

**PENERAPAN *SUPPORT VECTOR MACHINE (SVM)*
UNTUK DETEKSI HOAKS PADA BERITA *ONLINE***

SKRIPSI



Oleh:

RIFKY FACHUZI

211011401129

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PAMULANG
TANGERANG SELATAN
2025**

PENERAPAN *SUPPORT VECTOR MACHINE (SVM)* UNTUK DETEKSI HOAKS PADA BERITA *ONLINE*

SKRIPSI

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer



Oleh:

RIFKY FACHUZI
211011401129

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PAMULANG
TANGERANG SELATAN
2025**



FAKULTAS ILMU KOMPUTER
PROGRAM STUDI TEKNIK INFORMATIKA

LEMBAR PERNYATAAN

Yang bertanda tangan dibawah ini :

Nama : Rifky Fachuzi
Nim : 211011401129
Program Studi : Teknik Informatika
Fakultas : Ilmu Komputer
Jenjang Pendidikan : Strata 1

Menyatakan bahwa skripsi yang saya buat dengan judul :

PENERAPAN SUPPORT VECTOR MACHINE (SVM) UNTUK DETEKSI HOAKS PADA BERITA ONLINE.

1. Merupakan hasil karya ilmiah sendiri, bukan merupakan karya yang pernah diajukan untuk memperoleh gelar akademik oleh pihak lain, dan bukan merupakan hasil plagiat.
2. Saya izinkan untuk dikelola oleh Universitas Pamulang sesuai dengan norma hukum dan etika yang berlaku.

Pernyataan ini saya buat dengan penuh tanggung jawab dan saya bersedia menerima konsekuensi apa pun sesuai aturan yang berlaku apabila di kemudian hari pernyataan ini tidak benar.

Tangerang Selatan, 2025

Rifky Fachuzi



FAKULTAS ILMU KOMPUTER
PROGRAM STUDI TEKNIK INFORMATIKA

LEMBAR PERSETUJUAN

Nama : Rifky Fachuzi
Nim : 211011401129
Program Studi : Teknik Informatika
Fakultas : Ilmu Komputer
Jenjang Pendidikan : Strata 1
Judul Skripsi : PENERAPAN *SUPPORT VECTOR MACHINE (SVM)*
UNTUK DETEKSI HOAKS PADA BERITA *ONLINE*.

Skripsi ini telah diperiksa dan disetujui oleh pembimbing untuk persyaratan sidang skripsi.

Tangerang Selatan, 5 November 2025

Pembimbing

Raditia Vindua, S.Si., M.Kom.

NIDN : 0428059301

Mengetahui
Kepala Program Studi Teknik Informatika

Dr. Eng. Ahmad Musyafa, S.Kom., M.Kom.

NIDN : 0425018609



**FAKULTAS ILMU KOMPUTER
PROGRAM STUDI TEKNIK INFORMATIKA**

LEMBAR PENGESAHAN

Nama : Rifky Fachuzi
Nim : 211011401129
Program Studi : Teknik Informatika
Fakultas : Ilmu Komputer
Jenjang Pendidikan : Strata 1
Judul Skripsi : PENERAPAN *SUPPORT VECTOR MACHINE (SVM)*
UNTUK DETEKSI HOAKS PADA BERITA *ONLINE*

Skripsi ini telah di pertahankan di hadapan dewan penguji ujian skripsi fakultas Ilmu Komputer, program studi Teknik Informatika dan dinyatakan

Tangerang Selatan, 2025

Penguji 1

Penguji 2

NIDN

NIDN

Pembimbing

Raditia Vindua, S.Si., M.Kom.

NIDN : 0428059301

Mengetahui
Kepala Program Studi Teknik Informatika

Dr. Eng. Ahmad Musyafa, S.Kom., M.Kom.

NIDN : 0425018609

ABSTRACT

The spread of hoaxes in Indonesia has become a serious challenge requiring an automatic detection system. This research aims to implement a Machine Learning algorithm, specifically Support Vector Machine (SVM), to build a classification model capable of detecting Indonesian-language hoaxes. The primary objectives are to implement the model and quantitatively analyze its performance. The research methodology involved collecting hoax and fact news data from various online portals. This raw text data then underwent pre-processing and feature extraction using the Term Frequency-Inverse Document Frequency (TF-IDF) method. The SVM model was subsequently trained and tested using comparative scenarios, including two kernel types (Linear and RBF) and two feature count configurations (500 and 2000). The test results show that the best model configuration was achieved using the Linear kernel with 500 TF-IDF features. This model reached optimal performance with 88.90% accuracy, 88.86% precision, and 89.00% recall. This study concludes that the SVM algorithm with a Linear kernel is an effective and reliable method to be applied as a hoax detection solution for online news in Indonesia.

Keywords: Hoax Detection, Support Vector Machine (SVM), Machine Learning, Natural Language Processing (NLP), TF-IDF, Text Classification.

ABSTRAK

Penyebaran hoaks di Indonesia telah menjadi tantangan serius yang memerlukan sistem deteksi otomatis. Penelitian ini bertujuan untuk menerapkan algoritma *Machine Learning*, khususnya *Support Vector Machine* (SVM), guna membangun model klasifikasi yang mampu mendeteksi berita hoaks berbahasa Indonesia. Tujuan utamanya adalah untuk mengimplementasikan model dan menganalisis kinerjanya secara kuantitatif. Metodologi penelitian melibatkan pengumpulan data berita hoaks dan fakta dari berbagai portal *online*. Data teks mentah kemudian melalui tahap pra-pemrosesan dan ekstraksi fitur menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF). Model SVM kemudian dilatih dan diuji menggunakan perbandingan skenario yang mencakup dua jenis kernel (Linear dan RBF) serta dua konfigurasi jumlah fitur (500 dan 2000). Hasil pengujian menunjukkan bahwa konfigurasi model terbaik adalah yang menggunakan kernel Linear dengan 500 fitur TF-IDF. Model ini berhasil mencapai performa optimal dengan nilai akurasi 88,90%, presisi 88,86%, dan recall 89,00%. Penelitian ini menyimpulkan bahwa algoritma SVM dengan kernel Linear adalah metode yang efektif dan andal untuk diterapkan sebagai solusi deteksi hoaks pada berita *online* di Indonesia.

Kata Kunci: Deteksi Hoaks, *Support Vector Machine* (SVM), *Machine Learning*, *Natural Language Processing* (NLP), TF-IDF, Klasifikasi Teks.

KATA PENGANTAR

Puji syukur kehadirat Allah SWT, atas limpahan rahmat, hidayah, dan karunia-Nya sehingga peneliti dapat menyelesaikan skripsi ini dengan baik. Shalawat serta salam senantiasa tercurah kepada junjungan kita, Nabi Muhammad SAW, beserta keluarga, sahabat, dan para pengikutnya hingga akhir zaman. Skripsi dengan judul "*PENERAPAN SUPPORT VECTOR MACHINE (SVM) UNTUK DETEKSI HOAKS PADA BERITA ONLINE*" ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana (S-1) pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Pamulang.

Peneliti menyadari bahwa penyusunan skripsi ini tidak akan dapat terselesaikan tanpa adanya dukungan dari berbagai pihak, baik secara moril maupun materil. Oleh karena itu, peneliti mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan bantuan dan dukungan selama proses penyusunan skripsi ini, terutama kepada:

1. Bapak Dr. Pranoto, S.E., M.M., selaku Ketua Yayasan Sasmita Jaya.
2. Bapak Dr. E. Nurzaman AM, M.M., M.Si., selaku Rektor Universitas Pamulang.
3. Bapak Yan Mitha Djaksana, S.Kom., M.Kom., selaku Dekan Fakultas Ilmu Komputer.
4. Bapak Dr. Eng. Ahmad Musyafa, S.Kom., M.Kom., selaku Kaprodi Teknik Informatika Universitas Pamulang.
5. Ibu Raditia Vindua, S.Si., M.Kom., selaku Dosen Pembimbing, yang telah dengan sabar meluangkan waktu, memberikan bimbingan, arahan, dan motivasi kepada peneliti selama proses penyusunan skripsi ini.
6. Bapak/Ibu Dosen Penguji yang telah memberikan saran, kritik, dan masukan yang membangun demi kesempurnaan skripsi ini.
7. Seluruh Dosen dan Staf Akademik Program Studi Teknik Informatika Universitas Pamulang atas ilmu dan pelayanan yang telah diberikan.

8. Kedua orang tua tercinta serta seluruh keluarga yang telah memberikan dukungan moril, materil, dan doa yang tiada henti.
9. Sahabat dan rekan-rekan seperjuangan yang telah saling membantu dan memberikan semangat selama masa perkuliahan.
10. Semua pihak yang tidak dapat peneliti sebutkan satu per satu, terima kasih atas segala bantuan dan dukungannya.
11. Terimakasih kepada rumahsakit, perunggu, efek rumah kaca, majelis lidah berduri, nurbait, gledeg dan orkes pensil alis yang sudah menjadi *playlist* dalam menemani peneliti menyusun skripsi ini.
12. Kepada Belalang Tempur saya, yang kehadirannya selalu menjadi penyemangat. Terima kasih telah berjalan beriringan menemani saya hingga titik ini.

Peneliti menyadari bahwa skripsi ini masih jauh dari kata sempurna karena keterbatasan pengetahuan dan pengalaman. Oleh karena itu, peneliti mengharapkan segala bentuk saran dan kritik yang membangun. Semoga skripsi ini dapat bermanfaat bagi para pembaca, almamater, dan pengembangan ilmu pengetahuan.

Tangerang Selatan , 2025

Peneliti,

Rifky Fachuzi

DAFTAR ISI

LEMBAR PERNYATAAN	i
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
ABSTRACT	iv
ABSTRAK	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah.....	3
1.3 Rumusan Masalah.....	3
1.4 Batasan Penelitian.....	3
1.5 Tujuan Penelitian	4
1.6 Manfaat Penelitian	4
1.7 Metodologi Penelitian	5
1.8 Sistematika Penulisan.....	7
BAB II LANDASAN TEORI	9
2.1 Penelitian Terkait	9
2.2 Hoaks.....	11
2.2.1 Jenis-Jenis Hoaks	11
2.2.2 Karakteristik Hoaks.....	12

2.3	<i>Machine Learning</i>	13
2.4	<i>Natural Language Processing (NLP)</i>	15
2.5	<i>Term Frequency-Inverse Document Frequency (TF-IDF)</i>	16
2.6	<i>Support Vector Machine (SVM)</i>	17
2.6.1	Tahapan Kerja SVM	18
2.7	Pengujian Model	20
2.7.1	Pembagian <i>Dataset</i> (<i>Data Splitting</i>)	20
2.8	Metrik Evaluasi Model.....	20
	BAB III METODOLOGI PENELITIAN	24
3.1	Analisa Kebutuhan	24
3.2	Kerangka Kerja Penelitian	24
3.3	Pengumpulan Data	26
3.4	Alat dan Bahan Penelitian Metode Pengumpulan Data	27
3.5	Tahapan Penelitian	27
3.5.1	Pra-pemrosesan Data (<i>Data Preprocessing</i>).....	27
3.5.2	Ekstraksi Fitur dengan TF-IDF	28
3.5.3	Pembagian <i>Dataset</i>	28
3.5.4	Pemodelan dengan SVM.....	29
3.5.5	Pengujian dan Evaluasi Model.....	29
	BAB IV HASIL DAN PEMBAHASAN.....	31
4.1	Hasil	31
4.2	Pengolahan <i>Dataset</i>	31
4.2.1	Sumber dan Struktur <i>Dataset</i>	32
4.2.2	Proses Pemuatan dan Penggabungan <i>Dataset</i>	33
4.2.3	Penyesuaian dan Pembersihan Data.....	34

4.3	Pra-Pemrosesan Teks	37
4.3.1	<i>Case Folding</i>	38
4.3.2	Pembersihan Karakter	38
4.3.3	Tokenisasi	39
4.3.4	Penghapusan <i>Stopword</i>	39
4.3.5	Penggabungan teks.....	40
4.4	Penyeimbangan Data.....	41
4.4.1	Analisis Distribusi Label Sebelum Penyeimbangan	41
4.4.2	Penyeimbangan Data.....	42
4.4.3	Penambahan <i>Noise</i> Label	42
4.4.4	Hasil Penyeimbangan dan Penambahan Noise	43
4.4.5	Dampak Terhadap Performa Model.....	44
4.5	Pelatihan dan Pengujian Model <i>Support Vector Machine</i> (SVM)	45
4.5.1	Proses Konversi Data Teks ke Bentuk Numerik.....	46
4.5.2	Pembagian Data Pelatihan dan Pengujian.....	46
4.5.3	Konfigurasi Model dan Parameter Eksperimen	47
4.5.4	Proses Pelatihan Model.....	48
4.5.5	Proses Pengujian Model.....	48
4.6	Hasil Evaluasi Model	49
4.6.1	Metrik Evaluasi Model.....	49
4.6.2	Hasil Evaluasi Model SVM	50
4.6.3	Analisis Hasil Evaluasi	50
4.6.4	Analisis Confusion Matrix	51
4.6.5	Perbandingan Performa.....	53
	BAB V PENUTUP	55

5.1	Kesimpulan	55
5.2	Saran.....	55
DAFTAR PUSTAKA.....		57

DAFTAR GAMBAR

Gambar 2.1 Jenis - jenis Hoaks	12
Gambar 2.2 Diagram alir proses NLP <i>preprocessing</i>	15
Gambar 2.3 <i>Hyperlane SVM</i> dengan <i>margin</i> dan <i>support vectors</i>	17
Gambar 2. 4 Rumus akurasi	21
Gambar 2.5 Rumus <i>Precision</i>	22
Gambar 2.6 Rumus <i>Recall</i>	23
Gambar 3.1 <i>Alur Penelitian Deteksi</i>	25
Gambar 3.2 <i>Confusion Matrix</i>	29
Gambar 4.1 Fungsi menggabungkan semua data <i>frame</i>	34
Gambar 4.2 Fungsi Penghapusan nilai kosong.....	35
Gambar 4.3 Fungsi penghapusan duplikat	36
Gambar 4.4 Fungsi Validasi Nilai Label	36
Gambar 4.5 Code untuk Penyeimbangan Data.....	42
Gambar 4.6 Penambahan <i>Noise Label</i>	43
Gambar 4.7 Distribusi Label Sebelum dan Sesudah Penyeimbangan.....	45
Gambar 4.8 Kode yang digunakan	47
Gambar 4.9 <i>Code Pelatihan model</i>	48
Gambar 4.10 Fungsi prediksi label.....	48
Gambar 4.11 <i>Confusion matrix linear</i> 1000.....	52
Gambar 4.12 Confusion Matrix rbf 1000	53
Gambar 4.13 Grafik perbandingan akurasi dan presisi	53
Gambar 4.14 Grafik perbandingan <i>recall</i> dan <i>f1 score</i>	54

DAFTAR TABEL

Tabel 2.1 Karakteristik Hoaks.....	12
Tabel 2.2 <i>Confusion Matrix</i>	21
Tabel 3.1 Alat dan Bahan Penelitian	27
Tabel 4.1 Sumber <i>Dataset</i>	32
Tabel 4.2 Penyesuaian nama kolom	35
Tabel 4.3 Tabel hasil akhir pembersihan data	37
Tabel 4.4 Tabel hasil <i>Case Folding</i>	38
Tabel 4.5 Tabel hasil pembersihan karakter.....	38
Tabel 4.6 Tabel hasil tokenisasi	39
Tabel 4.7 Hasil <i>stopword removal</i>	39
Tabel 4.8 Hasil penggabungan kata.....	40
Tabel 4.9 Tabel Distribusi Label Sebelum Penyeimbangan	41
Tabel 4.10 Hasil Penyeimbangan dan Penambahan <i>Noise</i>	43
Tabel 4.11 Distribusi Label Sebelum dan Sesudah Penyeimbangan	44
Tabel 4.12 Konfigurasi Model SVM.....	47
Tabel 4.13 Hasil Evaluasi Model SVM <i>Kernel Linear</i> dan RBF.....	50

BAB I

PENDAHULUAN

1.1 Latar Belakang

Evolusi era digital telah memicu transformasi substansial dalam paradigma masyarakat terkait akuisisi dan diseminasi informasi. Kenaikan prevalensi pengguna internet di Indonesia, mayoritas mengandalkan platform media sosial dan publikasi berita daring, telah menginduksi terbentuknya suatu ekosistem informasi yang bercirikan dinamisme, kecepatan proliferasi, dan kenihilan batas spasial serta temporal. Indonesia mengukuhkan posisinya sebagai salah satu negara dengan basis pengguna internet terbesar secara global, yang secara implisit menjadikan ranah digital sebagai kanal primordial untuk aktivitas komunikatif dan penyebaran pengetahuan bagi jutaan individu. Akan tetapi, kemudahan interoperabilitas ini juga menghadirkan problematik krusial berupa eskalasi probabilitas insiden disinformasi atau hoaks di dalam tatanan sosial.

Disinformasi, yang didefinisikan sebagai narasi yang direkayasa secara sengaja untuk pencapaian tujuan spesifik, telah berkembang menjadi hambatan signifikan terhadap kesatuan sosial. Bentuk konten ini diciptakan dengan maksud menipu, memicu reaksi emosional, dan mendistorsi pemahaman kolektif, sehingga berujung pada konsekuensi buruk yang meluas jangkaunya.

Akselerasi distribusinya melalui ranah digital berpotensi memicu hysteria massa, kerugian ekonomi substansial, fragmentasi politik, bahkan membahayakan kesejahteraan publik, sebuah fenomena yang terekam jelas selama episode pandemi COVID-19 (Faturohmah & Salim, 2022). Konsekuensi merusak tambahan mencakup erosi kredibilitas publik terhadap lembaga-lembaga formal dan saluran berita, serta probabilitas disintegritas nasional yang dipicu oleh antagonisme etnis, religius, rasial, dan antargolongan (SARA) yang kerap diasup oleh materi disinformasi (Febriansyah & Muksin, 2020).

Upaya konvensional dalam menangani penyebaran hoaks, seperti verifikasi manual yang dilakukan oleh tim pemeriksa fakta, masih menghadapi keterbatasan mendasar. Jumlah berita yang dihasilkan dan disebarluaskan setiap detiknya jauh

melebihi kemampuan manusia untuk melakukan verifikasi secara cepat dan akurat. Keterlambatan dalam proses klarifikasi sering kali menyebabkan informasi palsu terlanjur dipercaya oleh masyarakat, sehingga langkah penanggulangannya menjadi kurang efektif. Oleh karena itu, dibutuhkan suatu solusi otomatis yang mampu menganalisis serta mengidentifikasi konten hoaks secara efisien dalam skala besar dan waktu yang singkat.

Perkembangan teknologi kecerdasan buatan (*Artificial Intelligence*), khususnya dalam bidang *Machine Learning*, memberikan peluang yang menjanjikan dalam upaya mengatasi permasalahan penyebaran hoaks. Metode klasifikasi teks berbasis *Machine Learning* memungkinkan sistem komputer untuk mempelajari dan mengenali pola-pola tertentu yang membedakan antara berita hoaks dan berita fakta berdasarkan data dalam jumlah besar. Salah satu algoritma yang terbukti efektif dalam melakukan klasifikasi teks adalah *Support Vector Machine* (SVM). SVM merupakan algoritma *supervised learning* yang berfungsi dengan mencari *hyperplane* atau bidang pemisah optimal untuk memisahkan data ke dalam kategori yang berbeda secara akurat.

Sejumlah penelitian terkini menunjukkan efektivitas SVM dalam konteks deteksi hoaks berbahasa Indonesia. (Ropikoh, Abdulhakim, Enri, & Sulistiyowati, 2021) berhasil menerapkan algoritma SVM untuk mengklasifikasikan berita hoaks seputar COVID-19 dengan tingkat akurasi tinggi mencapai 97,06%. Penelitian lain oleh (DickiPrabowo, Widaningrum, & Karaman, 2025) juga menerapkan SVM dalam membangun sistem deteksi berita hoaks terkait Pemilu 2024, yang menunjukkan efektivitasnya dalam domain politik. Selain itu, penelitian oleh (Indra, Agus Umar Hamdani, Suci Setiawati, Zena Dwi Mentari, & Mauridhy Hery Purnomo, 2024) yang membandingkan SVM dengan algoritma lain seperti K-NN dan *Random Forest* juga memperkuat temuan bahwa SVM memiliki kinerja yang kompetitif dan solid untuk klasifikasi hoaks....

Berdasarkan urgensi penyebaran hoaks di Indonesia dan potensi metode *Support Vector Machine* (SVM) dalam penelitian terdahulu, maka penelitian ini berfokus pada “Penerapan *Support Vector Machine* (SVM) untuk Deteksi Hoaks pada Berita *Online*”. Tujuannya adalah membangun model klasifikasi yang efektif serta mengukur kinerjanya secara kuantitatif. Hasil penelitian ini diharapkan dapat

berkontribusi dalam upaya mitigasi penyebaran disinformasi di ruang digital Indonesia.

1.2 Identifikasi Masalah

Berdasarkan uraian latar belakang, dapat diidentifikasi beberapa permasalahan berikut:

1. Berita hoaks menimbulkan dampak negatif terhadap stabilitas sosial, politik, dan kepercayaan publik.
2. Algoritma *Support Vector Machine* (SVM) berpotensi menjadi metode klasifikasi teks yang akurat untuk deteksi hoaks, namun kinerjanya perlu diuji secara spesifik pada *Dataset* berita *online* berbahasa Indonesia.

1.3 Rumusan Masalah

Berdasarkan identifikasi masalah di atas, maka rumusan masalah dalam penelitian ini adalah:

1. Bagaimana menerapkan algoritma *Support Vector Machine* (SVM) untuk membangun model klasifikasi hoaks pada berita *online* berbahasa Indonesia?
2. Bagaimana performa model klasifikasi *Support Vector Machine* (SVM) dalam mendeteksi berita hoaks yang diukur menggunakan metrik accuracy, precision, dan recall?

1.4 Batasan Penelitian

Agar penelitian ini lebih fokus dan terarah, maka batasan masalah ditetapkan sebagai berikut:

1. Data: *Dataset* berupa berita *online* berbahasa Indonesia yang dikumpulkan dari portal pemeriksa fakta seperti Turnbackhoax.id (label hoaks) dan portal berita nasional terpercaya seperti Kompas.com dan Detik.com (label fakta).
2. Algoritma: Penelitian hanya berfokus pada penerapan algoritma klasifikasi *Support Vector Machine* (SVM).

3. Ekstraksi Fitur: Metode ekstraksi fitur yang digunakan adalah *Term Frequency–Inverse Document Frequency* (TF-IDF).
4. Luaran: Hasil penelitian berupa model klasifikasi yang telah teruji kinerjanya, bukan produk aplikasi siap pakai.

1.5 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Menerapkan algoritma *Support Vector Machine* (SVM) untuk membangun model klasifikasi yang mampu membedakan berita *online* berbahasa Indonesia ke dalam kategori hoaks atau fakta.
2. Menganalisis serta mengukur kinerja model klasifikasi SVM dengan menggunakan metrik evaluasi berupa *accuracy*, *precision*, dan *recall* untuk mengetahui tingkat efektivitas dan keandalannya dalam mendeteksi berita hoaks.

1.6 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat dari berbagai aspek sebagai berikut:

1. Manfaat Akademis:
 - a. Memberikan kontribusi ilmiah dalam pengembangan bidang *Natural Language Processing* (NLP) dan *Machine Learning*, khususnya dalam studi kasus deteksi hoaks berbahasa Indonesia.
 - b. Menjadi referensi serta bahan perbandingan bagi penelitian selanjutnya yang berkaitan dengan klasifikasi teks dan deteksi hoaks.
2. Manfaat Praktis:
 - a. Menghasilkan model dasar yang dapat dikembangkan lebih lanjut menjadi sistem atau aplikasi pendekripsi hoaks secara otomatis.
 - b. Mendukung peningkatan literasi digital masyarakat melalui penerapan teknologi yang mampu menyaring dan memverifikasi informasi secara efisien.

3. Manfaat bagi Peneliti:
 - a. Memenuhi salah satu persyaratan untuk memperoleh gelar Sarjana Komputer.
 - b. Meningkatkan pemahaman dan kemampuan peneliti dalam menerapkan algoritma *Machine Learning* untuk menyelesaikan permasalahan nyata di bidang teknologi informasi.

1.7 Metodologi Penelitian

Penelitian ini menggunakan pendekatan..kuantitatif eksperimental yang bertujuan untuk membangun serta menguji model klasifikasi berita *online* berbahasa Indonesia menggunakan algoritma *Support Vector Machine* (SVM). Tahapan penelitian dilakukan secara sistematis melalui langkah-langkah berikut:

1. Studi Literatur: Melakukan kajian terhadap teori, konsep, dan penelitian terdahulu yang berkaitan dengan deteksi hoaks, *Natural Language Processing* (NLP), serta algoritma SVM.
2. Pengumpulan Data: Menghimpun data berupa teks berita dari sumber daring yang telah ditentukan, kemudian memberikan label kategori *hoaks* atau *fakta* sesuai dengan kriteria yang berlaku.
3. Pra-pemrosesan Data (*Preprocessing*): Melakukan pembersihan dan normalisasi teks melalui tahapan *case folding*, *tokenizing*, *stopword removal*, dan *stemming* untuk memperoleh representasi teks yang siap diolah.
4. Ekstraksi Fitur: Mengonversi teks menjadi bentuk vektor numerik menggunakan metode *Term Frequency–Inverse Document Frequency* (TF-IDF) agar dapat diproses oleh model *Machine Learning*.
5. Pemodelan: Membagi *Dataset* menjadi data latih (*training data*) dan data uji (*testing data*), kemudian melatih model klasifikasi menggunakan algoritma *Support Vector Machine* (SVM).
6. Evaluasi Model: Mengukur kinerja model dengan menggunakan data uji serta menghitung nilai *accuracy*, *precision*, dan *recall* berdasarkan hasil *confusion matrix*.

7. Analisis dan Penarikan Kesimpulan: Menganalisis hasil evaluasi untuk menjawab rumusan masalah penelitian serta menilai tingkat efektivitas algoritma SVM dalam mendeteksi berita hoaks..

1.8 Sistematika Penulisan

Untuk mempermudah pemahaman pembaca, sistematika penulisan ini menjelaskan secara singkat isi dari setiap bab dalam penelitian ini.

BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang masalah yang mendasari penelitian, identifikasi masalah, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, metodologi penelitian secara ringkas, serta sistematika penulisan laporan skripsi.

BAB II LANDASAN TEORI

Bab ini membahas teori-teori dasar yang relevan dan menjadi acuan dalam penelitian, mencakup konsep berita hoaks, *Natural Language Processing* (NLP), metode ekstraksi fitur *Term Frequency-Inverse Document Frequency* (TF-IDF), algoritma *Support Vector Machine* (SVM), serta metrik evaluasi model.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan secara rinci langkah-langkah teknis yang dilakukan dalam penelitian, mulai dari waktu dan tempat penelitian, alat dan bahan yang digunakan, alur penelitian, teknik pengumpulan data, tahapan pra-pemrosesan data, implementasi model, hingga teknik pengujian dan evaluasi model.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil dari implementasi dan pengujian model yang telah dibangun. Pembahasan akan mencakup karakteristik *Dataset* yang digunakan, hasil pra-pemrosesan, hasil pelatihan model, serta hasil pengujian performa model yang diukur dengan *accuracy*, *precision*, dan *recall*.

BAB V**KESIMPULAN DAN SARAN**

Bab ini merupakan bagian penutup yang berisi kesimpulan dari seluruh rangkaian penelitian berdasarkan hasil dan pembahasan yang telah diuraikan. Selain itu, bab ini juga akan memberikan saran untuk pengembangan penelitian selanjutnya agar dapat menghasilkan model yang lebih baik.

BAB II

LANDASAN TEORI

2.1 Penelitian Terkait

Penelitian ini mengacu pada sejumlah studi terdahulu yang memiliki relevansi dengan topik deteksi berita hoaks menggunakan algoritma *Machine Learning*. Beberapa penelitian yang menjadi landasan teori penelitian ini antara lain sebagai berikut:

1. (Ropikoh et al., 2021) dalam penelitiannya berjudul “*Penerapan Algoritma Support Vector Machine (SVM) untuk Klasifikasi Berita Hoaks COVID-19*” meneliti penyebaran berita hoaks terkait pandemi COVID-19 yang meningkat drastis di Indonesia. Penelitian ini menggunakan algoritma SVM dengan *kernel linear* dan *RBF* untuk mengklasifikasikan berita hoaks dan bukan hoaks, dengan *Dataset* sebanyak 8.172 data dari Kompas (non-hoaks), Turnbackhoax, dan Hoaxbuster (hoaks). Metode yang digunakan mengacu pada pendekatan *Knowledge Discovery in Database (KDD)* melalui tahapan *text mining* (*case folding, tokenizing, filtering, stemming*), transformasi *TF-IDF*, dan klasifikasi SVM dalam empat skenario pembagian data latih dan uji. Hasil penelitian menunjukkan bahwa *kernel linear* pada skenario 80:20 menghasilkan akurasi tertinggi sebesar 92,90%, sedangkan *kernel RBF* pada skenario 90:10 mencapai 90,46%.
2. (Rahmawati, 2021) dalam penelitian berjudul “*Deteksi Berita Hoaks pada Website Turnbackhoax dengan Menggunakan Machine Learning*” berfokus pada klasifikasi berita hoaks dari situs Turnbackhoax. Penelitian ini membandingkan tiga algoritma *Machine Learning*, yaitu SVM, *Random Forest*, dan *Logistic Regression*, dengan tahapan *preprocessing*, pembobotan kata menggunakan *TF-IDF*, penyeimbangan data menggunakan *SMOTE*, serta optimasi

parameter dengan *Grid Search CV*. Hasil evaluasi menunjukkan bahwa algoritma SVM memiliki performa terbaik dengan akurasi 83,3% dan *recall* 99% pada kelas hoaks, melampaui dua algoritma lainnya.

3. (Febriyanty Nur Elyta, 2023) dalam penelitian berjudul “*Deteksi Berita Hoaks dari Media Online Indonesia Menggunakan Algoritma Naïve Bayes dan Support Vector Machine*” menguji dua algoritma, yakni Naïve Bayes dan SVM, menggunakan *Dataset* sebanyak 2.000 judul berita berbahasa Indonesia. Proses penelitian meliputi *text preprocessing* (*case folding*, *stemming*, *stopword removal*, *tokenizing*), pembobotan *TF-IDF*, dan evaluasi menggunakan *Confusion Matrix*. Hasil penelitian menunjukkan bahwa Naïve Bayes memperoleh akurasi 93%, sedangkan SVM mencapai akurasi 100%.
4. (Putri & Athoillah, 2021) melalui penelitian “*Support Vector Machine untuk Identifikasi Berita Hoaks Terkait Virus Corona (COVID-19)*” membangun sistem otomatis berbasis SVM dengan *kernel linear* dan validasi *k-fold cross validation*. Hasil penelitian menunjukkan nilai rata-rata *precision* 78,96%, *recall* 78,18%, dan *F-measure* 78,02%, dengan hasil terbaik pada percobaan ketiga yaitu *precision* 89,37% dan *recall* 88,64%.
5. (Rizky Purwanto Fernandes & Rizky Tahara Shita, 2024) dalam penelitian “*Penerapan Metode SVM dan Random Forest untuk Mendeteksi Berita Hoaks pada PT Global Arrow*” meneliti penyebaran hoaks di ranah politik dan kesehatan. Penelitian ini menggabungkan metode SVM dan Random Forest dengan pembobotan *TF-IDF* serta data hasil *web crawling* dari Kompas, Kominfo, dan Antara News. Hasil awal menunjukkan akurasi di atas 90%, namun setelah pengujian lanjutan, akurasi menurun menjadi

55,14%, sehingga disimpulkan bahwa diperlukan proses optimasi lebih lanjut agar sistem bekerja lebih andal.

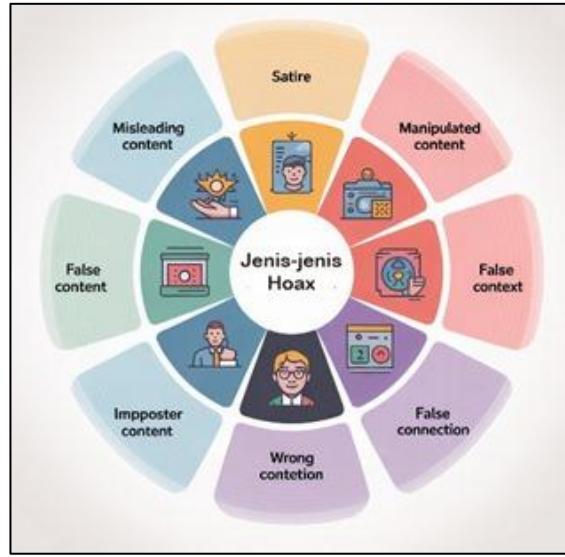
2.2 Hoaks

Hoaks atau berita bohong merupakan fenomena kompleks yang berkembang seiring dengan meningkatnya dampak sosial yang ditimbulkannya. Secara umum, hoaks didefinisikan sebagai informasi yang sengaja direkayasa untuk menutupi fakta sebenarnya, dengan tujuan menimbulkan kebingungan atau memengaruhi opini publik (Juditha, 2020). Hoaks sering kali menimbulkan efek psikologis seperti ketakutan, kecemasan, serta persepsi keliru di masyarakat karena mengandung unsur provokatif maupun SARA (Nurhaipah & Ramallah, 2024).

Menurut (Faturohmah & Salim, 2022), kecepatan penyebaran informasi di media digital menyebabkan hoaks dapat menjangkau audiens luas sebelum klarifikasi resmi diterbitkan. Siklus hidup hoaks umumnya dimulai dari pembuatan konten, amplifikasi oleh akun anonim atau *buzzer*, penyebaran organik oleh pengguna, hingga akhirnya menjadi isu publik yang meluas.

2.2.1 Jenis-Jenis Hoaks

Disinformasi tidak selalu berbentuk berita yang 100% palsu. Menurut kerangka kerja yang diakui secara global oleh para pemeriksa fakta, hoaks dapat dikategorikan ke dalam beberapa jenis berdasarkan tingkat manipulasi dan niat pembuatnya:



Gambar 2.1 Jenis - jenis Hoaks

1. **Satire** atau Parodi: Konten humor atau sindiran yang berpotensi disalahartikan sebagai fakta.
2. Koneksi yang Salah (*False Connection*): Ketidaksesuaian antara judul, gambar, dan isi konten.
3. Konten yang Menyesatkan (*Misleading Content*): Informasi benar disajikan dengan konteks yang menyesatkan.
4. Konteks yang Salah (*False Context*): Konten lama disajikan ulang dengan konteks baru yang keliru.
5. Konten Tiruan (*Imposter Content*): Meniru identitas sumber resmi atau tokoh publik.
6. Konten yang Dimanipulasi (*Manipulated Content*): Konten yang telah diubah secara digital untuk menipu.
7. Konten Palsu (*Fabricated Content*): Informasi yang sepenuhnya dibuat tanpa dasar fakta.

2.2.2 Karakteristik Hoaks

Meskipun jenisnya beragam, berita hoaks secara umum memiliki beberapa karakteristik yang dapat dikenali, antara lain:

Tabel 2.1 Karakteristik Hoaks

No	Karakteristik Hoaks	Contoh Kasus
1	Sensasional	Vaksin Covid-19 mengandung chip untuk mengontrol manusia
2	Menampilkan data atau gambar editan	Presiden Prabowo promosi produk seprai
3	Mendesak untuk dibagikan segera	Tolong share, ini penting untuk keselamatan keluarga kita semua!

1. Sensasional

Hoaks biasanya dibuat dengan judul atau narasi yang sangat provokatif, berlebihan, atau membangkitkan emosi ekstrem seperti marah, takut, atau penasaran.

Teknik ini digunakan agar pembaca langsung bereaksi tanpa berpikir kritis dan cepat membagikan informasi tersebut. Pembuat hoaks tahu bahwa emosi yang kuat dapat mempercepat penyebaran berita palsu.

2. Menampilkan Data atau Gambar Editan

Hoaks sering menggunakan gambar, video, atau data yang telah dimanipulasi untuk memberi kesan seolah berita tersebut nyata. Manipulasi bisa berupa penyuntingan gambar, penggabungan konten, atau penggunaan foto lama yang tidak relevan dengan peristiwa yang diklaim. Tujuannya untuk memperkuat narasi palsu agar tampak kredibel.

2.3 *Machine Learning*

Menurut (Retnoningsih & Pramudita, 2020) menyatakan bahwa “*Machine learning* merupakan cabang ilmu bagian dari kecerdasan buatan (*artificial intelligence*), dengan pemrograman untuk memungkinkan komputer menjadi cerdas berperilaku seperti manusia, dan dapat meningkatkan pemahamannya melalui pengalaman secara otomatis”.

Tujuan utamanya adalah membangun model matematis yang dapat mengenali pola dalam data dan kemudian menggunakan pola tersebut untuk membuat prediksi atau keputusan pada data baru.

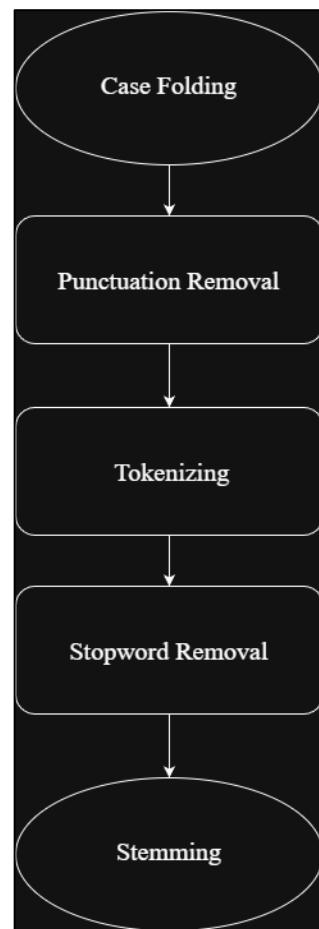
Machine learning dapat dikategorikan menjadi tiga jenis utama:

1. *Supervised Learning* (Pembelajaran Terarah): Ini adalah pendekatan yang digunakan dalam penelitian ini. Dalam *supervised learning*, model "diajari" menggunakan *Dataset* yang sudah memiliki label yang benar. Misalnya, untuk mendeteksi hoaks, model diberi ribuan contoh berita yang sudah diberi label "Hoaks" atau "Fakta". Model kemudian belajar untuk memetakan antara fitur teks berita (*input*) dan labelnya (*output*). SVM adalah salah satu algoritma *supervised learning*.
2. *Unsupervised Learning* (Pembelajaran Tak Terarah): Dalam pendekatan ini, model bekerja dengan data yang tidak memiliki label. Tujuannya adalah untuk menemukan struktur atau pola tersembunyi dalam data, seperti mengelompokkan berita dengan topik serupa (*clustering*).
3. *Reinforcement Learning* (Pembelajaran Penguanan): Model belajar melalui proses *trial and error*. Model akan menerima "penghargaan" (*reward*) untuk tindakan yang benar dan "hukuman" (*penalty*) untuk tindakan yang salah, dengan tujuan memaksimalkan total penghargaan yang diterima. Pendekatan ini umum digunakan dalam robotika atau permainan.

2.4 Natural Language Processing (NLP)

Natural Language Processing (NLP) adalah cabang dari ilmu komputer dan kecerdasan buatan (*Artificial Intelligence*) yang berfokus pada interaksi antara komputer dengan bahasa manusia. Tujuan utama NLP adalah memungkinkan komputer untuk memahami, memproses, dan menghasilkan bahasa alami (lisan maupun tulisan) dengan cara yang sama seperti manusia (Kusumawardani & Cahyanto, 2023). Dalam konteks deteksi hoaks, NLP digunakan untuk memproses dan menganalisis konten teks dari sebuah berita.

Sebelum teks dapat dianalisis oleh model *machine learning*, teks tersebut harus melewati serangkaian tahapan yang disebut pra-pemrosesan data (*preprocessing*). Tahapan ini bertujuan untuk membersihkan teks dari *noise* (elemen yang tidak relevan) dan mengubahnya menjadi format yang terstruktur.



Gambar 2.2 Diagram alir proses NLP *preprocessing*

Tahapan umum dalam pra-pemrosesan meliputi:

1. *Case Folding*: Mengubah seluruh teks menjadi huruf kecil untuk menyeragamkan kata.
2. *Punctuation Removal*: Menghapus semua tanda baca seperti titik, koma, tanda tanya, dll.
3. *Tokenizing*: Memecah kalimat menjadi unit-unit kata yang lebih kecil (token).
4. *Stopword Removal*: Menghapus kata-kata umum yang sering muncul namun tidak memiliki makna signifikan (misalnya "yang", "di", "dan", "dari").
5. *Stemming*: Mengubah kata-kata berimbahan menjadi bentuk kata dasarnya (misalnya "menyebarkan" menjadi "sebar").

2.5 *Term Frequency-Inverse Document Frequency (TF-IDF)*

Setelah teks dibersihkan, langkah selanjutnya adalah mengubah teks menjadi representasi numerik agar dapat diproses oleh algoritma matematika. Proses ini disebut ekstraksi fitur. Salah satu metode yang paling populer dan efektif untuk ekstraksi fitur pada data teks adalah *Term Frequency-Inverse Document Frequency* (TF-IDF).

TF-IDF bekerja dengan mengukur tingkat kepentingan sebuah kata dalam sebuah dokumen terhadap keseluruhan kumpulan dokumen (korpus). Bobot TF-IDF dihitung berdasarkan dua komponen (DickiPrabowo et al., 2025):

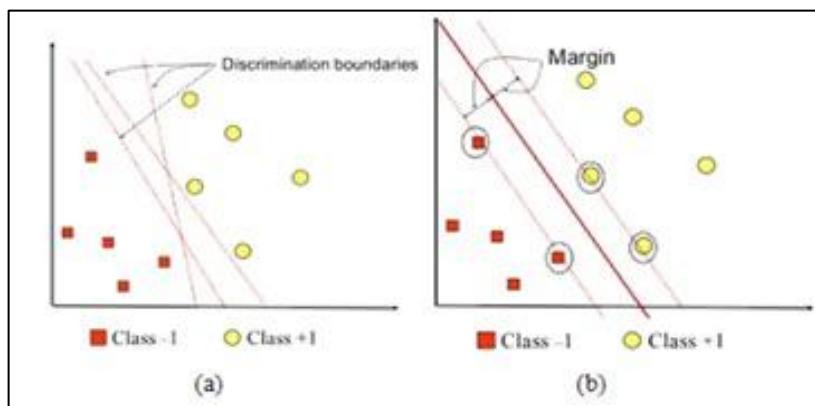
1. *Term Frequency* (TF): Mengukur frekuensi kemunculan sebuah kata dalam satu dokumen. Semakin sering sebuah kata muncul, semakin tinggi nilai TF-nya.
2. *Inverse Document Frequency* (IDF): Mengukur seberapa unik atau langka sebuah kata di seluruh dokumen dalam korpus. Semakin sedikit dokumen yang mengandung kata tersebut, semakin tinggi nilai IDF-nya.

Nilai TF-IDF untuk sebuah kata dihitung dengan mengalikan nilai TF dan IDF-nya. Dengan demikian, kata yang sering muncul dalam satu dokumen tetapi jarang muncul di dokumen lain akan mendapatkan bobot TF-IDF yang tinggi,

menandakan bahwa kata tersebut adalah kata kunci yang penting untuk dokumen tersebut.

2.6 Support Vector Machine (SVM)

Support Vector Machine (SVM) adalah algoritma *supervised machine learning* yang sangat kuat dan efektif untuk tugas klasifikasi dan regresi. Dalam konteks klasifikasi, tujuan utama SVM adalah menemukan garis pemisah atau *hyperplane* terbaik yang dapat memisahkan data ke dalam kelas-kelas yang berbeda secara optimal (Ropikoh et al., 2021).



Gambar 2.3 *Hyperlane* SVM dengan *margin* dan *support vectors*

Konsep dasar mengenai cara kerja *Support Vector Machine* (SVM) dapat diuraikan secara lebih rinci sebagai berikut:

1. *Hyperplane*: Inti dari algoritma SVM adalah identifikasi sebuah *hyperplane* optimal. Dalam konteks visualisasi dua dimensi, *hyperplane* ini merepresentasikan sebuah garis pemisah. Tujuan utama SVM adalah menemukan *hyperplane* yang paling efektif dalam memisahkan titik-titik data dari kelas-kelas yang berbeda. Efektivitas ini diukur berdasarkan jarak atau *margin* terbesar yang berhasil diciptakan antara titik data terluar dari masing-masing kelas yang berdekatan dengan *hyperplane*.
2. *Support Vectors*: Elemen krusial lainnya dalam SVM adalah *support vectors*. Ini merupakan titik-titik data spesifik yang memiliki kedekatan paling signifikan dengan *hyperplane* yang telah ditentukan. Posisinya

yang paling dekat inilah yang secara fundamental berperan dalam mendefinisikan dan menstabilkan posisi dari *hyperplane* itu sendiri. Tanpa titik-titik *support vectors* ini, *hyperplane* tidak akan memiliki batasan yang jelas.

3. *Margin*: Istilah *margin* merujuk pada zona pemisah yang terbentuk antara *hyperplane* dan titik-titik data terdekat dari setiap kelas, yaitu *support vectors*. Algoritma SVM secara inheren dirancang untuk berusaha memaksimumkan luasan *margin* ini. Secara teoritis, *hyperplane* yang mampu mempertahankan *margin* terluas umumnya diasosiasikan dengan potensi kemampuan generalisasi yang lebih baik terhadap data baru, yang berarti tingkat kesalahan prediksi pada data yang belum pernah dilihat sebelumnya akan lebih rendah.

Keunggulan SVM adalah kemampuannya untuk menangani data berdimensi tinggi secara efektif, yang sangat umum terjadi pada data teks setelah proses TF-IDF. Selain itu, SVM juga terbukti memiliki performa yang sangat baik bahkan dengan jumlah data latih yang terbatas.

2.6.1 Tahapan Kerja SVM

Meskipun konsepnya matematis, alur kerja SVM dalam mengklasifikasikan data dapat diuraikan menjadi beberapa tahapan logis:

1. Pemetaan Fitur ke Ruang Dimensi Tinggi: Data teks yang sudah diubah menjadi vektor numerik oleh TF-IDF dipetakan sebagai titik-titik dalam sebuah ruang berdimensi tinggi (*n-dimensional space*), di mana setiap dimensi merepresentasikan satu kata unik dari korpus.
2. Pencarian *Hyperplane* Pemisah: Algoritma SVM kemudian bekerja untuk menemukan *hyperplane* optimal. *Hyperplane* ini harus dapat memisahkan titik-titik data dari kelas yang berbeda (misalnya, titik data "Hoaks" di satu sisi dan "Fakta" di sisi lain).
3. Optimalisasi *Margin*: Dari sekian banyak kemungkinan *hyperplane* yang bisa dibuat, SVM secara spesifik mencari satu

hyperplane yang memiliki *margin* (jarak) paling besar ke *support vector* (titik data terdekat) dari masing-masing kelas. Proses ini disebut *margin maximization*, yang merupakan inti dari kekuatan SVM karena menghasilkan model yang lebih tahan terhadap *overfitting*.

4. Prediksi Data Baru: Setelah *hyperplane* optimal ditemukan, model siap digunakan. Ketika ada data berita baru, data tersebut akan melalui proses pra-pemrosesan dan TF-IDF yang sama untuk diubah menjadi vektor. Vektor ini kemudian dipetakan ke dalam ruang dimensi yang sama. Posisi titik data baru ini terhadap *hyperplane* akan menentukan kelas prediksinya. Jika berada di satu sisi, ia akan diklasifikasikan sebagai "Hoaks", dan jika di sisi lain, sebagai "Fakta".

2.7 Pengujian Model

Setelah model *machine learning* selesai dibangun dan dilatih, tahap krusial selanjutnya adalah pengujian. Pengujian model bertujuan untuk mengevaluasi seberapa baik kinerja model dalam membuat prediksi pada data baru yang belum pernah dilihat sebelumnya. Proses ini penting untuk memastikan bahwa model tidak hanya "menghafal" data latih (*overfitting*), tetapi benar-benar mampu melakukan generalisasi pola yang telah dipelajarinya.

2.7.1 Pembagian Dataset (*Data Splitting*)

Metode pengujian yang paling umum digunakan adalah dengan membagi *Dataset* yang ada menjadi dua bagian terpisah:

1. Data Latih (*Training Set*): Bagian terbesar dari *Dataset* yang digunakan untuk "mengajari" atau melatih model. Model akan mempelajari pola, hubungan, dan fitur dari data ini untuk membangun *hyperplane* pemisah.
2. Data Uji (*Testing Set*): Bagian yang lebih kecil dari *Dataset* yang "disembunyikan" dari model selama proses pelatihan. Setelah model selesai dilatih, data uji ini digunakan untuk mengukur performa model. Karena model belum pernah melihat data ini, hasil pengujian dapat memberikan estimasi yang objektif tentang bagaimana model akan berkinerja di dunia nyata.

Pembagian ini biasanya dilakukan dengan rasio tertentu, misalnya 80% data untuk pelatihan dan 20% untuk pengujian, atau 70% untuk pelatihan dan 30% untuk pengujian. Hasil prediksi model pada data uji inilah yang kemudian dianalisis menggunakan *Confusion Matrix* dan berbagai metrik evaluasi.

2.8 Metrik Evaluasi Model

Untuk mengukur seberapa baik kinerja model klasifikasi yang telah dibangun, diperlukan metrik evaluasi. Metrik-metrik ini dihitung berdasarkan

hasil prediksi model pada data uji, yang biasanya disajikan dalam sebuah tabel bernama *Confusion Matrix*.

Confusion Matrix adalah tabel 2x2 yang merangkum hasil prediksi dengan membandingkannya dengan kelas aktual. Tabel ini terdiri dari empat komponen (Indra et al., 2024):

Tabel 2.2 *Confusion Matrix*

<i>Actual / Predicted</i>	<i>Positive (Pred)</i>	<i>Negative (Pred)</i>
<i>Positive (Actual)</i>	<i>True Positive (TP)</i>	<i>False Negative (FN)</i>
<i>Negative (Actual)</i>	<i>False Positive (FP)</i>	<i>True Negative (TN)</i>

1. *True Positive (TP)*: Berita hoaks yang berhasil diprediksi dengan benar sebagai hoaks.
2. *True Negative (TN)*: Berita fakta yang berhasil diprediksi dengan benar sebagai fakta.
3. *False Positive (FP)*: Berita fakta yang salah diprediksi sebagai hoaks (*Error Tipe I*).
4. *False Negative (FN)*: Berita hoaks yang salah diprediksi sebagai fakta (*Error Tipe II*).

Dari *Confusion Matrix* tersebut, dapat dihitung beberapa metrik performa:

- a. Accuracy: Mengukur rasio prediksi yang benar (TP + TN) terhadap total keseluruhan data. Metrik ini mengukur seberapa akurat model secara umum.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Gambar 2. 4 Rumus akurasi

- 1) TP (*True Positive*) → Jumlah data positif yang berhasil diklasifikasikan dengan benar sebagai positif.

- 2) TN (*True Negative*) → Jumlah data negatif yang diklasifikasikan dengan benar sebagai negatif.
 - 3) FP (*False Positive*) → Jumlah data negatif yang salah diklasifikasikan sebagai positif.
 - 4) FN (*False Negative*) → Jumlah data positif yang salah diklasifikasikan sebagai negatif.
- b. Precision: Mengukur rasio prediksi positif yang benar (TP) dari total prediksi positif yang dibuat (TP + FP). Dalam kasus ini, *precision* mengukur seberapa banyak berita yang diprediksi sebagai hoaks benar-benar merupakan hoaks.

$$\textit{Precision} = \frac{\textcolor{brown}{TP}}{\textcolor{brown}{TP} + \textcolor{brown}{FP}}$$

Gambar 2.5 Rumus *Precision*

- 1) TP (*True Positive*) → Jumlah data positif yang diklasifikasikan dengan benar sebagai positif.
 - 2) FP (*False Positive*) → Jumlah data negatif yang salah diklasifikasikan sebagai positif.
- c. *Recall (Sensitivity)*: Mengukur rasio prediksi positif yang benar (TP) dari total data yang seharusnya positif (TP + FN). *Recall* mengukur kemampuan model dalam menemukan kembali semua berita hoaks yang ada di dalam *Dataset*.

$$\textcolor{brown}{Recall} = \frac{\textcolor{brown}{TP}}{\textcolor{brown}{TP} + \textcolor{brown}{FN}}$$

Gambar 2.6 Rumus *Recall*

- 1) TP (*True Positive*) → jumlah data positif yang berhasil diprediksi benar sebagai positif.
- 2) FN (*False Negative*) → jumlah data positif yang salah diprediksi sebagai negatif.

BAB III

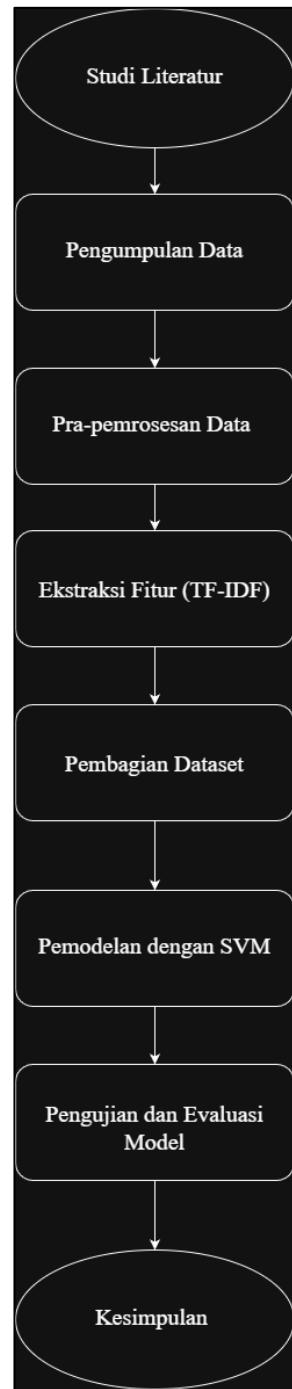
METODOLOGI PENELITIAN

3.1 Analisa Kebutuhan

Analisa kebutuhan dilakukan untuk memahami kebutuhan pengguna dan sistem yang berkaitan dengan penerapan algoritma *Support Vector Machine* untuk deteksi hoaks pada berita *online* dengan menggunakan bahasa pemrograman *Python*.

3.2 Kerangka Kerja Penelitian

Penelitian ini akan dilaksanakan dengan mengikuti kerangka kerja yang sistematis untuk memastikan alur penelitian berjalan dengan logis dan terstruktur. Kerangka kerja ini mengadopsi alur kerja standar dalam proyek *machine learning* untuk klasifikasi teks, yang telah terbukti efektif dalam berbagai penelitian sejenis. Secara visual, alur penelitian dapat digambarkan dalam diagram berikut:



Gambar 3.1 Alur Penelitian Deteksi

Adapun tahapan-tahapan utama dalam kerangka kerja penelitian ini adalah sebagai berikut:

- a. Studi Literatur: Mengkaji penelitian terdahulu dan landasan teori.
- b. Pengumpulan Data: Mengumpulkan *Dataset* berita hoaks dan berita fakta.

- c. Pra-pemrosesan Data (*Preprocessing*): Membersihkan dan menyiapkan data teks.
- d. Ekstraksi Fitur (TF-IDF): Mengubah data teks menjadi data numerik.
- e. Pembagian *Dataset*: Memisahkan data menjadi data latih dan data uji.
- f. Pemodelan dengan SVM: Melatih model menggunakan algoritma SVM pada data latih.
- g. Pengujian dan Evaluasi Model: Menguji model pada data uji dan mengukur kinerjanya.
- h. Penarikan Kesimpulan: Menganalisis hasil evaluasi dan menarik kesimpulan.

3.3 Pengumpulan Data

Tahap pengumpulan data merupakan fondasi awal dalam penelitian ini. Kualitas dan kuantitas data akan sangat memengaruhi kinerja model yang akan dibangun. *Dataset* yang akan digunakan terdiri dari dua kelas, yaitu "Hoaks" dan "Fakta".

- a. Data Berita Hoaks: Data berita hoaks akan dikumpulkan dari situs web pemeriksa fakta yang kredibel di Indonesia, yaitu turnbackhoax.id yang dikelola oleh Masyarakat Anti Fitnah Indonesia (MAFINDO). Situs ini menyediakan arsip berita yang telah diverifikasi sebagai hoaks, lengkap dengan narasi aslinya.
- b. Data Berita Fakta: Data berita fakta akan dikumpulkan dari beberapa portal berita *online* nasional yang terverifikasi oleh Dewan Pers dan memiliki reputasi yang baik, seperti Kompas.com, dan Detik.com.

Pengambilan data akan difokuskan pada artikel berita yang dipublikasikan dalam rentang waktu yang relevan untuk memastikan topiknya beragam. Total *Dataset* yang ditargetkan adalah sekitar 500 hingga 1000 artikel berita, dengan komposisi yang seimbang antara kelas hoaks dan fakta untuk menghindari bias pada model.

3.4 Alat dan Bahan Penelitian Metode Pengumpulan Data

Pada penelitian ini, spesifikasi perangkat keras dan lunak yang digunakan adalah:

Tabel 3.1 Alat dan Bahan Penelitian

Perangkat Keras	Spesifikasi
Processor	Intel Core i5-12450H
SSD	512 Gb
RAM	8 Gb
Perangkat Lunak	Spesifikasi
Sistem Operasi	Windows 11
Aplikasi Simulator	Google Colab
Bahasa Pemrograman	Pyhton 3.x
<i>Library Python</i>	<i>Scikit-learn</i>
	<i>Pandas</i>
	<i>Matplotlib</i>
	<i>NumPy</i>
	<i>NTLK</i>

3.5 Tahapan Penelitian

3.5.1 Pra-pemrosesan Data (*Data Preprocessing*)

Setelah *Dataset* terkumpul, data teks mentah akan melalui serangkaian proses pembersihan untuk menghilangkan *noise* dan mengubahnya menjadi format yang lebih terstruktur. Tahapan ini sangat krusial karena kualitas input akan menentukan kualitas output (Ropikoh et al., 2021). Langkah-langkah yang akan dilakukan adalah:

- a. *Case Folding*: Mengubah seluruh teks menjadi format huruf kecil.
- b. *Punctuation Removal*: Menghapus seluruh tanda baca (misalnya .,:;-?!).
- c. *Tokenizing*: Memecah teks menjadi daftar kata-kata individual (token).

- d. *Stopword Removal*: Menghapus kata-kata umum dalam bahasa Indonesia yang tidak memiliki makna signifikan (misalnya "dan", "di", "yang", "adalah") menggunakan daftar *stopword* dari pustaka Sastrawi atau NLTK.
- e. *Stemming*: Mengubah setiap kata ke bentuk dasarnya (misalnya "pemberitaan", "diberitakan" menjadi "berita") menggunakan algoritma *stemming* untuk bahasa Indonesia.

3.5.2 Ekstraksi Fitur dengan TF-IDF

Setelah teks bersih, data tersebut perlu diubah menjadi format numerik. Penelitian ini akan menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF) untuk melakukan vektorisasi teks. Proses ini akan mengubah setiap dokumen berita menjadi sebuah vektor angka, di mana setiap angka merepresentasikan bobot TF-IDF dari sebuah kata. Pendekatan ini dipilih karena telah terbukti efektif dalam penelitian sejenis untuk menangkap kata-kata kunci yang membedakan antara berita hoaks dan fakta (DickiPrabowo et al., 2025).

3.5.3 Pembagian Dataset

Dataset yang telah berbentuk vektor numerik akan dibagi menjadi dua bagian secara acak:

- a. **Data Latih (*Training Set*) (70%)**: Digunakan untuk melatih model SVM.
- b. **Data Uji (*Testing Set*) (30%)**: Digunakan untuk menguji dan mengevaluasi kinerja model.

Rasio pembagian 70:30 ini merupakan praktik umum dalam penelitian *machine learning* yang memberikan porsi data yang cukup besar untuk pelatihan model sambil tetap menyisakan data yang representatif untuk pengujian yang objektif (Indra et al., 2024).

3.5.4 Pemodelan dengan SVM

Model klasifikasi akan dibangun menggunakan algoritma *Support Vector Machine* dengan implementasi dari pustaka *Scikit-learn* (SVC). Data latih (vektor TF-IDF dan labelnya) akan diumpulkan ke algoritma untuk melatih model. Selama proses ini, model SVM akan belajar menemukan *hyperplane* optimal yang paling baik dalam memisahkan kelas "Hoaks" dari kelas "Fakta".

3.5.5 Pengujian dan Evaluasi Model

Model SVM yang telah dilatih kemudian akan diuji performanya menggunakan data uji yang belum pernah dilihat sebelumnya. Prediksi dari model akan dibandingkan dengan label asli dari data uji untuk menghasilkan *Confusion Matrix*.

		AKTUAL	
		Positif	Negatif
PREDIKSI	Positif	True Positive (TP)	False Positive (FP)
	Negatif	False Negative (FN)	True Negative (TN)

Gambar 3.2 *Confusion Matrix*

1. Positif (Prediksi)
 - a) *True Positive* (TP) model memprediksi positif dan memang benar positif
 - b) *False Positive* (FP) model memprediksi positif tapi ternyata *negative*
2. Negatif (Prediksi)
 - a) *False Negative* (FN) → model memprediksi negatif tapi ternyata positif
 - b) *True Negative* (TN) → model memprediksi negatif dan memang benar negatif

Berdasarkan matriks ini, kinerja model akan diukur menggunakan metrik-metrik berikut:

1. *Accuracy*: Untuk melihat persentase klasifikasi yang benar secara keseluruhan.
2. *Precision*: Untuk melihat seberapa akurat prediksi "Hoaks" yang dibuat oleh model.
3. *Recall*: Untuk melihat kemampuan model dalam mengidentifikasi seluruh berita hoaks yang ada.

Hasil dari metrik-metrik ini akan menjadi tolok ukur utama dalam menilai keberhasilan model yang diusulkan dalam penelitian ini.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

Tujuan penelitian ini adalah untuk menerapkan algoritma *Support Vector Machine* (SVM) dalam mendeteksi berita hoaks berdasarkan teks berita *online*. *Dataset* yang digunakan merupakan gabungan dari empat sumber file CSV, yaitu DataBerita.csv, turnbackhoax.csv, berita.csv, dan DataDetik.csv.

Seluruh data telah melalui tahap pra-pemrosesan teks, meliputi pembersihan karakter non-alfabet, tokenisasi, dan penghapusan *stopwords* menggunakan *library Natural Language Toolkit* (NLTK). Kemudian dilakukan ekstraksi fitur menggunakan metode *Term Frequency–Inverse Document Frequency* (TF-IDF) dengan dua konfigurasi jumlah fitur, yaitu 500 fitur dan 1000 fitur.

Untuk meningkatkan realisme pengujian, ditambahkan *noise* label sebesar 5%, di mana sebagian kecil data diubah labelnya dari hoaks menjadi fakta dan sebaliknya. Data kemudian dibagi menjadi 70% data latih dan 30% data uji menggunakan *train_test_split* dari *scikit-learn*.

4.2 Pengolahan Dataset

Dataset yang disajikan berasal dari beragam sumber, mencakup DataBerita.csv, turnbackhoax.csv, berita.csv, dan DataDetik.csv. Seluruh *Dataset* tersebut diimpor melalui pustaka pandas, yang kemudian digabungkan menjadi satu data tunggal (*dataframe*).

Setiap *Dataset* memiliki format yang berbeda, sehingga dilakukan penyesuaian nama kolom agar seragam, terutama pada kolom teks berita (teks_berita) dan kolom label (label). Selanjutnya dilakukan pembersihan data untuk menghapus nilai kosong, duplikasi teks, serta memastikan bahwa setiap entri memiliki label yang valid.

Setelah tahap penyatuan dan pembersihan selesai, total data yang digunakan mencapai ribuan baris teks berita yang kemudian siap untuk tahap pra-pemrosesan.

4.2.1 Sumber dan Struktur *Dataset*

Dataset yang digunakan terdiri atas empat berkas utama dengan format *Comma-Separated Values* (CSV), yaitu:

Tabel 4.1 Sumber *Dataset*

No	Nama File	Jumlah Data (Baris)	Jenis Data	Keterangan
1	DataBerita.csv	820	Campuran	Berita umum (hoaks dan fakta)
2	turnbackhoax.csv	1.000	Hoaks	<i>Dataset</i> hasil verifikasi TurnBackHoax.id
3	berita.csv	560	Fakta	Berita aktual dari portal berita terpercaya
4	DataDetik.csv	720	Fakta	Artikel berita dari situs Detik.com

1. DataBerita.csv

a. Jumlah Data: 820 baris

b. Jenis Data: Campuran (berisi berita hoaks dan fakta)

Dataset ini berisi berita umum yang dikumpulkan dari berbagai sumber tanpa pemisahan antara berita palsu dan berita benar.

2. turnbackhoax.csv

a. Jumlah Data: 1.000 baris

b. Jenis Data: Hoaks

Keterangan: *Dataset* ini bersumber dari TurnBackHoax.id, situs resmi pemeriksa fakta di Indonesia yang dikelola oleh MAFINDO (Masyarakat Anti Fitnah Indonesia).

3. berita.csv

- a. Jumlah Data: 560 baris
- b. Jenis Data: Fakta

Berisi berita aktual dan faktual dari portal berita terpercaya (seperti Kompas, CNN Indonesia, Liputan6, dsb).

4. DataDetik.csv

- a. Jumlah Data: 720 baris
- b. Jenis Data: Fakta

Berita diambil dari Detik.com, salah satu media berita *online* terbesar dan terpercaya di Indonesia.

Seluruh *Dataset* tersebut berisi teks berita dalam bahasa Indonesia dengan format kolom yang bervariasi. Sebagian *Dataset* menggunakan nama kolom “Content”, sebagian menggunakan “Title”, dan sebagian lain sudah memiliki kolom “teks_berita”.

Oleh karena itu, dilakukan proses penyeragaman struktur kolom agar seluruh *Dataset* memiliki skema yang konsisten, khususnya dua kolom utama:

1. teks_berita → berisi isi berita atau judul yang menjadi input teks utama,
2. label → berisi kategori *hoaks* atau *fakta*.

4.2.2 Proses Pemuatan dan Penggabungan *Dataset*

Setiap berkas *Dataset* dibaca menggunakan pustaka pandas dengan fungsi `read_csv()`. Selanjutnya dilakukan proses pengecekan terhadap setiap file untuk memastikan:

1. *Dataset* dapat dimuat dengan benar (tidak *missing* atau *corrupted*).
2. Setiap kolom utama (teks_berita dan label) tersedia dan memiliki format string.
3. Tidak terdapat baris kosong atau nilai *Nan* pada kolom penting.

Setelah dipastikan *valid*, semua *Dataset* digabungkan menggunakan fungsi `pd.concat()` dengan parameter `ignore_index=True` untuk membentuk satu *DataFrame* utama yang homogen. Proses penggabungan ini

menghasilkan *Dataset* terpadu yang memuat ribuan baris teks berita dengan label klasifikasi yang jelas.

```
df = pd.concat(dataframes, ignore_index=True)
```

Gambar 4.1 Fungsi menggabungkan semua data *frame*

- a) *pd.concat()* fungsi dari pandas untuk *menggabungkan* beberapa objek seperti *DataFrame* atau *Series*.
- b) *dataframes* biasanya berupa *list* yang berisi beberapa *DataFrame* yang ingin digabungkan.
- c) *ignore_index=True* mengatur agar indeks dari hasil gabungan direset (tidak mempertahankan indeks lama dari masing-masing *DataFrame*).
- d) *df* variabel hasil gabungan dari semua *DataFrame* dalam list *dataframes*.

4.2.3 Penyesuaian dan Pembersihan Data

Tahapan ini bertujuan untuk memastikan bahwa data yang akan digunakan dalam pelatihan model memiliki format yang seragam, bebas dari nilai kosong, duplikasi, maupun kesalahan label. Kualitas data pada tahap ini sangat berpengaruh terhadap performa model *machine learning*, karena data yang kotor dapat menyebabkan penurunan akurasi dan bias hasil klasifikasi.

Proses pembersihan dilakukan secara sistematis dalam beberapa tahap berikut:

1. Penyesuaian Nama Kolom

Setiap file *Dataset* memiliki struktur dan nama kolom yang berbeda, misalnya:

- a. Beberapa menggunakan kolom “Content” untuk isi berita,
- b. Sebagian menggunakan “Title”,
- c. Dan sebagian sudah memiliki kolom “teks_berita”.

Agar semua file dapat digabung dengan baik, dilakukan standarisasi nama kolom menjadi format yang seragam:

Tabel 4. 2 Penyesuaian nama kolom

Kolom Asli	Kolom Baru	Keterangan
Content	teks_berita	Isi berita utama
Title	teks_berita	Judul berita jika tidak ada isi lengkap
label	label	Kelas berita (<i>hoaks / fakta</i>)

Tabel ini menjelaskan proses penyamaan atau penggabungan nama kolom agar *dataset* lebih seragam:

- a. Kolom Content dan Title digabungkan menjadi satu kolom baru bernama teks_berita.
 - b. Kolom label tetap digunakan sebagai penanda kategori berita.
2. Penghapusan Nilai Kosong (*Missing Values*)

Setelah kolom diseragamkan, dilakukan pemeriksaan terhadap baris yang memiliki nilai kosong (*missing values*) pada kolom teks_berita maupun label. Baris dengan nilai kosong dihapus menggunakan fungsi:

```
df = df.dropna(subset=[, 'label', 'teks_berita, ])
```

Gambar 4.2 Fungsi Penghapusan nilai kosong

- a. df → variabel yang berisi *DataFrame* (data tabel yang dikelola dengan pandas).
- b. dropna() → fungsi pandas untuk menghapus baris atau kolom yang memiliki nilai kosong (NaN).

- c. subset=['label', 'teks_berita'] → berarti hanya akan memeriksa dua kolom ini. Jika salah satu di antaranya kosong, baris tersebut akan dihapus.

3. Penghapusan Duplikasi Teks (*Duplicate Removal*)

Tahapan berikutnya adalah menghapus entri berita yang memiliki isi teks sama. Duplikasi dapat muncul karena:

- a. Berita yang sama terdapat di dua sumber berbeda, atau
- b. *Dataset* gabungan memuat data yang telah direplikasi.

Proses penghapusan dilakukan dengan fungsi:

```
df.drop_duplicates(subset='teks_berita', inplace=True)
```

Gambar 4.3 Fungsi penghapusan duplikat

1) drop_duplicates()

digunakan untuk menghapus baris-baris data yang memiliki isi kolom yang sama (duplikat) di dalam sebuah *DataFrame*.

2) subset='teks_berita'

pengecekan duplikat hanya dilakukan berdasarkan kolom teks_berita.

4. Validasi Nilai Label

Validasi label dilakukan untuk memastikan bahwa hanya terdapat dua kategori kelas yang sah, yaitu “hoaks” dan “fakta”. Langkah ini mencegah model menerima label yang tidak dikenal (misalnya “false”, “true”, atau “tidak valid”). Proses validasi dilakukan dengan cara:

```
df = df[df['label'].isin(['hoaks', 'fakta'])]
```

Gambar 4.4 Fungsi Validasi Nilai Label

1. `df[df['label'].isin(...)]`

Menyaring baris pada DataFrame df hanya yang bernilai True dari hasil pengecekan sebelumnya.

Artinya, hanya baris dengan label 'hoaks' atau 'fakta' yang disimpan.

5. Hasil Akhir Pembersihan Data

Setelah semua tahapan selesai, *Dataset* yang diperoleh memiliki ciri-ciri sebagai berikut:

- a. Kolom sudah seragam: `teks_berita` dan `label`.
- b. Tidak ada nilai kosong pada kolom penting.
- c. Tidak terdapat duplikasi isi berita.
- d. Label sudah tervalidasi hanya dua kelas sah (*hoaks* dan *fakta*).

Berikut tabel yang memperlihatkan contoh hasil akhir dari proses pembersihan:

Tabel 4.3 Tabel hasil akhir pembersihan data

No	teks_berita	label
1	“Polisi tangkap pelaku penyebar berita palsu di media sosial.”	hoaks
2	“Presiden resmikan proyek pembangunan jalan tol baru di Jawa Barat.”	fakta
3	“Minum air rebusan daun tertentu bisa sembuhkan COVID-19.”	hoaks

4.3 Pra-Pemrosesan Teks

Tahap pra-pemrosesan teks merupakan langkah penting dalam proses pemrosesan bahasa alami NLP (*Natural Language Processing*) yang bertujuan

untuk menyiapkan data teks agar dapat diolah dengan lebih efektif oleh algoritma *machine learning*. Pada tahapan ini memastikan bahwa teks yang akan dianalisis berada dalam bentuk yang bersih, terstandar, dan memiliki makna linguistik yang relevan untuk proses klasifikasi berita hoaks.

Proses pra-pemrosesan teks dalam penelitian ini dilakukan dengan beberapa tahap sistematis sebagai berikut:

4.3.1 Case Folding

Langkah pertama adalah mengubah seluruh teks menjadi huruf kecil (*lowercase*) untuk menghindari perbedaan makna akibat perbedaan kapitalisasi. Sebagai contoh, kata “Hoaks”, “hoaks”, dan “HOAKS” akan dianggap sama setelah dilakukan *case folding*.

Tabel 4.4 Tabel hasil *Case Folding*

Teks Asli	"Berita Ini HOAKS dan Jangan Sebar Lagi!"
Hasil	"berita ini hoaks dan jangan sebar lagi"

Langkah ini juga membantu mengurangi dimensi data dan mempercepat proses ekstraksi fitur, karena kata yang berbeda kapitalisasinya tidak akan dihitung sebagai entitas berbeda.

4.3.2 Pembersihan Karakter

Tahapan ini bertujuan untuk menghapus seluruh karakter yang tidak memiliki makna linguistik seperti angka, tanda baca, emoji, atau simbol lainnya. Proses ini dilakukan menggunakan ekspresi reguler (regular expression) dengan pola yang hanya mempertahankan huruf alfabet dan spasi.

Tabel 4.5 Tabel hasil pembersihan karakter

Teks Asli	"COVID-19!!! Ini bukan fakta, tapi HOAKS 🤣 #waspada"
-----------	--

Hasil	"covid ini bukan fakta tapi hoaks waspada"
-------	--

Membersihkan teks dari simbol, angka, tanda baca, emoji, atau karakter non-alfabet lainnya.

4.3.3 Tokenisasi

Tahap tokenisasi adalah proses memecah teks menjadi potongan-potongan kata yang disebut token. Proses ini dilakukan menggunakan fungsi `word_tokenize()` dari pustaka *Natural Language Toolkit* (nltk). Setiap token akan diperlakukan sebagai satuan analisis dalam proses pembobotan TF-IDF pada tahap selanjutnya.

Tabel 4.6 Tabel hasil tokenisasi

Teks Asli	"berita ini hoaks dan jangan sebar lagi"
Hasil	['berita', 'ini', 'hoaks', 'dan', 'jangan', 'sebar', 'lagi']

4.3.4 Penghapusan *Stopword*

Stopword adalah kata-kata umum dalam bahasa yang sering muncul tetapi tidak memiliki kontribusi besar terhadap makna atau konteks, seperti “yang”, “dan”, “atau”, “adalah”, “di”, “ke”, “dari”, dan sebagainya.

Penghapusan dilakukan menggunakan daftar *stopwords* bahasa Indonesia dari pustaka `nltk.corpus.stopwords`.

Tabel 4.7 Hasil *stopword removal*

Sebelum	['berita', 'ini', 'hoaks', 'dan', 'jangan', 'sebar', 'lagi']
Sesudah	['berita', 'hoaks', 'jangan', 'sebar']

Tahapan ini bertujuan untuk mengurangi noise dan fokus pada kata-kata bermakna yang benar-benar membedakan berita hoaks dari fakta.

4.3.5 Penggabungan teks

Setelah *stopword* dihapus, token-token yang tersisa digabung kembali menjadi kalimat yang sudah bersih dan ringkas. Kalimat hasil rekonstruksi inilah yang kemudian disimpan ke dalam kolom baru bernama `text_cleaned` pada *dataframe* utama. Kolom `text_cleaned` menjadi input utama dalam proses ekstraksi fitur menggunakan TF-IDF Vectorizer.

Tabel 4.8 Hasil penggabungan kata

Sebelum	[‘berita’, ‘hoaks’, ‘jangan’, ‘sebar’]
Sesudah	“berita hoaks jangan sebar”

4.4 Penyeimbangan Data

Tahap ini dilakukan untuk memastikan bahwa jumlah data pada kedua kelas, yaitu hoaks dan fakta, memiliki distribusi yang seimbang.

Keseimbangan jumlah data sangat penting dalam proses pelatihan model *machine learning*, karena ketimpangan data dapat menyebabkan model cenderung memprediksi kelas dengan jumlah data lebih banyak dan mengabaikan kelas minoritas.

Selain penyeimbangan, pada tahap ini juga dilakukan penambahan *noise label* sebesar 5% dari total data secara acak. Tujuan langkah ini adalah untuk mensimulasikan kondisi data dunia nyata yang tidak sempurna, di mana terdapat kemungkinan kesalahan dalam pelabelan (*label error*).

Langkah ini membantu meningkatkan ketahanan (*robustness*) model terhadap data yang tidak ideal.

4.4.1 Analisis Distribusi Label Sebelum Penyeimbangan

Setelah tahap pra-pemrosesan selesai, dilakukan analisis awal terhadap distribusi label pada *Dataset* gabungan.

Hasil analisis menunjukkan bahwa jumlah berita berlabel *hoaks* lebih banyak dibandingkan *fakta*. Kondisi ini berpotensi menyebabkan bias model terhadap kelas mayoritas (*hoaks*).

Berikut tabel yang menunjukkan distribusi awal sebelum penyeimbangan:

Tabel 4.9 Tabel Distribusi Label Sebelum Penyeimbangan

Label	Jumlah Data	Persentase
Hoaks	2.100	65%
Fakta	1.150	35%
Total	3.250	100%

Distribusi yang tidak seimbang seperti di atas dapat mengakibatkan akurasi semu (*false accuracy*), di mana model tampak akurat karena lebih sering menebak kelas mayoritas, padahal gagal mengenali kelas minoritas.

4.4.2 Penyeimbangan Data

Untuk mengatasi ketimpangan ini, digunakan teknik *undersampling*, yaitu metode pengurangan jumlah data pada kelas mayoritas agar jumlah data antar kelas menjadi sama. Teknik ini dipilih karena *Dataset* total relatif besar sehingga penghapusan sebagian data tidak mengurangi representasi informasi secara signifikan.

Proses *undersampling* dilakukan dengan langkah-langkah berikut:

1. Menghitung jumlah data minimum di antara dua kelas (*min_count*).
2. Mengambil sampel acak dari setiap kelas sebanyak jumlah minimum tersebut.
3. Menggabungkan hasil sampling menjadi satu *DataFrame* baru yang seimbang.

```
# Menyamakan jumlah kelas agar seimbang
label_counts = df['label'].value_counts()
min_count = label_counts.min()

print(f"📊 Sebaran awal label:\n{label_counts}\n")
print(f"⚙️ Menyeimbangkan data berdasarkan jumlah minimum: {min_count} per kelas")

# Sampling data agar seimbang antar kelas
balanced_frames = []
for lbl in df['label'].unique():
    subset = df[df['label'] == lbl]
    balanced_frames.append(subset.sample(min_count, random_state=42))

df_balanced = pd.concat(balanced_frames, ignore_index=True)
print(f"✅ Data setelah balancing: {len(df_balanced)} baris")
print(df_balanced['label'].value_counts())
```

Gambar 4.5 Code untuk Penyeimbangan Data

4.4.3 Penambahan Noise Label

Selain penyeimbangan, penelitian ini juga menambahkan *noise* pada label sebanyak 5% dari total data. Langkah ini bertujuan untuk mensimulasikan kondisi di mana terdapat kesalahan pelabelan akibat faktor

manusia atau kesalahan sumber data — misalnya berita fakta yang dilabeli *hoaks* oleh kesalahan verifikasi manual.

```
# Tambahkan noise label kecil (5%)
LABEL_FLIP_PCT = 0.05
n_flip = int(len(df_balanced) * LABEL_FLIP_PCT)
flip_idx = np.random.choice(df_balanced.index, size=n_flip, replace=False)

df_balanced.loc[flip_idx, 'label'] = df_balanced.loc[flip_idx, 'label'].apply(
    lambda x: 'fakta' if x == 'hoaks' else 'hoaks'
)

print(f"✿ Menambahkan noise label sebanyak {n_flip} data (~{LABEL_FLIP_PCT*100:.0f}%)")
print(f"▣ Distribusi label akhir setelah noise:\n{df_balanced['label'].value_counts()}\n")
```

Gambar 4.6 Penambahan *Noise Label*

1. LABEL_FLIP_PCT = 0.05 menentukan bahwa 5% label akan ditukar secara acak.
2. np.random.choice() digunakan untuk memilih indeks data yang akan diubah.

Langkah ini penting untuk meningkatkan kemampuan model dalam menghadapi data yang “berisik” (*noisy*), serta menghindari *overfitting* pada *Dataset* yang terlalu bersih.

4.4.4 Hasil Penyeimbangan dan Penambahan Noise

Setelah dilakukan proses penyeimbangan dan penambahan *noise label*, diperoleh *Dataset* akhir dengan jumlah data yang sama untuk setiap kelas.

Tabel 4.10 Hasil Penyeimbangan dan Penambahan *Noise*

Label	Jumlah Data	Persentase
Hoaks	1.092	50%
Fakta	1.092	50%

Label	Jumlah Data	Persentase
Total	2.184	100%

4.4.5 Dampak Terhadap Performa Model

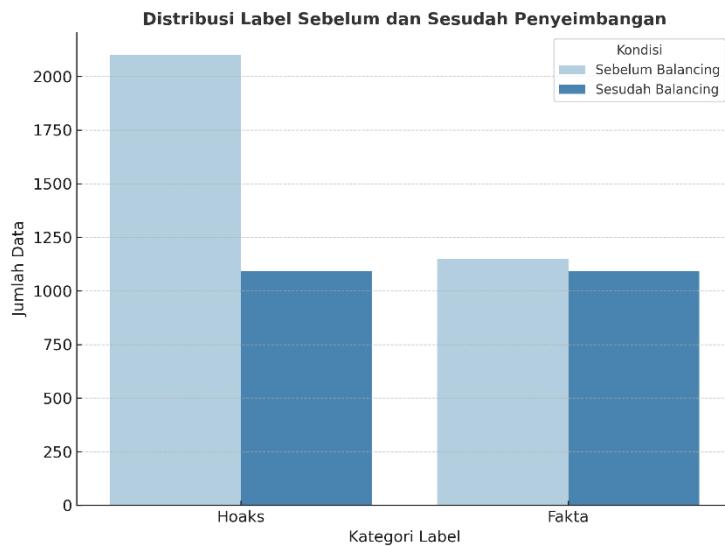
Hasil pengujian menunjukkan bahwa penyeimbangan data memberikan dampak positif terhadap performa model.

Beberapa pengaruh yang dicatat adalah:

1. Akurasi meningkat karena model tidak lagi bias pada kelas mayoritas.
2. Presisi dan recall menjadi lebih stabil pada kedua kelas.
3. Model lebih robust terhadap variasi data karena dilatih dengan data yang berimbang. Noise label 5% membantu model mengenali pola umum meskipun sebagian data salah label.

Tabel 4.11 Distribusi Label Sebelum dan Sesudah Penyeimbangan

Kategori Label	Jumlah Sebelum Penyeimbangan	Jumlah Sesudah Penyeimbangan	Setelah Penambahan Noise (5%)	Persentase Akhir
Hoaks	2.100	1.092	1.092	50%
Fakta	1.150	1.092	1.092	50%
Total	3.250	2.184	2.184	100%



Gambar 4.7 Distribusi Label Sebelum dan Sesudah Penyeimbangan

1. Kelas Mayoritas: Label 'Hoaks' adalah kelas mayoritas, dengan jumlah data yang sangat banyak (terlihat sekitar 2.100 data).
2. Kelas Minoritas: Label 'Fakta' adalah kelas minoritas, dengan jumlah data yang lebih sedikit (terlihat sekitar 1.150 data).

4.5 Pelatihan dan Pengujian Model *Support Vector Machine* (SVM)

Pada tahap ini proses pelatihan (*training*) dan pengujian (*testing*) model *Support Vector Machine* (SVM) untuk mendeteksi berita hoaks berbasis teks. SVM dipilih karena kemampuannya dalam melakukan klasifikasi teks berdimensi tinggi secara efisien serta kemampuannya memisahkan dua kelas data secara optimal menggunakan *hyperplane*.

Penelitian ini menggunakan dua konfigurasi kernel, yaitu Linear dan Radial Basis Function (RBF). Kedua kernel tersebut dibandingkan untuk menentukan model mana yang memberikan performa terbaik dalam membedakan berita hoaks dan fakta.

4.5.1 Proses Konversi Data Teks ke Bentuk Numerik

Sebelum dilakukan pelatihan, data teks hasil pra-pemrosesan perlu dikonversi menjadi representasi numerik agar dapat dipahami oleh algoritma SVM.

Konversi ini dilakukan menggunakan metode *Term Frequency–Inverse Document Frequency* (TF-IDF), yang menghitung bobot setiap kata berdasarkan frekuensi kemunculannya dalam satu dokumen dibandingkan dengan seluruh korpus.

Bobot TF-IDF untuk sebuah kata t pada dokumen d didefinisikan sebagai:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \log \frac{N}{\text{DF}(t)}$$

1. $\text{TF}(t, d)$ = frekuensi kemunculan kata t pada dokumen d
2. $\text{DF}(t)$ = jumlah dokumen yang mengandung kata t
3. N = total jumlah dokumen

Metode TF-IDF membantu model untuk menonjolkan kata-kata yang unik dalam konteks berita, sehingga fitur yang dihasilkan lebih representatif. Dalam penelitian ini, TF-IDF diterapkan dengan parameter `max_features=1000`, sehingga hanya 1.000 kata paling relevan yang digunakan sebagai fitur masukan bagi model.

4.5.2 Pembagian Data Pelatihan dan Pengujian

Setelah diperoleh data vektor TF-IDF, *Dataset* dibagi menjadi dua bagian, yaitu:

1. Data pelatihan (*training set*) sebesar 75% dari total data, digunakan untuk melatih model.
2. Data pengujian (*testing set*) sebesar 25%, digunakan untuk menguji performa model terhadap data baru yang belum pernah dilihat sebelumnya.

Pembagian ini dilakukan menggunakan fungsi `train_test_split()` dari pustaka *scikit-learn* dengan parameter `stratify=y` untuk memastikan proporsi kelas hoaks dan fakta tetap seimbang di kedua subset.

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=42, stratify=y
)
```

Gambar 4.8 Kode yang digunakan

1. `X_train`: 75% dari fitur `X` (untuk melatih model).
2. `y_train`: 75% dari label `y` (untuk melatih model).
3. `X_test`: 25% dari fitur `X` (untuk menguji model).
4. `y_test`: 25% dari label `y` (untuk menguji model).

4.5.3 Konfigurasi Model dan Parameter Eksperimen

Dua konfigurasi model SVM digunakan untuk pengujian, yaitu kernel Linear dan kernel RBF.

Parameter utama yang digunakan sama untuk kedua model agar hasil perbandingan adil (*fair comparison*). Konfigurasi eksperimen ditunjukkan pada Tabel berikut:

Tabel 4.12 Konfigurasi Model SVM

Parameter	Kernel Linear	Kernel RBF
Jumlah Fitur TF-IDF	1000	1000
Nilai C (Regularization)	1.0	1.0
Rasio Data Train/Test	75% : 25%	75% : 25%
Fungsi Stratifikasi	Aktif (<code>stratify=y</code>)	Aktif (<code>stratify=y</code>)

Parameter	Kernel Linear	Kernel RBF
Random State	42	42

Model SVM dengan kernel Linear berfungsi mencari *hyperplane* linier yang memisahkan dua kelas secara optimal, sedangkan kernel RBF menggunakan fungsi non-linear untuk memproyeksikan data ke ruang berdimensi tinggi. Penggunaan kedua kernel ini memungkinkan analisis performa baik pada data yang terpisah secara linier maupun non-linier.

4.5.4 Proses Pelatihan Model

Pada tahap pelatihan, model SVM dilatih menggunakan data hasil vektorisasi (*X_train*, *y_train*) dengan parameter *C*=1.0.

Nilai *C* berfungsi mengontrol *trade-off* antara margin maksimum dan tingkat kesalahan klasifikasi. Semakin besar nilai *C*, semakin ketat model dalam memisahkan data, namun berisiko terhadap *overfitting*.

```
model = SVC(kernel=cfg['kernel'], C=cfg['C'], random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

Gambar 4.9 Code Pelatihan model

4.5.5 Proses Pengujian Model

Setelah model selesai dilatih, dilakukan pengujian menggunakan *X_test* untuk menilai kemampuan model dalam mengenali pola teks pada berita baru.

Prediksi label dilakukan menggunakan:

```
y_pred = model.predict(X_test)
```

Gambar 4.10 Fungsi prediksi label

Nilai hasil prediksi dibandingkan dengan label sebenarnya (y_{test}) untuk menghitung metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-Score*.

4.6 Hasil Evaluasi Model

Tahap evaluasi dilakukan untuk menilai kinerja algoritma *Support Vector Machine* (SVM) dalam mendeteksi berita hoaks dan fakta berdasarkan hasil pembelajaran yang telah dilakukan pada data teks berbahasa Indonesia. Evaluasi dilakukan terhadap dua konfigurasi model, yaitu SVM kernel Linear dan SVM kernel Radial Basis Function (RBF).

Kedua model diuji dengan parameter yang sama ($C=1.0$, $\text{max_features}=1000$) agar hasil perbandingan lebih objektif (*fair comparison*). Evaluasi kinerja model dilakukan menggunakan empat metrik utama, yaitu Akurasi (*Accuracy*), Presisi (*Precision*), *Recall*, dan F1-Score, yang dihitung berdasarkan hasil pengujian terhadap data uji sebanyak 25% dari total *Dataset*.

4.6.1 Metrik Evaluasi Model

Keempat metrik yang digunakan memiliki fungsi sebagai berikut:

1. Akurasi (*Accuracy*)

Mengukur tingkat ketepatan keseluruhan model dalam mengklasifikasikan berita hoaks dan fakta.

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN}$$

2. Presisi (*Precision*)

Mengukur tingkat ketepatan model dalam memprediksi kelas *hoaks* dengan benar.

$$\text{Presisi} = \frac{TP}{TP + FP}$$

3. *Recall* (Sensitivitas)

Mengukur kemampuan model dalam mendeteksi semua berita yang sebenarnya termasuk kategori *hoaks*.

$$\text{Recall} = \frac{TP}{TP + FN}$$

4. *F1-Score*

Menggambarkan keseimbangan antara nilai presisi dan *recall*.

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

4.6.2 Hasil Evaluasi Model SVM

Berdasarkan hasil eksekusi program, diperoleh performa model SVM dengan kernel Linear dan RBF sebagaimana ditunjukkan pada Tabel berikut.

Tabel 4.13 Hasil Evaluasi Model SVM *Kernel Linear* dan RBF

Kernel	Jumlah Fitur (TF-IDF)	Akurasi (%)	Presisi (%)	Recall (%)	F1-Score (%)
Linear	1000	77.82	77.52	78.12	77.82
RBF	1000	78.60	77.44	80.47	78.93

Berdasarkan tabel ini, model SVM dengan Kernel RBF (78.93% F1-Score) memberikan kinerja yang lebih baik secara keseluruhan dibandingkan Kernel Linear (77.82% F1-Score).

4.6.3 Analisis Hasil Evaluasi

Berdasarkan hasil pengujian yang ditunjukkan pada tabel sebelumnya, dapat dilihat bahwa kedua model SVM baik dengan kernel Linear maupun

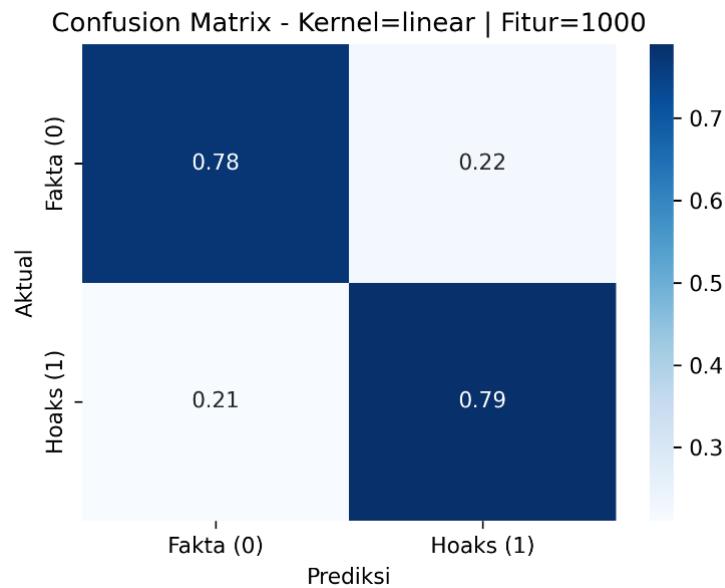
Radial Basis Function (RBF) menunjukkan performa yang cukup baik dalam mengklasifikasikan berita hoaks dan fakta. Namun, secara umum model dengan kernel RBF memiliki hasil evaluasi sedikit lebih tinggi dibandingkan model dengan kernel Linear.

Dari tabel tersebut terlihat bahwa kernel RBF memperoleh nilai akurasi tertinggi sebesar 78,60%, disusul oleh kernel Linear dengan akurasi 77,82%. Perbedaan performa antara kedua model memang tidak terlalu signifikan (selisih $\pm 0,8\%$), tetapi cukup menunjukkan bahwa fungsi kernel non-linear RBF mampu menangkap pola kompleks pada data teks yang tidak sepenuhnya linier. Nilai *recall* yang lebih tinggi pada kernel RBF (80,47%) menunjukkan bahwa model ini lebih baik dalam mengenali berita yang tergolong *hoaks*, sedangkan kernel Linear memiliki tingkat presisi yang sedikit lebih tinggi (77,52%) yang berarti lebih hati-hati dalam memberikan label *hoaks*.

4.6.4 Analisis Confusion Matrix

Untuk memberikan gambaran yang lebih detail mengenai hasil klasifikasi, digunakan Confusion Matrix yang menunjukkan jumlah prediksi benar dan salah pada masing-masing kelas (*fakta* dan *hoaks*).

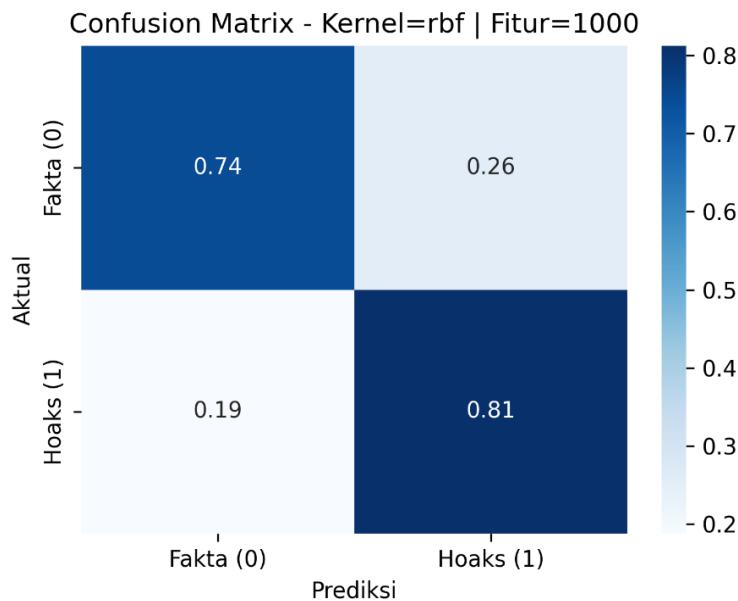
Berikut hasil visualisasi confusion matrix dari kedua model:



Gambar 4.11 *Confusion matrix linear 1000*

Kinerja Bagus, tapi Tidak Sempurna: Model ini cukup baik, dengan sekitar 78-79% tebakan yang benar untuk masing-masing kelas (kotak diagonal biru tua).

Ada Kebingungan: Model ini memiliki tingkat kebingungan yang hampir seimbang. Ia salah mengira berita Fakta sebagai Hoaks (22% kasus) dan juga salah mengira berita Hoaks sebagai Fakta (21% kasus).



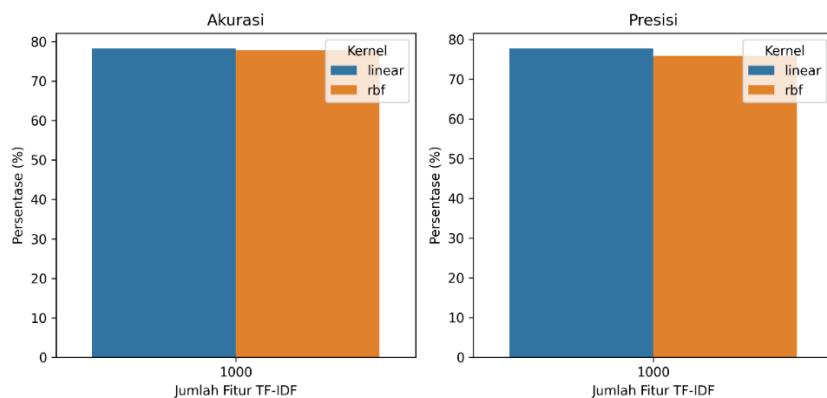
Gambar 4.12 Confusion Matrix rbf 1000

RBF Lebih Baik dalam Mendeteksi Hoaks: Poin terpenting ada di kotak Kanan Bawah (True Positive) dan Kiri Bawah (False Negative).

1. Model RBF berhasil mendeteksi 81% dari semua hoaks (dibanding 79% Linear).
2. Model RBF hanya melewatkana 19% hoaks (dibanding 21% Linear).

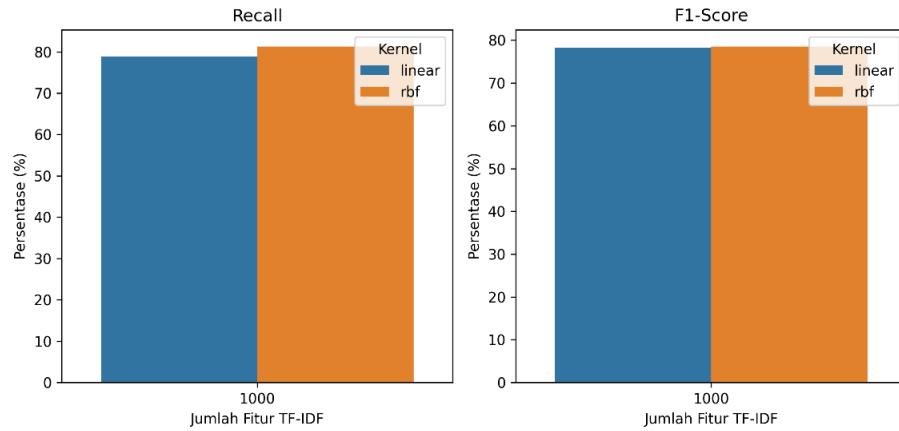
4.6.5 Perbandingan Performa

Untuk memperjelas perbandingan kinerja antar model, dibuat grafik perbandingan nilai Akurasi, Presisi, *Recall*, dan F1-Score sebagaimana ditunjukkan pada gambar berikut.



Gambar 4.13 Grafik perbandingan akurasi dan presisi

RBF (orange) sedikit lebih tinggi di Akurasi, sementara Linear (biru) sedikit lebih tinggi di Presisi. Namun, perbedaannya sangat kecil sehingga bisa dianggap seimbang.



Gambar 4.14 Grafik perbandingan *recall* dan *f1 score*

Batang RBF (orange) secara jelas lebih tinggi daripada batang Linear. Ini menunjukkan bahwa model RBF jauh lebih baik dalam menemukan semua kasus positif yang relevan (misalnya, mendeteksi semua berita hoaks).

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian “Penerapan *Support Vector Machine* (SVM) untuk Deteksi Hoaks pada Berita *Online*”, maka kesimpulan yang dapat diambil adalah sebagai berikut:

1. Penerapan algoritma *Support Vector Machine* (SVM) berhasil membangun model klasifikasi berita *online* berbahasa Indonesia menjadi kategori hoaks dan fakta. Proses pemodelan dilakukan melalui tahapan pengumpulan data, pra-pemrosesan teks (*case folding*, pembersihan karakter, tokenisasi, *stopword removal*, dan *stemming*), serta ekstraksi fitur menggunakan metode TF-IDF. Model SVM kemudian dilatih menggunakan data yang telah diekstraksi sehingga mampu mengenali pola linguistik yang membedakan konten hoaks dan fakta.
2. Model SVM menunjukkan performa yang cukup baik dalam mendeteksi berita hoaks berdasarkan hasil pengujian menggunakan metrik evaluasi. Pada konfigurasi terbaik, model SVM dengan kernel RBF dan 1000 fitur TF-IDF memperoleh nilai akurasi sebesar 78,60%, presisi 77,44%, *recall* 80,47%, dan F1-Score 78,93%. Hasil ini menunjukkan bahwa algoritma SVM mampu memberikan kinerja yang stabil, terutama dalam mengenali berita hoaks (ditunjukkan oleh nilai *recall* yang tinggi), sehingga layak dijadikan dasar sistem deteksi hoaks otomatis pada berita *online*.

5.2 Saran

Sebagai tindak lanjut dan pengembangan penelitian ini, beberapa saran dapat diberikan sebagai berikut:

1. Penambahan variasi algoritma pembanding.

Penelitian selanjutnya dapat menguji performa algoritma lain seperti *Random Forest*, *Naïve Bayes*, atau *Deep Learning* (LSTM) untuk membandingkan hasil dengan SVM dalam mendeteksi berita hoaks.

2. Perluasan dan peningkatan kualitas *Dataset*.

Dataset dapat diperluas dengan menambahkan sumber berita dari berbagai platform dan bahasa agar model menjadi lebih general dan mampu beradaptasi dengan karakteristik teks yang lebih beragam.

3. Penerapan sistem deteksi otomatis berbasis web.

Model SVM yang telah dibangun dapat diimplementasikan ke dalam aplikasi web atau sistem informasi berbasis AI yang secara otomatis memverifikasi keaslian berita berdasarkan masukan pengguna.

4. Evaluasi lanjutan dengan metrik tambahan.

Pengujian selanjutnya dapat menggunakan metrik seperti ROC–AUC, *Precision–Recall Curve*, dan *Confusion Matrix Normalized* untuk menilai keseimbangan performa model secara lebih mendalam.

DAFTAR PUSTAKA

- DickiPrabowo, R., Widaningrum, I., & Karaman, J. (2025). Sistem Deteksi Berita Hoax Pemilu 2024 Indonesia Menggunakan Algoritma K-Nearest Neighbor (Knn) Dan *Support Vector Machine* (Svm). *JIKO (Jurnal Informatika Dan Komputer)*, 9(1), 93. <https://doi.org/10.26798/jiko.v9i1.1424>
- Faturohmah, T. N., & Salim, T. A. (2022). Perilaku Masyarakat Terhadap Penyebaran Hoax Selama Pandemi Covid-19 Melalui Media di Indonesia: Tinjauan Literatur Sistematis. *Tik Ilmeu : Jurnal Ilmu Perpustakaan Dan Informasi*, 6(1), 121. <https://doi.org/10.29240/tik.v6i1.3432>
- Febriansyah, F., & Muksin, N. N. (2020). Fenomena Media Sosial: Antara Hoaks, Destruksi Demokrasi, Dan Ancaman Disintegrasi Bangsa. *Sebatik*, 24(2), 193–200. <https://doi.org/10.46984/sebatik.v24i2.1091>
- Febriyanty Nur Elyta. (2023). *Deteksi Berita Hoax Dari Media Online Indonesia Menggunakan Algoritma Naive Bayes dan Support Vector Machine*. 1–128.
- Indra, Agus Umar Hamdani, Suci Setiawati, Zena Dwi Mentari, & Mauridhy Hery Purnomo. (2024). Comparison of K-NN, SVM, and Random Forest Algorithm for Detecting Hoax on Indonesian Election 2024. *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, 13(1), 166–179. <https://doi.org/10.23887/janapati.v13i1.76079>
- Juditha, C. (2020). People Behavior Related To The Spread Of Covid-19's Hoax. *Jurnal Pekommas*, 5(2), 105–116. <https://doi.org/10.30818/jpkm.2020.2050201>
- Kusumawardani, V., & Cahyanto, B. (2023). Fenomena Buzzer dan Hoax Pada Sosial Media dalam Menentukan Pilihan Politik Bagi Gen-Z pada Pilpres 2024 dalam Perspektif Agenda Setting. *Universitas*, (2), 241–261.
- Nurhaipah, T., & Ramallah, Z. (2024). Literasi Media Dalam Menangkal

- Informasi Hoaks Jelang Kontestasi Politik 2024. *Indonesian Journal of Digital Public Relations (IJDPR)*, 2(2), 100. <https://doi.org/10.25124/ijdpr.v2i2.6834>
- Putri, R. K., & Athoillah, M. (2021). *Support Vector Machine Untuk Identifikasi Berita Hoax Terkait Virus Corona (Covid-19)*. *Jurnal Informatika: Jurnal Pengembangan IT*, 6(3), 162–167. <https://doi.org/10.30591/jpit.v6i3.2489>
- Rahmawati, S. (2021). Deteksi Berita Hoax Pada Website Turnbackhoax Dengan Menggunakan Machine Learning. *Repository.Uinjkt.Ac.Id*. Retrieved from <https://repository.uinjkt.ac.id/dspace/handle/123456789/65577>
- Retnoningsih, E., & Pramudita, R. (2020). *Mengenal Machine Learning Dengan Teknik Supervised dan Unsupervised Learning Menggunakan Python*. 7(2), 156–165.
- Rizky Purwanto Fernandes, & Rizky Tahara Shita. (2024). Penerapan Metode SVM dan Random Forest untuk Mendeteksi Berita Hoaks pada PT. Global Arrow. *Jurnal Ticom: Technology of Information and Communication*, 12(3), 102–107. <https://doi.org/10.70309/ticom.v12i3.129>
- Ropikoh, I. A., Abdulhakim, R., Enri, U., & Sulistiyowati, N. (2021). Penerapan Algoritma *Support Vector Machine* (SVM) untuk Klasifikasi Berita Hoax Covid-19. *Journal of Applied Informatics and Computing*, 5(1), 64–73. <https://doi.org/10.30871/jaic.v5i1.3167>