

# **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  
2026**

# **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  
2026**



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 tulis ilmiah sendiri, bukan merupakan karya yang pernah diajukan untuk memperoleh gelar akademik oleh pihak lain, dan bukan merupakan hasil plagiat.
2. Saya ijinkan 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 apapun sesuai aturan yang berlaku apabila di kemudian hari pernyataan ini tidak benar.

Tangerang Selatan, 26 Januari 2026

Materai 10000 IDR

Rifky Fachuzi



FAKULTAS ILMU KOMPUTER  
PROGRAM STUDI TEKNIK INFORMATIKA

## LEMBAR PERSETUJUAN

Yang bertanda tangan dibawah ini:

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, 26 Januari 2026

Pembimbing

Raditia Vindua, S.Si., M.Kom.  
NIDN : 0428059301

Mengetahui,  
Ketua Program Studi

Dr. Eng. Ahmad Musyafa, S.Kom., M.Kom.  
NIDN : 0425018609



FAKULTAS ILMU KOMPUTER  
PROGRAM STUDI TEKNIK INFORMATIKA

## LEMBAR PENGESAHAN

Yang bertanda tangan dibawah ini:

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 dipertahankan di hadapan dewan pengaji ujian skripsi fakultas Ilmu Komputer, program studi Teknik Informatika dan dinyatakan LULUS.

Tangerang Selatan, 26 Januari 2026

Pengaji 1

Pengaji 2

Shandi Noris, S.Kom., M.Kom.  
NIDN : 0431018601

Devi Yunita, S.Kom., M.Kom.  
NIDN : 0412069006

Pembimbing

Raditia Vindua, S.Si., M.Kom.  
NIDN : 0428059301

Mengetahui,  
Ketua Program Studi

Dr. Eng. Ahmad Musyafa, S.Kom., M.Kom.  
NIDN : 0425018609

## ABSTRACT

*The rapid growth of digital media has accelerated the distribution of online news, while simultaneously increasing the spread of hoax information that negatively affects society. Therefore, an automatic method is required to accurately detect hoax news. This study aims to apply the Support Vector Machine (SVM) algorithm to build a classification model that categorizes Indonesian online news into hoax and factual news, as well as to evaluate its performance. The dataset consists of 3,000 Indonesian online news articles, including 2,000 training data and 1,000 testing data, collected from fact-checking websites and trusted national news portals. The research stages include data collection, text preprocessing using Natural Language Processing techniques, feature extraction using Term Frequency–Inverse Document Frequency (TF-IDF), data balancing, and classification modeling using SVM. Model performance is evaluated using accuracy, precision, and recall metrics. The results of this study are expected to demonstrate that the SVM algorithm is effective for detecting hoax news in Indonesian online media and can serve as a foundation for the development of automated hoax detection systems in the future.*

**Keywords:** *hoax news, Support Vector Machine, text classification, TF-IDF, online news*

*Thesis: xiii + 57 pages, 13 references, reference years: 2020–2025*

## ABSTRAK

Perkembangan media digital telah meningkatkan penyebaran berita *online* secara cepat, namun juga memicu maraknya berita hoaks yang berdampak negatif terhadap masyarakat. Oleh karena itu, diperlukan metode otomatis yang mampu mendeteksi berita hoaks secara akurat. Penelitian ini bertujuan untuk menerapkan algoritma *Support Vector Machine* (SVM) dalam membangun model klasifikasi berita *online* berbahasa Indonesia ke dalam kategori hoaks dan fakta, serta mengevaluasi performanya. *Dataset* yang digunakan berjumlah 3.000 berita *online*, terdiri atas 2.000 data latih dan 1.000 data uji, yang bersumber dari situs pemeriksa fakta dan media berita nasional terpercaya. Tahapan penelitian meliputi pengumpulan data, pra-pemrosesan teks menggunakan teknik *Natural Language Processing*, ekstraksi fitur menggunakan *Term Frequency–Inverse Document Frequency* (TF-IDF), penyeimbangan data, serta pemodelan dan pengujian menggunakan algoritma SVM. Kinerja model dievaluasi menggunakan metrik *accuracy*, *precision*, dan *recall*. Hasil penelitian diharapkan mampu menunjukkan bahwa algoritma SVM dapat digunakan secara efektif untuk mendeteksi berita hoaks pada berita *online* berbahasa Indonesia dan menjadi dasar pengembangan sistem deteksi hoaks otomatis di masa mendatang.

**Kata Kunci:** berita hoaks, *Support Vector Machine*, klasifikasi teks, TF-IDF, berita *online*

Skripsi: xiii + 57 halaman, 13 pustaka, tahun pustaka: 2020–2025

## 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.

## DAFTAR ISI

LEMBAR PERNYATAAN .....	i
LEMBAR PERSETUJUAN .....	ii
LEMBAR PENGESAHAN .....	iii
ABSTRACT .....	iv
ABSTRAK .....	v
KATA PENGANTAR .....	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR .....	xi
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> .....	14
2.5	<i>Term Frequency-Inverse Document Frequency (TF-IDF)</i> .....	16
2.6	<i>Support Vector Machine (SVM)</i> .....	16
2.6.1	Tahapan Kerja SVM .....	18
2.7	Pengujian Model .....	19
2.7.1	Pembagian <i>Dataset</i> ( <i>Data Splitting</i> ) .....	19
2.8	Metrik Evaluasi Model.....	20
	BAB III METODOLOGI PENELITIAN.....	23
3.1	Analisa Kebutuhan .....	23
3.2	Kerangka Kerja Penelitian .....	23
3.2.1.	Pengumpulan Data .....	24
3.2.2.	Pra-Pemrosesan Data ( <i>Data Preprocessing</i> ).....	25
3.2.3.	Ekstraksi Fitur (TF-IDF) .....	28
3.2.4.	Penyeimbangan Data.....	29
3.2.5.	Pembagian <i>Dataset</i> .....	30
3.2.6.	Pemodelan dengan SVM.....	30
3.2.7.	Pengujian dan Evaluasi Model.....	31
	BAB IV HASIL DAN PEMBAHASAN .....	32
4.1	Hasil .....	32
4.2	<i>Dataset</i> .....	32
4.2.1	Sumber dan Struktur <i>Dataset</i> .....	32
4.2.2	Penggabungan <i>Dataset</i> .....	34
4.2.3	Penyesuaian dan Pembersihan <i>Dataset</i> .....	35
4.3	Pra-Pemrosesan Data .....	38

4.3.1.	Hasil Pra-Pemrosesan Teks.....	38
4.4	Ekstrasi Fitur (TF-IDF).....	39
4.4.1.	Konversi Data Teks ke Bentuk Numerik .....	39
4.5	Penyeimbangan Data.....	40
4.5.1	Analisis Distribusi Label Sebelum Penyeimbangan .....	41
4.5.2	Proses Penyeimbangan Data .....	41
4.5.3	Hasil Penyeimbangan Data .....	42
4.6	Pembagian <i>Dataset</i> .....	42
4.7	Pemodelan dengan <i>Support Vector Machine</i> (SVM).....	43
4.7.1	Konfigurasi Model dan Parameter Eksperimen .....	44
4.7.2	Proses Pelatihan Model .....	45
4.7.3	Proses Pengujian Model.....	45
4.8	Hasil Evaluasi Model .....	46
4.8.1.	<i>Confusion Matrix Model</i> .....	46
4.8.2	Perhitungan Metrik Evaluasi Model .....	48
4.8.3	Hasil Evaluasi Model SVM .....	51
4.8.4.	Analisis Hasil Evaluasi .....	52
4.8.5.	Perbandingan Performa.....	53
BAB V	PENUTUP.....	56
5.1	Kesimpulan .....	56
5.2	Saran.....	57
	DAFTAR PUSTAKA .....	57

## DAFTAR GAMBAR

<b>Gambar 2.1</b> Jenis-jenis Hoaks .....	12
<b>Gambar 2.2</b> Diagram alir proses NLP <i>preprocessing</i> .....	15
<b>Gambar 2.3</b> <i>Hyperlane</i> SVM dengan <i>margin</i> dan <i>support vectors</i> .....	17
<b>Gambar 2.4</b> Rumus akurasi .....	21
<b>Gambar 2.5</b> Rumus <i>Precision</i> .....	21
<b>Gambar 2.6</b> Rumus <i>Recall</i> .....	22
<b>Gambar 3.1</b> Kerangka Kerja Penelitian.....	24
<b>Gambar 3.2</b> <i>Code Python Case Folding</i> .....	26
<b>Gambar 3.3</b> <i>Code Python</i> Pembersihan Karakter.....	27
<b>Gambar 3.4</b> <i>Code Python</i> Tokenisasi .....	27
<b>Gambar 3.5</b> <i>Code Python</i> Tokenisasi .....	28
<b>Gambar 3.6</b> <i>Code Python</i> Penggabungan Teks .....	28
<b>Gambar 3.7</b> <i>Code Python</i> TF-IDF .....	29
<b>Gambar 4.1</b> Proses penggabungan <i>dataset</i> dari berbagai sumber .....	34
<b>Gambar 4.2</b> Proses Penghapusan nilai kosong .....	36
<b>Gambar 4.3</b> Fungsi penghapusan duplikat .....	36
<b>Gambar 4.4</b> Fungsi Validasi Nilai Label .....	37
<b>Gambar 4.5</b> Hasil Pemuatan dan Pembersihan <i>Dataset</i> .....	38
<b>Gambar 4.6</b> Hasil Pra-Pemrosesan .....	39
<b>Gambar 4.7</b> Hasil Ekstraksi Fitur TF-IDF .....	40
<b>Gambar 4.8</b> Distribusi Label Setelah Penyeimbangan Data.....	42
<b>Gambar 4.9</b> Hasil Pembagian <i>Dataset</i> Menjadi Data Pelatihan dan Data Pengujian.....	43
<b>Gambar 4.10</b> Proses Pelatihan Model SVM Menggunakan Data Pelatihan.....	45
<b>Gambar 4.11</b> Proses Prediksi Label Menggunakan Model SVM.....	45
<b>Gambar 4.12</b> <i>Confusion Matrix</i> Model SVM <i>Kernel RBF</i> .....	47
<b>Gambar 4.13</b> <i>Confusion Matrix</i> Model SVM <i>Kernel Linear</i> .....	47
<b>Gambar 4.14</b> Grafik Perbandingan Akurasi dan Presisi Model SVM.....	54

**Gambar 4.15** Grafik Perbandingan *Recall* dan *F1-Score* Model SVM..... 54

## DAFTAR TABEL

<b>Tabel 2.1</b> Karakteristik Hoaks .....	13
<b>Tabel 2.2</b> <i>Confusion Matrix</i> .....	20
<b>Tabel 3.1</b> Spesifikasi Perangkat Keras dan Perangkat Lunak .....	23
<b>Tabel 3.2</b> Contoh <i>Case Folding</i> .....	26
<b>Tabel 3.3</b> Pembersihan Karakter .....	26
<b>Tabel 3.4</b> Tokenisasi.....	27
<b>Tabel 3.5</b> Penghapusan <i>Stopword</i> .....	28
<b>Tabel 3.6</b> Penggabungan Teks.....	28
<b>Tabel 4.1</b> Sumber <i>Dataset</i> .....	33
<b>Tabel 4.2</b> Penyesuaian nama kolom .....	35
<b>Tabel 4.5</b> Distribusi Label <i>Dataset</i> Awal Sebelum Penyeimbangan .....	41
<b>Tabel 4.8</b> Konfigurasi Model SVM.....	44
<b>Tabel 4.9</b> Hasil Evaluasi Model SVM <i>Kernel Linear</i> dan RBF.....	52

# **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.

Dalam penelitian ini digunakan *dataset* berita *online* berbahasa Indonesia sebanyak 3.000 data, yang terdiri atas 2.000 data latih (*training data*) dan 1.000 data uji (*testing data*), mencakup berita hoaks dan berita fakta yang bersumber dari portal pemeriksa fakta dan media berita nasional terpercaya. Diharapkan, hasil penelitian ini dapat berkontribusi dalam upaya mitigasi penyebaran disinformasi di ruang digital Indonesia.

## 1.2 Identifikasi Masalah

Berdasarkan uraian latar belakang, dapat diidentifikasi beberapa permasalahan sebagai berikut:

1. Berita hoaks menimbulkan dampak negatif terhadap stabilitas sosial, politik, dan kepercayaan publik, sehingga diperlukan upaya deteksi hoaks secara otomatis dan akurat pada berita *online* berbahasa Indonesia.
2. Algoritma *Support Vector Machine* (SVM) berpotensi menjadi metode klasifikasi teks yang akurat untuk deteksi hoaks, namun kinerjanya perlu diuji secara empiris menggunakan *dataset* berita *online* berbahasa Indonesia dengan pembagian data latih dan data uji yang jelas.

## 1.3 Rumusan Masalah

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

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 3.000 berita *online* berbahasa Indonesia, yang terdiri atas 2.000 data latih (*training data*) dan 1.000 data uji (*testing data*), dikumpulkan dari portal pemeriksa fakta seperti Turnbackhoax.id (label hoaks) serta 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 yang ingin dicapai 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*) sesuai dengan pembagian yang telah ditetapkan, 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 mendekripsi berita hoaks.

## 1.8 Sistematika Penulisan

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

<b>BAB I</b>	<b>PENDAHULUAN</b> 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.
<b>BAB II</b>	<b>LANDASAN TEORI</b> Bab ini membahas teori-teori dasar yang relevan dan menjadi acuan dalam penelitian, mencakup konsep berita hoaks, <i>Natural Language Processing</i> (NLP), metode ekstraksi fitur <i>Term Frequency-Inverse Document Frequency</i> (TF-IDF), algoritma <i>Support Vector Machine</i> (SVM), serta metrik evaluasi model.
<b>BAB III</b>	<b>METODOLOGI PENELITIAN</b> 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.
<b>BAB IV</b>	<b>HASIL DAN PEMBAHASAN</b> Bab ini menyajikan hasil dari implementasi dan pengujian model yang telah dibangun. Pembahasan akan mencakup karakteristik <i>Dataset</i> yang digunakan, hasil pra-pemrosesan, hasil pelatihan model, serta hasil pengujian performa model yang diukur dengan <i>accuracy</i> , <i>precision</i> , dan <i>recall</i> .

<b>BAB V</b>	<b>PENUTUP</b> 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. **“PENERAPAN ALGORITMA SUPPORT VECTOR MACHINE (SVM) UNTUK KLASIFIKASI BERITA HOAKS COVID-19”** (Ropikoh et al., 2021) Penelitian ini mengkaji penyebaran berita hoaks terkait pandemi COVID-19 yang meningkat signifikan di Indonesia. Algoritma *Support Vector Machine* (SVM) dengan *kernel linear* dan *Radial Basis Function* (RBF) digunakan untuk mengklasifikasikan berita hoaks dan non-hoaks. *Dataset* yang digunakan berjumlah 8.172 data yang bersumber dari Kompas (non-hoaks), Turnbackhoax, dan Hoaxbuster (hoaks). Metodologi penelitian mengacu pada pendekatan *Knowledge Discovery in Database* (KDD) yang meliputi tahapan *text mining* (*case folding*, *tokenizing*, *filtering*, dan *stemming*), transformasi fitur menggunakan TF-IDF, serta klasifikasi menggunakan SVM dengan empat skenario pembagian data latih dan data uji. Hasil penelitian menunjukkan bahwa *kernel linear* pada skenario pembagian data 80:20 menghasilkan akurasi tertinggi sebesar 92,90%, sedangkan *kernel RBF* pada skenario 90:10 menghasilkan akurasi sebesar 90,46%.
2. **“DETEKSI BERITA HOAKS PADA WEBSITE TURNBACKHOAX DENGAN MENGGUNAKAN MACHINE LEARNING”** (Rahmawati, 2021) Penelitian ini berfokus pada klasifikasi berita hoaks yang bersumber dari situs Turnbackhoax. Algoritma yang digunakan dalam penelitian ini meliputi *Support Vector Machine* (SVM), *Random Forest*, dan *Logistic Regression*. Tahapan penelitian mencakup *text preprocessing*, pembobotan kata menggunakan TF-IDF, penyeimbangan data menggunakan SMOTE,

- serta optimasi parameter dengan metode *Grid Search Cross Validation*. Hasil evaluasi menunjukkan bahwa algoritma SVM memiliki performa terbaik dibandingkan algoritma lainnya, dengan tingkat akurasi sebesar 83,3% dan nilai *recall* mencapai 99% pada kelas hoaks.
3. **“DETEKSI BERITA HOAKS DARI MEDIA *ONLINE* INDONESIA MENGGUNAKAN ALGORITMA *NAÏVE BAYES* DAN *SUPPORT VECTOR MACHINE*”** (Febriyanty Nur Elyta, 2023) Penelitian ini membandingkan performa algoritma *Naïve Bayes* dan *Support Vector Machine* (SVM) dalam mendekripsi berita hoaks dari media *online* Indonesia. *Dataset* yang digunakan terdiri atas 2.000 judul berita berbahasa Indonesia. Proses penelitian meliputi tahapan *text preprocessing* seperti *case folding*, *stemming*, *stopword removal*, dan *tokenizing*, kemudian dilanjutkan dengan pembobotan fitur menggunakan TF-IDF serta evaluasi model menggunakan *Confusion Matrix*. Hasil penelitian menunjukkan bahwa algoritma *Naïve Bayes* memperoleh akurasi sebesar 93%, sedangkan algoritma SVM mencapai akurasi sebesar 100%.
  4. **“*SUPPORT VECTOR MACHINE* UNTUK IDENTIFIKASI BERITA HOAKS TERKAIT VIRUS CORONA (COVID-19)”** (Putri & Athoillah, 2021) Penelitian ini membangun sistem otomatis untuk identifikasi berita hoaks terkait COVID-19 menggunakan algoritma *Support Vector Machine* (SVM) dengan *kernel linear*. Proses evaluasi model dilakukan menggunakan metode *k-fold cross validation*. Hasil pengujian menunjukkan nilai rata-rata *precision* sebesar 78,96%, *recall* sebesar 78,18%, dan *F-measure* sebesar 78,02%. Hasil terbaik diperoleh pada percobaan ketiga dengan nilai *precision* sebesar 89,37% dan *recall* sebesar 88,64%.
  5. **“PENERAPAN METODE SVM DAN *RANDOM FOREST* UNTUK MENDETEKSI BERITA HOAKS PADA PT GLOBAL ARROW”** (Rizky Purwanto Fernandes & Rizky Tahara Shita, 2024) Penelitian ini meneliti deteksi berita hoaks pada domain politik dan kesehatan dengan mengombinasikan algoritma *Support Vector Machine* (SVM) dan *Random Forest*. *Dataset* diperoleh melalui proses *web crawling* dari beberapa sumber berita seperti Kompas, Kominfo, dan Antara News, kemudian

dilakukan pembobotan fitur menggunakan TF-IDF. Hasil awal penelitian menunjukkan tingkat akurasi di atas 90%. Namun, pada pengujian lanjutan, akurasi model menurun hingga 55,14%, sehingga disimpulkan bahwa diperlukan proses optimasi lebih lanjut agar sistem klasifikasi dapat bekerja secara lebih andal.

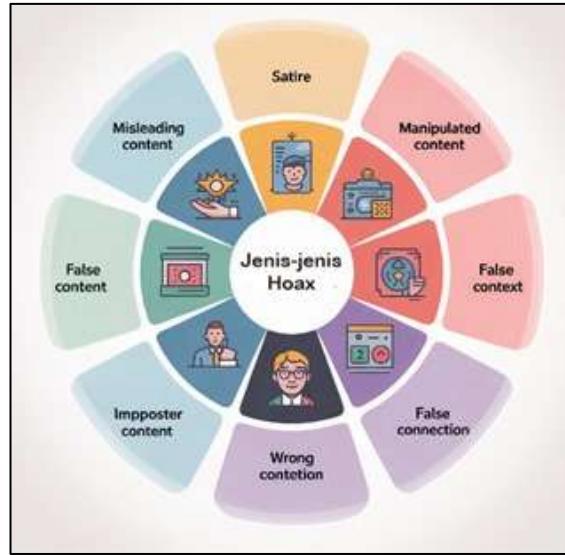
## 2.2 Hoaks

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:

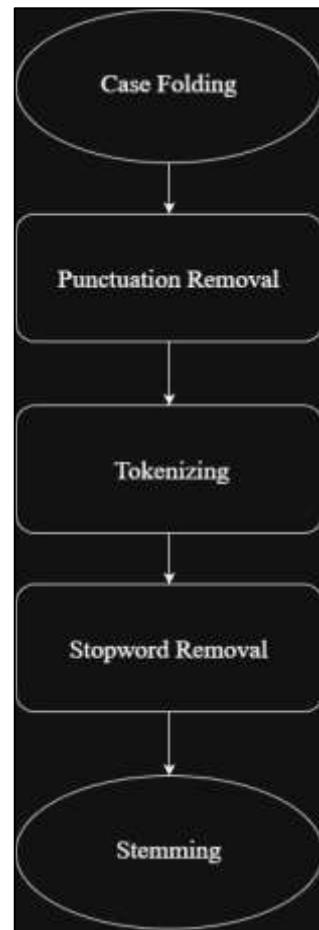
1. *Supervised Learning* (Pembelajaran Terarah): Ini adalah pendekatan yang digunakan dalam penelitian ini. Dalam *supervised learning*, model "diajari" menggunakan *Dataset* yang sudah memiliki label yang benar. Misalnya, untuk mendeteksi hoaks, model diberi ribuan contoh berita yang sudah diberi label "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 berimbuhan menjadi bentuk kata dasarnya (misalnya "menyebarluaskan" 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).

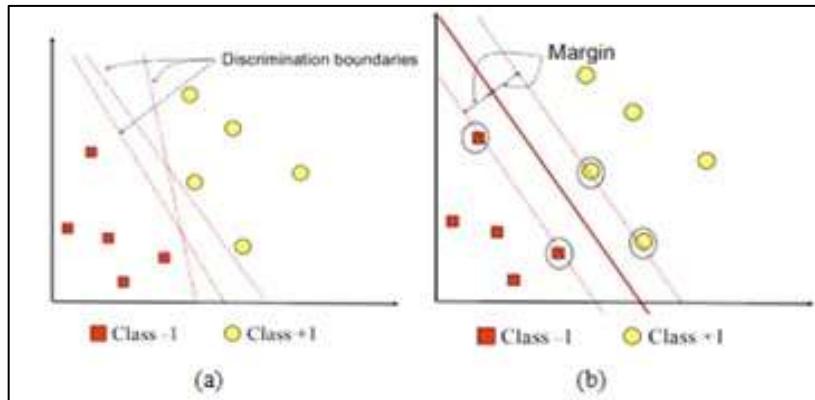
TF-IDF bekerja dengan mengukur tingkat kepentingan sebuah kata dalam sebuah dokumen terhadap keseluruhan kumpulan dokumen (korpus). Bobot TF-IDF dihitung berdasarkan dua komponen (DickiPrabowo et al., 2025):

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 *inherent* 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 dataset umumnya dilakukan dengan rasio tertentu antara data latih dan data uji. Pada penelitian ini, dataset dibagi menjadi 2.000 data latih (*training data*) dan 1.000 data uji (*testing data*) dari total 3.000 data yang digunakan. Hasil prediksi model pada data uji selanjutnya dianalisis

menggunakan *Confusion Matrix* dan berbagai metrik evaluasi untuk menilai kinerja model klasifikasi.

## 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:

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

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

**Gambar 2.4** Rumus akurasi

- a. TP (*True Positive*) → Jumlah data positif yang berhasil diklasifikasikan dengan benar sebagai positif.
  - b. TN (*True Negative*) → Jumlah data negatif yang diklasifikasikan dengan benar sebagai negatif.
  - c. FP (*False Positive*) → Jumlah data negatif yang salah diklasifikasikan sebagai positif.
  - d. FN (*False Negative*) → Jumlah data positif yang salah diklasifikasikan sebagai negatif.
2. *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.

$$Precision = \frac{TP}{TP + FP}$$

**Gambar 2.5** Rumus *Precision*

- a. TP (*True Positive*) → Jumlah data positif yang diklasifikasikan dengan benar sebagai positif.
  - b. FP (*False Positive*) → Jumlah data negatif yang salah diklasifikasikan sebagai positif.
3. *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*.

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

**Gambar 2.6** Rumus Recall

- a. TP (*True Positive*) → jumlah data positif yang berhasil diprediksi benar sebagai positif.
- b. 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 *system* yang berkaitan dengan pengembangan implementasi algoritma *Support Vector Machine* untuk deteksi hoaks pada berita *online* dengan menggunakan bahasa pemrograman Python.

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

**Tabel 3.1** Spesifikasi Perangkat Keras dan Perangkat Lunak

Perangkat Keras	Spesifikasi
<i>Processor</i>	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>

#### **3.2 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** Kerangka Kerja Penelitian

### 3.2.1. Pengumpulan Data

Tahap pengumpulan data merupakan tahap awal yang sangat penting dalam penelitian ini karena kualitas dan kuantitas data berpengaruh langsung terhadap kinerja model klasifikasi yang dibangun. *Dataset* yang digunakan dalam penelitian ini terdiri atas dua kelas, yaitu berita hoaks dan berita fakta.

1. Data Berita Hoaks

Data berita hoaks diperoleh dari situs web pemeriksa fakta TurnBackHoax.id yang dikelola oleh Masyarakat Anti Fitnah Indonesia (MAFINDO). Situs ini menyediakan arsip berita yang telah diverifikasi dan diklasifikasikan sebagai hoaks, lengkap dengan narasi berita serta klarifikasi faktanya. Data yang diambil berupa teks berita yang memuat judul dan isi utama berita.

## 2. Data Berita Fakta

Data berita fakta diperoleh dari beberapa portal berita *online* nasional yang telah terverifikasi oleh Dewan Pers dan memiliki reputasi tinggi, yaitu Kompas.com dan Detik.com. Artikel berita yang diambil merupakan berita faktual yang dipublikasikan secara resmi oleh media tersebut.

Proses pengumpulan data dilakukan dengan menghimpun artikel berita *online* berbahasa Indonesia dari masing-masing sumber, kemudian dilakukan pelabelan kelas sesuai dengan kategori hoaks atau fakta berdasarkan sumber asalnya. Total *dataset* yang digunakan dalam penelitian ini berjumlah 3.000 artikel berita, yang terdiri atas 2.000 data latih (*training data*) dan 1.000 data uji (*testing data*), dengan komposisi kelas yang relatif seimbang antara berita hoaks dan berita fakta. Pembagian ini dilakukan untuk menghindari ketidakseimbangan kelas (*class imbalance*) yang dapat memengaruhi kinerja model klasifikasi.

### 3.2.2. Pra-Pemrosesan Data (*Data Preprocessing*)

Tahap pra-pemrosesan data merupakan tahapan awal dalam pemrosesan teks yang bertujuan untuk menyiapkan data agar dapat diolah secara optimal oleh algoritma *machine learning*. Pada penelitian ini, pra-pemrosesan dilakukan untuk membersihkan dan menstandarkan teks berita sehingga informasi yang tidak relevan dapat dihilangkan sebelum dilakukan ekstraksi fitur dan pemodelan.

Tahapan pra-pemrosesan data yang diterapkan dalam penelitian ini meliputi beberapa langkah sebagai berikut.

### 3.2.2.1. 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 3.2** Contoh *Case Folding*

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



**Gambar 3.2** Code Python *Case Folding*

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

### 3.2.2.2. 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 3.3** Pembersihan Karakter

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



```
1 text = re.sub(r'[^a-zA-Z\s]', '', text)
```

**Gambar 3.3** *Code Python* Pembersihan Karakter

### 3.2.2.3. 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 3.4** Tokenisasi

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



```
1 tokens = word_tokenize(text)
```

**Gambar 3.4** *Code Python* Tokenisasi

### 3.2.2.4. 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 3.5 Penghapusan Stopword**

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

```

1 tokens = [w for w in tokens if w not in stop_words and len(w) > 2]

```

**Gambar 3.5 Code Python Tokenisasi**

### 3.2.2.5. 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 3.6 Penggabungan Teks**

Sebelum	['berita', 'hoaks', 'jangan', 'sebar']
Sesudah	"berita hoaks jangan sebar"

```

1 return ' '.join(tokens)

```

**Gambar 3.6 Code Python Penggabungan Teks**

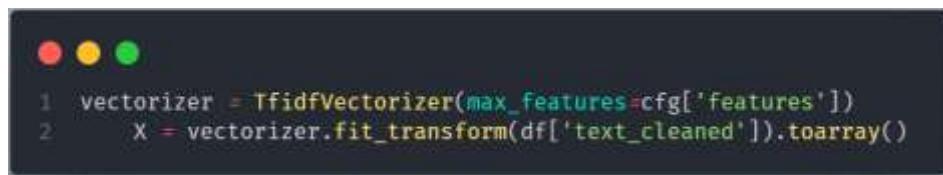
### 3.2.3. Ekstraksi Fitur (TF-IDF)

Setelah proses pra-pemrosesan selesai, data teks yang telah dibersihkan perlu diubah ke dalam bentuk numerik agar dapat diproses oleh algoritma *machine learning*. Pada penelitian ini, proses ekstraksi fitur dilakukan menggunakan metode *Term Frequency–Inverse Document*

*Frequency* (TF-IDF) untuk merepresentasikan teks berita dalam bentuk vektor numerik.

Metode TF-IDF bekerja dengan memberikan bobot pada setiap kata berdasarkan frekuensi kemunculannya dalam suatu dokumen (*term frequency*) serta tingkat keunikannya terhadap seluruh dokumen dalam dataset (*inverse document frequency*). Dengan pendekatan ini, kata-kata yang sering muncul namun tidak bersifat umum akan memiliki bobot yang lebih tinggi, sehingga mampu merepresentasikan karakteristik dokumen secara lebih informatif.

Ekstraksi fitur TF-IDF diterapkan pada data teks hasil prapemrosesan yang tersimpan dalam kolom `text_cleaned`. Hasil dari proses ini berupa matriks vektor numerik yang selanjutnya digunakan sebagai input pada tahap pemodelan menggunakan algoritma *Support Vector Machine* (SVM). Penggunaan metode TF-IDF dipilih karena telah terbukti efektif dalam penelitian terdahulu untuk membedakan karakteristik teks antara berita hoaks dan berita fakta (DickiPrabowo et al., 2025).



```

1 vectorizer = TfidfVectorizer(max_features=cfg['features'])
2 X = vectorizer.fit_transform(df['text_cleaned']).toarray()

```

Gambar 3.7 Code Python TF-IDF

### 3.2.4. Penyeimbangan Data

Penyeimbangan data dilakukan untuk mengatasi ketidakseimbangan jumlah data pada masing-masing kelas dalam *dataset*. Ketidakseimbangan data dapat menyebabkan model cenderung memprediksi kelas mayoritas dan menurunkan kemampuan model dalam mengenali kelas minoritas.

Pada penelitian ini, penyeimbangan data dilakukan pada *dataset* hasil ekstraksi fitur dan sebelum pembagian data latih dan data uji. Tahapan ini bertujuan untuk menghasilkan distribusi kelas yang lebih seimbang sehingga proses pelatihan model dapat berjalan secara optimal.

### 3.2.5. Pembagian Dataset

*Dataset* hasil ekstraksi fitur yang telah direpresentasikan dalam bentuk vektor numerik selanjutnya dibagi menjadi dua bagian, yaitu data latih (*training set*) dan data uji (*testing set*). Pembagian *dataset* ini bertujuan untuk melatih model klasifikasi serta mengevaluasi kinerjanya secara objektif menggunakan data yang belum pernah dilihat sebelumnya.

Pada penelitian ini, dataset dibagi menjadi 2.000 data latih (*training data*) dan 1.000 data uji (*testing data*) dari total 3.000 data yang digunakan. Data latih digunakan dalam proses pelatihan model *Support Vector Machine* (SVM), sedangkan data uji digunakan untuk mengukur performa model melalui proses pengujian dan evaluasi.

Pembagian *dataset* dilakukan secara acak untuk memastikan bahwa data latih dan data uji memiliki distribusi kelas yang seimbang antara berita hoaks dan berita fakta, sehingga model yang dihasilkan tidak mengalami bias terhadap salah satu kelas.

### 3.2.6. Pemodelan dengan SVM

Pemodelan dilakukan dengan membangun model klasifikasi berita menggunakan algoritma *Support Vector Machine* (SVM). Pada penelitian ini, implementasi algoritma SVM menggunakan pustaka *Scikit-learn* dengan modul *Support Vector Classifier* (SVC). Data latih yang telah direpresentasikan dalam bentuk vektor numerik hasil ekstraksi fitur TF-IDF, beserta label kelasnya (hoaks dan fakta), digunakan sebagai masukan dalam proses pelatihan model.

Pada tahap pelatihan, model SVM mempelajari pola dan karakteristik data latih untuk menentukan *hyperplane* pemisah yang optimal, sehingga mampu membedakan antara kelas berita hoaks dan berita fakta secara maksimal. Model yang dihasilkan dari proses ini selanjutnya digunakan pada tahap pengujian untuk mengevaluasi kinerjanya dalam mendeteksi berita hoaks menggunakan data uji.

### 3.2.7. Pengujian dan Evaluasi Model

Model *Support Vector Machine* (SVM) yang telah dilatih selanjutnya diuji menggunakan data uji (*testing data*) yang tidak digunakan selama proses pelatihan. Tahap pengujian ini bertujuan untuk mengetahui kemampuan model dalam mengklasifikasikan berita hoaks dan berita fakta pada data yang belum pernah dilihat sebelumnya.

Hasil prediksi model terhadap data uji dibandingkan dengan label asli untuk membentuk *Confusion Matrix*. *Confusion Matrix* digunakan untuk menggambarkan jumlah prediksi yang benar dan salah yang dihasilkan oleh model, sehingga dapat memberikan gambaran menyeluruh mengenai kinerja model klasifikasi.

Berdasarkan *Confusion Matrix* tersebut, kinerja model dievaluasi menggunakan beberapa metrik evaluasi, yaitu *accuracy*, *precision*, dan *recall*. *Accuracy* digunakan untuk mengukur tingkat ketepatan klasifikasi secara keseluruhan, *precision* digunakan untuk mengukur ketepatan model dalam memprediksi kelas berita hoaks, sedangkan *recall* digunakan untuk mengukur kemampuan model dalam mendeteksi seluruh berita hoaks yang ada dalam data uji.

Hasil pengujian dan perhitungan metrik evaluasi ini selanjutnya disajikan dan dibahas secara rinci pada BAB IV sebagai dasar dalam menilai keberhasilan model klasifikasi yang dibangun pada penelitian ini.

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Hasil**

Bab ini menyajikan hasil penerapan algoritma *Support Vector Machine* (SVM) dalam mendeteksi berita hoaks berdasarkan teks berita *online* berbahasa Indonesia. *Dataset* yang digunakan merupakan gabungan dari beberapa sumber data, yaitu TurnBackHoax.id sebagai sumber berita hoaks serta Kompas.com dan Detik.com sebagai sumber berita fakta, dengan total data sebanyak 3.000 artikel berita.

Pada bab ini ditampilkan hasil pengolahan *dataset*, hasil pra-pemrosesan teks, penyeimbangan data, serta hasil pelatihan dan pengujian model SVM. Selain itu, bab ini juga menyajikan hasil evaluasi kinerja model menggunakan beberapa metrik evaluasi, yaitu *accuracy*, *precision*, dan *recall*, yang diperoleh dari pengujian terhadap data uji. Seluruh hasil tersebut selanjutnya dianalisis dan dibahas untuk menilai efektivitas model dalam mendeteksi berita hoaks.

#### **4.2 Dataset**

*Dataset* yang digunakan dalam penelitian ini merupakan *dataset* hasil penggabungan dari beberapa sumber data berita *online* berbahasa Indonesia. Proses pengolahan *dataset* pada tahap ini bertujuan untuk menghasilkan data yang siap digunakan pada tahap penyeimbangan data, pelatihan, dan pengujian model klasifikasi.

##### **4.2.1 Sumber dan Struktur Dataset**

*Dataset* yang digunakan dalam penelitian ini terdiri atas empat berkas utama dengan format *Comma-Separated Values* (CSV), yang diperoleh dari beberapa sumber berita *online* berbahasa Indonesia. Rincian sumber *dataset* yang digunakan ditunjukkan pada Tabel 4.1.

**Tabel 4.1** Sumber *Dataset*

No	Nama File	Jumlah Data (Baris)	Jenis Data	Keterangan
1	DataBerita.csv	614	Campuran	Berita umum (hoaks dan fakta)
2	turnbackhoax.csv	3.071	Hoaks	<i>Dataset</i> hasil verifikasi TurnBackHoax.id
3	berita.csv	541	Fakta	Berita aktual dari portal berita terpercaya
4	DataDetik.csv	4.945	Fakta	Artikel berita dari situs Detik.com

Berkas DataBerita.csv berisi berita umum yang dikumpulkan dari berbagai sumber dan mencakup berita hoaks maupun berita fakta tanpa pemisahan kelas secara langsung. *Dataset* turnbackhoax.csv merupakan kumpulan berita hoaks yang bersumber dari TurnBackHoax.id, yaitu situs pemeriksa fakta resmi di Indonesia yang dikelola oleh Masyarakat Anti Fitnah Indonesia (MAFINDO).

*Dataset* berita.csv dan DataDetik.csv berisi berita faktual yang berasal dari portal berita nasional terpercaya. *Dataset* berita.csv mencakup berita dari beberapa media daring seperti Kompas.com, CNN Indonesia, dan Liputan6, sedangkan DataDetik.csv secara khusus berisi artikel berita yang diperoleh dari Detik.com.

Seluruh *dataset* yang digunakan memuat teks berita dalam bahasa Indonesia, namun memiliki variasi struktur kolom. Sebagian *dataset* menggunakan kolom *Content*, sebagian menggunakan *Title*, dan sebagian lainnya telah memiliki kolom *teks\_berita*. Oleh karena itu, dilakukan penyeragaman struktur *dataset* agar seluruh data memiliki skema yang konsisten.

Setelah proses penyesuaian struktur dilakukan, setiap *dataset* disusun ke dalam dua kolom utama, yaitu:

1. *teks\_berita*, yang berisi teks berita atau judul berita yang digunakan sebagai masukan utama model, dan
2. *label*, yang menunjukkan kategori kelas berita, yaitu hoaks atau fakta.

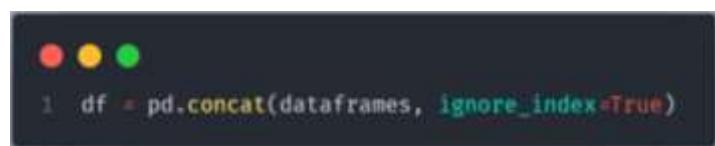
Berdasarkan hasil penggabungan seluruh *dataset* dari masing-masing sumber, diperoleh total data awal sebanyak 9.171 data. Selanjutnya dilakukan proses penyesuaian dan pembersihan data, termasuk penghapusan data duplikat dan data yang tidak memenuhi kriteria penelitian.

#### **4.2.2 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.



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

**Gambar 4.1** Proses penggabungan *dataset* dari berbagai sumber

1. `pd.concat()` fungsi dari pandas untuk *menggabungkan* beberapa objek seperti *DataFrame* atau *Series*.

2. *dataframes* biasanya berupa *list* yang berisi beberapa *DataFrame* yang ingin digabungkan.
3. *ignore\_index=True* mengatur agar indeks dari hasil gabungan di-reset (tidak mempertahankan indeks lama dari masing-masing *DataFrame*).
4. *df* variabel hasil gabungan dari semua *DataFrame* dalam list *dataframes*.

#### 4.2.3 Penyesuaian dan Pembersihan *Dataset*

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
<i>Content</i>	teks_berita	Isi berita utama
<i>Title</i>	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']).drop_duplicates(subset='teks_berita')
```

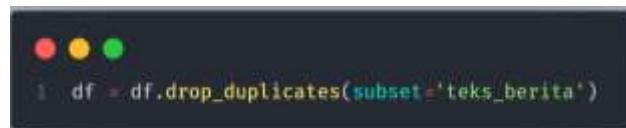
**Gambar 4.2** Proses Penghapusan nilai kosong

- 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.



```
df = df.drop_duplicates(subset='teks_berita')
```

**Gambar 4.3** Fungsi penghapusan duplikat

1) `drop_duplicates()`

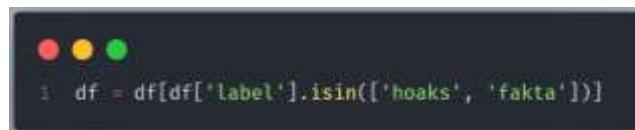
digunakan untuk menghapus baris-baris data yang memiliki isi kolom yang sama (duplikat) di dalam sebuah *DataFrame*.

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

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 seluruh tahapan pembersihan dan penyesuaian struktur *dataset* dilakukan, diperoleh *dataset* hasil pembersihan yang siap digunakan pada tahap analisis selanjutnya. Proses pembersihan ini dilakukan secara otomatis menggunakan bahasa pemrograman Python pada lingkungan Google Colaboratory.

Berdasarkan hasil eksekusi kode, *dataset* yang berhasil dimuat dan dibersihkan berasal dari empat berkas utama, yaitu `DataBerita.csv`, `turnbackhoax.csv`, `berita.csv`, dan `DataDetik.csv`. Jumlah data dari masing-masing berkas serta total data setelah proses pembersihan ditunjukkan pada gambar berikut.

```

--- 1. Memuat Dataset ---
✓ DataBerita.csv dimuat (614 data) | label unik: ['fakta' 'hoaks']
✓ turnbackhoax.csv dimuat (3071 data) | label unik: ['hoaks']
✓ berita.csv dimuat (541 data) | label unik: ['fakta']
✓ DataDetik.csv dimuat (4945 data) | label unik: ['fakta']
📊 Distribusi label gabungan:
label
fakta    4113
hoaks    3092
Name: count, dtype: int64

```

**Gambar 4.5** Hasil Pemuatan dan Pembersihan *Dataset*

Proses pembersihan data menghasilkan total 7.205 data yang telah memenuhi kriteria kelayakan data, dengan karakteristik sebagai berikut:

- Seluruh *dataset* telah diseragamkan ke dalam dua kolom utama, yaitu *teks\_berita* dan *label*.
- Tidak terdapat nilai kosong (*missing value*) pada kolom teks maupun *label*.
- Data duplikat berhasil diidentifikasi dan dihapus sehingga setiap baris data bersifat unik.
- Label* data telah tervalidasi dan hanya terdiri dari dua kelas, yaitu *hoaks* dan *fakta*.

Gambar 4.5 memperlihatkan ringkasan hasil pemuatan dan pembersihan *dataset*, termasuk jumlah data dari masing-masing sumber serta total data yang siap digunakan sebelum dilakukan tahap penyeimbangan data dan seleksi *dataset* penelitian.

### 4.3 Pra-Pemrosesan Data

Bagian ini menyajikan hasil dari proses pra-pemrosesan data teks yang telah dijelaskan pada BAB III. Pra-pemrosesan diterapkan pada *dataset* hasil pengolahan sebelumnya untuk menghasilkan data teks yang lebih bersih dan siap digunakan pada tahap ekstraksi fitur.

#### 4.3.1. Hasil Pra-Pemrosesan Teks

Hasil pra-pemrosesan teks ditunjukkan melalui perbandingan antara teks berita sebelum dan sesudah dilakukan pra-pemrosesan. Proses ini menghasilkan teks yang telah dibersihkan dari karakter non-alfabet,

diseragamkan ke dalam bentuk huruf kecil, serta dihilangkan kata-kata umum yang tidak memiliki kontribusi signifikan terhadap proses klasifikasi.

Gambar berikut menyajikan contoh perubahan teks berita setelah pra-pemrosesan dilakukan.

```
--- 2. Pra-pemrosesan ---
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tag to /root/nltk_data...
[nltk_data]   Package punkt_tag is already up-to-date!

★ Contoh 1 data.preprocessing:
❶ Sebelum : JEMBER, KOPPIAS.com -Dinas Kesehatan (Dinkes) Kabupaten Jember, Jawa Timur, mengungkap hasil uji laboratorium terhadap makakan
❷ Sesudah : Jember kumpas.com dinas kesehatan dinkes kabupaten jember jawa timur mengungkap hasil uji laboratorium makakan bersilai gratis
```

**Gambar 4.6 Hasil Pra-Pemrosesan**

Berdasarkan hasil pra-pemrosesan tersebut, teks berita menjadi lebih ringkas dan konsisten karena penghapusan karakter non-alfabet dan kata-kata umum. Proses ini mengurangi variasi kata yang tidak relevan serta mempersiapkan data teks agar lebih efektif pada tahap ekstraksi fitur menggunakan TF-IDF.

#### 4.4 Ekstrasi Fitur (TF-IDF)

Setelah data teks melalui tahap pra-pemrosesan, diperlukan suatu metode untuk mengubah data teks tersebut ke dalam bentuk numerik agar dapat diproses oleh algoritma *machine learning*. Tahap ini dikenal sebagai ekstraksi fitur, yang bertujuan untuk merepresentasikan teks berita dalam bentuk vektor angka yang mencerminkan karakteristik penting dari setiap dokumen.

Pada penelitian ini, metode *Term Frequency–Inverse Document Frequency* (TF-IDF) digunakan sebagai teknik ekstraksi fitur. TF-IDF dipilih karena mampu memberikan bobot yang lebih tinggi pada kata-kata yang bersifat informatif dan jarang muncul pada keseluruhan dokumen, sehingga representasi fitur yang dihasilkan lebih efektif dalam membedakan berita hoaks dan berita fakta.

##### 4.4.1. Konversi Data Teks ke Bentuk Numerik

Setelah tahap pra-pemrosesan selesai, data teks yang telah dibersihkan perlu dikonversi ke dalam bentuk numerik agar dapat diproses oleh algoritma *Support Vector Machine* (SVM). Proses konversi ini

dilakukan menggunakan metode *Term Frequency–Inverse Document Frequency* (TF-IDF).

TF-IDF bekerja dengan memberikan bobot pada setiap kata berdasarkan tingkat kepentingannya di dalam suatu dokumen dan keseluruhan kumpulan dokumen (*corpus*). Kata yang sering muncul dalam satu dokumen tetapi jarang muncul pada dokumen lain akan memiliki bobot yang lebih tinggi, sehingga dianggap lebih representatif dalam membedakan kelas berita hoaks dan fakta.

Pada penelitian ini, proses ekstraksi fitur TF-IDF diterapkan menggunakan pustaka *scikit-learn* dengan parameter `max_features = 1000`, yang berarti hanya 1.000 kata paling relevan yang dipilih sebagai fitur. Proses ini menghasilkan matriks fitur numerik dari 7.205 data teks berita, yang selanjutnya digunakan pada tahap penyeimbangan data dan pemodelan.

Hasil proses ekstraksi fitur TF-IDF ditunjukkan pada Gambar 4.7, yang memperlihatkan jumlah data yang berhasil dikonversi serta jumlah fitur yang digunakan.

```
--- 3. Ekstraksi Fitur (TF-IDF) ---
✓ TF-IDF selesai | Jumlah data: 7205 | Jumlah fitur: 1000
```

**Gambar 4.7** Hasil Ekstraksi Fitur TF-IDF

Gambar ini menampilkan hasil eksekusi proses ekstraksi fitur TF-IDF pada Google Colaboratory yang menunjukkan jumlah data teks yang diproses sebanyak 7.205 data dengan jumlah fitur yang digunakan sebesar 1.000 fitur.

## 4.5 Penyeimbangan Data

Tahap penyeimbangan data dilakukan untuk memastikan bahwa jumlah data pada masing-masing kelas, yaitu hoaks dan fakta, memiliki distribusi yang seimbang sebelum proses pelatihan model dilakukan. Keseimbangan distribusi

kelas merupakan aspek penting dalam pembelajaran *machine learning* karena ketimpangan data dapat menyebabkan model bias terhadap kelas mayoritas dan menurunkan kemampuan generalisasi terhadap kelas minoritas.

Pada penelitian ini, penyeimbangan data dilakukan melalui seleksi *dataset* penelitian, yaitu dengan memilih jumlah data yang sama untuk masing-masing kelas dari *dataset* awal hasil pengumpulan dan pembersihan data. Pendekatan ini dipilih karena *dataset* awal memiliki ukuran yang relatif besar, sehingga pemilihan subset data yang seimbang dianggap cukup representatif tanpa perlu melakukan teknik penyeimbangan sintetis seperti oversampling atau SMOTE.

#### **4.5.1 Analisis Distribusi Label Sebelum Penyeimbangan**

Setelah tahap pra-pemrosesan dan penggabungan *dataset* selesai, diperoleh dataset awal hasil pembersihan sebanyak 7.205 data berita. *Dataset* ini kemudian dianalisis untuk mengetahui distribusi label sebelum dilakukan proses penyeimbangan data.

Hasil analisis menunjukkan bahwa distribusi data antar kelas tidak seimbang, di mana jumlah data dengan label hoaks lebih banyak dibandingkan data dengan label fakta. Kondisi ketidakseimbangan ini berpotensi menyebabkan model lebih dominan mempelajari pola dari kelas mayoritas dan mengabaikan karakteristik kelas minoritas.

**Tabel 4.3** Distribusi Label *Dataset* Awal Sebelum Penyeimbangan

Label	Jumlah Data	Percentase
Hoaks	3092	42,91%
Fakta	4113	57,09%
<b>Total</b>	<b>7.205</b>	<b>100%</b>

#### **4.5.2 Proses Penyeimbangan Data**

Untuk mengatasi ketidakseimbangan distribusi kelas pada *dataset* awal, dilakukan proses penyeimbangan data melalui seleksi *dataset*

penelitian. Pada tahap ini, dipilih masing-masing 1.500 data berita hoaks dan 1.500 data berita fakta dari *dataset* awal.

Pemilihan data dilakukan secara acak menggunakan bahasa pemrograman Python untuk memastikan bahwa data yang dipilih bersifat representatif dan tidak menimbulkan bias tertentu. Proses ini tidak melibatkan pembuatan data sintetis maupun perubahan label, melainkan hanya pemilihan subset data yang seimbang dari *dataset* awal.

Dengan pendekatan ini, diperoleh *dataset* penelitian yang memiliki distribusi kelas seimbang dan siap digunakan pada tahap pelatihan dan pengujian model.

#### 4.5.3 Hasil Penyeimbangan Data

Setelah proses penyeimbangan data melalui seleksi *dataset* penelitian dilakukan, diperoleh *dataset* akhir yang memiliki distribusi kelas seimbang antara berita hoaks dan berita fakta, masing-masing sebanyak 1.500 data.

```
--- 4. Penyeimbangan Data (1500 Hoaks : 1500 Fakta) ---
✓ Balancing selesai.
📊 Distribusi label setelah balancing:
label_num
0    1500
1    1500
Name: count, dtype: int64
```

**Gambar 4.8** Distribusi Label Setelah Penyeimbangan Data

Gambar 4.8 menunjukkan hasil distribusi label setelah proses penyeimbangan data dilakukan, di mana jumlah data pada masing-masing kelas, yaitu hoaks dan fakta, telah seimbang dengan jumlah masing-masing sebanyak 1.500 data. Hasil ini diperoleh berdasarkan *output* eksekusi kode Python pada Google Colaboratory.

#### 4.6 Pembagian Dataset

Setelah diperoleh *dataset* akhir hasil penyeimbangan data sebanyak 3.000 data (1.500 berita hoaks dan 1.500 berita fakta) serta direpresentasikan dalam bentuk vektor numerik menggunakan TF-IDF, langkah selanjutnya adalah melakukan pembagian *dataset* menjadi data pelatihan dan data pengujian.

Pembagian *dataset* dilakukan untuk mengevaluasi kemampuan model dalam menggeneralisasi data yang belum pernah dilihat sebelumnya. Pada penelitian ini, *dataset* dibagi menjadi dua bagian sebagai berikut:

1. Data pelatihan (*training set*) sebanyak 2.000 data, yang digunakan untuk melatih model *Support Vector Machine* (SVM).
2. Data pengujian (*testing set*) sebanyak 1.000 data, yang digunakan untuk menguji performa model terhadap data yang belum pernah dilihat sebelumnya.

Pembagian data dilakukan menggunakan fungsi `train_test_split()` dari pustaka scikit-learn dengan parameter `test_size=1000` dan `stratify=y`. Penggunaan parameter `stratify` bertujuan untuk menjaga proporsi kelas hoaks dan fakta tetap seimbang pada data pelatihan maupun data pengujian.

Dengan pembagian ini, distribusi kelas pada kedua subset tetap representatif dan memungkinkan evaluasi performa model dilakukan secara objektif.

```
--- 5. Pembagian Dataset (2000 Train : 1000 Test) ---
✓ Data latih : 2000
✓ Data uji   : 1000
📊 Distribusi y_train:
1    1000
0    1000
Name: count, dtype: int64
📊 Distribusi y_test:
0    500
1    500
Name: count, dtype: int64
```

**Gambar 4.9** Hasil Pembagian *Dataset* Menjadi Data Pelatihan dan Data Pengujian

#### 4.7 Pemodelan dengan *Support Vector Machine* (SVM)

Pada tahap ini dilakukan proses pelatihan (*training*) dan pengujian (*testing*) model *Support Vector Machine* (SVM) untuk mendeteksi berita hoaks berbasis teks. Model SVM dipilih karena kemampuannya dalam menangani data berdimensi tinggi, seperti data teks hasil ekstraksi fitur TF-IDF, serta

efektivitasnya dalam memisahkan dua kelas data menggunakan *hyperplane optimal*.

Dalam penelitian ini, dilakukan perbandingan dua jenis kernel SVM, yaitu *kernel Linear* dan *kernel Radial Basis Function* (RBF). Perbandingan ini bertujuan untuk mengetahui *kernel* yang memberikan performa terbaik dalam mengklasifikasikan berita hoaks dan fakta pada *dataset* penelitian.

#### **4.7.1 Konfigurasi Model dan Parameter Eksperimen**

Model SVM diuji menggunakan dua konfigurasi *kernel*, yaitu *Linear* dan RBF, dengan parameter yang disamakan agar perbandingan performa dilakukan secara adil (*fair comparison*).

**Tabel 4.4** Konfigurasi Model SVM

Parameter	<i>Kernel Linear</i>	<i>Kernel RBF</i>
Jumlah fitur TF-IDF	1000	1000
Nilai C ( <i>Regularization</i> )	1.0	1.0
Pembagian data	2.000 : 1.000	2.000 : 1.000
Sratifikasi kelas	Aktif ( <i>stratify=y</i> )	Aktif ( <i>stratify=y</i> )
<i>Random state</i>	42	42

*Kernel Linear* digunakan untuk membangun *hyperplane linier* yang memisahkan dua kelas data, sedangkan kernel RBF digunakan untuk menangani data yang tidak terpisah secara *linier* dengan memproyeksikan data ke ruang berdimensi lebih tinggi.

#### 4.7.2 Proses Pelatihan Model

Pada tahap pelatihan, model SVM dilatih menggunakan 2.000 data pelatihan ( $X_{train}$  dan  $y_{train}$ ) hasil ekstraksi fitur TF-IDF. Proses pelatihan dilakukan menggunakan pustaka scikit-learn dengan parameter  $C = 1.0$ .

Parameter  $C$  berfungsi sebagai pengontrol tingkat regularisasi yang mengatur keseimbangan antara margin pemisah dan kesalahan klasifikasi. Nilai  $C$  yang digunakan bersifat default untuk menghindari overfitting dan menjaga generalisasi model.



```

1 model = SVC(kernel=cfg['kernel'], C=cfg['C'], random_state=42)
2 model.fit(X_train, y_train)

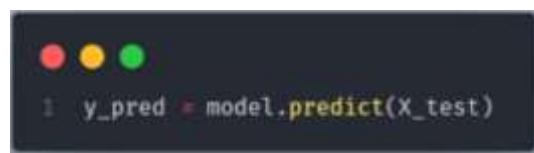
```

**Gambar 4.10** Proses Pelatihan Model SVM Menggunakan Data Pelatihan

#### 4.7.3 Proses Pengujian Model

Setelah model selesai dilatih, dilakukan proses pengujian menggunakan 1.000 data pengujian ( $X_{test}$ ) untuk mengevaluasi kemampuan model dalam mengklasifikasikan data yang belum pernah dilihat sebelumnya.

Model menghasilkan prediksi label menggunakan fungsi prediksi dari SVM, yang kemudian dibandingkan dengan label sebenarnya ( $y_{test}$ ) untuk menghitung metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-score*.



```

1 y_pred = model.predict(X_test)

```

**Gambar 4.11** Proses Prediksi Label Menggunakan Model SVM

## 4.8 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 dengan *kernel Linear* dan SVM dengan *kernel Radial Basis Function* (RBF).

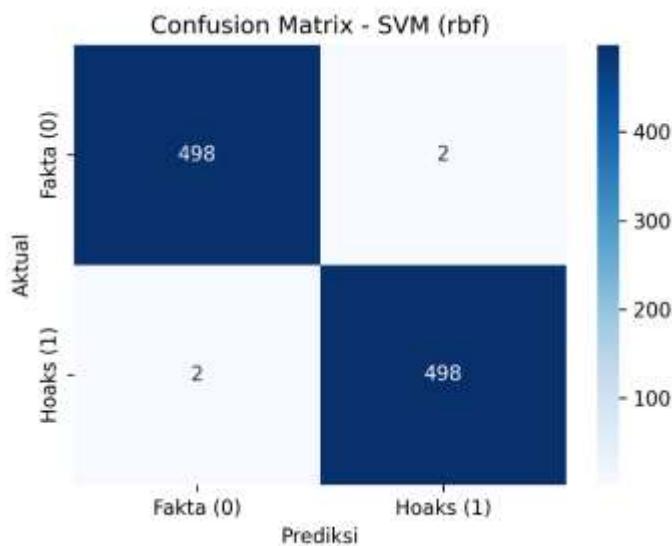
Kedua model diuji menggunakan parameter yang sama, yaitu nilai regularisasi  $C=1.0$  dan jumlah fitur TF-IDF sebanyak 1000, sehingga perbandingan kinerja antar model dapat dilakukan secara objektif (*fair comparison*). Proses evaluasi dilakukan menggunakan data uji sebanyak 1000 data (sekitar 33% dari total *dataset*), sementara 2000 data lainnya digunakan sebagai data latih.

Kinerja model diukur menggunakan empat metrik evaluasi utama, yaitu *Accuracy*, *Precision*, *Recall*, dan *F1-Score*, yang dihitung berdasarkan hasil prediksi model terhadap data uji. Keempat metrik tersebut digunakan untuk memberikan gambaran menyeluruh mengenai kemampuan model dalam mengklasifikasikan berita hoaks dan berita fakta secara tepat.

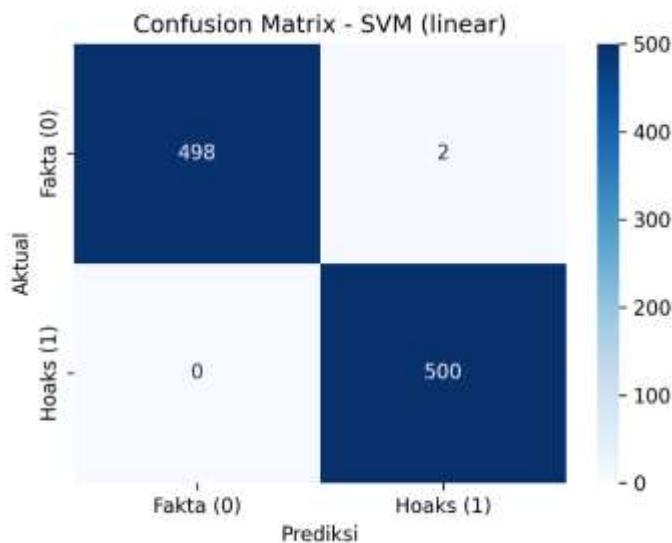
### 4.8.1. Confusion Matrix Model

Sebagai langkah awal dalam evaluasi kinerja model klasifikasi, digunakan *confusion matrix* untuk menggambarkan perbandingan antara label aktual dengan label hasil prediksi model. *Confusion matrix* memberikan informasi dasar mengenai jumlah prediksi yang benar dan salah pada masing-masing kelas, sehingga menjadi dasar dalam perhitungan metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-score*.

Pada penelitian ini, *confusion matrix* digunakan untuk mengevaluasi dua konfigurasi model *Support Vector Machine* (SVM), yaitu SVM dengan *kernel Linear* dan SVM dengan *kernel Radial Basis Function* (RBF).



**Gambar 4.12** *Confusion Matrix Model SVM Kernel RBF*



**Gambar 4.13** *Confusion Matrix Model SVM Kernel Linear*

*Confusion matrix* terdiri dari empat komponen utama sebagai berikut:

1. *True Positive* (TP)

Merupakan jumlah data berita hoaks yang diprediksi sebagai hoaks oleh model dan memang benar berlabel hoaks. Nilai ini menunjukkan kemampuan model dalam mengenali berita hoaks secara tepat.

## 2. *True Negative* (TN)

Merupakan jumlah data berita fakta yang diprediksi sebagai fakta oleh model dan memang benar berlabel fakta. Nilai ini mencerminkan kemampuan model dalam mengidentifikasi berita faktual dengan benar.

## 3. *Fakse Positive* (FP)

Merupakan jumlah data berita fakta yang salah diprediksi sebagai hoaks oleh model. Kesalahan ini menunjukkan tingkat kesalahan model dalam mengklasifikasikan berita fakta sebagai hoaks.

## 4. *False Negative* (FN)

Merupakan jumlah data berita hoaks yang salah diprediksi sebagai fakta oleh model. Kesalahan ini sangat penting diperhatikan karena menunjukkan berita hoaks yang gagal terdeteksi oleh sistem.

Berdasarkan Gambar 4.12 dan Gambar 4.13, dapat dilihat bahwa kedua model SVM mampu melakukan klasifikasi dengan tingkat kesalahan yang sangat rendah. Model SVM *kernel Linear* menunjukkan jumlah kesalahan yang lebih sedikit dibandingkan kernel RBF, khususnya pada prediksi berita hoaks, yang ditunjukkan oleh nilai *False Negative* yang lebih kecil.

Nilai-nilai TP, TN, FP, dan FN yang diperoleh dari *confusion matrix* ini selanjutnya digunakan sebagai dasar perhitungan metrik evaluasi model yang dibahas pada subbab berikutnya.

### 4.8.2 Perhitungan Metrik Evaluasi Model

Pada subbab ini dilakukan perhitungan metrik evaluasi secara manual berdasarkan nilai TP, TN, FP, dan FN yang diperoleh dari *confusion matrix* (Subbab 4.8.1). Perhitungan dilakukan untuk dua model SVM, yaitu *kernel Linear* dan *kernel RBF*.

Rumus yang digunakan :

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

$$Precision = \frac{TP}{TP + FP}$$

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

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

### 1. Perhitungan Metrik SVM *Kernel Linear*

Berdasarkan *confusion matrix* SVM (*linear*):

TN = 498 (Fakta diprediksi Fakta)

FP = 2 (Fakta diprediksi Hoaks)

FN = 0 (Hoaks diprediksi Fakta)

TP = 500 (Hoaks diprediksi Hoaks)

Total data uji:

$$TP + TN + FP + FN = 500 + 498 + 2 + 0 = 1000$$

#### a. Accuracy

$$\begin{aligned} Accuracy &= \frac{TP + TN}{TP + TN + FP + FN} \\ &= \frac{500 + 498}{1000} \\ &= \frac{998}{1000} \\ &= 0.998 = 99.8\% \end{aligned}$$

#### b. Precision

$$\begin{aligned} Precision &= \frac{TP}{TP + FP} \\ &= \frac{500}{500 + 2} \end{aligned}$$

$$\begin{aligned}
 &= \frac{500}{502} \\
 &= 0.9960 = 99.60\%
 \end{aligned}$$

c. *Recall*

$$\begin{aligned}
 Recall &= \frac{TP}{TP + FN} \\
 &= \frac{500}{500 + 0} \\
 &= \frac{500}{500} \\
 &= 1.0 = 100\%
 \end{aligned}$$

d. *F1-Score*

$$\begin{aligned}
 F1 &= 2 \times \frac{Precision \times Recall}{Precision + Recall} \\
 &= 2 \times \frac{0.9960 \times 1.0}{0.9960 + 1.0} \\
 &= 0.9980 = 99.80\%
 \end{aligned}$$

## 2. Perhitungan Metrik SVM *Kernel RBF*

Berdasarkan *confusion matrix* SVM (rbf):

$TN = 498$  (Fakta diprediksi Fakta)

$FP = 2$  (Fakta diprediksi Hoaks)

$FN = 2$  (Hoaks diprediksi Fakta)

$TP = 498$  (Hoaks diprediksi Hoaks)

Total data uji:

$$TP + TN + FP + FN = 498 + 498 + 2 + 2 = 1000$$

a. *Accuracy*

$$\begin{aligned}
 Accuracy &= \frac{TP + TN}{TP + TN + FP + FN} \\
 &= \frac{498 + 498}{1000}
 \end{aligned}$$

$$\begin{aligned}
 &= \frac{996}{1000} \\
 &= 0.996 = 99.6\%
 \end{aligned}$$

b. *Precision*

$$\begin{aligned}
 Precision &= \frac{TP}{TP + FP} \\
 &= \frac{498}{498 + 2} \\
 &= \frac{498}{500} \\
 &= 0.996 = 99.6\%
 \end{aligned}$$

c. *Recall*

$$\begin{aligned}
 Recall &= \frac{TP}{TP + FN} \\
 &= \frac{498}{498 + 2} \\
 &= \frac{498}{500} \\
 &= 0.996 = 99.6\%
 \end{aligned}$$

d. *F1-Score*

$$\begin{aligned}
 F1 &= 2 \times \frac{Precision \times Recall}{Precision + Recall} \\
 &= 2 \times \frac{0.996 \times 0.996}{0.996 + 0.996} \\
 &= 0.996 = 99.6\%
 \end{aligned}$$

Berdasarkan hasil perhitungan manual di atas, diperoleh nilai metrik evaluasi untuk model SVM *kernel Linear* dan RBF. Nilai tersebut selanjutnya dirangkum dalam bentuk tabel pada subbab 4.8.3 Hasil Evaluasi Model SVM untuk memudahkan perbandingan performa kedua model.

#### 4.8.3 Hasil Evaluasi Model SVM

Berdasarkan hasil pengujian menggunakan data uji sebanyak 1.000 data, diperoleh nilai evaluasi performa model *Support Vector Machine* (SVM) dengan *kernel Linear* dan *kernel Radial Basis Function* (RBF). Hasil evaluasi tersebut dirangkum pada Tabel 4.9.

**Tabel 4.5** Hasil Evaluasi Model SVM *Kernel Linear* dan RBF

<b>Kernel</b>	<b>Nilai C</b>	<b>Jumlah Fitur (TF-IDF)</b>	<b>Akurasi (%)</b>	<b>Presisi (%)</b>	<b>Recall (%)</b>	<b>F1-Score (%)</b>
<i>Linear</i>	1.0	1000	99.8	99.6	100.0	99.8
RBF	1.0	1000	99.6	99.6	99.6	99.6

Berdasarkan Tabel 4.9, dapat dilihat bahwa kedua model SVM menunjukkan performa yang sangat baik dalam mendeteksi berita hoaks dan fakta. Model SVM dengan *kernel Linear* memperoleh nilai akurasi dan F1-Score tertinggi, yaitu sebesar 99.8%, sedangkan model SVM kernel RBF memperoleh nilai F1-Score sebesar 99.6%.

Perbedaan performa kedua model relatif kecil, yang menunjukkan bahwa baik *kernel Linear* maupun RBF sama-sama efektif dalam melakukan klasifikasi berita hoaks pada *dataset* yang digunakan.

#### 4.8.4. Analisis Hasil Evaluasi

Berdasarkan hasil evaluasi yang ditunjukkan pada Tabel 4.9, dapat disimpulkan bahwa kedua model *Support Vector Machine* (SVM), baik dengan *kernel Linear* maupun *kernel Radial Basis Function* (RBF), menunjukkan performa yang sangat baik dalam mengklasifikasikan berita hoaks dan fakta pada *dataset* yang digunakan.

Model SVM dengan *kernel Linear* memperoleh nilai akurasi sebesar 99,8% dan nilai *F1-Score* sebesar 99,8%, sedangkan model SVM dengan *kernel RBF* memperoleh nilai akurasi dan *F1-Score* masing-masing sebesar 99,6%. Perbedaan performa antara kedua model relatif kecil, yang menunjukkan bahwa kedua pendekatan *kernel* sama-sama efektif untuk permasalahan klasifikasi teks berbasis TF-IDF.

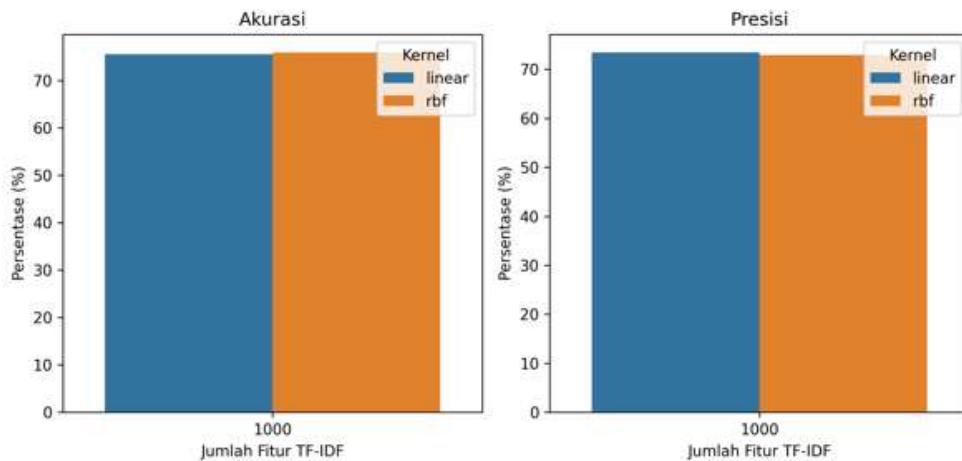
Jika ditinjau lebih lanjut, *kernel Linear* menunjukkan keunggulan pada nilai *recall* sebesar 100%, yang berarti seluruh data berita hoaks pada data uji berhasil terdeteksi tanpa adanya kesalahan *false negative*. Hal ini menunjukkan bahwa model *kernel Linear* sangat efektif dalam mendeteksi berita hoaks, sehingga risiko hoaks yang tidak teridentifikasi dapat diminimalkan.

Sementara itu, *kernel RBF* menunjukkan performa yang sedikit lebih rendah dengan adanya sejumlah kecil kesalahan klasifikasi, namun tetap mempertahankan nilai presisi dan *recall* yang tinggi. Hal ini menunjukkan bahwa *kernel RBF* masih mampu menangkap pola *non-linear* dalam data teks, meskipun pada *dataset* ini pola *linier* sudah cukup representatif.

Secara keseluruhan, hasil evaluasi menunjukkan bahwa penggunaan TF-IDF dengan jumlah fitur 1.000 telah mampu menghasilkan representasi fitur yang efektif, sehingga kedua model SVM dapat bekerja secara optimal. Namun, berdasarkan hasil pengujian dan analisis *confusion matrix*, model SVM dengan *kernel Linear* dipilih sebagai model terbaik dalam penelitian ini karena memberikan performa paling stabil dan tingkat kesalahan klasifikasi paling rendah.

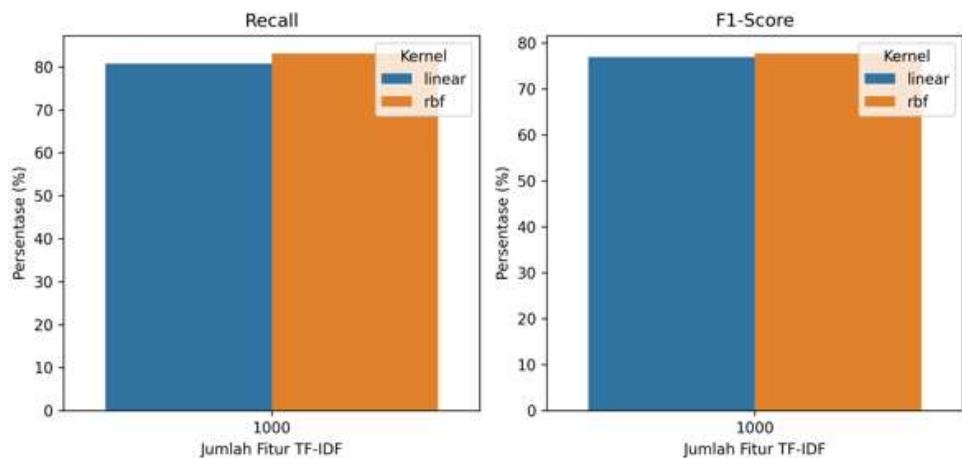
#### 4.8.5. Perbandingan Performa

Untuk memperjelas perbandingan kinerja antara model *Support Vector Machine* (SVM) dengan *kernel Linear* dan *kernel Radial Basis Function* (RBF), dilakukan visualisasi performa menggunakan grafik perbandingan berdasarkan empat metrik evaluasi, yaitu akurasi, presisi, *recall*, dan *F1-score*, sebagaimana ditunjukkan pada Gambar 4.14 dan Gambar 4.15.



**Gambar 4.14** Grafik Perbandingan Akurasi dan Presisi Model SVM

Berdasarkan Gambar 4.15, terlihat bahwa nilai akurasi dan presisi yang dihasilkan oleh kedua model relatif berdekatan. Model SVM dengan *kernel* RBF menunjukkan nilai akurasi yang sedikit lebih tinggi dibandingkan *kernel* Linear, sedangkan *kernel* Linear memiliki nilai presisi yang sedikit lebih tinggi. Perbedaan nilai pada kedua metrik ini tidak terlalu signifikan, sehingga dapat disimpulkan bahwa kedua model memiliki kemampuan yang seimbang dalam mengklasifikasikan berita hoaks dan fakta secara umum.



**Gambar 4.15** Grafik Perbandingan *Recall* dan *F1-Score* Model SVM

Pada Gambar 4.16, terlihat bahwa model SVM dengan *kernel* RBF menghasilkan nilai *recall* dan *F1-score* yang lebih tinggi dibandingkan

*kernel Linear.* Nilai *recall* yang lebih tinggi menunjukkan bahwa *kernel RBF* memiliki kemampuan yang lebih baik dalam mendeteksi seluruh data berita hoaks yang ada pada *dataset uji*. Hal ini berdampak langsung pada peningkatan nilai *F1-score*, yang mencerminkan keseimbangan antara presisi dan *recall*.

Secara keseluruhan, hasil perbandingan menunjukkan bahwa meskipun kedua model SVM memiliki performa yang baik dan relatif seimbang, *kernel RBF* cenderung lebih unggul pada metrik *recall* dan *F1-score*, sehingga lebih efektif dalam meminimalkan kesalahan dalam mendeteksi berita hoaks.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

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

1. Algoritma *Support Vector Machine* (SVM) diterapkan melalui tahapan metodologi yang sistematis, dimulai dari pengumpulan *dataset* berita *online* berbahasa Indonesia, pra-pemrosesan teks (*case folding*, pembersihan karakter, tokenisasi, penghapusan *stopword*, dan *stemming*), serta ekstraksi fitur menggunakan metode *Term Frequency–Inverse Document Frequency* (TF-IDF). *Dataset* hasil pra-pemrosesan kemudian diseleksi dan diseimbangkan sehingga diperoleh 3.000 data dengan distribusi 1.500 berita hoaks dan 1.500 berita fakta. Selanjutnya, data dibagi menjadi 2.000 data latih dan 1.000 data uji, kemudian digunakan untuk melatih model SVM dengan *kernel Linear* dan *Radial Basis Function* (RBF). Melalui tahapan tersebut, model SVM berhasil dibangun dan mampu mengklasifikasikan berita *online* ke dalam kategori hoaks dan fakta berdasarkan karakteristik teks.
2. Berdasarkan hasil pengujian yang telah dilakukan, model klasifikasi *Support Vector Machine* (SVM) menunjukkan performa yang sangat baik dalam mendeteksi berita hoaks pada berita *online* berbahasa Indonesia. Evaluasi kinerja model dilakukan menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score* dengan menggunakan data uji sebanyak 1.000 data. Pada penggunaan 1.000 fitur TF-IDF, model SVM dengan *kernel Linear* memperoleh nilai akurasi sebesar 99,8%, presisi 99,6%, *recall* 100%, dan *F1-score* 99,8%. Sementara itu, model SVM dengan *kernel Radial Basis Function* (RBF) menghasilkan nilai akurasi sebesar 99,6%, presisi 99,6%, *recall* 99,6%, dan *F1-score* 99,6%. Nilai *recall*

yang sangat tinggi menunjukkan bahwa model mampu mendeteksi hampir seluruh berita hoaks secara tepat, sedangkan nilai presisi yang tinggi menunjukkan rendahnya kesalahan dalam memberikan label hoaks. Dengan demikian, dapat disimpulkan bahwa algoritma SVM memiliki performa yang sangat efektif dan andal dalam mendeteksi berita hoaks berdasarkan teks berita *online*.

## 5.2 Saran

Berdasarkan hasil penelitian dan keterbatasan yang ada, beberapa saran untuk pengembangan penelitian selanjutnya adalah sebagai berikut:

1. Penambahan algoritma pembanding

Penelitian selanjutnya disarankan untuk membandingkan performa SVM dengan algoritma klasifikasi lain, seperti *Naïve Bayes*, *Random Forest*, atau metode *Deep Learning* (misalnya LSTM atau BERT) guna memperoleh gambaran performa yang lebih komprehensif dalam deteksi hoaks.

2. Perluasan dan peningkatan kualitas *dataset*

*Dataset* yang digunakan dapat diperluas baik dari segi jumlah data maupun variasi sumber berita, termasuk dari media sosial atau *platform* daring lainnya. Hal ini bertujuan untuk meningkatkan kemampuan generalisasi model terhadap berbagai gaya bahasa dan topik berita.

3. Pengembangan sistem deteksi hoaks berbasis aplikasi

Model klasifikasi yang telah dibangun dapat dikembangkan lebih lanjut ke dalam bentuk aplikasi berbasis web atau sistem pendukung keputusan, sehingga dapat dimanfaatkan secara langsung oleh masyarakat dalam memverifikasi kebenaran berita.

4. Penggunaan metrik evaluasi tambahan

Penelitian selanjutnya disarankan untuk menggunakan metrik evaluasi tambahan seperti ROC–AUC, *Precision–Recall Curve*, atau *confusion matrix* ternormalisasi, guna memberikan analisis performa model yang lebih mendalam dan menyeluruh.

## 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>