), serta keberadaan algoritma media sosial yang memperkuat polarisasi informasi turut memperburuk situasi ini[10]. Oleh karena itu, diperlukan pendekatan berbasis teknologi yang dapat membantu mengatasi masalah ini secara efektif. Penerapan *machine learning* dalam deteksi berita hoax dapat menjadi solusi yang efisien karena mampu menganalisis data dalam jumlah besar secara cepat dan akurat, serta dapat terus belajar dari pola baru yang muncul dalam berita-berita yang beredar.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang ditelah dipaparkan sebelumnya, berikut rumusan masalah dari penelitian ini :

1. Bagaimana cara mendeteksi berita hoax menggunakan *machine learning*?
2. Sejauh mana akurasi model yang digunakan dalam mengklasifikasikan berita hoax?
3. Bagaimana memastikan antarmuka *website* deteksi berita hoax yang dikembangkan bersifat *user-friendly*?

1.3 Batasan Masalah

Meninjau dari uraian masalah sebelumnya, selanjutnya penelitian ini akan dibatasi supaya memperpadat isi laporan peneliti. Sehingga dapat disimpulkan batasan masalah sebagai berikut :

1. Berfokus pada berita dari media online, bukan media cetak atau televisi.
2. Hanya menggunakan berita dalam bahasa Indonesia.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, tujuan yang ingin dicapai dari penelitian ini adalah :

1. Mengembangkan model berbasis *machine learning* untuk deteksi berita hoax.
2. Menganalisis keakuratan model yang dibuat.
3. Mengembangkan antarmuka *website* yang ramah pengguna guna meningkatkan kemudahan penggunaan dan efektivitas sistem.

1.5 Manfaat Penelitian

Berdasarkan tujuan yang ingin dicapai, harapan dari penelitian ini, yaitu :

1. Diharapkan hasil dari penelitian ini bisa menjadi referensi untuk akademisi lainnya dalam mengembangkan model *machine learning* terutama pada pengklasifikasian.
2. Bagi masyarakat, diharapkan hasil penelitian ini bisa membantu dalam membedakan berita hoax dan berita asli, sehingga dapat meningkatkan literasi digital dan mengurangi penyebaran informasi yang salah.
3. Bagi media, diharapkan hasil penelitian ini bisa mempermudah proses verifikasi berita sebelum dipublikasikan, sehingga dapat menjaga kredibilitas media dalam menyampaikan informasi yang akurat.
4. Bagi peneliti, diharapkan hasil penelitian ini bisa menjadi dasar untuk penelitian lebih lanjut dalam pengembangan sistem deteksi berita hoax yang lebih akurat dan efisien menggunakan teknologi *machine learning*.

1.6 Sistematika Penulisan

Sistematika penulisan laporan skripsi ini disusun untuk memberikan kerangka kerja yang jelas mengenai alur dan fokus penelitian, dan terbagi menjadi lima bab. Laporan dimulai dengan Bab I Pendahuluan, yang berfungsi menguraikan latar belakang penelitian, merumuskan masalah, menentukan tujuan dan manfaat, serta mendefinisikan batasan batasan masalah. Kemudian, Bab II Landasan Teori dan Penelitian Terdahulu menyajikan kajian pustaka dan dasar teori yang mendasari penelitian ini termasuk algoritma *Naive Bayes*, teknik TF IDF, *N Gram*, dan *Confusion Matrix* sehingga relevansi penelitian dan *research gap* dapat diidentifikasi. Selanjutnya, Bab III Metodologi Penelitian berisi rincian pendekatan yang digunakan, mulai dari tahapan pengumpulan data, proses *preprocessing* data teks, hingga desain sistem dan pelatihan model klasifikasi. Bab IV Hasil dan Pembahasan memaparkan hasil penelitian, mencakup implementasi sistem, evaluasi kinerja model pada dua domain data, analisis perbandingan, dan interpretasi dari temuan-temuan yang dicapai, serta mengaitkan hasilnya dengan rumusan masalah dan tujuan penelitian. Terakhir, Bab V Kesimpulan dan Saran menutup laporan dengan menyajikan rangkuman temuan utama dan evaluasi dari hasil model, diikuti dengan saran-saran konstruktif untuk penelitian lanjutan, perbaikan metode, dan potensi penerapan praktis dari hasil penelitian ini.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Pada Tabel 1 adalah rangkuman beberapa penelitian terdahulu yang berkaitan dengan topik penelitian ini.

Tabel 1 Penelitian Terkait

No	Nama dan Tahun	Judul	Metode	Hasil
1	C. S. Sriyano & E. B. Setiawan (2021) [18]	Pendeteksian Berita Hoax Menggunakan <i>Naive Bayes Multinomial</i> Pada Twitter dengan Fitur Pembobotan TF-IDF	<i>Multinomial</i> <i>Naive Bayes & TF-IDF</i>	Fokus pada data Twitter, sedangkan penelitian ini menggunakan dataset berita Indonesia yang lebih umum dengan akurasi 72,06%
2	A. Zahra & M. N. Fauzan (2022) [17]	Sistem Identifikasi <i>Fake News</i> Menggunakan Metode <i>Multinomial Naive Bayes</i>	<i>Multinomial</i> <i>Naive Bayes & Twitter API</i>	Data hanya dari Twitter API (500 data), sedangkan penelitian ini menggunakan dataset lebih besar dari Kaggle dengan akurasi 83%
3	R. Qubra & R. A. Saputra (2024) [5]	<i>Classification of Hoax News Using the Naive Bayes Method</i>	<i>Naive Bayes & CountVectorizer</i> + TF-IDF	Menggunakan CountVectorizer, sedangkan penelitian ini memakai TF-IDF & N-Gram dengan akurasi 94,73%
4	B. Ferguson & W. Istiono (2024) [7]	<i>Hoax News Detection in Indonesian Political Headlines Using Multinomial Naive Bayes</i>	<i>Multinomial</i> <i>Naive Bayes & TF-IDF</i>	Fokus Pada Headline beritanya saja, dan topik yang dibicarakan hanya sebatas berita politik yang ada di indonesia

Berdasarkan hasil penelitian-penelitian terdahulu pada Tabel 1, dapat disimpulkan bahwa penggunaan metode *Multinomial Naive Bayes* (MNB) secara

konsisten memberikan hasil yang baik dalam klasifikasi teks, terutama ketika dikombinasikan dengan metode pembobotan TF-IDF. Namun, masih terdapat ruang untuk peningkatan terutama dalam konteks bahasa Indonesia, karena adanya variasi gaya bahasa, kosakata informal, serta campuran istilah lokal yang sering menyebabkan penurunan akurasi model. Oleh karena itu, penelitian ini mencoba memperbaiki kelemahan tersebut dengan menggabungkan TF-IDF dan N-Gram, serta mengimplementasikannya dalam sistem berbasis web yang interaktif.

2.2 Ketertarikan Penelitian

Penelitian ini memiliki keterkaitan dengan penelitian-penelitian terdahulu yang juga menggunakan algoritma MNB dalam mendekripsi berita hoax. Namun, penelitian ini berbeda karena berfokus pada berita berbahasa Indonesia dengan *dataset* yang lebih besar dan bervariasi dari Kaggle, serta mengombinasikan TF-IDF dan N-Gram sebagai fitur utama. Selain itu, penelitian ini dikembangkan lebih lanjut menjadi sebuah *website* interaktif yang memudahkan pengguna dalam mengidentifikasi berita hoax secara langsung, sehingga diharapkan dapat memberikan kontribusi praktis sekaligus memperluas penerapan metode MNB pada konteks yang lebih nyata.

2.3 Identifikasi Berita Hoax

Identifikasi berita hoax merupakan proses untuk menentukan apakah suatu informasi yang tersebar dalam bentuk berita termasuk kategori hoax atau tidak. Hoax sendiri merujuk pada informasi palsu yang dibuat secara sengaja untuk menyesatkan atau memprovokasi pembaca. Dalam konteks media daring, hoax dapat disebarluaskan secara masif dan cepat melalui berbagai platform seperti situs berita, media sosial, atau pesan instan. Oleh karena itu, identifikasi berita hoax menjadi langkah penting dalam menjaga kualitas informasi yang diterima masyarakat, serta mencegah dampak negatif yang ditimbulkan oleh penyebaran informasi palsu.

Dalam pengembangan sistem identifikasi otomatis, proses ini dilakukan dengan cara menganalisis konten teks dari berita yang beredar. Analisis ini dapat mencakup aspek linguistik, seperti pemilihan kata yang provokatif, penggunaan

sumber yang tidak kredibel, dan pola penyusunan kalimat yang tidak objektif. Sistem identifikasi berita hoax akan memanfaatkan data-data berita yang sudah diklasifikasikan sebelumnya untuk mengenali pola tertentu yang sering muncul dalam berita hoax. Dengan cara ini, sistem dapat membedakan antara berita hoax dan non-hoax berdasarkan karakteristik teks yang dianalisis.

Teknologi komputasi seperti klasifikasi teks menjadi pendekatan yang umum digunakan dalam membangun sistem identifikasi berita hoax. Dalam konteks ini, algoritma klasifikasi seperti MNB dapat diterapkan untuk melakukan proses identifikasi secara otomatis dan efisien. Model ini akan dilatih menggunakan data berita yang telah diberi label (hoax atau non-hoax), dan nantinya digunakan untuk memprediksi kategori berita baru. Dengan pendekatan ini, proses verifikasi informasi menjadi lebih cepat, akurat, dan dapat diakses oleh publik secara luas terutama jika diintegrasikan ke dalam platform berbasis web.

2.4 Berita Hoax

Berita hoax memiliki beberapa karakteristik seperti judul yang provokatif, konten berisi sara, ujaran kebencian, dan sumber yang tidak jelas[3]. Menurut kamus *oxford*, kata hoax diartikan sebagai suatu tindakan yang bertujuan untuk membuat seseorang mempercayai sesuatu yang tidak benar terutama sesuatu yang tidak menyenangkan[15]. Hoax adalah istilah popular yang sering digunakan secara luas, biasanya didefinisikan sebagai informasi yang bertentangan dengan fakta[15].

Data Kemekominfo tahun 2018 menyebutkan bahwa ada sekitar 800.000 situs di Indonesia yang telah terindikasi sebagai penyebar informasi palsu (hoax). Adapun hasil riset dari *DailySocial* menyebutkan bahwa sebesar 44,19% masyarakat Indonesia tidak bisa mendeteksi berita hoax. Sebanyak 73% responden selalu membaca seluruh informasi namun hanya 55% yang selalu memverifikasi keakuratan. hasil riset *DailySocial* tentang sebaran berita hoax terbanyak di jumpai pada media sosial. Persentasenya adalah pada platform Facebook (82,25%), *WhatsApp* (56,55%), dan Instagram (29,48%)[16].

Selain judul provokatif, konten SARA, ujaran kebencian, dan sumber tidak jelas, berita hoax juga sering kali memiliki karakteristik seperti :

1. Memanipulasi fakta dengan cara menggabungkan fakta dan informasi palsu agar terlihat lebih meyakinkan.
2. Menggunakan kata-kata yang membangkitkan emosi para pembaca sehingga pembaca dapat dengan cepat mempercayainya.
3. Sering kali menggunakan gambar atau video palsu yang telah diedit atau diambil dari luar konteks berita.
4. Menggunakan klaim palsu atau mengatasnamakan lembaga resmi untuk meningkatkan kredibilitasnya.

Dengan memahami karakteristik-karakteristik umum berita hoax seperti penggunaan judul provokatif, sumber tidak kredibel, serta klaim yang bersifat emosional, proses analisis teks secara otomatis menjadi lebih efektif. Fitur-fitur linguistik ini nantinya akan menjadi dasar dalam ekstraksi ciri (*feature extraction*) untuk membantu model mengenali pola-pola khas berita palsu.

2.5 Prapemrosesan Data

Pra-pemrosesan data adalah tahap awal yang penting dalam pengolahan teks, terutama untuk tugas klasifikasi seperti deteksi hoax, karena teks mentah sering mengandung noise seperti URL, angka, simbol, dan variasi huruf yang tidak relevan. Pada penelitian ini, pra-pemrosesan mencakup normalisasi huruf (lowercasing), penghapusan elemen non-alfabet, stopword removal, dan stemming untuk mengubah kata berimbuhan menjadi bentuk dasar. Stemming membantu menyatukan berbagai bentuk morfologis kata sehingga variasi kata yang memiliki arti serupa dapat dipetakan ke satu fitur yang konsisten. Beberapa penelitian menunjukkan bahwa kombinasi teknik pra-pemrosesan yang tepat dapat berpengaruh pada hasil klasifikasi, karena dapat mengurangi jumlah fitur tidak relevan dan meningkatkan kualitas representasi kata sebelum pembobotan TF-IDF dilakukan[22].

Meskipun operasi seperti stemming dapat mengubah bentuk kata secara linguistik, efeknya terhadap metode klasifikasi berbasis statistik justru sering bersifat positif karena mempermudah model dalam mengenali pola distribusi kata dalam korpus. Pra-pemrosesan yang efektif dapat memperbaiki akurasi model dengan menjaga fokus pada kata-kata informatif yang membantu membedakan

antar kelas, seperti yang dibuktikan oleh beberapa studi yang menerapkan langkah-langkah pra-pemrosesan dalam analisis sentimen dan klasifikasi teks berbasis TF-IDF dan algoritma klasifikasi lainnya[23].

2.6 Naive Bayes Multinomial

Naive Bayes adalah salah satu algoritma klasifikasi berbasis probabilitas yang didasarkan pada teorema bayes. Algoritma ini disebut “*Naive*” karena mengasumsikan bahwa setiap fitur dalam data bersifat independen satu sama lain, meskipun dalam kenyataannya tidak selalu demikian[17]. Dalam konteks klasifikasi teks, seperti deteksi berita hoax, model *Naive Bayes* bekerja dengan menghitung probabilitas suatu teks termasuk dalam kategori tertentu berdasarkan frekuensi kemunculan kata-kata dalam dokumen tersebut[18].

Salah satu varian dari *Naive Bayes* yang paling banyak digunakan dalam klasifikasi teks adalah *Multinomial Naive Bayes* (MNB). Model ini bekerja dengan menghitung frekuensi kata dalam dokumen dan membandingkannya dengan probabilitas distribusi kata dalam setiap kelas[17]. MNB sangat cocok untuk klasifikasi berita hoax, karena algoritama ini dapat mengidentifikasi pola kata yang sering muncul dalam berita hoax dan membandingkannya dengan berita non-hoax.

Naive Bayes adalah salah satu metode klasifikasi yang menggunakan pendekatan probabilitas dan statistika, dikembangkan berdasarkan teori yang dikemukakan oleh ilmuwan Inggris, Thomas Bayes. Algoritma ini memprediksi kemungkinan suatu kejadian di masa depan berdasarkan pola yang terbentuk dari data sebelumnya, sehingga dikenal sebagai Teorema Bayes. Terdapat beberapa varian model dalam algoritma *Naive Bayes*, dan dalam penelitian ini digunakan *Naive Bayes Multinomial*. Model ini memperhitungkan frekuensi kemunculan kata dalam suatu dokumen untuk menentukan kategorinya. Secara umum, *Naive Bayes* bekerja dengan memanfaatkan konsep probabilitas bersyarat (*conditional probability*), di mana model menghitung kemungkinan suatu dokumen termasuk ke dalam kelas tertentu berdasarkan distribusi kata yang muncul di dalamnya. Pendekatan ini sederhana namun sangat efisien, terutama pada data teks berukuran besar. Jika terdapat suatu dokumen d dan sekumpulan kelas c , maka probabilitas dokumen d termasuk dalam kelas c dapat dihitung menggunakan rumus:

$$P(c|term\ dokumen\ d) = P(c) \times P(t_1|c) \times P(t_2|c) \times P(t_3|c) \times \dots \times P(t_n|c) \quad (2.1)$$

$P(c|term\ dokumen\ d)$ merupakan probabilitas bahwa suatu dokumen termasuk dalam kelas c . t_n adalah kata ke- n dalam dokumen d , sementara $P(t_n|c)$ menunjukkan probabilitas kemunculan kata ke- n dalam kelas c . Adapun $P(c)$ merupakan probabilitas awal (*prior probability*) dari kelas c , yang dapat dihitung menggunakan rumus berikut:

$$P(c) = \frac{N_c}{N} \quad (2.2)$$

$P(c)$ merepresentasikan proporsi jumlah dokumen dalam kelas c terhadap total keseluruhan dokumen N . Sementara itu, rumus *Naive Bayes Multinomial* yang diterapkan dengan teknik pembobotan kata menggunakan TF-IDF dapat dinyatakan sebagai berikut:

$$P(t_n|c) = \frac{M_{ct} + 1}{(\sum_{W' \in V} W'_{ct}) + B'} \quad (2.3)$$

$P(t_n|c)$ merupakan probabilitas kemunculan kata ke- n dalam suatu dokumen yang tergolong dalam kelas c . W_{ct} adalah nilai bobot TF-IDF atau W dari kata t dalam kategori c . Sementara itu, $\sum_{W' \in V} W'_{ct}$ menggambarkan total bobot dari seluruh kata yang terdapat dalam kategori c . Adapun B' merepresentasikan jumlah bobot kata unik di seluruh dokumen, di mana nilai *Inverse Document Frequency* (IDF) tidak dikalikan dengan *Term Frequency* (TF)[11][18].

Dalam penelitian ini, ambang batas (*threshold*) probabilitas yang digunakan untuk menentukan apakah suatu teks dikategorikan sebagai hoax atau real ditetapkan sebesar 0,5 atau 50%. Nilai ini dipilih berdasarkan prinsip *maximum probability decision* pada algoritma MNB, di mana sistem secara otomatis menetapkan suatu teks ke dalam kelas dengan nilai probabilitas tertinggi. Artinya, apabila probabilitas teks termasuk ke dalam kelas hoax lebih besar dari 0,5, maka teks tersebut akan diklasifikasikan sebagai hoax. sebaliknya, jika kurang dari 0,5 maka dikategorikan sebagai real. Pemilihan nilai ambang 0,5 ini didasarkan pada konvensi umum dalam klasifikasi biner yang menempatkan titik keseimbangan antara dua kelas dengan peluang sama besar. Meskipun demikian, nilai ambang ini

bersifat fleksibel dan dapat disesuaikan (misalnya 0,6 atau 0,7) jika sistem diharapkan lebih sensitif dalam mendeteksi berita hoax. Pendekatan ini sejalan dengan praktik yang diterapkan dalam berbagai penelitian terdahulu yang menggunakan algoritma *Naive Bayes* untuk klasifikasi teks berbasis probabilitas

2.7 Term Frequency-Inverse Document Frequency

TF-IDF (*Term Frequency-Inverse Document Frequency*) adalah teknik dalam *preprocessing* teks yang digunakan untuk menentukan bobot suatu kata dalam dokumen. Metode ini berfungsi untuk mengidentifikasi kata-kata yang paling penting dalam dokumen tertentu berdasarkan frekuensi kemunculannya.

Konsep dasar TF-IDF terdiri dari dua komponen utama, pertama *Team Freaquency* (TF), berfungsi untuk mengukur seberapa sering suatu kata muncul dalam sebuah dokumen. Kedua *Inverse Document Frequency* (IDF), berperan untuk mengurangi bobot kata yang terlalu sering muncul di banyak dokumen, karena kata tersebut dianggap kurang signifikan dalam membedakan dokumen satu dengan yang lainnya[18].

Bobot suatu kata t dalam dokumen d dapat dihitung menggunakan persamaan berikut:

$$W_{dt} = tf_{dt} \times id_{ft} \quad (2.4)$$

di mana :

W_{dt} : bobot kata t dalam dokumen d.

tf_{dt} : jumlah kemunculan kata t dalam dokumen d.

id_{ft} : nilai *Inverse Document Frequency* (IDF).

Nilai IDF dapat dihitung menggunakan rumus:

$$id_{ft} = \log \left(\frac{D}{df_t} \right) \quad (2.5)$$

di mana :

D : jumlah total dokumen dalam koleksi.

df_t : jumlah dokumen yang mengandung kata t.

Pendekatan ini memungkinkan sistem menghilangkan kata-kata umum yang sering muncul di banyak dokumen namun tidak memiliki makna signifikan

dalam klasifikasi teks, sehingga meningkatkan akurasi dalam analisis dokumen seperti deteksi berita hoax[18]. Penggunaan TF-IDF terbukti mampu meningkatkan akurasi model dalam mendeteksi berita hoax karena metode ini memberikan bobot lebih besar pada kata-kata yang jarang muncul namun signifikan, seperti istilah “konspirasi”, “terbukti”, atau “viral”, yang sering muncul dalam berita palsu namun jarang di berita asli.

2.8 N-Gram

N-Gram adalah model probabilistik yang digunakan untuk memprediksi elemen berikutnya dalam suatu teks berdasarkan urutan elemen sebelumnya. Elemen yang dimaksud dapat berupa huruf, karakter, atau kata, tergantung pada kebutuhan aplikasi yang digunakan. Metode ini memiliki keunggulan utama dalam independensinya terhadap struktur bahasa tertentu, sehingga dapat diterapkan dalam berbagai bahasa. Salah satu penerapan umumnya adalah dalam koreksi ejaan, di mana N-Gram digunakan untuk membentuk urutan N kata dalam sebuah string guna menganalisis hubungan antar kata[18].

Dalam tahap transformasi, metode N-Gram diterapkan dengan cara memodifikasi pemisahan atau segmentasi teks berdasarkan kata. Menurut referensi [19], N-Gram merupakan sekumpulan n-karakter yang diekstraksi dari suatu string. Teknik ini umumnya digunakan dalam pembangkitan kata atau karakter dalam pemrosesan teks. Selain itu, penerapan N-Gram juga berperan dalam meningkatkan akurasi ekstraksi fitur, khususnya dalam perhitungan TF-IDF, dengan mengurangi potensi kesalahan dalam proses tersebut. Penggunaan N-Gram tidak hanya membantu dalam pengolahan teks, tetapi juga meningkatkan akurasi dan efektivitas hasil yang diperoleh[20].

N-Gram menganalisis dan membentuk urutan kata atau karakter dalam suatu teks. Berdasarkan jumlah kata dalam satu unit analisis, N-Gram dikategorikan menjadi beberapa jenis, seperti Uni-Gram, Bi-Gram, dan Tri-Gram. Uni-Gram (1-Gram) memisahkan teks menjadi kata-kata individual tanpa mempertimbangkan hubungan antar kata, misalnya kalimat "Berita hoax sangat banyak" akan diproses menjadi ["Berita", "hoax", "sangat", "banyak"]. Bi-Gram (2-Gram) mempertimbangkan hubungan antara dua kata yang berurutan, sehingga kalimat

yang sama akan diubah menjadi ["Berita hoax", "hoax sangat", "sangat banyak"]. Sementara itu, Tri-Gram (3-Gram) menganalisis tiga kata yang berurutan, menghasilkan ["Berita hoax sangat", "hoax sangat banyak"] seperti yang terlihat pada Tabel 2. Semakin besar nilai N, semakin kompleks hubungan antar kata yang dapat dianalisis, namun juga meningkatkan kebutuhan komputasi dalam pemrosesan data.

Tabel 2 Hasil Proses Tokenisasi Berdasarkan Jenis N-Gram

Jenis N-Gram	Hasil Tokenisasi
Uni-Gram	Berita hoax sangat banyak
Bi-Gram	Berita hoax sangat banyak
Tri-Gram	Berita hoax sangat hoax sangat banyak

Sebagai contoh, pada frasa “berita hoax”, penggunaan unigram hanya mengenali kata “berita” dan “hoax” secara terpisah, sedangkan bigram mampu memahami konteks hubungan antar kata. Dengan demikian, model dapat mengenali pola semantik yang lebih kompleks, yang sangat berguna dalam mendekripsi kalimat hoax yang sering menggunakan pasangan kata tertentu secara berulang.

2.9 *Confusion Matrix*

Confusion Matrix merupakan metode analisis dan evaluasi yang digunakan untuk mengukur kinerja suatu sistem klasifikasi. Evaluasi ini dilakukan dengan menghitung beberapa metrik seperti akurasi, *precision*, *recall*, dan *F1-score*. *Confusion Matrix* sering digunakan dalam konsep data *mining* untuk mengevaluasi sejauh mana model mampu melakukan prediksi dengan benar[18][19].

Dalam konteks penelitian ini, *Confusion Matrix* menjadi alat penting untuk menilai seberapa baik algoritma MNB dalam membedakan berita hoax dan non-hoax. Melalui analisis terhadap nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN), peneliti dapat memahami aspek mana yang perlu diperbaiki dari model. Misalnya, nilai FP yang tinggi menandakan bahwa sistem masih sering salah mengklasifikasikan berita asli sebagai hoax, yang dapat berdampak negatif terhadap kredibilitas sumber informasi. Oleh karena itu, evaluasi

berbasis *Confusion Matrix* memberikan gambaran menyeluruh tentang performa model, tidak hanya dari sisi akurasi, tetapi juga keseimbangan dalam prediksi antar kelas. Untuk memperjelas definisi operasional dari setiap komponen matriks tersebut dalam konteks klasifikasi berita, rinciannya disajikan pada Tabel 3.

Tabel 3 *Confusion Matrix* untuk Klasifikasi Hoax

Prediksi/ Aktual	Hoax (Aktual)	Bukan Hoax (Aktual)
Hoax (Prediksi)	<i>True Positive</i> (TP), Model mengklasifikasikan data sebagai hoax, dan data tersebut memang benar merupakan hoax.	<i>False Positive</i> (FP), Model memprediksi berita sebagai hoax, padahal sebenarnya bukan hoax.
Bukan Hoax (Prediksi)	<i>False Negative</i> (FN), Model memprediksi berita sebagai bukan hoax, padahal sebenarnya berita tersebut merupakan hoax.	<i>True Negative</i> (TN), Model memprediksi berita sebagai bukan hoax, dan hasil aslinya juga memang bukan hoax.

Tahap ini bertujuan untuk mengevaluasi performa model yang telah dikembangkan dengan menggunakan beberapa metrik utama, yaitu akurasi, precision, recall, dan F1-score. Metrik ini digunakan untuk menilai sejauh mana model mampu memberikan hasil klasifikasi yang sesuai dengan data sebenarnya.

1. *Accuracy*

Akurasi adalah ukuran sejauh mana hasil prediksi model sesuai dengan nilai aktualnya. Semakin tinggi nilai akurasi, semakin baik performa klasifikasi yang dilakukan oleh sistem. Akurasi dihitung dengan rumus berikut :

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (2.6)$$

2. *Precision*

Precision mengukur tingkat ketepatan hasil klasifikasi dengan membandingkan jumlah prediksi benar dalam suatu kelas terhadap keseluruhan prediksi yang diberikan model. *Precision* dihitung dengan rumus :

$$Precision = \frac{TP}{TP + FP} \quad (2.7)$$

3. Recall

Recall mengukur seberapa baik model dapat menemukan seluruh data yang benar dalam suatu kategori. Semakin tinggi *recall*, semakin baik model dalam mengenali kelas yang sebenarnya. Rumus perhitungannya adalah :

$$Recall = \frac{TP}{TP + FN} \quad (2.8)$$

4. F1-Score

F1-Score adalah rata-rata harmonis dari *precision* dan *recall*, yang digunakan untuk menyeimbangkan antara keduanya dalam evaluasi model. F1-Score dihitung dengan rumus :

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (2.9)$$

Metrik-metrik di atas digunakan untuk memahami sejauh mana model mampu membedakan berita hoax dan bukan hoax dengan akurasi yang tinggi. Dengan menggunakan *Confusion Matrix*, kelemahan dalam klasifikasi dapat diidentifikasi dan diperbaiki untuk meningkatkan performa sistem secara keseluruhan[18][21].

2.10 Perbandingan Metode Machine Learning

Dalam penelitian deteksi berita hoax, terdapat berbagai metode *machine learning* yang sering digunakan untuk melakukan klasifikasi teks. Beberapa di antaranya adalah *Multinomial Naive Bayes*, *Support Vector Machine* (SVM), *Decision Tree*, dan *Random Forest*. Setiap algoritma memiliki karakteristik, keunggulan, dan keterbatasan tersendiri yang memengaruhi kinerja sistem dalam mengenali pola data teks. Oleh karena itu, pemilihan metode klasifikasi yang tepat menjadi langkah penting untuk mencapai hasil yang optimal.

Secara umum, algoritma MNB dikenal memiliki efisiensi komputasi yang tinggi dan performa yang stabil pada data teks berukuran besar. Sementara itu,

algoritma SVM sering dipilih karena akurasi tinggi pada dataset yang seimbang, meskipun proses pelatihannya memerlukan waktu yang lebih lama. Di sisi lain, metode Decision Tree mudah dipahami dan divisualisasikan, namun cenderung mengalami overfitting jika tidak dilakukan pruning. Adapun Random Forest mampu memberikan hasil yang lebih robust karena menggabungkan beberapa pohon keputusan, tetapi membutuhkan sumber daya komputasi yang lebih besar. Untuk mempermudah pemahaman mengenai karakteristik masing-masing metode tersebut, ringkasan perbandingan kelebihan dan kekurangan dari setiap algoritma disajikan pada Tabel 4.

Tabel 4 Perbandingan Beberapa Algoritma *Machine Learning*.

Algoritma	Kelebihan	Kekurangan
<i>Multinomial Naive Bayes</i>	Cepat, ringan, efisien untuk teks, akurasi tinggi pada dataset besar	Asumsi independensi antar fitur sering tidak realistik
<i>Support Vector Machine</i>	Akurasi tinggi pada data seimbang, efektif untuk fitur berdimensi tinggi	Proses <i>training</i> lambat dan sulit dioptimasi untuk data besar
<i>Decision Tree</i>	Mudah dipahami, hasil dapat divisualisasikan dengan baik	Mudah mengalami <i>overfitting</i>
<i>Random Forest</i>	Akurasi tinggi, tahan terhadap <i>noise</i>	Membutuhkan banyak waktu dan data untuk <i>training</i>

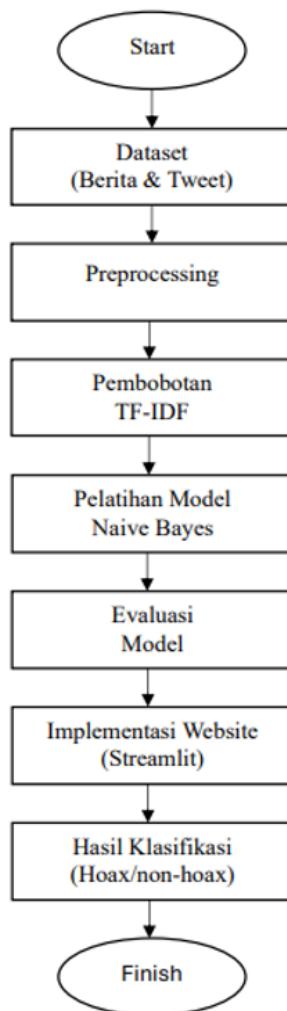
Tabel 4 menunjukkan perbandingan umum dari keempat algoritma berdasarkan aspek efisiensi, akurasi, serta kompleksitas penanganan data teks. Berdasarkan hasil analisis tersebut serta karakteristik data yang digunakan dalam penelitian ini, algoritma MNB dipilih sebagai metode utama. Keputusan ini didasarkan pada keseimbangan yang optimal antara efisiensi komputasi, akurasi, dan kemudahan implementasi yang ditawarkan MNB. Selain itu, algoritma ini terbukti handal dalam memproses data teks dengan dimensi fitur yang besar, seperti pada Bahasa Indonesia yang memiliki variasi morfologi kata yang beragam. Dengan kompleksitas algoritma yang relatif rendah, metode ini juga sangat cocok diintegrasikan ke dalam sistem berbasis web untuk kebutuhan deteksi berita hoax secara *real-time*.

BAB III

METODOLOGI

3.1 Alur Penelitian

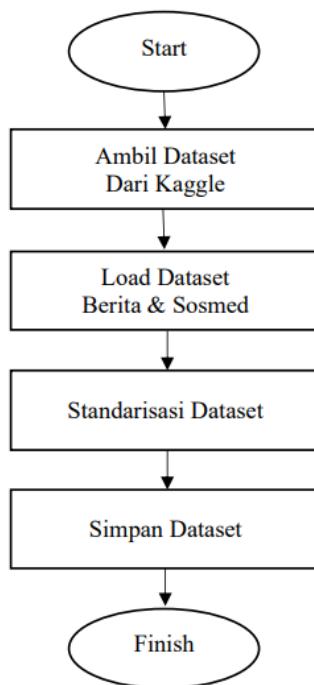
Penelitian ini dilaksanakan melalui tahapan sistematis untuk membangun model deteksi berita hoax berbasis algoritma MNB. Prosesnya mencakup pengumpulan data, pra-pemrosesan teks, ekstraksi fitur menggunakan metode TF-IDF, serta pelatihan dan evaluasi model. Seluruh tahapan dirancang berdasarkan prinsip *Natural Language Processing* (NLP) dan *machine learning* guna menghasilkan sistem klasifikasi yang akurat.



Gambar 1. Diagram Alur Penelitian

3.2 Pengumpulan Dataset

Tahap pengumpulan data dilakukan untuk memperoleh sumber data yang digunakan dalam proses pelatihan dan pengujian model deteksi berita hoax. Data yang digunakan dalam penelitian ini merupakan data sekunder yang diambil dari dua dataset publik di platform Kaggle, masing-masing berfokus pada berita daring dan unggahan media sosial (tweet). Dataset pertama berisi kumpulan berita berbahasa Indonesia yang telah dilabeli hoax dan non-hoax, bersumber dari berbagai situs berita serta laman verifikasi seperti Turnbackhoax.id. Dataset ini digunakan untuk melatih model dalam mengenali struktur dan pola kebahasaan khas pada berita palsu. Sementara itu, dataset kedua berisi kumpulan tweet masyarakat Indonesia yang juga telah diberi label secara manual berdasarkan isi dan konteks kalimat. Dataset ini merepresentasikan pola penyebaran hoax di media sosial yang bersifat lebih informal dan opiniatif.



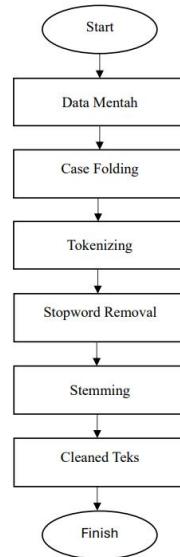
Gambar 2. Alur Pengumpulan dan Standarisasi Dataset

Sementara itu, dataset kedua berisi kumpulan tweet masyarakat Indonesia yang diperoleh dari platform *Kaggle* dan telah diberi label secara manual berdasarkan isi dan konteks kalimat, apakah mengandung unsur hoax atau tidak.

Dataset ini mencerminkan bentuk penyebaran hoax di media sosial yang cenderung lebih bebas, informal, dan sering kali mengandung opini pribadi. Kedua dataset kemudian digabungkan untuk memperkaya variasi data dan meningkatkan kemampuan generalisasi model. Sebelum digunakan dalam proses pelatihan, seluruh data terlebih dahulu melalui tahap pembersihan awal untuk menghapus elemen yang tidak relevan seperti *emoji*, tanda baca, tautan, serta karakter khusus. Proses pengumpulan dan seleksi data ini menjadi tahap penting dalam menjamin kualitas dan *representativitas* dataset terhadap fenomena penyebaran hoax di Indonesia.

3.3 Preprocessing

Tahap pra-pemrosesan data merupakan langkah krusial dalam penelitian ini karena berpengaruh langsung terhadap kualitas hasil klasifikasi. Data teks yang diperoleh dari berbagai sumber umumnya tidak terstruktur dan mengandung banyak noise. Oleh sebab itu, dilakukan proses pembersihan dan normalisasi agar data siap diolah oleh algoritma MNB. Tahapan utama dalam pra-pemrosesan meliputi *case folding*, *tokenizing*, *stopword removal*, dan *stemming*..



Gambar 3. Alur *Preprocessing* Data

1. Case Folding

Pada tahap ini, seluruh teks diubah menjadi huruf kecil (lowercase) untuk menyamakan format kata. Langkah ini diperlukan karena algoritma pembelajaran mesin memperlakukan huruf besar dan kecil sebagai entitas yang berbeda, misalnya kata “Hoax” dan “hoax” akan dianggap dua kata yang berbeda jika tidak dinormalkan terlebih dahulu.

2. Tokenizing

Tahapan ini bertujuan untuk memecah teks menjadi unit-unit kata (*token*). Proses ini memungkinkan setiap kata dianalisis secara individual dalam proses pembobotan. Sebagai contoh, kalimat “Berita ini tidak benar” akan diubah menjadi daftar token [“berita”, “ini”, “tidak”, “benar”].

3. Stopword Removal

Setelah teks diubah menjadi token, dilakukan proses penghapusan kata-kata umum yang tidak memiliki kontribusi penting terhadap makna kalimat, seperti “yang”, “dan”, “di”, “ke”, dan sebagainya. Proses ini dilakukan menggunakan daftar *stopword* bahasa Indonesia dari pustaka Sastrawi.

4. Stemming

Tahapan ini dilakukan untuk mengubah setiap kata menjadi bentuk dasarnya. Misalnya, kata “menyebarluaskan”, “penyebarluasan”, dan “disebarluaskan” semuanya akan dikembalikan ke bentuk dasar “sebar”. Proses ini juga dilakukan menggunakan library Sastrawi, yang dirancang khusus untuk bahasa Indonesia.

Keseluruhan tahapan pra-pemrosesan yang telah diuraikan sebelumnya bertujuan untuk mentransformasi data teks yang tidak terstruktur menjadi bentuk yang lebih bersih dan baku. Proses ini tidak hanya sekadar menghapus karakter yang dianggap sebagai *noise*, tetapi juga menstandarisasi variasi morfologi kata agar memiliki format yang seragam, sehingga pola dalam data lebih mudah dipelajari oleh model klasifikasi. Tanpa proses ini, variasi kata yang berlebihan

dapat menurunkan akurasi sistem secara signifikan. Sebagai ilustrasi nyata untuk memperjelas perbedaan struktur data antara input mentah dan output bersih, contoh hasil transformasi teks tersebut dapat dilihat secara rinci pada Tabel 5.

Tabel 5 Contoh Hasil Prapemrosesan Teks

Teks Asli	Hasil Pra-Pemrosesan
“Breaking News! Pemerintah dikabarkan menyembunyikan data kasus baru Covid-19 di daerah tertentu.”	“pemerintah kabar sembunyi data kasus baru covid daerah tentu”
“Informasi ini sudah terbukti palsu, jangan percaya sebelum cek ke sumber resmi.”	“informasi bukti palsu percaya cek sumber resmi”

Berdasarkan Tabel 5, terlihat bahwa teks yang telah melalui tahap prapemrosesan menjadi lebih ringkas dan terstruktur dibandingkan teks aslinya. Elemen-elemen pengganggu seperti tanda baca, *stopword*, dan variasi imbuhan telah dieliminasi, sehingga data kini lebih fokus pada kata-kata kunci utama.

3.4 Pembobotan TF-IDF

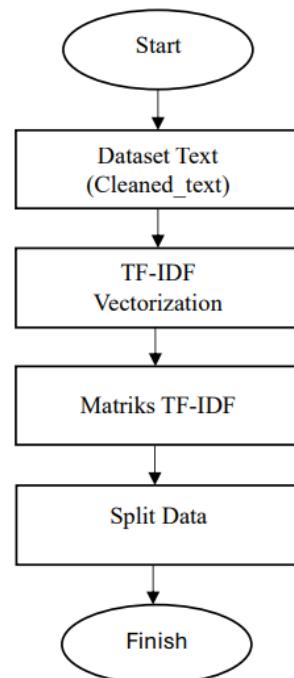
Tahapan ekstraksi fitur bertujuan mengubah data teks hasil pra-pemrosesan menjadi bentuk numerik agar dapat diproses oleh algoritma MNB. Penelitian ini menerapkan metode *Term Frequency–Inverse Document Frequency* (TF-IDF) untuk memberikan bobot pada setiap kata berdasarkan tingkat kepentingannya. Prinsipnya, kata yang unik dalam satu dokumen akan mendapatkan bobot tinggi, sedangkan kata umum yang muncul di seluruh *dataset* akan diberi bobot rendah. Semakin tinggi nilai TF-IDF, semakin besar kontribusi kata tersebut terhadap konteks dokumen. Contoh hasil pembobotan ini dapat dilihat pada Tabel 6.

Tabel 6 Contoh Hasil Pembobotan TF-IDF

Kata	Dokumen 1 (Hoax)	Dokumen 2 (Real)	Bobot TF-IDF
vaksin	0.247	0.102	0.247
bahaya	0.233	0.035	0.223
terbukti	0.131	0.029	0.131
pemerintah	0.089	0.112	0.112

Hasil pembobotan pada Tabel 6 menunjukkan perbedaan nilai bobot kata antara dokumen yang dikategorikan sebagai hoax dan non-hoax, di mana nilai-nilai

numerik tersebut menggambarkan tingkat kepentingan atau kontribusi setiap kata terhadap isi dokumen. Semakin tinggi nilai TF-IDF, semakin besar pengaruh kata tersebut dalam membentuk konteks teks. Sebagai contoh, kata “vaksin” memiliki bobot 0,247 pada dokumen hoax dan 0,102 pada non-hoax, menunjukkan bahwa kata ini lebih sering digunakan dalam narasi hoax untuk menimbulkan kekhawatiran publik. Begitu pula kata “bahaya” dengan bobot 0,223 pada dokumen hoax dan 0,035 pada non-hoax, yang menandakan kecenderungan penggunaannya dalam berita palsu untuk memperkuat efek emosional pembaca. Sementara itu, kata “pemerintah” memiliki bobot relatif seimbang (0,089 dan 0,112) karena sering muncul di kedua jenis teks, baik resmi maupun hoax. Adapun kata “terbukti” memiliki bobot lebih tinggi pada dokumen hoax (0,131) dibandingkan non-hoax (0,029), mencerminkan kecenderungan penggunaannya untuk memberikan kesan validitas palsu terhadap informasi yang tidak benar. Dari hasil ini dapat disimpulkan bahwa metode TF-IDF efektif dalam mengidentifikasi kata-kata yang memiliki kontribusi besar terhadap pembentukan narasi hoax, sekaligus menjadi dasar bagi algoritma MNB dalam proses klasifikasi teks secara otomatis.

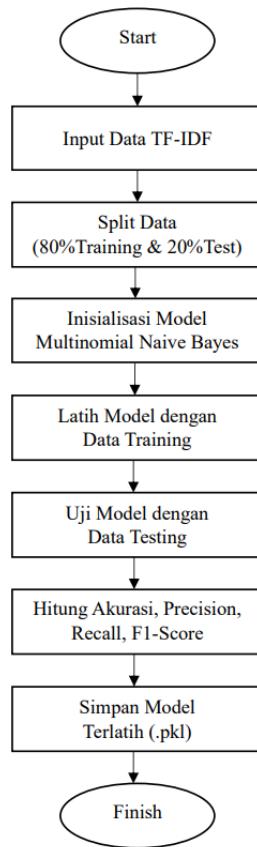


Gambar 4. Alur Pembobotan TF-IDF

Tahapan pembobotan TF-IDF secara keseluruhan dapat digambarkan melalui alur pada Gambar 4. Proses dimulai dari data teks hasil pra-pemrosesan, kemudian setiap kata dalam dokumen dihitung frekuensi kemunculannya *Term Frequency* (TF). Setelah itu, sistem menghitung *Inverse Document Frequency* (IDF) untuk menilai seberapa jarang kata tersebut muncul di seluruh korpus. Nilai TF dan IDF kemudian dikalikan untuk menghasilkan bobot akhir TF-IDF, yang merepresentasikan tingkat kepentingan setiap kata terhadap konteks dokumen. Hasil akhir berupa matriks TF-IDF digunakan sebagai input bagi algoritma MNB pada tahap pelatihan model.

3.5 Pelatihan Model *Multinomial Naive Bayes*

Tahap pelatihan model merepresentasikan fase paling krusial dalam penelitian ini, di mana sistem secara aktif mempelajari pola distribusi probabilistik dari data teks yang telah dikonversi menjadi representasi numerik melalui metode TF-IDF. Algoritma *Multinomial Naive Bayes* (MNB) dipilih sebagai inti klasifikasi karena keunggulannya dalam menangani data berdimensi tinggi (seperti matriks teks) namun tetap mempertahankan efisiensi komputasi yang optimal. Pendekatan ini memungkinkan model untuk mengkalkulasi peluang suatu berita tergolong *hoax* atau *real* berdasarkan bobot frekuensi kata yang telah diekstraksi. Proses pelatihan ini tidak dilakukan secara tunggal, melainkan melalui serangkaian tahapan sistematis yang dimulai dari pembagian data latih dan uji, proses *fitting* algoritma, perhitungan metrik evaluasi, hingga finalisasi berupa penyimpanan model (*saving model*) agar dapat digunakan kembali (*reusable*). Secara visual, alur kerja teknis pelatihan dan evaluasi model ini diilustrasikan secara rinci pada Gambar 5.



Gambar 5. *Flowchart Proses Pelatihan Model Multinomial Naive Bayes*

Pada tahap awal, dataset yang telah melalui proses pembobotan TF-IDF dibagi menjadi dua bagian utama, data latih (*training-set*) sebesar 80% dan data uji (*testing-set*) sebesar 20%. Pembagian data ini menggunakan fungsi (*train_test_split*) dari pustaka *Scikit-learn*, dengan parameter *stratify* diaktifkan untuk memastikan distribusi label antara data latih dan data uji tetap seimbang, sehingga mengurangi bias pada proses pembelajaran. Data latih kemudian digunakan untuk melatih model agar menghitung dan menyimpan probabilitas kemunculan setiap kata pada masing-masing kelas (Hoax dan Real), yang menjadi parameter pembelajaran model.

Proses pelatihan MNB ini dilakukan secara terpisah untuk dua *dataset*, yakni dataset berita daring dan dataset media sosial (Twitter). Pemisahan model ini mutlak dilakukan karena perbedaan karakteristik bahasa yang signifikan, berita formal cenderung baku dan terstruktur, sementara teks media sosial lebih informal, tidak terstruktur, dan sering menggunakan *slang*. Setelah model selesai dilatih pada

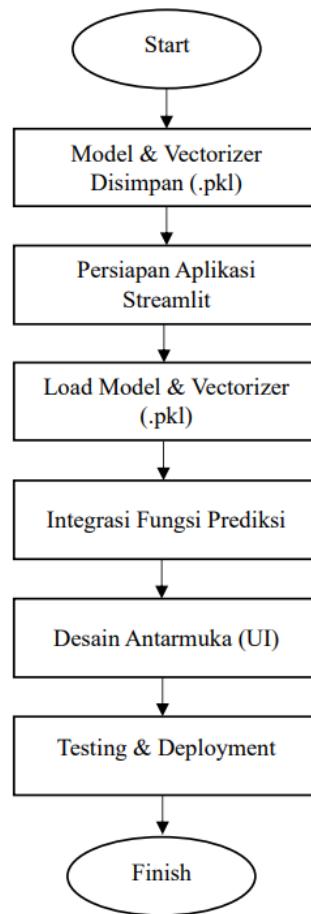
masing-masing domain, dilakukan tahap pengujian untuk mengukur kapabilitas model dalam menggeneralisasi data baru yang belum pernah dilihat sebelumnya. Pengujian ini menggunakan data uji dan dilakukan dengan menghitung tingkat akurasi, *precision*, *recall*, dan *F1-score*. Nilai-nilai metrik ini membantu menilai sejauh mana model mampu membedakan antara konten *hoax* dan *non-hoax* secara efektif. Hasil akhir pelatihan dan pengujian model ini kemudian disimpan dalam format .pkl (pickle) untuk dimuat kembali dan diimplementasikan pada sistem deteksi *hoax* berbasis *web* menggunakan Streamlit.

3.6 Evaluasi dan Analisis Hasil Pemodelan

Evaluasi dan analisis hasil pemodelan dilakukan untuk menilai performa model MNB dalam mengklasifikasikan teks ke dalam kategori hoax dan non-hoax. Penilaian dilakukan menggunakan metrik evaluasi utama yaitu akurasi, *precision*, *recall*, dan F1-score yang dihitung berdasarkan *Confusion Matrix*. Metrik akurasi digunakan untuk mengetahui tingkat ketepatan prediksi model secara keseluruhan, *precision* untuk mengukur ketepatan model dalam memprediksi data hoax, *recall* untuk mengetahui sejauh mana model mampu mendeteksi seluruh data yang benar-benar hoax, dan *F1-score* sebagai rata-rata harmonis antara *precision* dan *recall* yang menunjukkan keseimbangan kinerja model. Proses evaluasi dilakukan menggunakan fungsi (*classification_report*) dan *confusion matrix* dari *library Scikit-learn*, yang menghasilkan nilai numerik untuk setiap metrik. Apabila hasil evaluasi menunjukkan nilai akurasi dan *F1-score* yang tinggi, maka model dapat dianggap efektif dalam mendeteksi berita hoax.

3.7 Implementasi Website

Tahap implementasi sistem dimulai dengan menyimpan model MNB dan vektor TF-IDF terbaik ke dalam format .pkl menggunakan pustaka *Joblib* untuk menghindari proses pelatihan ulang. Selanjutnya, model tersebut diintegrasikan ke dalam antarmuka berbasis web menggunakan kerangka kerja *Streamlit*. Secara garis besar, alur teknis mulai dari penyimpanan model hingga *deployment* aplikasi digambarkan pada Gambar 6.



Gambar 6. *Flowchart* Implementasi Sistem pada Aplikasi Web

Berdasarkan alur Pada Gambar 6, sistem bekerja dengan memuat kembali (*load*) file model dan vectorizer ke dalam aplikasi. Setelah model aktif, fungsi prediksi diintegrasikan untuk memproses teks masukan pengguna melalui tahapan pra-pemrosesan hingga klasifikasi (hoax atau valid). Hasil prediksi tersebut kemudian divisualisasikan pada antarmuka web melalui komponen interaktif *Streamlit*.

3.8 Perangkat dan Sumber Daya Penelitian

Dalam penelitian ini, digunakan berbagai perangkat dan sumber daya untuk mendukung seluruh tahapan penelitian, mulai dari pengolahan data, pelatihan model machine learning, hingga implementasi sistem berbasis web. Perangkat yang digunakan mencakup perangkat keras dan perangkat lunak, yang dirangkum pada Tabel 7.

Tabel 7 Perangkat dan Sumber Daya Penelitian

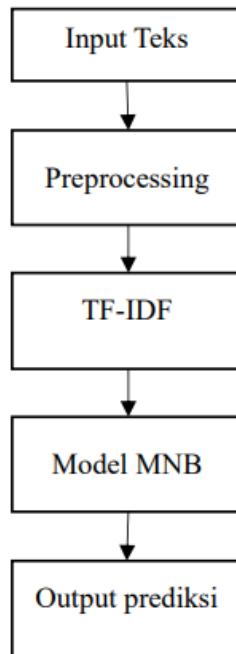
Jenis	Detail
Perangkat Keras (Hardware)	Laptop Lenovo IdeaPad Gaming 3 dengan prosesor AMD Ryzen 5 5500H (6 Core, 12 Thread), GPU NVIDIA GeForce RTX 2050, RAM 8 GB, SSD 256 GB, sistem operasi Windows 11 64-bit.
Perangkat Lunak (Software)	Python 3.10, Jupyter Notebook, Streamlit, Visual Studio Code.
Library dan Modul Python	Scikit-learn, Pandas, NumPy, Matplotlib, Seaborn, Pickle, dan Sastrawi untuk stemming bahasa Indonesia
Sumber Data	Dua dataset publik dari Kaggle (berita daring dan tweet media sosial berbahasa Indonesia).

BAB IV

HASIL DAN PEMBAHASAN

4.1 Gambaran Umum Sistem

Bab ini menjelaskan realisasi sistem deteksi *hoax* yang mengintegrasikan seluruh komponen utama mulai dari pra-pemrosesan, ekstraksi fitur TF-IDF, hingga model MNB ke dalam aplikasi berbasis *Streamlit*. Sistem dirancang untuk memproses input teks pengguna guna menghasilkan klasifikasi kategori (*hoax* atau *real*) beserta tingkat akurasinya. Secara sistematis, mekanisme aliran data dari tahap input hingga munculnya output prediksi digambarkan pada Diagram Alur Proses Deteksi Hoax di Gambar 7.

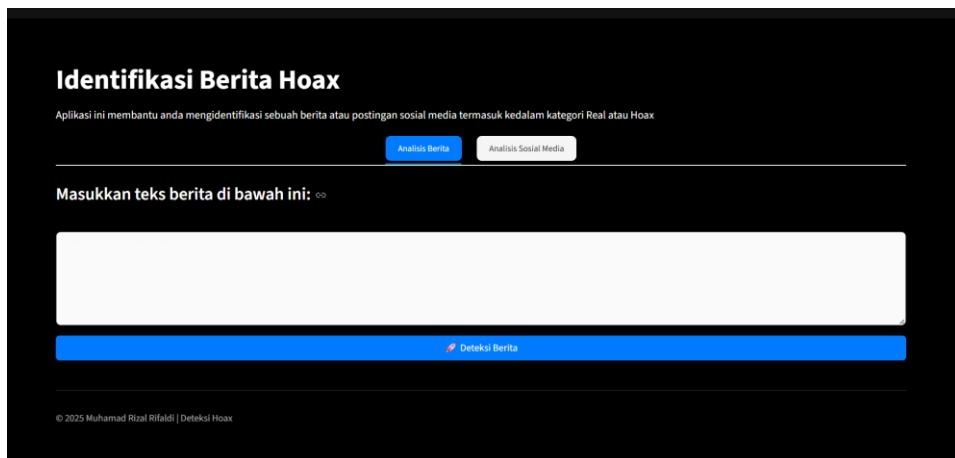


Gambar 7. Diagram Alur Deteksi Hoax Menggunakan Algoritma MNB

4.1.1 Input Data Teks

Tahapan awal dimulai dari proses input teks yang dilakukan oleh pengguna melalui antarmuka *website* berbasis *Streamlit*. Pengguna dapat memasukkan teks berita formal maupun postingan media sosial ke dalam kolom *text area* yang telah

disediakan. Teks inilah yang kemudian menjadi objek analisis bagi sistem untuk menentukan apakah konten tersebut termasuk kategori hoax atau real.



Gambar 8. Tampilan Awal Website

Gambar 8. memperlihatkan tampilan halaman utama sistem deteksi hoax berbasis web yang dibangun menggunakan *framework Streamlit*. Pengguna dapat memilih mode analisis, yakni *Analisis Berita* atau *Analisis Sosial Media*, lalu memasukkan teks ke kolom input yang tersedia. Setelah tombol “Deteksi Berita” ditekan, sistem secara otomatis menjalankan seluruh proses mulai dari pembersihan teks, pembobotan fitur dengan TF-IDF, hingga klasifikasi menggunakan model MNB. Hasilnya ditampilkan secara *real-time* dalam bentuk label Hoax atau Real lengkap dengan nilai probabilitas keyakinan model.

4.1.2 *Preprocessing* Teks

Setelah teks diinput, sistem menjalankan proses pra-pemrosesan (*preprocessing*) untuk menstandarkan format data sebelum diproses lebih lanjut oleh algoritma *machine learning*. Tujuan utama tahap ini adalah menghilangkan *noise* dan menjaga konsistensi teks agar dapat dianalisis secara optimal. Tahapan *preprocessing* meliputi :

1. Pembersihan (*Cleaning*) dilakukan untuk menghilangkan elemen-elemen non-leksikal seperti URL, angka, serta tanda baca yang tidak memiliki makna semantik. Langkah ini memastikan bahwa sistem hanya memproses kata yang benar-benar memiliki arti kontekstual dan

relevan dengan isi teks. Dengan demikian, model dapat lebih fokus pada informasi penting tanpa terganggu oleh simbol atau karakter yang tidak diperlukan.

2. Penyeragaman Huruf (*Lowercasing*) bertujuan untuk mengubah seluruh karakter dalam teks menjadi huruf kecil. Proses ini mencegah perbedaan interpretasi antara kata yang sama namun ditulis dengan kapitalisasi berbeda, misalnya antara “Hoax” dan “hoax”. Tahapan ini sederhana namun sangat penting untuk menjaga konsistensi data sebelum dilakukan analisis lebih lanjut.
3. Selanjutnya, dilakukan penghapusan *Stopword* (*Stopword Removal*), yaitu penghapusan kata-kata umum yang sering muncul tetapi tidak memiliki nilai informasi yang signifikan dalam konteks klasifikasi, seperti “yang”, “dan”, “adalah”, atau “itu”. Dengan menghapus kata-kata tersebut, dimensi fitur menjadi lebih kecil dan model dapat lebih fokus pada kata-kata kunci yang relevan terhadap deteksi hoax.
4. Tahapan terakhir adalah *Stemming*, yang dilakukan menggunakan *library* Sastrawi untuk konteks Bahasa Indonesia. Proses ini mengembalikan setiap kata ke bentuk dasarnya (root word), misalnya kata “menyebarluaskan” menjadi “sebar”. *Stemming* membantu sistem mengenali berbagai variasi morfologi dari kata yang sama sehingga tidak dianggap sebagai fitur berbeda. Hasil akhir dari seluruh proses *preprocessing* ini berupa teks yang bersih, seragam, dan siap untuk melalui tahap ekstraksi fitur menggunakan metode TF-IDF.

4.1.3 Penerapan Pembobotan TF-IDF pada Data Teks

Setelah penyelesaian tahapan pra-pemrosesan, data teks yang telah bersih perlu ditransformasikan menjadi representasi numerik agar dapat dipahami dan diproses oleh algoritma pembelajaran mesin. Untuk tujuan tersebut, penelitian ini menerapkan metode TF-IDF sebagai teknik ekstraksi fitur utama. Secara fundamental, metode ini bekerja dengan menyeimbangkan frekuensi kemunculan kata yang unik dan spesifik pada suatu dokumen akan diberikan bobot tinggi karena dianggap informatif, sementara kata yang muncul secara masif di hampir seluruh

dokumen akan dinilai memiliki relevansi yang rendah (bobot kecil). Berdasarkan hasil analisis pembobotan pada *dataset*, terlihat pola yang jelas bahwa fitur dengan nilai signifikansi tertinggi didominasi oleh terminologi politik dan pemerintahan, yang mengindikasikan bahwa metode TF-IDF berhasil menangkap konteks topik utama dalam berita yang diolah. Rincian mengenai sepuluh kata dengan akumulasi bobot TF-IDF terbesar yang menjadi fitur penentu dalam klasifikasi dapat dicermati pada Tabel 8.

Tabel 8 Hasil Pembobotan Kata Menggunakan Ekstraksi Fitur TF-IDF

Kata	Nilai TF-IDF
Sebut	0.022581
Kata	0.021812
Partai	0.019775
Jadi	0.019381
Politik	0.017351
Jokowi	0.016882
Presiden	0.016825
Indonesia	0.015083
Jelas	0.014453
Laku	0.014391

Berdasarkan Tabel 8, terlihat bahwa fitur dengan nilai TF-IDF tertinggi didominasi oleh terminologi yang erat kaitannya dengan konteks politik dan pemerintahan, seperti 'presiden', 'jokowi', 'partai', dan 'politik'. Tingginya bobot pada kata-kata tersebut mengindikasikan bahwa istilah ini memiliki tingkat kekhususan (*specificity*) yang tinggi terhadap dokumen dalam *dataset*. Artinya, kata-kata ini berperan vital dalam membedakan karakteristik antar dokumen. Selain itu, munculnya kata kerja seperti 'sebut' dan 'jadi' di peringkat atas memperlihatkan pola linguistik yang umum dalam penulisan berita, khususnya narasi kutipan atau klaim, yang kerap ditemukan baik pada berita kategori *real* maupun *hoax*. Dengan demikian, hasil ini membuktikan bahwa metode TF-IDF berhasil mengidentifikasi kata kunci yang informatif dan mampu merepresentasikan isi teks secara numerik guna mendukung akurasi klasifikasi pada tahap selanjutnya.

4.1.4 Klasifikasi Menggunakan Model *Multinomial Naive Bayes*

Untuk memahami mekanisme klasifikasi model MNB, dilakukan analisis terhadap probabilitas kemunculan kata pada kelas *Real* dan *Hoax*. Analisis ini bertujuan mengidentifikasi kata-kata yang secara statistik menjadi fitur pembeda dominan (diskriminatif) di antara kedua kategori tersebut. Daftar sepuluh kata dengan kontribusi probabilitas terbesar dalam menentukan keputusan model disajikan pada Tabel 9.

Tabel 9 Probabilitas Kata terhadap Kelas Real dan Hoax pada Model MNB

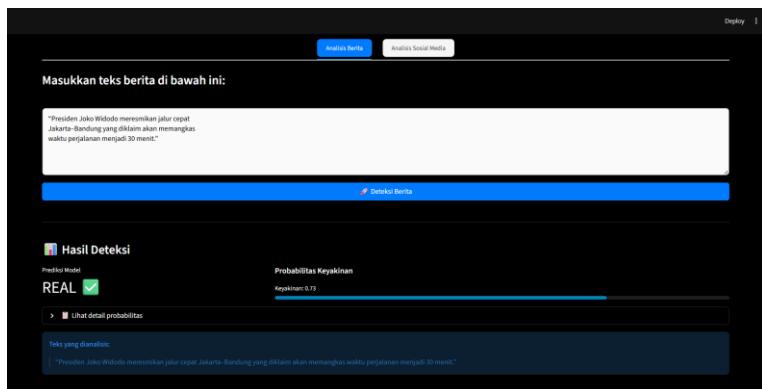
Kata	P(Kata Real)	P(Kata Hoax)	Selisih
partai	0.002526	0.000150	0.002376
politik	0.002186	0.000178	0.002009
kata	0.002395	0.000793	0.001602
milu	0.001656	0.000107	0.001549
ketua	0.001705	0.000250	0.001456
narasi	0.000055	0.002229	0.002174
referensi	0.000017	0.002074	0.002057
konten	0.000035	0.002086	0.002051
kategori	0.000039	0.002065	0.002026
sumber	0.000108	0.002118	0.002010

Berdasarkan probabilitas kata pada Tabel 9, terlihat bahwa model MNB mampu memetakan perbedaan distribusi terminologi yang signifikan antara kelas *Real* dan *Hoax*. Kata-kata seperti 'partai', 'politik', 'ketua', dan 'milu' (bentuk dasar dari pemilu) memiliki probabilitas dominan pada kelas *Real*. Tingginya frekuensi kata-kata ini mengindikasikan bahwa berita asli cenderung menggunakan diction formal yang berkaitan dengan peristiwa politik atau pemerintahan. Sebaliknya, kata-kata seperti 'narasi', 'referensi', 'konten', 'kategori', dan 'sumber' justru memiliki probabilitas jauh lebih tinggi pada kelas *Hoax*. Kemunculan kata-kata ini merefleksikan karakteristik penyebaran berita bohong yang sering kali berfokus pada penyajian klaim sepihak, pencatutan sumber yang tidak valid, atau penggunaan istilah meta-data yang sering beredar di media sosial. Disparitas atau selisih probabilitas inilah yang menjadi landasan utama model dalam mengambil keputusan. Dengan memanfaatkan pola distribusi tersebut, algoritma MNB dapat

mengklasifikasikan dokumen secara efektif dengan mengandalkan bobot numerik yang dihasilkan dari proses pelatihan.

4.1.5 Output Prediksi

Tahap final dari sistem ini adalah penyajian output prediksi yang informatif. Sistem dirancang untuk menampilkan label klasifikasi (*Hoax* atau *Real*) yang didampingi dengan nilai probabilitas (*confidence score*). Nilai ini berfungsi sebagai indikator kuantitatif untuk mengukur tingkat keyakinan model terhadap keputusan yang diambil. Visualisasi antarmuka hasil prediksi tersebut ditunjukkan pada Gambar 9.



Gambar 9. Interface Halaman Hasil Prediksi

Sebagaimana terlihat pada tampilan di atas, seluruh hasil prediksi divisualisasikan melalui antarmuka *Streamlit* secara *real-time*. Desain antarmuka mengedepankan aspek *user-friendly* dengan memanfaatkan indikator visual yang intuitif, seperti *progress bar* dan kode warna (misalnya hijau untuk kategori *Real* dan merah untuk indikasi *Hoax*). Pendekatan visual ini tidak hanya meningkatkan keterbacaan, tetapi juga memudahkan pengguna memahami keputusan model tanpa memerlukan pemahaman teknis mendalam.

Dengan demikian, integrasi antara performa klasifikasi algoritma MNB dan antarmuka interaktif ini berhasil menjembatani kompleksitas teknis dengan kebutuhan pengguna akhir, menjadikan sistem deteksi *hoax* ini tidak hanya akurat, tetapi juga efisien dan mudah dioperasikan.

4.2 Implementasi Sistem

Implementasi sistem deteksi hoax ini dilakukan dengan mengintegrasikan seluruh tahapan penelitian mulai dari preprocessing teks, ekstraksi fitur menggunakan TF-

IDF, hingga klasifikasi dengan algoritma MNB ke dalam satu pipeline terstruktur. Proses implementasi dijalankan menggunakan bahasa pemrograman Python dan memanfaatkan sejumlah pustaka pendukung seperti *scikit-learn* untuk klasifikasi, pandas dan numpy untuk manipulasi data, serta Sastrawi untuk stemming Bahasa Indonesia. Seluruh tahapan diimplementasikan pada framework Streamlit, sehingga pengguna dapat berinteraksi langsung dengan sistem melalui antarmuka berbasis web.

4.2.1 Implementasi *Load* dan Standarisasi *Dataset*

Tahap awal implementasi dimulai dengan proses pemuatan (*loading*) dan standarisasi *dataset* yang bersumber dari portal berita daring (CNN, Kompas, Tempo, Turnbackhoax) serta media sosial (Twitter). Mengingat adanya variasi struktur kolom antar sumber data, diterapkan fungsi khusus bernama *standardize_dataset()* untuk menyeragamkan format menjadi tiga atribut utama *text*, *label*, dan *sumber*. Implementasi teknis untuk standarisasi dan penggabungan data ini diperlihatkan pada Gambar 10.



```
# Fungsi umum untuk standarisasi dataset
def standardize_dataset(df, sumber):
    teks = df['text'].fillna('')
    label = df['label'].astype(int)
    return pd.DataFrame({'text': teks, 'label': label, 'sumber': sumber})

# ===== Dataset Berita =====
cnn = pd.read_csv("dataset_cnn_10k_cleaned.csv")
kompas = pd.read_csv("dataset_kompas_4k_cleaned.csv")
tempo = pd.read_csv("dataset_tempo_6k_cleaned.csv")
turnbackhoax = pd.read_csv("dataset_turnbackhoax_10_cleaned.csv")

data_berita = pd.concat([
    standardize_dataset(cnn, "CNN"),
    standardize_dataset(kompas, "Kompas"),
    standardize_dataset(tempo, "Tempo"),
    standardize_dataset(turnbackhoax, "Turnbackhoax")
], ignore_index=True)

# ===== Dataset Sosial Media =====
data_sosmed_raw = pd.read_csv("dataset_sosmed_hoax.csv")
data_sosmed = standardize_dataset(data_sosmed_raw, "Twitter")

print("Dataset berita:", data_berita.shape)
print("Dataset sosmed:", data_sosmed.shape)
```

Gambar 10. Implementasi Kode *Load Dataset*

Berdasarkan kode di atas, fungsi *pd.concat()* digunakan untuk menggabungkan seluruh data berita yang telah distandarisasi menjadi satu kesatuan *dataframe*, sedangkan data media sosial diproses sebagai entitas terpisah. Hasil akhir dari tahapan ini adalah terbentuknya dua variabel dataset utama, yakni *data_berita* dan *data_sosmed*, yang kini memiliki struktur seragam. Keseragaman format ini krusial untuk memastikan kelancaran tahap pra-pemrosesan selanjutnya.

4.2.2 Implementasi Pra-pemrosesan Teks

Tahap pra-pemrosesan merupakan langkah fundamental untuk menstandarisasi data teks mentah menjadi format baku yang bebas *noise*. Dalam sistem ini, seluruh operasi pembersihan dikonsolidasikan ke dalam satu fungsi utama bernama *clean_text()*. Fungsi ini secara sekvensial mengeksekusi penyeragaman huruf (*lowercasing*), eliminasi elemen non-leksikal (URL, angka, tanda baca) menggunakan *Regular Expression* (Regex), penghapusan kata umum (*stopword*), hingga pengembalian kata ke bentuk dasar (*stemming*) menggunakan pustaka Sastrawi. Implementasi kode program untuk fungsi tersebut dapat dilihat pada Gambar 11.

```
def clean_text(text):
    if pd.isna(text):
        return ""
    s = text.lower()
    s = re.sub(r"http\S+|www\S+", " ", s)    # hapus URL
    s = re.sub(r"\S*[^\w\A-Z\s]", " ", s)      # hapus angka & simbol
    s = re.sub(r"\s+", " ", s).strip()          # hapus spasi ganda

    tokens = [w for w in s.split() if w not in stop_words]
    tokens = [stem_cache[w] if w in stem_cache else stemmer.stem(w) for w in tokens]
    for w in tokens:
        if w not in stem_cache:
            stem_cache[w] = stemmer.stem(w)
    return " ".join(tokens)
```

Gambar 11. Kode Implementasi Fungsi *clean_text* pada Tahap Pra-pemrosesan Teks

Berdasarkan kode di atas, fungsi *clean_text()* dirancang untuk memproses teks secara otomatis melalui serangkaian filter. Logika program dimulai dengan pemeriksaan data kosong (pd.isna), dilanjutkan dengan pembersihan karakter menggunakan pola *regex* untuk menghapus atribut yang tidak relevan. Setelah teks bersih, proses berlanjut ke tahap tokenisasi dan penyaringan *stopword* untuk membuang kata-kata yang tidak memiliki nilai diskriminatif. Terakhir, algoritma menerapkan *stemming* dengan mekanisme *caching* (stem_cache) untuk mempercepat proses pengembalian kata dasar. Struktur kode yang terintegrasi ini menjamin bahwa setiap input teks yang masuk ke model memiliki format leksikal yang konsisten dan berkualitas.

4.2.3 Implementasi Ekstraksi Fitur dan Split Data

Setelah teks melalui tahap pembersihan, data dikonversi menjadi representasi numerik menggunakan teknik ekstraksi fitur TF-IDF. Proses ini diimplementasikan

menggunakan pustaka TfidfVectorizer dengan konfigurasi kombinasi *unigram* dan *bigram* (ngram_range=(1,2)) untuk menangkap konteks frasa, serta pembatasan fitur maksimal sebanyak 20.000 token guna menjaga efisiensi komputasi. Implementasi kode untuk vektorisasi dan persiapan data latih tersebut ditampilkan pada Gambar 12.

```
# TF-IDF Vectorizer dengan unigram + bigram
vectorizer_berita = TfidfVectorizer(
    ngram_range=(1,2),
    max_features=20000,
    sublinear_tf=True
)

# Transformasi teks bersih menjadi matriks TF-IDF
X_berita = vectorizer_berita.fit_transform(data_berita["clean_text"])
y_berita = data_berita["label"].values

# Pembagian data 80% Latih, 20% uji
Xb_train, Xb_test, yb_train, yb_test = train_test_split(
    X_berita, y_berita, test_size=0.2, random_state=42, stratify=y_berita
)
```

Gambar 12. Kode Implementasi Ekstraksi Fitur TF-IDF dan Pembagian Data

Kode di atas menghasilkan matriks TF-IDF (X_berita) yang merepresentasikan bobot kepentingan kata terhadap dokumen. Selanjutnya, dataset dibagi menggunakan fungsi *train_test_split* dengan proporsi 80% untuk data latih dan 20% untuk data uji. Pembagian ini dilakukan dengan parameter *stratify=y_berita* untuk menjamin bahwa rasio kelas *Hoax* dan *Real* tetap seimbang (proporsional) di kedua subset data, sehingga model dapat belajar dari distribusi yang representatif dan hasil evaluasinya lebih objektif.

4.2.4 Implementasi dan Pelatihan Model *Multinomial Naive Bayes*

Tahap inti dari penelitian ini adalah pelatihan model klasifikasi menggunakan algoritma *Multinomial Naive Bayes* (MNB). Algoritma ini dipilih karena efisiensi komputasinya yang tinggi dalam menangani distribusi frekuensi kata pada data teks. Proses pelatihan dieksekusi menggunakan fungsi *.fit()* untuk mempelajari pola linguistik dari data latih, kemudian dilanjutkan dengan pengujian generalisasi model terhadap data uji (*test set*) untuk menghasilkan prediksi kelas. Implementasi kode program untuk pelatihan dan pengujian tersebut disajikan pada Gambar 13.

```

# Inisialisasi model
mnb_berita = MultinomialNB()

# Training
mnb_berita.fit(Xb_train, yb_train)

# Prediksi di data test
yb_pred = mnb_berita.predict(Xb_test)

# Evaluasi
print("==> Evaluasi Model (MNB - Berita) ==")
print("Akurasi:", accuracy_score(yb_test, yb_pred))
print("\nClassification Report:")
print(classification_report(yb_test, yb_pred, target_names=["REAL", "HOAX"]))
print("\nConfusion Matrix:")
print(confusion_matrix(yb_test, yb_pred))

```

Gambar 13. Kode Implementasi Pelatihan dan Evaluasi Model MNB

Kode di atas tidak hanya melakukan prediksi, tetapi juga secara otomatis menjalankan evaluasi kinerja model. Melalui fungsi *classification_report* dan *confusion_matrix*, sistem menghasilkan metrik terukur yang meliputi akurasi, presisi, *recall*, dan matriks kebingungan. Hasil evaluasi ini menjadi landasan kuantitatif utama untuk memvalidasi kemampuan model dalam membedakan berita *hoax* dan *real* secara akurat sebelum sistem diimplementasikan ke tahap produksi.

4.2.5 Implementasi Fungsi Prediksi dan Penyimpanan Model

Tahap krusial dalam implementasi sistem adalah penyusunan fungsi prediksi yang mengorkestrasikan seluruh tahapan pemrosesan. Fungsi *prediksi_sosmed()* dirancang untuk menerima input teks mentah, melakukan pembersihan melalui *clean_text()*, mentransformasikannya ke format vektor, dan akhirnya menghitung probabilitas kategori menggunakan metode *predict_proba()*. Implementasi logika prediksi tersebut dapat dilihat pada Gambar 14.

```

def prediksi_sosmed(teks, model=mnb_sosmed, vec=vectorizer_sosmed):
    teks_clean = clean_text(teks)
    fitur = vec.transform([teks_clean])

    probas = model.predict_proba(fitur)[0]
    pred = model.predict(fitur)[0]

    print("▣ [SOSMED]")
    print("Teks (potongan):", teks[:120], "...")
    print("Probabilitas => REAL:", probas[0], " | HOAX:", probas[1])
    print("Prediksi:", "HOAX ✗" if pred == 1 else "REAL ✓")
    print("-"*80)
    return "HOAX" if pred == 1 else "REAL"

```

Gambar 14. Implementasi Fungsi Prediksi

Kode di atas menunjukkan penerapan prinsip *Maximum A Posteriori* (MAP) di mana keputusan klasifikasi didasarkan pada nilai probabilitas tertinggi. Sistem menggunakan ambang batas (*threshold*) netral sebesar 0.5 jika probabilitas kelas

hoax melebihi angka tersebut, maka teks dikategorikan sebagai *hoax*, dan sebaliknya. Pendekatan ini menjamin objektivitas model dalam menentukan prediksi berdasarkan tingkat keyakinan yang terukur secara statistik.

Setelah fungsi prediksi tervalidasi, langkah selanjutnya adalah menyimpan model dan *vectorizer* hasil pelatihan ke dalam penyimpanan lokal agar bersifat persisten. Menggunakan pustaka *Joblib*, objek model diserialisasi ke dalam format file *.pkl* sehingga dapat digunakan kembali tanpa perlu melatih ulang model dari awal setiap kali aplikasi dijalankan. Proses penyimpanan model ini ditunjukkan pada Gambar 15.

```
# Simpan model berita & vectorizer
joblib.dump(mnb_berita, "model_hoax_berita.pkl")
joblib.dump(vectorizer_berita, "tfidf_vectorizer_berita.pkl")

# Simpan model sosmed & vectorizer
joblib.dump(mnb_sosmed, "model_hoax_sosmed.pkl")
joblib.dump(vectorizer_sosmed, "tfidf_vectorizer_sosmed.pkl")
```

Gambar 15. Implementasi Penyimpanan Model

Untuk mengintegrasikan model yang telah disimpan ke dalam antarmuka aplikasi *Streamlit*, diperlukan mekanisme pemuatan ulang (*loading*). Proses ini memastikan bahwa file *.pkl* dapat diakses dan dijalankan oleh sistem secara *real-time*. Implementasi kode untuk memanggil kembali model dan *vectorizer* tersebut disajikan pada Gambar 16.

```
# Load model & vectorizer berita
mnb_berita_loaded = joblib.load("model_hoax_berita.pkl")
vectorizer_berita_loaded = joblib.load("tfidf_vectorizer_berita.pkl")

# Load model & vectorizer sosmed
mnb_sosmed_loaded = joblib.load("model_hoax_sosmed.pkl")
vectorizer_sosmed_loaded = joblib.load("tfidf_vectorizer_sosmed.pkl")
```

Gambar 16. Implementasi Load Model

Dengan berhasilnya proses pemuatan model di atas, maka seluruh arsitektur *pipeline* deteksi *hoax* mulai dari akuisisi input, pra-pemrosesan, inti pemodelan, hingga integrasi sistem telah terbentuk secara utuh. Sistem kini siap beroperasi sepenuhnya sebagai aplikasi berbasis web yang interaktif dan efisien.

4.3 Hasil Pengujian

Pada tahap ini dilakukan serangkaian pengujian untuk mengetahui performa sistem dalam mendeteksi hoax pada dua domain teks, yaitu berita daring dan media sosial. Pengujian dilakukan setelah seluruh proses pra-pemrosesan, vektorisasi, dan pelatihan model selesai dilaksanakan. Hasil pengujian mencakup tampilan dataset yang digunakan, performa model pada data uji, serta kemampuan model dalam mengklasifikasikan teks baru yang tidak termasuk dalam data pelatihan. Selain itu, pengujian juga dilakukan pada data penelitian terdahulu untuk melihat konsistensi model dalam mengenali pola-pola hoax yang pernah diuji oleh peneliti sebelumnya. Seluruh hasil ini bertujuan untuk memberikan gambaran menyeluruh mengenai sejauh mana model MNB yang dikembangkan mampu menangani variasi teks nyata dan mempertahankan akurasi pada berbagai skenario pengujian.

4.3.1 Hasil dan Contoh Dataset yang Dipakai

Tahapan ini memvisualisasikan hasil akuisisi dan standarisasi dari dua jenis dataset utama yang digunakan. Dataset pertama bersumber dari portal berita daring terpercaya (CNN Indonesia, Kompas, Tempo, Turnbackhoax) dengan total 31.353 entri data, sedangkan dataset kedua berasal dari media sosial Twitter yang berjumlah 1.126 entri dengan distribusi kelas yang seimbang. Sampel struktur data berita yang telah diseragamkan formatnya dapat dilihat pada Tabel 10.

Tabel 10 Dataset Berita yang digunakan

No	Text	Label	Sumber
1	Bambang PDIP Respons Gerakan Rumah Ganjar: Kok...	0	CNN
2	Sumber : Media Daring dailymail.co.uk Akun ...	1	Turnback
3	Komisi I Respons Bjorka Acak-acak Data Pemerin...	0	CNN
4	Profil Afriansyah Noor, Meniti Karier dari Pen...	0	Tempo
5	Jokowi Kumpulkan Menteri di Istana: Pencabutan...	0	CNN

Selain dataset berita, penelitian ini juga mengolah data yang bersumber dari media sosial untuk menangkap variasi gaya bahasa yang lebih dinamis. Contoh sampel data dari platform Twitter, lengkap dengan label klasifikasi dan atribut sumbernya, disajikan pada Tabel 11.

Tabel 11 Dataset Sosmed yang digunakan

No	Text	Label	Sumber
1	Keingetan kowe, cah bagus.. Lagi sekolah tent...	1	Twitter
2	Varian baru virus corona B117 yang pertama kal...	0	Twitter
3	Sejuta Lebih Warga Ber-KTP Kuningan Sudah Diva...	0	Twitter
4	Polisi menangkap puluhan pemuda yang tengah as...	1	Twitter
5	Perjalanan keluar daerah, Kota Surabaya tetap...	0	Twitter

Berdasarkan Tabel 10 dan Tabel 11, terlihat perbedaan karakteristik tekstual yang signifikan antara kedua sumber data. *Dataset* berita didominasi oleh ragam bahasa jurnalistik yang baku dan terstruktur, sementara *dataset* media sosial cenderung menggunakan gaya bahasa informal yang variatif. Kombinasi kedua jenis data ini sangat krusial untuk melatih model agar memiliki kemampuan generalisasi yang baik dalam mendekripsi *hoax* di berbagai konteks linguistik.

4.3.2 Hasil Pra-Pemrosesan Teks

Tahapan pra-pemrosesan (*preprocessing*) diterapkan secara sistematis dan komprehensif untuk menstandarisasi data mentah menjadi format baku yang bebas dari *noise*. Proses ini melibatkan serangkaian operasi pembersihan bertingkat, dimulai dari normalisasi huruf (*lowercasing*) untuk menjaga konsistensi karakter, diikuti dengan eliminasi elemen non-leksikal seperti URL, tanda baca, dan angka yang tidak berkontribusi pada konteks kalimat. Selanjutnya, dilakukan penyaringan *stopword* untuk membuang kata-kata umum yang redundant, serta penerapan *stemming* menggunakan pustaka Sastrawi yang secara spesifik menangani morfologi Bahasa Indonesia untuk mengembalikan kata berimbuhan ke bentuk dasar (*root word*). Transformasi mendalam ini secara fundamental bertujuan untuk mereduksi dimensi fitur data dan menghilangkan ambiguitas, sehingga model klasifikasi nantinya dapat bekerja lebih efisien dengan hanya memproses kata-kata yang memiliki signifikansi semantik tinggi.

Efektivitas dari tahapan pembersihan tersebut dapat dilihat secara komparatif pada tabel di bawah ini. Tabel 12 menampilkan contoh hasil pra-pemrosesan pada dataset berita yang memiliki struktur bahasa formal, sedangkan

Tabel 13 memperlihatkan hasil normalisasi pada teks media sosial yang cenderung lebih informal dan tidak terstruktur.

Tabel 12 Contoh Hasil Pra-pemrosesan Data Berita

No	Teks Asli	Hasil Pra-Pemrosesan	Label
1	Anies di Milad BKMT: Pengajian Menghasilkan Ibu-ibu yang Tahu Tugasnya	anies milad bkmt aji hasil ibu ibu tahu mantan tugas	0
2	Edy Soal Pilgub Sumut: Kalau yang Maju Abal-abal Pasti Kalah	edy soal pilgub sumut kalau maju abal abal pasti kalah	0
3	PKB Bakal Daftarkan Menaker Ida Fauziyah Jadi Cawapres	pkb bakal daftar menaker ida fauziyah jadi cawapres	0
4	Gede Pasek Doakan AHY Jadi Capres atau Cawapres	gede pasek doa ahy jadi capres cawapres ketua	0
5	PKN Siapkan Jabatan Khusus Buat Anas Urbaningrum Usai Bebas	pkn siap jabat khusus buat anas urbaningrum usai bebas	0

Tabel 13 Contoh Hasil Pra-Pemrosesan Data Sosmed

No	Teks Asli	Hasil Pra-Pemrosesan	Label
1	Aturan 20 Menit Makan di Tempat Tak Terpantau	atur menit makan tempat tak pantau	0
2	#BeritaTerkini #PPKM https://t.co/sPOewNypIu	beritaterkini ppkm	0
3	Laju penyebaran Covid turun.	laju sebar covid turun	0
4	@rbkunwas\n#indonesia #imigrasi #imigrasiindonesia	rbkunwas indonesia imigrasi imigrasiindonesia	0
5	Persiapan 2024 👉👉👉👉 kacauuu\n#PPKM #IkatanCinta	siap kacauuu ppkm ikatancintaep ikatancinta	0

Tabel 12 dan Tabel 13 secara jelas mengilustrasikan transformasi data sebelum dan sesudah tahap pra-pemrosesan. Terlihat bahwa seluruh elemen non-leksikal, simbol, serta kata-kata umum yang dianggap sebagai *noise* berhasil dieliminasi, sementara kata-kata kunci utama tetap dipertahankan. Hal ini membuktikan bahwa mekanisme pembersihan yang diterapkan tidak hanya efektif dalam menyederhanakan struktur teks, tetapi juga mampu menjaga integritas konteks semantik yang menjadi basis utama analisis TF-IDF. Dengan representasi data yang lebih ringkas dan bersih ini, model MNB dapat beroperasi jauh lebih efisien dalam mengidentifikasi pola linguistik spesifik yang membedakan antara

berita *hoax* dan *real*, yang pada akhirnya berkontribusi signifikan terhadap peningkatan akurasi sistem deteksi yang dibangun.

4.3.3 Hasil Vektorisasi TF-IDF

Vektorisasi teks merupakan proses penting yang mengubah data teks hasil pra-pemrosesan menjadi representasi numerik agar dapat diproses oleh algoritma *machine learning*. Dalam penelitian ini digunakan metode TF-IDF, yaitu teknik pembobotan kata yang menghitung tingkat kepentingan sebuah term berdasarkan frekuensinya dalam dokumen (TF) dan tingkat kekhususannya terhadap seluruh korpus (IDF). Perhitungan bobot TF-IDF mengikuti rumus (2.4), dengan nilai IDF dihitung menggunakan rumus (2.5). Implementasi metode ini dilakukan menggunakan *TfidfVectorizer* dari pustaka scikit-learn dengan kombinasi unigram dan bigram (*ngram_range*=(1,2)), sehingga model tidak hanya memahami makna kata tunggal tetapi juga pasangan kata berurutan seperti “konspirasi global” atau “berita palsu”. Jumlah fitur dibatasi hingga 20.000 term paling representatif untuk menjaga efisiensi komputasi tanpa mengurangi informasi penting di dalam teks.

Proses vektorisasi ini mentransformasikan teks menjadi matriks TF-IDF dua dimensi yang siap diproses oleh algoritma. Matriks tersebut memiliki dimensi fitur yang konsisten sebanyak 20.000 kata unigram dan bigram terpilih. Berdasarkan skema pembagian data, matriks dipisahkan menjadi dua subset data pelatihan dengan dimensi (25.082, 20.000) dan data pengujian dengan dimensi (6.271, 20.000). Statistik rinci mengenai hasil vektorisasi serta distribusi kelas *Real* dan *Hoax* pada setiap subset data tersebut dirangkum dalam Tabel 14.

Tabel 14 Hasil Vektorisasi dan Pembagian Dataset Berita

Jenis Data	Jumlah Sampel	Jumlah Fitur	Real	Hoax
Data Latih	25.082	20.000	16.777	8.305
Data Uji	6.271	20.000	4.195	2.076

Data pada Tabel 14 mengonfirmasi bahwa pembagian dataset dilakukan secara proporsional (*stratified split*), sehingga rasio kelas *Real* dan *Hoax* tetap terjaga keseimbangannya baik pada data latih maupun data uji. Pendekatan metodologi yang sama juga diterapkan pada dataset media sosial. Namun, karena karakteristik bahasa media sosial yang cenderung informal dan tidak baku, proses

vektorisasi menghasilkan jumlah fitur unik yang sedikit berbeda, yakni sebanyak 19.466 term. Rincian hasil vektorisasi dan distribusi data untuk kategori media sosial tersebut disajikan pada Tabel 15.

Tabel 15 Hasil Vektorisasi dan Pembagian Dataset Sosmed

Jenis Data	Jumlah Sampel	Jumlah Fitur	Real	Hoax
Data Latih	900	19.466	450	450
Data Uji	226	19.466	113	113

Rincian pada tabel di atas mengonfirmasi bahwa penerapan metode *train-test split* dengan proporsi 80:20 dan teknik stratifikasi label telah berhasil menjaga keseimbangan distribusi kelas. Hal ini terlihat dari jumlah sampel yang simetris sempurna, di mana data latih memiliki total 900 sampel terbagi rata menjadi 450 *Real* dan 450 *Hoax*, sedangkan data uji berjumlah 226 sampel 113 *Real* dan 113 *Hoax*. Konsistensi proporsi ini sangat krusial untuk mencegah terjadinya bias pada model saat mempelajari pola dari dataset media sosial yang memiliki volume data terbatas.

4.3.4 Hasil Evaluasi Model *Multinomial Naive Bayes*

Bagian ini memaparkan hasil pengujian model MNB pada dataset berita daring yang melibatkan 6.271 sampel uji 4.195 *Real* dan 2.076 *Hoax*. Validasi performa diukur secara menyeluruh menggunakan metrik standar meliputi akurasi, presisi, *recall*, dan *F1-score*. Rincian hasil statistik evaluasi serta pemetaan prediksi pada *confusion matrix* disajikan secara berturut-turut dalam Tabel 16 dan Tabel 17.

Tabel 16 Hasil Evaluasi *Model Multinomial Naive Bayes* Data Berita Daring

Kelas	Precision	Recall	F1-Score	Support
Real	0.99	0.99	0.99	4195
Hoax	0.97	0.97	0.97	2076
Akurasi Total			0.981	6271

Tabel 17 *Confusion Matrix* Model Berita Daring

	Prediksi Real	Prediksi Hoax
Aktual Real	4,137	58
Aktual Hoax	60	2,016

Berdasarkan pada hasil di atas, model menunjukkan performa klasifikasi yang sangat presisi pada data berita dengan akurasi mencapai 98.1%. Nilai *F1-score* yang seimbang (0.99 untuk *Real* dan 0.97 untuk *Hoax*) serta rendahnya tingkat kesalahan pada *confusion matrix* hanya 58 kasus *false positive* dan 60 kasus *false negative* membuktikan konsistensi model dalam membedakan pola teks yang terstruktur. Hal ini mengonfirmasi bahwa kombinasi algoritma MNB dengan fitur *unigram-bigram* TF-IDF sangat efektif untuk mendeteksi *hoax* pada ragam bahasa jurnalistik yang baku.

Selanjutnya, pengujian diperluas pada domain media sosial menggunakan 226 sampel uji (113 *Real* dan 113 *Hoax*). Berbeda dengan berita daring, dataset ini memiliki tantangan tersendiri berupa gaya bahasa informal. Hasil evaluasi kinerja model terhadap data media sosial tersebut, beserta rincian matriks kebingungannya, dirangkum dalam Tabel 18 dan Tabel 19.

Tabel 18 Hasil Evaluasi *Model Multinomial Naive Bayes* Data Media Sosial

Kelas	Precision	Recall	F1-Score	Support
Real	0.83	0.79	0.95	563
Hoax	0.80	0.84	0.94	563
Akurasi Total			0.814	226

Tabel 19 Confusion Matrix Model Media Sosial

	Prediksi Real	Prediksi Hoax
Aktual Real	89	24
Aktual Hoax	18	95

Hasil evaluasi menunjukkan bahwa meskipun akurasi pada media sosial 81.4% lebih rendah dibandingkan berita daring, model tetap menunjukkan performa yang kompetitif. Penurunan ini merupakan konsekuensi logis dari kompleksitas linguistik teks informal yang kaya akan *noise* dan ketidakbukan. Namun, keseimbangan nilai *precision* dan *recall* pada kisaran 0.80–0.84 menegaskan bahwa model tetap stabil dan objektif tanpa bias signifikan terhadap salah satu kelas. Hal ini membuktikan bahwa algoritma MNB memiliki tingkat ketahanan (*robustness*) yang memadai dalam menangani variasi gaya bahasa yang dinamis pada platform sosial.

4.4 Uji Coba pada Data baru

Data yang digunakan berupa teks yang disusun oleh peneliti secara mandiri dengan memasukkan karakteristik bahasa yang provokatif dan klaim menyesatkan. Model yang digunakan untuk pengujian ini adalah Model MNB untuk Berita Daring. Beberapa contoh berita diuji dan hasilnya disajikan pada Tabel 20 berikut.

Tabel 20 Hasil Uji Coba Prediksi Model Berita pada Data Baru

No	Teks Berita Uji	Label Aktual	Probabilitas Hoax	Prediksi Model
1	"Presiden Joko Widodo meresmikan jalur cepat Jakarta–Bandung yang diklaim akan memangkas waktu perjalanan menjadi 30 menit."	Real	26.74%	Real
2	"Minum air rebusan kabel listrik terbukti menyembuhkan penyakit jantung tanpa obat dokter."	Hoax	97.57%	Hoax
3	"Pemerintah akan membagikan uang tunai Rp15 juta kepada semua warga yang memiliki KTP elektronik."	Hoax	56.74%	Hoax
4	"Rupiah melemah tajam sebesar 0,5% di pasar global hari ini, Jumat, 27 Okt 2025, akibat tekanan inflasi. Analis memprediksi pelemahan akan berlanjut."	Real	40.37%	Real

Berdasarkan Tabel 20, Model MNB terbukti mampu memprediksi berita baru dengan akurat, yang membuktikan kemampuan sistem untuk mengenali hoax di luar data latih. Teks yang jelas-jelas palsu (Baris 2, klaim kesehatan) berhasil diklasifikasikan sebagai hoax dengan keyakinan yang sangat tinggi, yaitu Probabilitas Hoax 97.57%. Hasil ini menunjukkan bahwa model berhasil mengenali pola kata yang bersifat sensasional, yang sering ada di konten palsu. Teks yang nyata (Baris 1 dan 4) secara konsisten diprediksi sebagai real.

Meskipun Baris 4 adalah berita asli, adanya kata-kata yang memicu emosi, seperti "melemah tajam," membuat nilai Probabilitas Hoax menjadi lebih tinggi (40.37%) dibanding Baris 1 (26.74%). Hal ini menunjukkan bahwa model sangat sensitif terhadap kosakata tertentu. Selanjutnya, kasus klaim uang tunai pada Baris 3 diprediksi sebagai hoax dengan keyakinan sedang, yaitu Probabilitas Hoax 56.74%. Keyakinan yang tidak terlalu tinggi ini terjadi karena adanya penggunaan

kata-kata formal, seperti "Pemerintah" dan "KTP elektronik," yang juga sering muncul di berita asli, sehingga membuat model sedikit ragu. Secara keseluruhan, hasil uji kasus ini memvalidasi bahwa model MNB mampu membedakan informasi asli dan palsu dengan baik. Selanjutnya dilakukan Uji pada *caption* Sosial Media yang disajikan pada Tabel 21.

Tabel 21 Hasil Uji Coba Prediksi Model Sosmed pada Data Baru

No	Teks Berita Uji	Label Aktual	Probabilitas	Prediksi
			Hoax	Model
1	“Temen gue dapet transferan Rp10 juta cuma isi survei doang, buruan ikutan linknya nih!”	Hoax	39.51%	Real
2	“Cuaca panas banget hari ini, semoga ga mati listrik lagi 😞”	Real	47.43%	Real
3	“Wah makanan bergizi gratis ini katanya bikin banyak anak sekolah keracunan! 🤢”	Hoax	53.07%	Hoax
4	“barusan dapet info dari baca berita, pemerintah udah buat putusan buat Ahmad Sahroni, Nafa Urbach, Eko Patrio, Uya Kuya, dan Adies Kadir. Ada yang melanggar etik, ada juga yang enggak”	Real	55.64%	Hoax

Berdasarkan hasil pada Tabel 21, model menunjukkan performa yang cukup baik dalam mengenali pola linguistik hoax di media sosial, meskipun masih terjadi beberapa kesalahan klasifikasi. Teks pada Baris 1 seharusnya termasuk hoax karena mengandung unsur penipuan dengan klaim “transferan Rp10 juta cuma isi survei,” namun diprediksi sebagai real dengan probabilitas hoax 39.51%. Sebaliknya, teks pada Baris 4 yang bersifat informatif justru diklasifikasikan sebagai hoax dengan probabilitas 55.64%. Kesalahan ini kemungkinan terjadi karena banyaknya nama tokoh publik yang sering muncul dalam konten hoax pada data pelatihan, sehingga model salah mengasosiasikan konteks tersebut sebagai indikator hoax. Sementara itu, dua teks lainnya (Baris 2 dan 3) berhasil diprediksi dengan benar. Teks pada Baris 2 dikategorikan sebagai real karena hanya berisi ekspresi pribadi tanpa klaim faktual, sedangkan Baris 3 teridentifikasi tepat sebagai hoax karena memuat unsur sensasional dan provokatif. Secara keseluruhan, hasil uji coba ini menunjukkan

bahwa model MNB cukup andal dalam mengenali pola umum hoax di media sosial, terutama pada konten dengan gaya bahasa emosional atau berlebihan, meski masih perlu penyempurnaan agar lebih sensitif terhadap hoax dengan kalimat netral atau ambigu.

4.5 Uji Coba pada Data penelitian Sebelumnya

Untuk memastikan model yang dikembangkan bekerja dengan baik, dilakukan uji coba perbandingan (*benchmarking*) menggunakan sampel teks yang sebelumnya telah diuji oleh peneliti terdahulu. Tujuan dari tahapan ini adalah untuk memverifikasi apakah model MNB yang dioptimalkan dalam penelitian ini mampu menunjukkan performa yang lebih baik dibandingkan dengan model sejenis yang ada.

Penting untuk dicatat bahwa penelitian terdahulu hanya menggunakan headline atau judul berita sebagai teks uji. Dalam penelitian ini, teks-teks uji tersebut diuji kembali menggunakan Model MNB untuk Berita Daring, yang dilatih menggunakan *full-text* dan diperkuat dengan N-Gram. Hasil pengujian disajikan pada Tabel 22 berikut.

Tabel 22 Perbandingan Hasil Prediksi Model Penelitian Terdahulu dan Model Peneliti

No	Headline	Model D.F.N. Anisa (2023)	Model Peneliti (2025)
1	“Dari Ratusan Ribu Tahanan Tidak Ada Satupun Aktivis Islam yang Dibebaskan di Tengah Wabah COVID-19”	Hoax	Hoax
2	“Ganjil Genap Di Jakarta Ditiadakan Mulai Senin, 7 September 2020”	Hoax	Real
3	“Imigran China Sengaja Membawa Virus, Tujuannya Membasmi Rakyat +62”	Hoax	Hoax

Berdasarkan Tabel 22, model yang dikembangkan menunjukkan hasil yang stabil untuk berita yang sudah jelas palsu (Baris 1 dan 3), di mana kedua model sepakat memprediksi Hoax. namun, ada perbedaan penting di Baris 2. Berita tentang "Ganjil Genap Di Jakarta Ditiadakan Mulai Senin, 7 September 2020"

faktanya adalah hoax. Model peneliti lain[1]. berhasil memprediksi ini dengan benar. tetapi, Model Peneliti justru salah memprediksi berita hoax dan mengklasifikasikannya sebagai berita Real. Kesalahan ini sebagian besar disebabkan karena Model dilatih menggunakan *fulltext* bertita, sehingga mengalami kesulitan saat hanya diuji menggunakan headline atau judul saja, karena informasi konteks yang minim.

Kesalahan ini juga menunjukkan keterbatasan model. Model MNB bekerja dengan mengandalkan bobot kata yang sering muncul. Oleh karena itu model menganggap kata-kata resmi seperti "Ganjil Genap," "Jakarta," dan "September 2020" sebagai indikasi konten Real karena kata-kata ini sering ada di berita-berita resmi selama pelatihan. Model hanya melihat jenis kata yang digunakan (analisis leksikal), bukan memverifikasi kebenaran fakta atau tanggalnya. Kegagalan ini menunjukkan bahwa model belum mampu membedakan konteks faktual pada detail spesifik. Temuan ini menjadi masukan penting untuk penelitian yang akan datang.

4.6 Pengaruh Panjang Teks terhadap Tingkat Keyakinan Prediksi Model

Untuk menganalisis pengaruh panjang teks terhadap tingkat keyakinan model dalam mengklasifikasikan berita real, dilakukan pengujian menggunakan dua jenis input teks, yaitu headline dan fulltext. Pengujian ini dilakukan dengan memasukkan beberapa sampel berita real ke dalam sistem deteksi hoax berbasis MNB, kemudian dihitung rata-rata probabilitas prediksi yang dihasilkan oleh model untuk masing-masing jenis teks.

Tabel 23 Perbandingan Rata-rata Probabilitas Prediksi pada Berita Real

Jenis Berita	Jumlah Kata	Rata-rata Probabilitas
Headline	< 25	85.70%
Fulltext	> 25	93.75%

Berdasarkan hasil pengujian yang ditampilkan pada Tabel 23, dapat dilihat bahwa panjang teks memiliki pengaruh yang signifikan terhadap tingkat keyakinan model dalam mengklasifikasikan berita real. Headline yang memiliki jumlah kata terbatas (<20 kata) menghasilkan rata-rata probabilitas real sebesar 85.70%. Nilai ini menunjukkan bahwa meskipun model sudah mampu mengenali sebagian pola

berita real, keterbatasan konteks pada headline menyebabkan tingkat keyakinan model belum optimal. Sementara itu, penggunaan full text dengan jumlah kata lebih dari 100 menghasilkan rata-rata probabilitas real yang lebih tinggi, yaitu sebesar 93.75%. Hal ini membuktikan bahwa semakin lengkap konteks teks yang diberikan, semakin akurat distribusi fitur TF-IDF yang terbentuk, sehingga model MNB mampu melakukan klasifikasi dengan tingkat keyakinan yang lebih tinggi.

BAB V

PENUTUP

5.1 Simpulan

Berdasarkan hasil penelitian terkait penerapan MNB dengan metode pembobotan TF-IDF dalam mendekripsi berita hoax, dapat disimpulkan bahwa pendekatan *machine learning* mampu melakukan deteksi berita hoax secara efektif melalui tahapan yang terstruktur. Tahapan tersebut meliputi pengumpulan dataset berita daring dan media sosial, pra-pemrosesan teks berupa *cleaning*, *lowercasing*, *stopword removal*, dan *stemming*, hingga pembentukan fitur numerik menggunakan TF-IDF dengan dukungan N-Gram (unigram dan bigram). Representasi numerik tersebut selanjutnya diproses menggunakan algoritma MNB untuk mengklasifikasikan teks ke dalam kategori hoax atau real.

Hasil pengujian menunjukkan bahwa model memiliki performa klasifikasi yang sangat baik, dengan akurasi 98,1% pada dataset berita daring dan 81,4% pada dataset media sosial. Pengujian variasi panjang teks juga membuktikan bahwa penggunaan full text menghasilkan tingkat keyakinan prediksi yang lebih tinggi dibandingkan headline, sehingga kelengkapan konteks teks berpengaruh terhadap akurasi model. Selain itu, implementasi sistem berbasis Streamlit berhasil menghadirkan antarmuka yang sederhana dan mudah digunakan, sehingga sistem tidak hanya akurat secara teknis, tetapi juga bersifat *user-friendly* dan dapat dimanfaatkan sebagai alat bantu verifikasi awal berita hoax.

5.2 Saran

Untuk pengembangan selanjutnya, sistem deteksi berita hoax ini masih dapat ditingkatkan dari sisi kemudahan penggunaan dan kemampuan analisis. Salah satu pengembangan yang disarankan adalah penambahan fitur unggah dokumen berita dalam format PDF, sehingga pengguna tidak perlu menyalin teks secara manual. Selain itu, pengayaan dataset dari berbagai sumber dan platform media sosial perlu dilakukan agar sistem lebih adaptif terhadap variasi bahasa. Terakhir, sistem juga dapat dikembangkan menjadi alat bantu verifikasi informasi yang lebih praktis dan mendukung peningkatan literasi digital masyarakat.

DAFTAR PUSTAKA

- [1] D. F. N. Anisa, I. Mukhlash, and M. Iqbal, “Deteksi Berita Online Hoax Covid-19 Di Indonesia Menggunakan Metode Hybrid Long Short Term Memory dan Support Vector Machine,” *J. Sains Dan Seni ITS*, vol. 11, no. 3, pp. A101–A108, Mar. 2023, doi: 10.12962/j23373520.v11i3.83227.
- [2] T. A. Roshinta, E. Kumala, and I. F. Dinata, “Sistem Deteksi Berita Hoax Berbahasa Indonesia Bidang Kesehatan,” *remik*, vol. 7, no. 2, pp. 1167–1173, Apr. 2023, doi: 10.33395/remik.v7i2.12369.
- [3] A. Ripa’i, F. Santoso, and F. Lazim, “Deteksi Berita Hoax dengan Perbandingan Website Menggunakan Pendekatan Deep Learning Algoritma BERT,” *G-Tech J. Teknol. Terap.*, vol. 8, no. 3, pp. 1749–1758, Jul. 2024, doi: 10.33379/gtech.v8i3.4541.
- [4] Y. Kurnia, E. D. Kusuma, L. W. Kusuma, Suwitno, and W. Apridius, “Perbandingan Naïve Bayes dan CNN yang Dioptimasi PSO pada Identifikasi Berita Hoax Politik Indonesia,” *Bit-Tech*, vol. 6, no. 3, pp. 340–352, Apr. 2024, doi: 10.32877/bt.v6i3.1225.
- [5] R. Qubra and R. A. Saputra, “Classification of Hoax News Using the Naïve Bayes Method,” *Int. J. Softw. Eng. Comput. Sci. IJSECS*, vol. 4, no. 1, pp. 40–48, Apr. 2024, doi: 10.35870/ijsecs.v4i1.2068.
- [6] R. Yunanto, A. P. Purfini, and A. Prabuwisesa, “Survei Literatur: Deteksi Berita Palsu Menggunakan Pendekatan Deep Learning,” *J. Manaj. Inform. JAMIKA*, vol. 11, no. 2, pp. 118–130, Sep. 2021, doi: 10.34010/jamika.v11i2.5362.
- [7] B. Baldomero Ferguson and W. Istiono, “Hoax News Detection in Indonesian Political Headlines Using Multinomial Naive Bayes,” *JITSI J. Ilm. Teknol. Sist. Inf.*, vol. 5, no. 4, pp. 168–179, Dec. 2024, doi: 10.62527/jitsi.5.4.287.
- [8] N. Agustina, A. Adrian, and M. Hermawati, “Implementasi Algoritma Naïve Bayes Classifier untuk Mendeteksi Berita Palsu pada Sosial Media,” *Fakt. Exacta*, vol. 14, no. 4, p. 206, Jan. 2022, doi: 10.30998/faktorexacta.v14i4.11259.
- [9] H. Amanda, N. Faiza, and L. S. Harahap, “Implementasi Multilayer Perceptron untuk Klasifikasi Berita Hoax dalam Media Sosial”.
- [10] R. R. Sani, Y. A. Pratiwi, S. Winarno, E. D. Udayanti, and F. A. Zami, “Analisis Perbandingan Algoritma Naive Bayes Classifier dan Support Vector

Machine untuk Klasifikasi Hoax pada Berita Online Indonesia,” vol. 13, no. 2, 2022.

- [11] R. S. Sutton and A. G. Barto, “Reinforcement Learning: An Introduction”.
- [12] Y. Han, S. Karunasekera, and C. Leckie, “Graph Neural Networks with Continual Learning for Fake News Detection from Social Media,” Aug. 14, 2020, *arXiv*: arXiv:2007.03316. doi: 10.48550/arXiv.2007.03316.
- [13] T. M. Mitchell, *Machine Learning*. in McGraw-Hill series in computer science. New York: McGraw-Hill, 1997.
- [14] L. Song, F. Chen, S. R. Young, C. D. Schuman, G. Perdue, and T. E. Potok, “Deep Learning for Vertex Reconstruction of Neutrino-Nucleus Interaction Events with Combined Energy and Time Data,” Feb. 02, 2019, *arXiv*: arXiv:1902.00743. doi: 10.48550/arXiv.1902.00743.
- [15] R. N. Rahayu, “Analisis Berita Hoax Covid - 19 di Media Sosial di Indonsia,” no. 09.
- [16] R. N. Rahayu, “E-ISSN 2686 5661 VOL.2 NO. 07 - FEBRUARY 2021 Intelektiva : Jurnal Ekonomi, Sosial & Humaniora,” no. 07.
- [17] A. Zahra and M. N. Fauzan, “Sistem Identifikasi ‘Fake News’ menggunakan Metode Multinomial Naïve Bayes,” *J. Sist. Dan Teknol. Inf. JustIN*, vol. 10, no. 4, p. 489, Dec. 2022, doi: 10.26418/justin.v10i4.52441.
- [18] C. S. Sriyano and E. B. Setiawan, “Pendeteksian Berita Hoax Menggunakan Naive Bayes Multinomial Pada Twitter dengan Fitur Pembobotan TF-IDF”.
- [19] N. Arifin, U. Enri, and N. Sulistiyowati, “Penerapan Algoritma Support Vector Machine (SVM) dengan TF-IDF N-Gram untuk Text Classification,” *STRING Satuan Tulisan Ris. Dan Inov. Teknol.*, vol. 6, no. 2, p. 129, Dec. 2021, doi: 10.30998/string.v6i2.10133.
- [20] E. Indrayuni, A. Nurhadi, and D. A. Kristiyanti, “Implementasi Algoritma Naive Bayes, Support Vector Machine, dan K-Nearest Neighbors untuk Analisa Sentimen Aplikasi Halodoc,” *Fakt. Exacta*, vol. 14, no. 2, p. 64, Aug. 2021, doi: 10.30998/faktorexacta.v14i2.9697.
- [21] D. Krstinić, M. Braović, L. Šerić, and D. Božić-Štulić, “Multi-label Classifier Performance Evaluation with Confusion Matrix,” in *Computer Science & Information Technology*, AIRCC Publishing Corporation, Jun. 2020, pp. 01–14. doi: 10.5121/csit.2020.100801.

- [22] N. Widhiyanta, I. Muhandhis, R. S. Jannah, and L. A. Wulansari, “Analisis Sentimen Ulasan Produk Moisturizer Skintific di Tokopedia Menggunakan Support Vector Machine,” *J. Sist. Inf. Dan Bisnis Cerdas*, vol. 18, no. 1, pp. 129–142, 2025.
- [23] V. B. Lestari and C. A. Hutagalung, “Evaluation of TF-IDF Extraction Techniques in Sentiment Analysis of Indonesian-Language Marketplaces Using SVM, Logistic Regression, and Naive Bayes,” *J-KOMA (Journal of Computer Science and Applications)*, vol. 08, no. 01, pp. 36–44, 2025.