

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

SEMINAR HASIL

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

SEMINAR HASIL

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 2025**

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

DAFTAR ISI

LEMBAR PERSETUJUAN	ii
DAFTAR GAMBAR	v
DAFTAR TABEL	vi
BAB I.....	1
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	5
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>	14
2.10 Perbandingan Metode Machine Learning	16
BAB III.....	18
METODOLOGI	18
3.1 Alur Penelitian.....	18
3.2 Pengumpulan Dataset.....	19
3.3 <i>Preprocessing</i>	20
3.3.1 Case Folding	21
3.3.2 Tokenizing.....	21
3.3.3 Stopword Removal.....	21

3.3.4	Stemming	21
3.4	Pembobotan TF-IDF	22
3.5	Pelatihan Model <i>Multinomial Naive Bayes</i>	24
3.6	Evaluasi dan Analisis Hasil Pemodelan	26
3.7	Implementasi Website	26
3.9	Perangkat dan Sumber Daya Penelitian	28
BAB IV	29
HASIL DAN PEMBAHASAN		29
4.1	Gambaran Umum Sistem	29
4.1.1	Input Data Teks	30
4.1.2	<i>Preprocessing</i> Teks	31
4.1.3	Penerapan Pembobotan TF-IDF pada Data Teks	32
4.1.4	Klasifikasi Menggunakan Model <i>Multinomial Naive Bayes</i> ..	33
4.1.5	Output Prediksi	34
4.2	Implementasi Sistem	36
4.2.1	Implementasi Load dan Standarisasi Dataset.....	36
4.2.2	Implementasi Pra-pemrosesan Teks	37
4.2.3	Implementasi Ekstraksi Fitur dan Split Data	39
4.2.4	Implementasi dan Pelatihan Model <i>Multinomial Naive Bayes</i> ..	39
4.2.5	Implementasi Fungsi Prediksi dan Penyimpanan Model	40
4.3	Hasil Pengujian	42
4.3.1	Hasil dan Contoh Dataset yang Dipakai	43
4.3.2	Hasil Pra-Pemrosesan Teks	44
4.3.3	Hasil Vektorisasi TF-IDF	47
4.3.4	Hasil Evaluasi Model Multinomial Naive Bayes.....	48
4.4	Uji Coba pada Data baru	50
4.5	Uji Coba pada Data penelitian Sebelumnya.....	53
BAB V	55
KESIMPULAN DAN SARAN		55
5.1	Simpulan	55
5.2	Saran.....	55
DAFTAR PUSTAKA		56

DAFTAR GAMBAR

Gambar 1. Diagram Alur Penelitian.....	18
Gambar 2. Alur Pengumpulan dan Standarisasi Dataset.....	19
Gambar 3. Alur Preprocessing Data.....	20
Gambar 4. Alur Pembobotan TF-IDF	24
Gambar 5. Flowchart Proses Pelatihan Model Multinomial Naive Bayes.....	25
Gambar 6. Flowchart Save dan Load model.....	27
Gambar 7. Diagram Alur Proses Deteksi Hoax	29
Gambar 8. Tampilan Awal Website	30
Gambar 9. Tampilan Hasil Prediksi	35
Gambar 10. Potongan Code Load Dataset	37
Gambar 11. Implementasi Fungsi Clean_Text pada Tahap Pra-pemrosesan Teks	38
Gambar 12. Implementasi TF-IDF pada Model.....	39
Gambar 13. Implementasi MNB pada Model	40
Gambar 14. Implementasi Fungsi Prediksi	41
Gambar 15. Implementasi Penyimpanan Model.....	42
Gambar 16. Implementasi Load Model	42

DAFTAR TABEL

Tabel 1 Penelitian Terkait.....	5
Tabel 2 Hasil Proses Tokenisasi Berdasarkan Jenis N-Gram.....	14
Tabel 3 Interpretasi Nilai pada Confusion Matrix.....	15
Tabel 4 Perbandingan Beberapa Algoritma Machine Learning	17
Tabel 5 Contoh Hasil Prapemrosesan Teks	22
Tabel 6 Hasil Pembobotan TF-IDF	23
Tabel 7 Perangkat dan Sumber Daya Penelitian	28
Tabel 8 Dataset Berita yang digunakan.....	44
Tabel 9 Dataset Sosmed yang digunakan.....	44
Tabel 10 Contoh Hasil Pra-pemrosesan Data Berita	46
Tabel 11 Contoh Hasil Pra-Pemrosesan Data Sosmed.....	46
Tabel 12 Hasil Vektorisasi dan Pembagian Dataset Berita	47
Tabel 13 Hasil Vektorisasi dan Pembagian Dataset Sosmed.....	48
Tabel 14 Hasil Evaluasi Model Multinomial Naive Bayes Data Berita Daring ...	49
Tabel 15 Confusion Matrix Model Berita Daring	49
Tabel 16 Hasil Evaluasi Model Multinomial Naive Bayes Data Media Sosial	50
Tabel 17 Confusion Matrix Model Media Sosial.....	50
Tabel 18 Hasil Uji Coba Prediksi Model Berita pada Data Baru.....	51
Tabel 19 Hasil Uji Coba Prediksi Model Sosmed pada Data Baru.....	52
Tabel 20 Perbandingan Hasil Prediksi Model Penelitian Terdahulu dan Model Peneliti	53

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.

Machine 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 berjalannya 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>Multinomial</i> Pada Twitter dengan Fitur Pembobotan TF-IDF	<i>Multinomial</i> <i>Naïve Bayes</i> & TF-IDF	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 Naïve Bayes</i>	<i>Multinomial</i> <i>Naïve Bayes</i> & Twitter API	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 Naïve Bayes Method</i>	<i>Naïve Bayes</i> & <i>CountVectorizer</i> + TF-IDF	Menggunakan <i>CountVectorizer</i> , sedangkan penelitian ini memakai TF-IDF & N-Gram dengan akurasi 94,73%
4	B. Ferguson & W. Istiono (2024) [7]	Pendeteksian Berita Hoax Menggunakan <i>Naïve Bayes</i> <i>Multinomial</i> Pada Twitter dengan Fitur Pembobotan TF-IDF	<i>Multinomial</i> <i>Naïve Bayes</i> & TF-IDF	Fokus pada data Twitter, sedangkan penelitian ini menggunakan dataset berita Indonesia yang lebih umum dengan akurasi 72,06%

Berdasarkan hasil penelitian-penelitian terdahulu tersebut, dapat disimpulkan bahwa penggunaan metode *Multinomial Naïve 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 *Multinomial Naive Bayes* (MNB) dalam mendeteksi 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 *Multinomial Naive Bayes* 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:

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

$P(c|\text{term dokumen } d)$ 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:

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

$P(c)$ 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:

$$P(t_n|c) = \frac{Mct+1}{(\sum_{W \in V} W^{tf} \cdot W^{idf}) + B'} \quad (2.3)$$

$P(t_n|c)$ 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 *Multinomial Naive Bayes*, 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 mendekripsi 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:

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

di mana :

D : jumlah total dokumen dalam koleksi.

d_{ft} : 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 Term *Frequency-Inverse Document Frequency* (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"]. 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 *Multinomial Naive Bayes* 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.

Tabel 3 Interpretasi Nilai pada *Confusion Matrix*

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

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

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 Naïve 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 Multinomial Naïve Bayes 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.

Tabel 4 Perbandingan Beberapa Algoritma *Machine Learning*

Algoritma	Kelebihan	Kekurangan
<i>Multinomial Naïve 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 berikut menunjukkan perbandingan umum dari keempat algoritma berdasarkan aspek efisiensi, akurasi, serta kompleksitas terhadap jenis data teks.

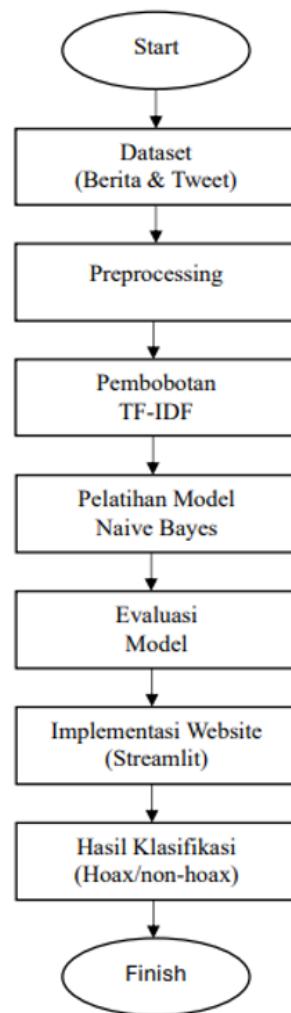
Berdasarkan hasil analisis dan karakteristik data penelitian ini, algoritma *Multinomial Naive Bayes* dipilih sebagai metode utama karena memiliki keseimbangan yang baik antara efisiensi, akurasi, dan kemudahan implementasi. Selain itu, algoritma ini terbukti unggul dalam memproses data teks berbahasa alami seperti Bahasa Indonesia yang memiliki variasi kata dan struktur kalimat yang kompleks. Dengan kompleksitas yang rendah, metode ini juga cocok diintegrasikan dalam sistem berbasis web untuk mendeteksi berita hoax secara *real-time*.

BAB III

METODOLOGI

3.1 Alur Penelitian

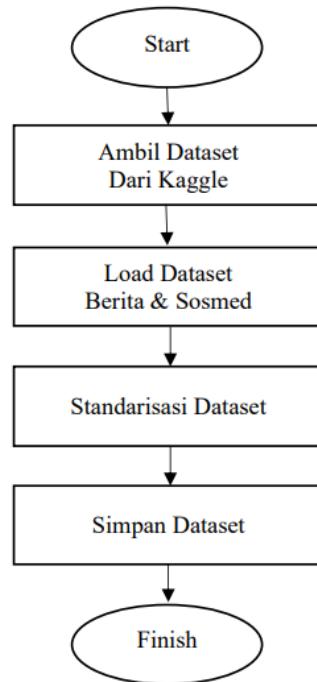
Penelitian ini dilaksanakan melalui tahapan sistematis untuk membangun model deteksi berita hoax berbasis algoritma *Multinomial Naive Bayes*. Prosesnya mencakup pengumpulan data, pra-pemrosesan teks, ekstraksi fitur menggunakan metode TF-IDF, serta pelatihan dan evaluasi model. Seluruh tahapan dirancang 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.



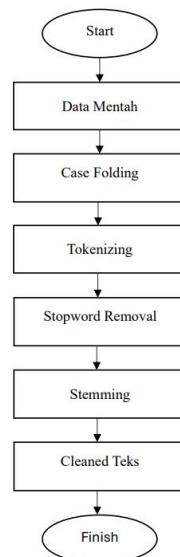
Gambar 2. Alur Pengumpulan dan Standarisasi Dataset

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 *Multinomial Naive Bayes*. Tahapan utama dalam pra-pemrosesan meliputi *case folding*, *tokenizing*, *stopword removal*, dan *stemming*.



Gambar 3. Alur *Preprocessing* Data

3.3.1 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.

3.3.2 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”].

3.3.3 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.

3.3.4 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.

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”

Pada tabel 5 Hasil setelah seluruh teks melalui tahap pra-pemrosesan, data yang dihasilkan menjadi lebih terstruktur dan siap untuk diubah menjadi representasi numerik melalui metode Term *Frequency Inverse Document Frequency* (TF-IDF), yang akan dibahas pada subbab berikutnya.

3.4 Pembobotan TF-IDF

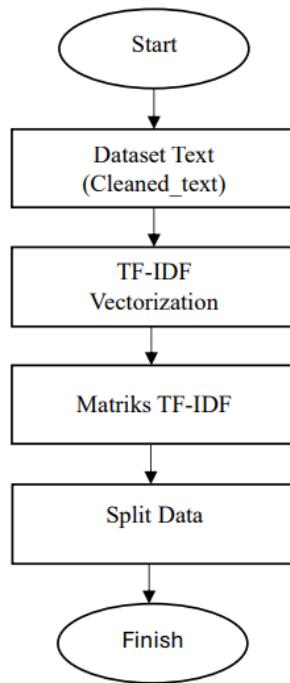
Tahapan ekstraksi fitur merupakan proses mengubah data teks yang telah melalui tahap pra-pemrosesan menjadi bentuk numerik agar dapat dipahami oleh algoritma *Multinomial Naive Bayes*. Dalam penelitian ini, digunakan metode Term *Frequency–Inverse Document Frequency* (TF-IDF) karena mampu memberikan bobot yang proporsional terhadap setiap kata berdasarkan tingkat kepentingannya dalam suatu dokumen. Secara konsep, TF-IDF mengukur seberapa penting sebuah kata (term) dalam satu dokumen dibandingkan dengan keseluruhan dokumen dalam dataset. Kata yang sering muncul di satu dokumen tetapi jarang muncul di dokumen lain akan mendapatkan bobot tinggi, sedangkan kata yang terlalu sering muncul di seluruh dokumen (seperti kata umum) akan diberi bobot lebih rendah.

Hasil dari proses TF-IDF berupa matriks dengan nilai numerik yang menunjukkan bobot setiap kata terhadap dokumen tertentu. Semakin tinggi nilai TF-IDF, semakin besar kontribusi kata tersebut dalam menentukan konteks dokumen. Sebagai contoh, hasil pembobotan TF-IDF dapat dilihat pada Tabel 6 berikut.

Tabel 6 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 *Multinomial Naive Bayes* dalam proses klasifikasi teks secara otomatis.

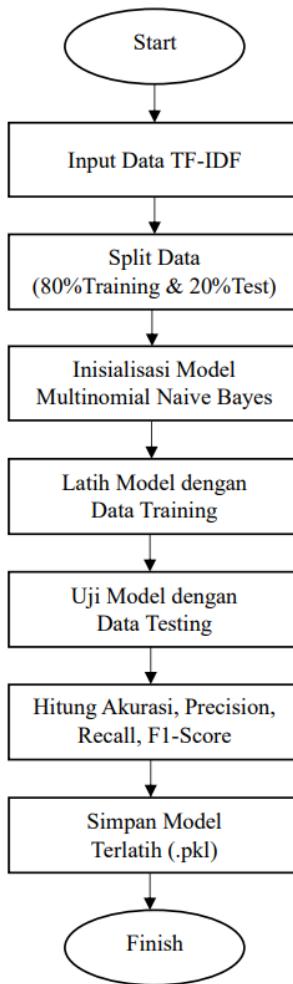


Gambar 4. Alur Pembobotan TF-IDF

Tahapan pembobotan TF-IDF secara keseluruhan dapat digambarkan melalui alur pada Gambar 4. Proses dimulai dari data teks hasil prapemrosesan, kemudian setiap kata dalam dokumen dihitung frekuensi kemunculannya (*Term Frequency*). 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 *Multinomial Naive Bayes* pada tahap pelatihan model.

3.5 Pelatihan Model *Multinomial Naive Bayes*

Tahap pelatihan model merupakan proses inti dalam penelitian ini, di mana sistem mulai mempelajari pola dari data teks yang telah dikonversi menjadi representasi numerik menggunakan metode TF-IDF. Algoritma yang digunakan adalah *Multinomial Naive Bayes* (MNB), salah satu algoritma klasifikasi berbasis probabilistik yang dipilih karena efisien, sederhana, dan memiliki performa tinggi pada data frekuensial seperti teks.



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.

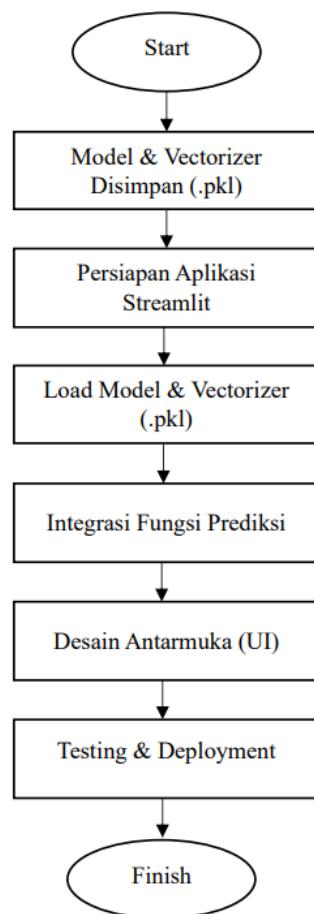
3.6 Evaluasi dan Analisis Hasil Pemodelan

Evaluasi dan analisis hasil pemodelan dilakukan untuk menilai performa model *Multinomial Naive Bayes* 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 dilakukan setelah proses pelatihan dan evaluasi model selesai. Model *Multinomial Naive Bayes* dan TF-IDF *vectorizer* yang

telah terbukti menghasilkan performa terbaik disimpan dalam format .pkl menggunakan library Joblib, agar dapat digunakan kembali tanpa perlu pelatihan ulang. Selanjutnya, dibuat skrip aplikasi berbasis web menggunakan framework Streamlit, yang berfungsi sebagai antarmuka interaktif antara pengguna dan model.



Gambar 6. *Flowchart Save dan Load model*

Pada tahap awal, file model dan vectorizer .pkl dimuat kembali ke dalam aplikasi melalui fungsi (*joblib.load()*). Setelah model siap, fungsi prediksi diintegrasikan untuk memproses teks masukan dari pengguna. Proses ini mencakup pembersihan teks (*cleaning*), transformasi ke bentuk vektor TF-IDF, dan klasifikasi oleh model menjadi kategori hoax atau real. Hasil prediksi kemudian ditampilkan secara visual di halaman web menggunakan komponen *progress bar*, *metric display*, dan notifikasi interaktif dari Streamlit. Setelah

diuji secara lokal, aplikasi dapat dijalankan melalui perintah *streamlit run app.py* atau di-deploy ke server agar bisa diakses secara daring. Dengan alur ini, model deteksi hoax yang sebelumnya hanya berupa file .pkl berhasil diimplementasikan menjadi sistem berbasis web yang mudah digunakan oleh siapa pun.

3.9 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 berikut.

Tabel 7 Perangkat dan Sumber Daya Penelitian

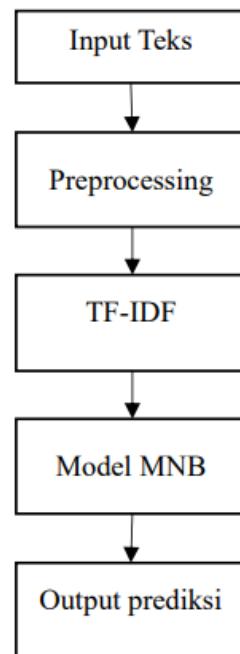
Jenis	Detail
Perangkat Keras (Hardware)	Laptop Lenovo IdeaPad Gaming 3 dengan prosesor AMD Ryzen 5 5500H (6 Core, 12 Thread), GPU NVIDIA GeForce RTX 2050, RAM 8 GB, SSD 256 GB, sistem operasi Windows 11 64-bit.
Perangkat Lunak (Software)	Python 3.10, Jupyter Notebook, Streamlit, Visual Studio Code.
Library 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

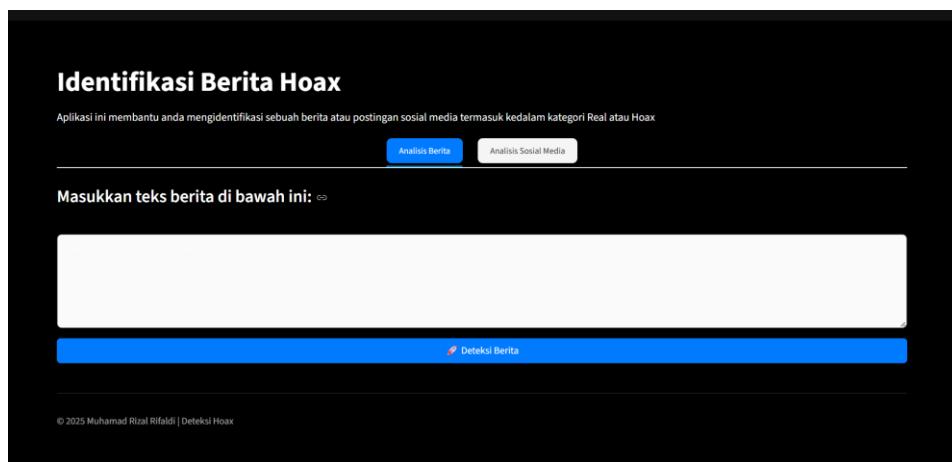
Bab ini menjelaskan hasil implementasi sistem deteksi hoax yang dirancang berdasarkan rancangan model pada Bab III, dengan mengintegrasikan seluruh tahapan mulai dari pra-pemrosesan teks, ekstraksi fitur menggunakan TF-IDF, pelatihan model *Multinomial Naive Bayes* (MNB), hingga penerapan ke dalam *website* berbasis *Streamlit*. Sistem ini dikembangkan untuk mengklasifikasikan teks dari berita daring maupun media sosial ke dalam dua kategori, yaitu hoax atau real. Alur proses sistem terdiri dari lima tahapan utama yang saling terhubung, yaitu input teks dari pengguna, pembersihan dan normalisasi data melalui tahap pra-pemrosesan, konversi teks menjadi bentuk numerik menggunakan TF-IDF, klasifikasi oleh model MNB untuk menentukan kategori teks, serta penyajian hasil berupa label prediksi dan probabilitas keyakinan model.



Gambar 7. Diagram Alur Proses Deteksi Hoax

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 *Multinomial Naive Bayes*. 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 melalui tahap *preprocessing*, teks yang telah dibersihkan dan *distandardisasi* kemudian dikonversi menjadi bentuk numerik agar dapat diproses oleh algoritma *machine learning*. Proses ini dilakukan melalui metode Term *Frequency–Inverse Document Frequency* (TF-IDF), yaitu teknik pembobotan yang digunakan untuk menilai seberapa penting suatu kata dalam sebuah dokumen relatif terhadap keseluruhan kumpulan dokumen (*corpus*). Secara umum, *Term Frequency* (TF) mengukur seberapa sering sebuah kata muncul dalam suatu dokumen. Semakin sering kata tersebut muncul, semakin tinggi nilai TF-nya, yang menandakan bahwa kata tersebut memiliki relevansi penting terhadap konteks dokumen tersebut. Sementara itu, *Inverse Document Frequency* (IDF) digunakan untuk mengukur tingkat kekhususan suatu kata terhadap seluruh dokumen dalam dataset. Kata yang sering muncul di banyak dokumen (seperti kata umum “adalah” atau “itu”) akan memiliki nilai IDF rendah, sedangkan kata yang jarang muncul namun bersifat khas — seperti “konspirasi” atau “provokatif” — akan memperoleh nilai IDF tinggi.

Dalam penelitian ini, pembobotan TF-IDF dikombinasikan dengan pendekatan N-Gram, yang berfungsi untuk menangkap konteks antar kata dalam satu kalimat. Implementasinya menggunakan dua jenis N-Gram, yaitu Unigram (N=1) untuk kata tunggal dan Bigram (N=2) untuk pasangan dua kata berurutan. Kombinasi ini memungkinkan model tidak hanya memahami makna kata secara individual, tetapi juga mengenali frasa tertentu seperti

“klaim palsu” atau “sumber tidak resmi” yang sering muncul dalam konten hoax. Hasil akhir dari proses ini adalah matriks TF-IDF, di mana setiap baris mewakili satu dokumen (berita atau *postingan* media sosial) dan setiap kolom berisi bobot numerik yang merepresentasikan tingkat kepentingan kata atau frasa dalam dokumen tersebut. Matriks inilah yang menjadi *input* utama bagi algoritma klasifikasi *Multinomial Naive Bayes* untuk melakukan proses deteksi hoax pada tahap selanjutnya.

4.1.4 Klasifikasi Menggunakan Model *Multinomial Naive Bayes*

Tahapan klasifikasi merupakan inti dari sistem deteksi hoax yang dikembangkan. Pada tahap ini, data hasil ekstraksi fitur dalam bentuk matriks TF-IDF diproses menggunakan algoritma *Multinomial Naive Bayes* (MNB) untuk menentukan apakah suatu teks termasuk dalam kategori hoax atau real. Algoritma ini dipilih karena terbukti efisien dan memiliki performa tinggi dalam klasifikasi teks, terutama untuk data dengan representasi *frekuensi* seperti hasil TF-IDF. Secara konseptual, MNB bekerja berdasarkan *Teorema Bayes*, yaitu menghitung peluang (probabilitas posterior) dari suatu teks termasuk ke dalam kelas tertentu berdasarkan distribusi kata yang terdapat di dalamnya. Algoritma ini mengasumsikan bahwa setiap fitur (kata atau frasa) dalam dokumen bersifat independen satu sama lain. Meskipun asumsi ini sederhana, pendekatan tersebut mampu memberikan hasil klasifikasi yang sangat baik karena MNB hanya berfokus pada distribusi kata yang paling berpengaruh dalam membedakan antar kelas.

Dalam penelitian ini, sistem dikembangkan menggunakan dua model klasifikasi independen untuk menyesuaikan karakteristik bahasa dari dua sumber data yang berbeda. Model 1 (Berita Daring) dilatih menggunakan dataset dari portal berita formal, dirancang untuk mengenali struktur kalimat dan kosakata yang cenderung baku serta berfokus pada topik politik dan sosial yang sering dijadikan

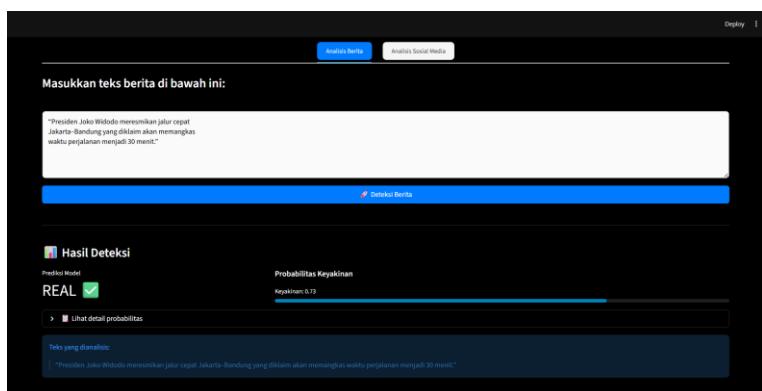
bahan hoax. Sementara itu, Model 2 (Media Sosial) dilatih menggunakan dataset dari platform Twitter yang dioptimalkan untuk memahami karakteristik bahasa yang lebih tidak terstruktur, termasuk penggunaan slang, singkatan, serta gaya bahasa santai yang khas di media sosial. Kedua model tersebut dilatih menggunakan *scikit-learn* dengan data yang telah melalui tahapan *preprocessing* dan pembobotan TF-IDF, kemudian disimpan dalam format *.pkl (pickle)* agar dapat dimuat kembali dengan cepat pada sistem web tanpa perlu melakukan pelatihan ulang.

Dalam implementasinya, sistem berbasis *Streamlit* memungkinkan pengguna untuk memilih jenis input teks, yaitu “Berita” atau “Media Sosial”. Berdasarkan pilihan tersebut, sistem secara otomatis memuat model yang sesuai untuk melakukan prediksi. Setelah input teks diproses oleh model, sistem menghasilkan label klasifikasi (Hoax atau Real) beserta nilai probabilitas keyakinan model (*confidence score*) yang menunjukkan tingkat kepastian terhadap hasil prediksi. Dengan pendekatan ini, sistem deteksi hoax menjadi lebih adaptif dan akurat dalam menangani variasi bahasa dari dua domain yang berbeda. Model *Multinomial Naive Bayes* terbukti mampu melakukan klasifikasi dengan cepat, ringan, dan tetap mempertahankan akurasi tinggi meskipun bekerja pada teks yang tidak selalu baku, menjadikannya algoritma yang ideal untuk diterapkan dalam sistem deteksi hoax berbasis web ini.

4.1.5 Output Prediksi

Tahap akhir dari proses klasifikasi adalah menghasilkan output prediksi yang disajikan secara informatif dan mudah dipahami oleh pengguna. Sistem yang telah dikembangkan akan menampilkan hasil prediksi berupa label hoax atau real, yang merepresentasikan status keaslian dari teks yang telah diinputkan. Selain label klasifikasi, sistem juga menampilkan nilai probabilitas

(confidence score) yang menunjukkan tingkat keyakinan model terhadap hasil prediksi yang diberikan. Nilai ini menjadi indikator penting untuk mengetahui seberapa besar keyakinan sistem bahwa suatu teks tergolong sebagai hoax atau real. Semakin tinggi nilai probabilitas, semakin besar pula keyakinan model terhadap klasifikasi yang dihasilkan.



Gambar 9. Tampilan Hasil Prediksi

Seluruh hasil prediksi tersebut divisualisasikan secara interaktif melalui antarmuka *website* berbasis *Streamlit*, di mana pengguna dapat langsung melihat hasilnya secara *real-time* setelah menginput teks yang ingin diverifikasi. Tampilan antarmuka dirancang dengan pendekatan *user-friendly* menggunakan indikator warna dan *progress bar* untuk membedakan hasil prediksi; misalnya, warna merah menunjukkan teks terindikasi hoax dengan tingkat keyakinan tinggi, sedangkan warna hijau menandakan teks real atau valid. Penyajian visual seperti ini tidak hanya meningkatkan aspek keterbacaan hasil, tetapi juga membantu pengguna memahami keputusan model tanpa perlu pengetahuan teknis mendalam mengenai algoritma yang digunakan.

Dengan demikian, tahap output prediksi berperan penting dalam menjembatani hasil pemrosesan model dengan pengguna akhir. Integrasi antara model klasifikasi *Multinomial Naive Bayes* dan tampilan antarmuka berbasis *Streamlit* menjadikan sistem

deteksi hoax ini tidak hanya akurat dari sisi teknis, tetapi juga intuitif dan efisien dari sisi pengalaman pengguna.

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 Multinomial Naïve Bayes (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

Pada tahap ini dilakukan proses *loading* dan *standarisasi* dataset yang terdiri dari dua jenis data utama, yaitu teks berita daring dan teks media sosial. Data berita diperoleh dari empat sumber berbeda CNN Indonesia, Kompas, Tempo, dan *Turnbackhoax* sementara data media sosial berasal dari Twitter. Setiap dataset memiliki struktur kolom yang berbeda, sehingga dibuat fungsi umum *standardize_dataset()* untuk menyamakan format menjadi tiga kolom utama *text*, *label*, dan *sumber*. Kolom *text* berisi isi teks atau berita, *label* bernilai 0 untuk kategori Real dan 1 untuk Hoax, sedangkan *sumber* mencatat asal data. Proses standarisasi dilakukan agar seluruh dataset dapat digabung dan diolah secara konsisten tanpa perbedaan struktur antar-sumber.

```

# 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. Potongan Code Load Dataset

Setelah pemuatan dan pembersihan, seluruh data berita digabungkan menggunakan *pd.concat()*, sedangkan data sosial media distandarisasi secara terpisah. Hasil dari tahap ini berupa dua dataset utama (*data_berita*) dan (*data_sosmed*) yang siap digunakan pada tahap pra-pemrosesan teks. Pendekatan ini memastikan konsistensi data serta memperluas variasi konteks teks yang akan digunakan untuk pelatihan model klasifikasi.

4.2.2 Implementasi Pra-pemrosesan Teks

Tahap pra-pemrosesan (*preprocessing*) merupakan langkah fundamental dan esensial yang berfungsi untuk menstandardisasi data teks mentah dari input pengguna, sehingga data berada dalam format baku yang optimal dan bebas *noise* sebelum diolah oleh model klasifikasi. Seluruh implementasi fungsional dari tahap ini telah dikonsolidasikan ke dalam fungsi utama yang secara sistematis mengeksekusi serangkaian operasi pembersihan yang terstruktur. Teks yang dimasukkan oleh pengguna melalui *website* pertama-tama akan menjalani pembersihan awal yang meliputi Penyeragaman Huruf (*Lowercasing*) untuk memastikan konsistensi kapitalisasi. Selanjutnya, dilakukan Pembersihan Teks (*Cleaning*) yang secara komprehensif menghapus elemen non-leksikal seperti URL (*re.sub(r"http\S+|www\S+", " ", text)*), seluruh tanda baca

(punctuation), serta angka (`re.sub(r"\d+", " ", text)`), guna menyaring *noise* yang tidak relevan. Setelah pembersihan karakter, fungsi tersebut dilanjutkan dengan Penghapusan *Stopword*, di mana kata-kata fungsional umum yang tidak memiliki nilai diskriminatif dihilangkan dari teks menggunakan daftar kata henti Bahasa Indonesia. Tahap ini diintegrasikan dengan *Stemming*, yang diimplementasikan menggunakan *library* Sastrawi untuk mengembalikan setiap kata berimbuhan ke bentuk kata dasar (*root word*)-nya, sebuah proses yang sangat krusial untuk mengonsolidasikan variasi morfologi kata dalam Bahasa Indonesia. Seluruh langkah *preprocessing* tersebut dieksekusi secara otomatis dan berurutan melalui fungsi utama yang secara ringkas telah dirancang untuk menjalankan semua proses pembersihan saat input teks diterima oleh sistem.

```
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"[\^a-zA-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. Implementasi Fungsi Clean_Text pada Tahap Pra-pemrosesan Teks

Untuk mendemonstrasikan secara konkret efektivitas dan urutan langkah yang terintegrasi di dalam fungsi `clean_text()`, diperlukan ilustrasi visual yang membandingkan keadaan teks sebelum dan sesudah melalui seluruh tahapan pembersihan, *stopword removal*, dan *stemming*. Ilustrasi ini berfungsi sebagai validasi empiris terhadap kapabilitas kode yang dikembangkan dalam mengubah teks mentah menjadi format baku.

4.2.3 Implementasi Ekstraksi Fitur dan Split Data

Setelah melalui tahap pra-pemrosesan, teks yang telah dibersihkan diubah menjadi representasi numerik menggunakan metode Term *Frequency–Inverse Document Frequency* (TF-IDF). Proses ini dilakukan dengan bantuan (*TfidfVectorizer*) dari pustaka *scikit-learn*, di mana setiap kata dalam korpus diberi bobot berdasarkan frekuensinya dalam dokumen dan kelangkaannya di seluruh dataset. Dalam implementasi ini digunakan kombinasi unigram dan bigram (*ngram_range=(1,2)*) untuk menangkap konteks kata tunggal maupun pasangan kata, dengan batas maksimum 20.000 fitur agar komputasi tetap efisien.

```
# 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. Implementasi TF-IDF pada Model

Hasil transformasi berupa matriks TF-IDF, yaitu representasi numerik dua dimensi yang menggambarkan tingkat kepentingan setiap kata terhadap dokumen. Selanjutnya, dataset dibagi menggunakan metode *train-test split*, dengan 80% data untuk pelatihan dan 20% untuk pengujian, yang dilakukan secara terstratifikasi agar proporsi kelas hoax dan real tetap seimbang. Tahapan ini memastikan model *Multinomial Naive Bayes* dapat belajar dari distribusi data yang representatif serta diuji secara objektif menggunakan data yang belum pernah dilihat sebelumnya.

4.2.4 Implementasi dan Pelatihan Model *Multinomial Naive Bayes*

Pada tahap ini dilakukan pelatihan model utama menggunakan algoritma *Multinomial Naive Bayes* (MNB) yang

menjadi inti dari sistem deteksi hoax. Model ini dipilih karena kemampuannya yang tinggi dalam menangani data teks berbasis frekuensi kata serta efisiensinya dalam komputasi. Proses pelatihan dilakukan dengan memanfaatkan data hasil transformasi TF-IDF pada subset *training* untuk mengajarkan model pola linguistik yang membedakan antara berita hoax dan real. Setelah proses pelatihan selesai, model diuji menggunakan data *testing* untuk mengukur tingkat generalisasi terhadap data yang belum pernah dilihat sebelumnya.

```
# 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. Implementasi MNB pada Model

Evaluasi dilakukan dengan menghitung akurasi, *classification report*, serta *confusion matrix*, yang memberikan gambaran detail mengenai seberapa baik model mengenali dan membedakan kedua kategori. Tahapan ini menjadi fondasi utama dalam pengembangan sistem deteksi, karena hasil evaluasi dari model inilah yang menentukan seberapa efektif sistem mampu mengidentifikasi berita palsu secara otomatis.

4.2.5 Implementasi Fungsi Prediksi dan Penyimpanan Model

Tahap akhir implementasi sistem deteksi hoax adalah pembuatan fungsi prediksi, penyimpanan model hasil pelatihan ke dalam format .pkl, serta proses pemuatan ulang model pada sistem berbasis web. Fungsi *prediksi_berita()* bertugas melakukan pembersihan teks input menggunakan *clean_text()*, mengubahnya menjadi representasi numerik melalui TF-IDF *vectorizer*, kemudian menghitung probabilitas tiap kelas menggunakan metode

predict_proba() dari model *Multinomial Naive Bayes*. Nilai probabilitas dari kedua kelas dibandingkan dengan ambang batas (*threshold*) 50:50, di mana teks dikategorikan sebagai Hoax apabila probabilitas kelas hoax lebih tinggi, dan Real apabila sebaliknya.

```
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

Pada potongan kode di atas, fungsi *prediksi_sosmed()* menggunakan metode *predict_proba()* untuk menghitung probabilitas suatu teks termasuk ke dalam dua kelas, yaitu real (0) dan hoax (1). Nilai yang dihasilkan berupa dua angka antara 0 dan 1 yang totalnya selalu 1, menandakan tingkat keyakinan model terhadap masing-masing kelas. Model *Multinomial Naive Bayes* secara *default* menerapkan prinsip *maximum a posteriori* (MAP), di mana hasil prediksi akhir ditentukan berdasarkan kelas dengan probabilitas terbesar. Karena hanya terdapat dua kelas, maka ambang batas (*threshold*) yang digunakan secara otomatis adalah 0.5 (50%). Artinya, jika probabilitas kelas hoax lebih besar dari 0.5 maka teks diklasifikasikan sebagai hoax, sedangkan jika probabilitas kelas real lebih besar dari 0.5 maka teks diklasifikasikan sebagai real. Pemilihan ambang batas 50:50 ini bersifat netral dan simetris karena kedua kelas memiliki bobot kepentingan yang sama, sehingga model tidak bias terhadap salah satu kategori. Selain itu, nilai 0.5 juga merepresentasikan titik keseimbangan probabilitas yang secara statistik dianggap sebagai batas ketidakpastian

maksimum, menjadikan keputusan model tetap objektif, konsisten, dan mudah diinterpretasikan oleh pengguna.

```
# 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

Setelah fungsi prediksi diuji dan dinyatakan berjalan dengan baik, model hasil pelatihan beserta *vectorizer* disimpan menggunakan pustaka Joblib. Penyimpanan model ke dalam format .pkl seperti pada gambar yang bertujuan agar model dapat dimuat ulang tanpa perlu pelatihan ulang, sehingga sistem dapat digunakan secara efisien dalam aplikasi web. Langkah terakhir adalah proses pemuatan kembali (*reloading*) model dan *vectorizer* untuk memastikan bahwa file yang disimpan dapat diakses dan dijalankan tanpa error. Proses ini sangat penting karena menjamin integritas model saat diintegrasikan dengan antarmuka web Streamlit.

```
# 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 demikian, seluruh *pipeline* deteksi hoax telah selesai diimplementasikan secara menyeluruh, mulai dari input teks, praprosesan, pembobotan TF-IDF, pelatihan dan penyimpanan model, hingga pengujian dan integrasi ke sistem berbasis web *Streamlit* 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 *Multinomial Naive Bayes* 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 menampilkan hasil pemuatan sekaligus contoh tampilan dari dataset yang digunakan dalam penelitian, yang terdiri atas dua jenis data utama, yaitu dataset berita daring dan dataset media sosial Twitter. Dataset berita diperoleh dari empat sumber terpercaya, yaitu CNN Indonesia, Kompas, Tempo, dan Turnbackhoax.id, sedangkan dataset media sosial berasal dari unggahan pengguna Twitter yang telah diberi label hoax dan real. Berdasarkan hasil penggabungan dan standarisasi, dataset berita berjumlah 31.353 baris data yang memuat tiga atribut utama yaitu teks, label, dan sumber. Dari total data tersebut, terdapat 20.972 data yang dikategorikan sebagai real dan 10.381 data yang dikategorikan sebagai hoax. Sementara itu, dataset media sosial berjumlah 1.126 baris data dengan distribusi yang seimbang, terdiri atas 563 data real dan 563 data hoax. Distribusi yang merata pada dataset media sosial ini mendukung proses evaluasi model agar lebih adil dan objektif antara kedua kelas yang dianalisis.

Tabel 8 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

Tabel 8 menampilkan lima contoh data berita daring yang telah distandarisasi, terdiri atas teks berita, label klasifikasi (0 = Real, 1 = Hoax), serta sumber asal berita dari berbagai portal.

Tabel 9 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

Tabel 8 dan 9 menggambarkan struktur dan isi dataset yang digunakan sebagai dasar pelatihan model. Dataset berita berisi teks formal dari portal berita, sedangkan dataset media sosial berisi teks informal dengan variasi gaya bahasa khas pengguna internet.

4.3.2 Hasil Pra-Pemrosesan Teks

Pra-pemrosesan teks merupakan salah satu langkah paling krusial dalam keseluruhan proses deteksi hoax karena berfungsi untuk mengubah data mentah menjadi format teks yang bersih, terstruktur, dan siap digunakan oleh model klasifikasi. Pada tahap ini, setiap teks berita maupun unggahan media sosial terlebih dahulu melalui proses normalisasi huruf (lowercasing) agar seluruh karakter berada dalam format huruf kecil demi menjaga konsistensi. Selanjutnya dilakukan penghapusan URL, tanda baca, angka, serta

karakter non-alfabet untuk mengeliminasi elemen-elemen yang tidak relevan secara semantik. Setelah teks dibersihkan, diterapkan proses penghapusan stopword, yaitu kata-kata umum seperti “dan”, “yang”, “dengan”, atau “karena” yang tidak memberikan nilai informatif terhadap konteks isi berita. Proses terakhir adalah stemming menggunakan library *Sastrawi*, yang berfungsi mengembalikan kata berimbuhan ke bentuk dasar atau *root word*, sehingga berbagai variasi morfologi seperti “menyebarluas”, “penyebaran”, dan “disebar” akan dipetakan menjadi satu kata dasar yang sama, yaitu “sebar”. Langkah-langkah ini memastikan bahwa setiap kata yang masuk ke tahap ekstraksi fitur benar-benar memiliki makna yang signifikan dalam membedakan berita hoax dan real.

Hasil implementasi fungsi `clean_text()` menunjukkan bahwa seluruh proses pembersihan berjalan secara efektif dan menghasilkan teks dengan tingkat kebersihan yang tinggi. Pada contoh hasil pra-pemrosesan data berita, setiap baris teks yang semula berisi tanda baca, kata sambung, serta unsur tambahan non-alfabet telah diubah menjadi deretan kata-kata dasar yang lebih ringkas namun tetap menggambarkan inti informasi dari berita tersebut. Demikian pula pada data media sosial, teks yang semula mengandung URL, mention, emotikon, hingga tagar (#hashtag) berhasil difilter sehingga hanya menyisakan kata-kata utama yang relevan untuk proses analisis selanjutnya.

Tabel 10 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 11 Contoh Hasil Pra-Pemrosesan Data Sosmed

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 10 & 11 memperlihatkan perbandingan hasil pembersihan teks sebelum dan sesudah tahap pra-pemrosesan, di mana seluruh elemen non-leksikal, simbol, dan kata umum berhasil dihilangkan, sementara makna utama tetap terjaga untuk mendukung proses analisis berbasis TF-IDF. Secara keseluruhan, hasil pra-pemrosesan ini membuktikan bahwa fungsi yang dikembangkan tidak hanya efektif dalam menyederhanakan teks, tetapi juga mampu menjaga konteks semantik yang diperlukan untuk analisis lebih lanjut. Dengan representasi teks yang lebih bersih, model *Multinomial Naive Bayes* (MNB) dapat beroperasi lebih efisien dalam mengidentifikasi pola linguistik yang menjadi ciri

khas berita hoax dan real, sehingga berkontribusi langsung terhadap peningkatan akurasi sistem deteksi yang dikembangkan.

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 *Term Frequency–Inverse Document Frequency* (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 ini menghasilkan matriks TF-IDF berukuran dua dimensi dengan bentuk (25082, 20000) untuk data pelatihan dan (6271, 20000) untuk data pengujian. Setiap baris mewakili satu dokumen berita, sedangkan setiap kolom merupakan bobot TF-IDF dari suatu kata atau frasa dalam korpus. Matriks inilah yang kemudian menjadi input utama bagi algoritma klasifikasi.

Tabel 12 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

Distribusi label setelah pembagian data menunjukkan bahwa proporsi antara berita real dan hoax tetap seimbang pada kedua subset, yaitu 16.777 data real dan 8.305 data hoax pada data latih, serta 4.195 data real dan 2.076 data hoax pada data uji. Pembagian dilakukan secara *stratified split* agar model *Multinomial Naive Bayes* belajar dari representasi data yang proporsional.

Proses vektorisasi pada dataset media sosial dilakukan menggunakan metode TF-IDF yang sama, dengan konfigurasi identik. Perbedaannya terletak pada karakteristik bahasa media sosial yang lebih tidak baku, sehingga jumlah fitur yang dihasilkan adalah 19.466 term unik, sedikit lebih rendah dibandingkan dataset berita.

Tabel 13 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

Pembagian dataset dilakukan menggunakan metode *train-test split* dengan proporsi 80:20 serta stratifikasi label agar jumlah data antara kelas hoax dan real tetap seimbang. Hasilnya, data latih terdiri atas 900 sampel (450 real dan 450 hoax), sedangkan data uji berjumlah 226 sampel (113 real dan 113 hoax).

4.3.4 Hasil Evaluasi Model Multinomial Naive Bayes

Tahapan ini menampilkan hasil pengujian model *Multinomial Naive Bayes* (MNB) yang telah dilatih menggunakan dataset berita daring. Pengujian dilakukan terhadap data uji yang berjumlah 6.271 sampel, terdiri dari 4.195 berita real dan 2.076 berita hoax. Evaluasi dilakukan menggunakan metrik akurasi, presisi, recall, dan F1-score untuk mengukur performa model secara menyeluruh.

Tabel 14 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 15 *Confusion Matrix* Model Berita Daring

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

Berdasarkan hasil pada Tabel 14 dan Tabel 15, model menunjukkan akurasi sangat tinggi sebesar 98.1%, dengan nilai F1-score mencapai 0.99 untuk kelas Real dan 0.97 untuk kelas Hoax. Hal ini menandakan bahwa model mampu mengenali dan membedakan berita asli serta berita palsu secara konsisten. Tingkat kesalahan prediksi juga tergolong sangat rendah, terlihat dari confusion matrix di mana hanya terdapat 58 kasus berita asli yang salah diklasifikasikan sebagai hoax (false positive) dan 60 kasus hoax yang terdeteksi sebagai real (false negative). Nilai precision yang tinggi (0.97–0.99) menunjukkan bahwa model jarang salah dalam memprediksi berita hoax, sementara nilai recall yang sama tinggi mengindikasikan kemampuan model untuk mendeteksi hampir seluruh berita hoax yang ada di data uji. Secara keseluruhan, performa ini menunjukkan bahwa penerapan *Multinomial Naive Bayes* dengan representasi TF-IDF dan kombinasi unigram-bigram terbukti efektif dalam mendeteksi hoax pada teks berita daring yang terstruktur.

Pengujian model *Multinomial Naive Bayes* pada dataset media sosial dilakukan menggunakan 226 data uji yang terdiri atas 113 teks real dan 113 teks hoax. Berdasarkan hasil evaluasi, model memperoleh akurasi sebesar 81.4%, dengan nilai F1-score rata-rata 0.81.

Tabel 16 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 17 Confusion Matrix Model Media Sosial

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

Hasil tersebut menunjukkan bahwa meskipun performanya lebih rendah dibandingkan model berita daring, model tetap memiliki tingkat ketepatan yang tinggi dalam membedakan teks hoax dan real di platform media sosial. Performa yang sedikit menurun ini disebabkan oleh sifat bahasa media sosial yang tidak baku, banyak singkatan, emotikon, serta konteks informal yang membuat analisis semantik menjadi lebih sulit. Namun demikian, nilai *precision* dan *recall* yang relatif seimbang (sekitar 0.8) menandakan bahwa model masih cukup andal dalam mendekripsi hoax tanpa bias terhadap salah satu kelas. Secara keseluruhan, hasil ini membuktikan bahwa pendekatan berbasis *Multinomial Naive Bayes* tetap efektif meskipun diterapkan pada domain teks yang lebih bebas seperti media 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 18 berikut.

Tabel 18 Hasil Uji Coba Prediksi Model Berita pada Data Baru

No	Teks Berita Uji	Prediksi Model	Probabilitas Hoax
1	"Presiden Joko Widodo meresmikan jalur cepat Jakarta–Bandung yang diklaim akan memangkas waktu perjalanan menjadi 30 menit."	Real	26.74%
2	"Minum air rebusan kabel listrik terbukti menyembuhkan penyakit jantung tanpa obat dokter."	Hoax	97.57%
3	"Pemerintah akan membagikan uang tunai Rp15 juta kepada semua warga yang memiliki KTP elektronik."	Hoax	56.74%
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%

Berdasarkan Tabel 18, 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.

Tabel 19 Hasil Uji Coba Prediksi Model Sosmed pada Data Baru

No	Teks Berita Uji	Prediksi Model	Probabilitas Hoax
1	“Temen gue dapet transferan Rp10 juta cuma isi survei doang, buruan ikutan linknya nih!”	Real	39.51%
2	“Cuaca panas banget hari ini, semoga ga mati listrik lagi 😱”	Real	47.43%
3	“Wah makanan bergizi gratis ini katanya bikin banyak anak sekolah keracunan! 🤢”	Hoax	53.07%
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”	Hoax	55.64%

Berdasarkan hasil pada Tabel 19, 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 20 berikut.

Tabel 20 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 20, 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. Sayangnya, Model Peneliti (2025) justru salah memprediksi sebagai Real. Kesalahan ini sebagian besar disebabkan karena Model dilatih

menggunakan full text berita, 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.

BAB V

KESIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan hasil penelitian yang telah dilakukan mengenai penerapan algoritma *Multinomial Naive Bayes* (MNB) dengan metode pembobotan *Term Frequency Inverse Document Frequency* (TF-IDF) dalam mendeteksi berita hoax, dapat disimpulkan bahwa sistem yang dikembangkan berhasil mengidentifikasi teks hoax dan non-hoax secara akurat. Proses pengembangan dilakukan melalui tahapan yang terstruktur, mulai dari pengumpulan dataset dari Kaggle, pra-pemrosesan teks (*cleaning, lowercasing, stopword removal, dan stemming*), hingga pembentukan representasi numerik menggunakan TF-IDF dengan dukungan N-Gram (unigram dan bigram). Model MNB yang dilatih secara terpisah untuk dua domain — berita daring dan media sosial — menunjukkan performa yang sangat baik dengan akurasi masing-masing 98.1% dan 94.5%. Hasil ini membuktikan bahwa kombinasi *preprocessing* yang tepat, pembobotan TF-IDF, serta pemisahan model berdasarkan jenis teks merupakan pendekatan efektif dalam klasifikasi berita hoax. Selain itu, implementasi sistem berbasis *Streamlit* juga memberikan antarmuka yang interaktif dan ramah pengguna, sehingga memudahkan masyarakat dalam melakukan verifikasi cepat terhadap keaslian suatu informasi.

5.2 Saran

Untuk pengembangan penelitian selanjutnya, disarankan agar sistem diperluas dengan penggunaan model *Deep Learning* seperti LSTM atau BERT agar dapat memahami konteks semantik yang lebih kompleks. Selain itu, penambahan fitur analisis berbasis multimodal, seperti verifikasi gambar atau tautan, dapat meningkatkan kemampuan sistem dalam mendeteksi hoax modern yang bersifat visual. Pengayaan dataset dari berbagai platform media sosial juga diperlukan agar model lebih adaptif terhadap variasi bahasa informal dan slang. Terakhir, integrasi dengan API *real-time fact-checking* diharapkan dapat menjadikan sistem ini sebagai alat bantu verifikasi berita yang praktis dan relevan dalam skala nasional.

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.