



ARTICLE

Sistem Deteksi Berita Hoax Pemilu 2024 Indonesia Menggunakan Algoritma KNN Dan SVM

Indonesian 2024 Election Hoax News Detection System Using KNN And SVM Algorithm

Reza DickiPrabowo, Ida Widaningrum,* dan Jamilah Karaman

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Ponorogo, Indonesia

*Penulis Korespondensi: iwidaningrum.as@gmail.com

(Disubmit 14-08-24; Diterima 23-09-24; Dipublikasikan online pada 05-02-25)

Abstrak

Kemajuan teknologi pada era saat ini memberikan dampak yang signifikan, baik secara positif maupun negatif. Fenomena informasi palsu atau hoax semakin merajalela, terutama dalam konteks politik. Penyebaran informasi hoax telah menjadi strategi yang umum digunakan dalam kontestasi politik di berbagai negara, termasuk dalam pemilihan umum. Dengan melihat tren peningkatan serangan hoax politik sebelum, selama, dan setelah pemilu, penelitian ini menyoroti urgensi pengembangan teknologi deteksi hoax yang lebih canggih dan akurat. Telah dilakukan penelitian sebelumnya menggunakan metode identifikasi namun dinilai belum optimal, sebab akurasi yang dihasilkan masih kurang tinggi. Penelitian ini bertujuan menggunakan metode gabungan algoritma KNN dan SVM untuk mengembangkan sistem deteksi berita hoax yang lebih efektif dengan menggunakan kedua metode gabungan tersebut. Metode gabungan yang telah dilakukan menunjukkan akurasi yang sangat baik, mencapai sekitar 93.31%. KNN memberikan keunggulan dalam kesederhanaan dan kemampuannya menangkap pola non-linear, sementara SVM menonjol dalam kinerja di ruang berdimensi tinggi dengan margin yang besar. Dengan kombinasi kekuatan algoritma yang berbeda, model stacking menghasilkan model yang lebih akurat dan efektif daripada penggunaan satu algoritma.

Kata kunci: Sistem Deteksi; Pemilu 2024; K-Nearest Neighbor (KNN); Support Vector Machine (SVM)

Abstract

Technological advances in the current era have had a significant impact, both positive and negative. The phenomenon of false information or hoaxes is increasingly rampant, especially in the political context. The spread of hoax information has become a strategy commonly used in political contestations in various countries, including in general elections. By looking at the increasing trend of political hoax attacks before, during and after the election, this research highlights the urgency of developing more sophisticated and accurate hoax detection technology. Previous research has been carried out using identification methods but it is considered not optimal, because the resulting accuracy is still not high. This research aims to use a combined method of the KNN and SVM algorithms to develop a more effective hoax news detection system using the two combined methods. The combined method that has been carried out shows very good accuracy, reaching around 93.31%. KNN provides advantages in its simplicity and ability to capture non-linear patterns, while SVM excels in performance in high-dimensional spaces by a large margin. By combining the strengths of different algorithms, model stacking produces a more accurate and effective model than using one algorithm.

KeyWords: Detection system; Pemilu 2024; K-Nearest Neighbor (KNN); Support Vector Machine (SVM)

This is an Open Access article - copyright on authors, distributed under the terms of the Creative Commons Attribution-ShareAlike 4.0 International License (CC BY SA) (<http://creativecommons.org/licenses/by-sa/4.0/>)

How to Cite: R. DickiPrabowo *et al.*, "Sistem Deteksi Berita Hoax Pemilu 2024 Indonesia Menggunakan Algoritma KNN Dan SVM", *JIKO (JURNAL INFORMATIKA DAN KOMPUTER)*, Volume: 9, No.1, Pages 93–111, Februari 2025, doi: 10.26798/jiko.v9i1.1424.

1. Pendahuluan

Kemajuan teknologi saat ini memberikan dampak yang positif maupun negatif. Informasi kini dapat disebarluaskan dengan sangat cepat, memungkinkan siapa saja untuk membuat dan mendistribusikannya melalui internet. Internet telah menjadi platform utama untuk berbagai jenis informasi, seperti berita, musik, gambar, produk, film, dan lainnya [1]. Pengguna media sosial saat ini sangat aktif menggunakan internet, di mana setiap individu dapat menerima dan mengakses berbagai informasi dan berita melalui platform tersebut. Informasi palsu, yang sering disebut sebagai *hoax*, kini telah menjadi bagian dari kehidupan sehari-hari masyarakat. Dalam konteks politik, penyebaran *hoax* di negara maju dan berkembang telah digunakan sebagai strategi untuk mencapai kemenangan [2]. Di era digital saat ini, politik dan *hoax* menjadi dua elemen yang semakin erat kaitannya. Oleh karena itu, menanamkan nilai-nilai ideologis sangat penting untuk mengurangi penyebaran *hoax* [3].

Fenomena *hoax* semakin meluas di media sosial, terutama dalam ranah politik [4]. Menjelang Pemilu 2024, Presidium Masyarakat Anti Fitnah melaporkan peningkatan serangan *hoax* politik, mencapai 664 kasus dalam tiga bulan pertama tahun 2023. Angka ini menunjukkan peningkatan sekitar 24 persen dibandingkan dengan periode yang sama tahun sebelumnya. Berdasarkan temuan Kominfo, *hoax* terkait pemilu meningkat hampir 10 kali lipat antara Januari 2023 hingga Februari 2024. Setelah pemilu berakhir, Kominfo menangani 203 *hoax* pemilu yang tersebar di media sosial dengan total sebaran sebanyak 2.882 konten [5]. *Hoax* ini disebarluaskan melalui berbagai platform media sosial seperti *YouTube*, *Facebook*, *TikTok*, dan aplikasi pesan lainnya, yang tidak hanya menargetkan tokoh politik, tetapi juga partai dan para pendukungnya. Konten *hoax* yang beredar mirip dengan tahun-tahun sebelumnya, mencakup tuduhan korupsi, politik identitas, serta isu-isu SARA yang berpotensi memecah belah masyarakat dan memicu konflik sosial [6, 7]. Dengan semakin maraknya tren *hoax*, berbagai gagasan mulai muncul untuk mencegah penyebarannya [8]. Banyak tips telah tersedia untuk membantu orang menghindari penyebaran berita *hoax*, dan sejumlah platform media sosial kini menyediakan fitur untuk melaporkan konten yang dicurigai sebagai *hoax*. Saat ini, teknologi yang diintegrasikan dalam sistem deteksi untuk mencegah *hoax* masih tergolong jarang, dan sebagian besar pencegahan *hoax* lebih banyak berbentuk situs web yang digunakan untuk melaporkan berita *hoax*.

Mengidentifikasi berita *hoax* secara manual adalah tugas yang sangat merepotkan dan memakan waktu cukup lama [9]. Oleh karena itu, diperlukan penerapan teknologi seperti Kecerdasan Buatan (AI). AI merupakan sistem komputer yang dilatih menggunakan pembelajaran mesin untuk menyelesaikan masalah-masalah tertentu. Dengan teknologi ini, diharapkan proses deteksi dapat dilakukan dengan lebih cepat dan lebih akurat [10]. Sebelumnya, telah dilakukan penelitian untuk mengembangkan sistem deteksi berita *hoax* di situs web menggunakan metode *Term Frequency Inverse Document Frequency* (TF-IDF) dan *Support Vector Machine* (SVM) [11]. Dalam pengujian dengan *10-fold cross validation*, sistem ini menunjukkan bahwa algoritma SVM memiliki nilai *F-Measure* dan akurasi yang cukup tinggi, yaitu masing-masing 0.804 dan 80.48% [12]. Namun, akurasi yang dihasilkan masih dianggap kurang memadai. Hal ini dapat mempengaruhi keakuratan pencarian dalam sistem deteksi, yang dapat menyebabkan data tidak ditampilkan atau bahkan hilang.

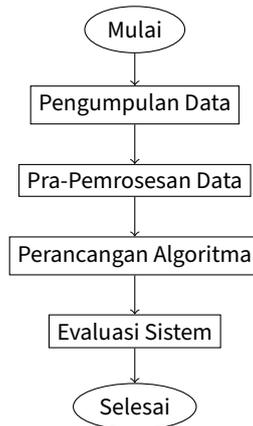
Oleh karena itu peneliti melakukan penelitian sistem deteksi berita *hoax* pemilu 2024 dengan menggunakan dua metode gabungan baru dengan algoritma KNN dan SVM. Diharapkan penelitian ini dapat mencapai akurasi yang lebih tinggi dibandingkan penelitian sebelumnya. Penelitian ini menerapkan metode *K-Nearest Neighbor* (KNN), yang menganalisis frekuensi kemunculan kata dalam suatu kalimat. Klasifikasi kalimat dilakukan berdasarkan nilai frekuensi kata yang paling tinggi. *Support Vector Machine* (SVM) adalah metode pembelajaran terawasi yang digunakan untuk mengklasifikasikan berbagai jenis data. Secara umum, konsep SVM berfokus pada menemukan *hyperplane* optimal yang memisahkan dua kelas. SVM menentukan *hyperplane* dengan memperhitungkan *support vectors* dan margin. *Support vectors* adalah data yang paling dekat dengan *hyperplane*, sedangkan margin mengukur jarak antara *hyperplane* pemisah dan data. Tujuan SVM adalah menciptakan fungsi klasifikasi dalam bentuk $\text{sign}(x)$, $f(x) = y$, untuk memastikan bahwa data dapat diklasifikasikan dengan akurat selama proses pengujian [13].

2. Metode

Penelitian ini akan berfokus pada pengembangan sistem deteksi berita *hoax* dengan menggabungkan metode *K-Nearest Neighbor* (KNN) dan *Support Vector Machine* (SVM). Proses pengembangan sistem meliputi tahapan pengumpulan data, pemrosesan awal data, penerapan KNN dan SVM, dengan hasil akhir berupa sistem deteksi berita *hoax*. Data penelitian diperoleh dengan *Web Scrapping* dari *website*.

2.1 Tahap Perancangan Sistem

Proses perancangan sistem ditampilkan pada Gambar 1.



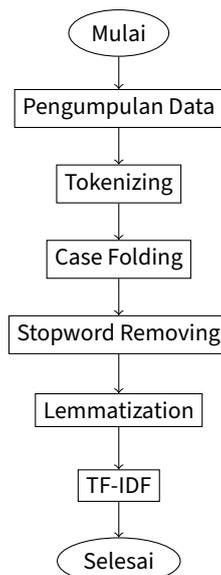
Gambar 1. Tahap Alur Perancangan Sistem

2.1.1 Pengumpulan Data

Data yang digunakan dalam penelitian ini, adalah data sekunder yang diambil melalui *Web Scrapping* dari *website kumparan.com, detik.com, turnbackhoax.id, dan liputan6.com* dari bulan Agustus 2023 hingga Juni 2024. Data yang berlabel salah *hoax*, disinformasi, misinformasi, dan klarifikasi dikategorikan sebagai berita *hoax* dan data yang berlabel benar, berita, dan edukasi dikategorikan sebagai *non-hoax*.

2.1.2 Pre-Pemrosesan Data

Berikut ini merupakan urutan tahapan pra-pemrosesan data :



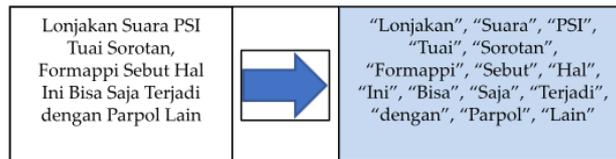
Gambar 2. Flowchart Alur Pra-Pemrosesan Data

Data yang telah dikumpulkan akan diproses untuk diubah ke dalam format yang sesuai untuk perhitung-

an TF-IDF. TF-IDF memberikan bobot pada setiap kata dalam dokumen dan digunakan untuk memilih fitur atau kata-kata yang memiliki pengaruh signifikan dalam dokumen tersebut. Pra-pemrosesan data mencakup beberapa langkah, yaitu *tokenisasi*, *case folding*, *filtering*, *lemmatization*, dan pembobotan TF-IDF sebagai tahap akhir. Proses ini menghasilkan kumpulan kata-kata dari berita *hoax* yang terpilih, yang kemudian disimpan dalam format dokumen CSV. Dokumen ini akan digunakan sebagai dataset untuk klasifikasi menggunakan *K-Nearest Neighbor* (KNN), dengan mengukur jarak terdekat antara berita *hoax* yang sudah diproses dan berita *hoax* yang sedang diuji.

1. *Tokenizing*

Berikut ini merupakan tahapan *tokenizing* :

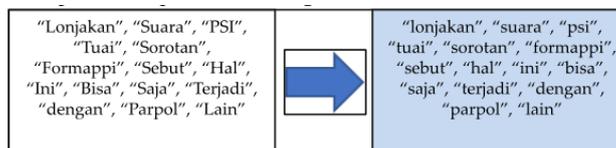


Gambar 3. Tahap *Tokenizing*

Langkah untuk memecah teks menjadi bagian-bagian seperti kata, frasa, simbol, atau elemen lain yang dikenal sebagai *token*. Tujuannya untuk menyelidiki kata-kata dalam suatu kalimat [14].

2. *Case Folding*

Berikut merupakan tahapan *case folding* :

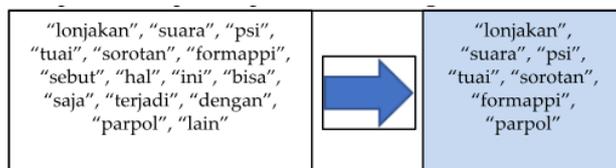


Gambar 4. Tahap *Case Folding*

Mengubah semua huruf dalam dokumen menjadi huruf kecil (dari 'a' sampai 'z'). Karakter yang bukan huruf dianggap sebagai pemisah dan dihilangkan dari dokumen untuk mengurangi gangguan atau noise dalam proses ekstraksi informasi [14].

3. *Stopword Removing*

Berikut ini merupakan tahapan *stopwords removing* :

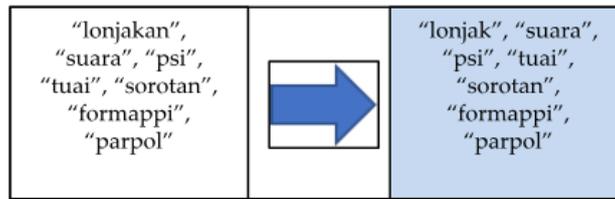


Gambar 5. Tahap *Stopword Removing*

Proses penghapusan *stopwords* bertujuan untuk mengurangi jumlah kata dalam teks dengan menghilangkan kata-kata seperti kata depan, kata penghubung, dan kata ganti. Beberapa contoh stopwords dalam bahasa Indonesia termasuk "yang", "ini", "dari", "ke", "di", dan "dari" [15], [16].

4. *Lemmatization*

Berikut adalah tahap *lemmatization* :

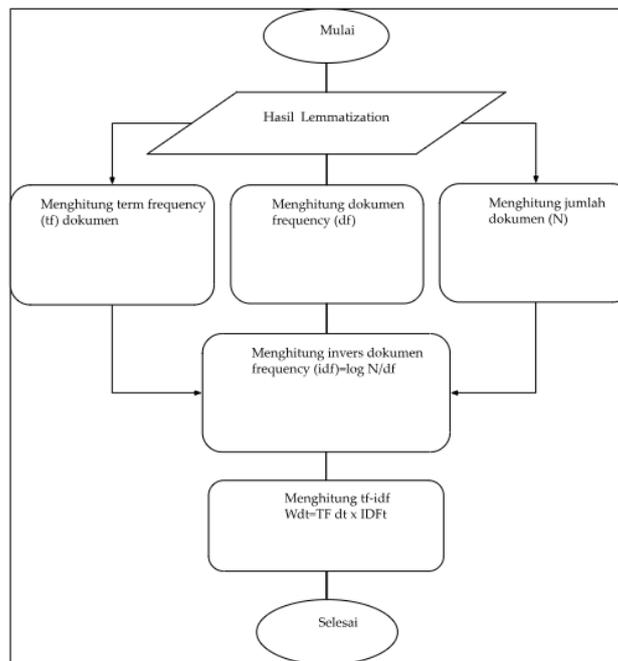


Gambar 6. Tahap Lemmatization

Lemmatization bertujuan untuk mengubah kata-kata yang muncul dalam teks menjadi bentuk dasar atau bentuk yang ada dalam kamus yang disebut sebagai *lemma*. Proses ini dilakukan dengan cara menghilangkan akhiran dan awalan kata. Dengan mengurangi variasi kata yang berbeda, *lemmatization* membantu mengurangi kompleksitas teks yang sedang dianalisis [17].

5. Pembobotan TF-IDF

Berikut merupakan tahapan pembobotan kata :



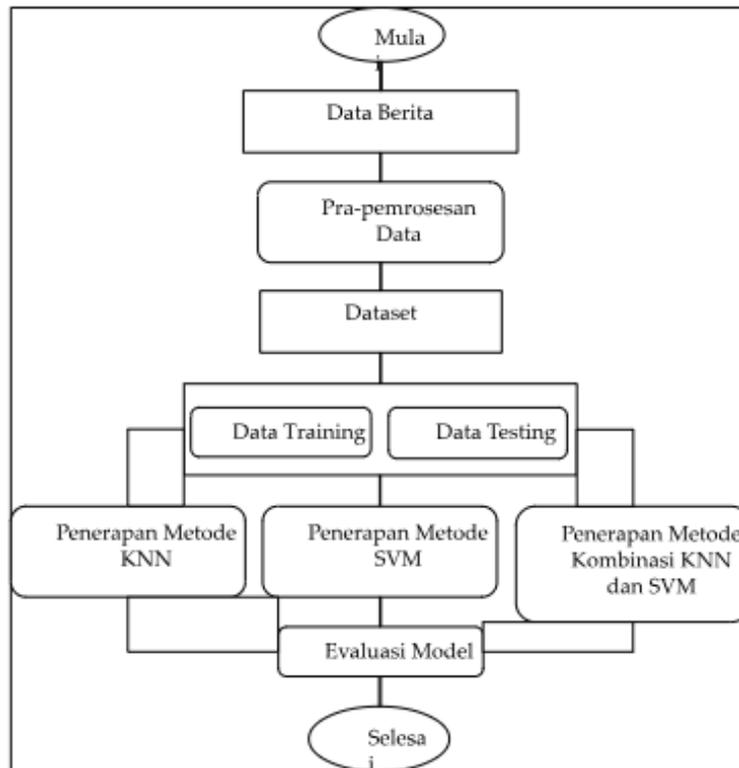
Gambar 7. Tahap Alur Perhitungan TF-IDF

TF-IDF adalah kombinasi dari dua formula, yaitu TF dan IDF, yang dirancang untuk menentukan bobot setiap kata dalam dokumen. Proses ini melibatkan perhitungan frekuensi kemunculan kata dalam dokumen (t_i) dan jumlah total kata dalam dokumen (t) sebagai bagian dari formula TF. Selain itu, sistem juga menghitung jumlah dokumen yang memuat kata yang bobotnya dihitung ($d: t_i \in d$), yang merupakan elemen dari perhitungan IDF [18].

2.2 Perancangan Algoritma

2.2.1 Tahap Perancangan Algoritma

Berikut ini merupakan alur tahapan perancangan algoritma :



Gambar 8. Tahap Perancangan Algoritma

Setelah menyelesaikan tahap pra-pemrosesan dataset seperti yang dijelaskan pada subbab sebelumnya, langkah berikutnya adalah membagi data menjadi dua bagian: data pelatihan dan data pengujian, dengan proporsi 70% untuk pelatihan dan 30% untuk pengujian. Dengan rasio 70:30 dapat memberikan keseimbangan yang baik antara jumlah data yang tersedia untuk pelatihan dan pengujian, memungkinkan untuk menghasilkan evaluasi yang stabil terhadap kinerja model dan memperkirakan kemampuan model untuk digeneralisasikan pada data baru.

Penerapan metode *K-Nearest Neighbors* (KNN) melibatkan inisialisasi model dengan memilih jumlah tetangga terdekat (k). Proses pembelajaran dilakukan dengan menghitung jarak antara data menggunakan *Euclidean Distance*, dan prediksi kelas data dilakukan dengan memilih mayoritas kelas dari tetangga terdekat. Evaluasi model KNN menggunakan metrik seperti akurasi, presisi, recall, dan F1-score.

Sedangkan penerapan *Support Vector Machine* (SVM) melibatkan inisialisasi model dengan memilih *kernel* yang sesuai (linear, polynomial, atau radial basis function). Model SVM membangun *hyperplane* terbaik untuk memisahkan kelas. Proses pembelajaran melibatkan penentuan *hyperplane* yang optimal [19]. Evaluasi model SVM menggunakan metrik evaluasi yang sama dengan KNN.

Kombinasi Metode KNN dan SVM melibatkan integrasi hasil dari kedua model untuk meningkatkan performa deteksi berita *hoax*. Pendekatan ini bertujuan untuk memanfaatkan kelebihan masing-masing metode, sehingga dapat memberikan hasil yang lebih *robust* dan akurat dalam menanggulangi permasalahan disinformasi. Evaluasi hasil dari kombinasi metode ini akan memberikan gambaran komprehensif terkait keefektifan sistem deteksi berita *hoax* yang dikembangkan.

2.2.2 Tahap Penerapan Sistem

Setelah melatih model KNN dan SVM, model disimpan dengan menggunakan `library joblib.dumb()` yang digunakan untuk menyimpan dan memuat model pembelajaran mesin yang telah dilatih dan memungkinkan untuk menyimpan model secara persisten setelah dilatih, sehingga dapat menggunakannya kembali tanpa perlu melatih ulang setiap kali ingin menggunakannya. Sebagai penghubung *input* dari *user* dengan model menggunakan *framework flask*.

2.2.3 Tahap Pengujian Fungsionalitas Sistem

Pengujian fungsionalitas menggunakan pendekatan *white box* melibatkan pengujian internal sistem secara mendetail, termasuk alur logika, dan kode. Pengujian ini memastikan bahwa setiap bagian dari seluruh komponen sistem beroperasi sesuai dengan yang diharapkan. Dengan memeriksa implementasi algoritma dan proses internal untuk memastikan integritas dan keandalan sistem. Tujuan utama dari pengujian *white box* adalah untuk memastikan bahwa setiap bagian dan jalur dalam kode bekerja dengan baik serta memberikan hasil yang sesuai dengan *input* yang diterima.

2.3 Evaluasi Sistem

Evaluasi sistem merupakan langkah penting dalam menilai kinerja deteksi berita hoax yang dikembangkan. Setelah melalui tahapan pemisahan data, pelatihan model, dan pengujian menggunakan metode *K-Nearest Neighbors* (KNN) dan *Support Vector Machine* (SVM), maka dilakukan tahap evaluasi sistem untuk memberikan gambaran seberapa baik sistem tersebut.

Setelah model KNN dilatih dan diuji pada dataset uji, metrik evaluasi seperti akurasi, presisi, *recall*, dan skor F1 akan memberikan gambaran tentang performa model dalam mengklasifikasikan berita. Akurasi mencerminkan tingkat presisi model secara keseluruhan, presisi menunjukkan seberapa banyak berita yang tergolong *hoax* sebenarnya *hoax*, *recall* mengukur seberapa banyak berita *hoax* yang berhasil terdeteksi, dan skor F1 memberikan keseimbangan antara presisi dan *recall*.

Sementara itu, evaluasi model SVM juga dilakukan dengan menggunakan metrik evaluasi yang serupa. Akurasi, presisi, *recall*, dan skor F1 akan memberikan gambaran komprehensif tentang seberapa baik SVM dapat mengklasifikasikan berita pada dataset pengujian. Dalam konteks kombinasi metode KNN dan SVM, evaluasi sistem akan menjadi lebih kompleks karena melibatkan pengintegrasian hasil dari kedua model.

3. Hasil

3.1 Pengumpulan Data

Data yang dikumpulkan berasal dari sumber berita *hoax*, yaitu *Turnbackhoax.id*, yang dikelola oleh Masyarakat Anti *Hoax* Indonesia (MAFINDO), sebuah organisasi resmi yang telah berdiri sejak November 2016. Sedangkan data non-*hoax* berasal dari situs penyedia berita terpercaya yaitu *kumparan.com*, *detik.com*, *liputan6.com*, dan *cnn indonesia*.

Setiap berita yang diperoleh dari situs sudah dilabeli untuk menunjukkan berita tersebut termasuk berita non-*hoax* atau *hoax*. Data berita yang dideteksi berasal dari judul berita dari kanal berita. Sebanyak 4283 data dikumpulkan, terdiri dari 3040 berita non-*hoax* dan 1243 berita *hoax*. Data diambil dari periode Agustus 2023 hingga Juni 2024. Data yang terkumpul disimpan dalam dokumen yang dinamai *dataset.csv*.

```

Dataset:
  id      text      label
4278 4279 Nusron: Kami Bersyukur Saran Kiai Agar Pak Pra... 0
4279 4280 Ketua Harian Gerindra Sebut Relawan Laskar Pra... 0
4280 4281 Fahri Hamzah Ungkit Harun Masiku, Senior PDIP:... 0
4281 4282 Dukung Prabowo-Gibran, Agus Gumiwang Akan Cuti... 0
4282 4283 Sandiaga Yakin TPN Ganjar-Mahfud Semakin Kuat ... 0

Jumlah data dalam dataset:
4283
    
```

Gambar 9. Jumlah Data

3.2 Pra-Pemrosesan Data

Data yang telah dimuat kemudian dipersiapkan untuk memudahkan proses pembelajaran mesin. Tahapan pra-pemrosesan data terdiri dari 5 tahapan, yaitu *tokenizing*, *case folding*, *stopword removal*, dan *lemmatization*. Proses pra-pemrosesan data dilakukan dengan memanfaatkan *Google Colab*. Seluruh proses tersebut dilakukan secara berurutan.

3.2.1 Tokenizing

Proses *tokenizing* adalah tahap pemecahan teks menjadi unit-unit seperti kata, frasa, simbol, atau elemen lain yang disebut sebagai *token*. Tabel 1 memperlihatkan hasil dari proses *tokenizing*.

Tabel 1. Tokenizing

Data Asli	Hasil <i>Tokenizing</i>
KPU Jatuhkan Denda dan Pidana Kepada Gibran Serta Dicoret dari Cawapres	“Kpu”, ”Jatuhkan”, ”Denda”, ”dan”, ”Pidana”, ”Kepada”, ”Gibran”, ”Serta”, ”Dicoret”, ”dari”, ”Cawapres”
Prabowo beri 20 juta bagi pemilihnya di Pilpres 2024	“Prabowo”, ”beri”, ”20”, ”juta”, ”bagi”, ”pemilihnya”, ”di”, ”Pilpres”, ”2024”
Golkar Resmi Dukung Bobby Nasution di Pilgub Sumut	“Golkar”, ”Resmi”, ”Dukung”, ”Bobby”, ”Nasution”, ”di”, ”Pilgub”, ”Sumut”

Tahap tokenisasi mengubah kalimat menjadi unit-unit kata yang diatur dalam sebuah *array*, seperti yang ditunjukkan pada Tabel 1. Kalimat dipecah menjadi kata-kata secara individual. Setelah proses *tokenisasi* selesai, langkah berikutnya adalah tahap *case folding*.

Hasil dari proses *tokenizing* yang dilakukan di *Google Colab* sebagai berikut :

```

Tokenizing :
Data Asli: KPU Jatuhkan Denda dan Pidana Kepada Gibran Serta Dicoret dari Cawapres
Hasil Tokenizing : ['KPU', 'Jatuhkan', 'Denda', 'dan', 'Pidana', 'Kepada', 'Gibran', 'Serta', 'Dicoret', 'dari', 'Cawapres']

Data Asli: Prabowo beri 20 juta bagi pemilihnya di Pilpres 2024
Hasil Tokenizing : ['Prabowo', 'beri', '20', 'juta', 'bagi', 'pemilihnya', 'di', 'Pilpres', '2024']

Data Asli: Golkar Resmi Dukung Bobby Nasution di Pilgub Sumut
Hasil Tokenizing : ['Golkar', 'Resmi', 'Dukung', 'Bobby', 'Nasution', 'di', 'Pilgub', 'Sumut']

Data Asli: TNI Siapkan 133.416 Personel Bantu Pengamanan Pilkada 2024
Hasil Tokenizing : ['TNI', 'Siapkan', '133.416', 'Personel', 'Bantu', 'Pengamanan', 'Pilkada', '2024']

Data Asli: KPU Resmi Umumkan Prabowo-Gibran Pemenang Pilpres 2024!
Hasil Tokenizing : ['KPU', 'Resmi', 'Umunkan', 'Prabowo-Gibran', 'Pemenang', 'Pilpres', '2024', '!']

Data Asli: Golkar Resmi Usung Khofifah-Emil di Pilkada Jatim
Hasil Tokenizing : ['Golkar', 'Resmi', 'Usung', 'Khofifah-Emil', 'di', 'Pilkada', 'Jatim']
    
```

Gambar 10. Hasil Pemrosesan *Tokenizing*

3.2.2 Case Folding

Case Folding adalah proses yang mengubah semua huruf kapital dalam teks menjadi huruf kecil (dari 'A' hingga 'z'). Hasil dari proses *case folding* dapat dilihat pada Tabel 2.

Tabel 2. Case Folding

Hasil <i>Tokenizing</i>	Hasil <i>Case Folding</i>
“Kpu”, ”Jatuhkan”, ”Denda”, ”dan”, ”Pidana”, ”Kepada”, ”Gibran”, ”Serta”, ”Dicoret”, ”dari”, ”Cawapres”	“kpu”, ”jatuhkan”, ”denda”, ”dan”, ”pidana”, ”kepada”, ”gibran”, ”serta”, ”dicoret”, ”dari”, ”cawapres”
“Prabowo”, ”beri”, ”20”, ”juta”, ”bagi”, ”pemilihnya”, ”di”, ”Pilpres”, ”2024”	“prabowo”, ”beri”, ”20”, ”juta”, ”bagi”, ”pemilihnya”, ”di”, ”pilpres”, ”2024”
“Golkar”, ”Resmi”, ”Dukung”, ”Bobby”, ”Nasution”, ”di”, ”Pilgub”, ”Sumut”	“golkar”, ”resmi”, ”dukung”, ”bobby”, ”nasution”, ”di”, ”pilgub”, ”sumut”

Teks atau kalimat sering kali tidak konsisten dalam penggunaan huruf kapital dan huruf kecil, sehingga perlu dilakukan konversi dari huruf kapital ke huruf kecil. Langkah berikutnya adalah melakukan penghapusan kata-kata umum (*stopword removal*). Hasil dari proses *case folding* yang dilakukan di *Google Colab* sebagai berikut :

```

Case Folding :
Data Asli: KPU Jatuhkan Denda dan Pidana Kepada Gibran Serta Dicoret dari Cawapres
Hasil Case Folding: ['kpu', 'jatuhkan', 'denda', 'dan', 'pidana', 'kepada', 'gibran', 'serta', 'dicoret', 'dari', 'cawapres']

Data Asli: Prabowo beri 20 juta bagi pemilihnya di Pilpres 2024
Hasil Case Folding: ['prabowo', 'beri', '20', 'juta', 'bagi', 'pemilihnya', 'di', 'pilpres', '2024']

Data Asli: Golkar Resmi Dukung Bobby Nasution di Pilgub Sumut
Hasil Case Folding: ['golkar', 'resmi', 'dukung', 'bobby', 'nasution', 'di', 'pilgub', 'sumut']

Data Asli: TNI Siapkan 133.416 Personel Bantu Pengamanan Pilkada 2024
Hasil Case Folding: ['tni', 'siapkan', '133.416', 'personel', 'bantu', 'pengamanan', 'pilkada', '2024']

Data Asli: KPU Resmi Umumkan Prabowo-Gibran Pemenang Pilpres 2024!
Hasil Case Folding: ['kpu', 'resmi', 'umumkan', 'prabowo-gibran', 'pemenang', 'pilpres', '2024', '!']

Data Asli: Golkar Resmi Usung Khofifah-Emil di Pilkada Jatim
Hasil Case Folding: ['golkar', 'resmi', 'usung', 'khofifah-emil', 'di', 'pilkada', 'jatim']
    
```

Gambar 11. Hasil Pemrosesan Case Folding

3.2.3 Stopword Removing

Tabel 3. Stopword Removal

Hasil Case Folding	Hasil Stopword Removal
“kpu”, ”jatuhkan”, ”denda”, ”dan”, ”pidana”, ”kepada”, ”gibran”, ”serta”, ”dicoret”, ”dari”, ”cawapres”	“kpu”, ”jatuhkan”, ”denda”, ”pidana”, ”gibran”, ”dicoret”, ”cawapres”
“prabowo”, ”beri”, ”20”, ”juta”, ”bagi”, ”pemilihnya”, ”di”, ”pilpres”, ”2024”	“prabowo”, ”20”, ”juta”, ”pemilihnya”, ”pilpres”, ”2024”
“golkar”, ”resmi”, ”dukung”, ”bobby”, ”nasution”, ”di”, ”pilgub”, ”sumut”	“golkar”, ”resmi”, ”dukung”, ”bobby”, ”nasution”, ”pilgub”, ”sumut”

Kata-kata seperti “yang”, “dan”, “untuk”, “saya”, dan sejenisnya dihapus. Setelah proses penghapusan *stopword* selesai, langkah selanjutnya adalah melanjutkan ke tahapan proses *lemmatization*.

Hasil dari proses *stopword removal* yang dilakukan di *Google Colab* sebagai berikut :

```

Stopword Removal :
Data Asli: KPU Jatuhkan Denda dan Pidana Kepada Gibran Serta Dicoret dari Cawapres
Hasil Stopwords Removal: ['kpu', 'jatuhkan', 'denda', 'pidana', 'gibran', 'dicoret', 'cawapres']

Data Asli: Prabowo beri 20 juta bagi pemilihnya di Pilpres 2024
Hasil Stopwords Removal: ['prabowo', '20', 'juta', 'pemilihnya', 'pilpres', '2024']

Data Asli: Golkar Resmi Dukung Bobby Nasution di Pilgub Sumut
Hasil Stopwords Removal: ['golkar', 'resmi', 'dukung', 'bobby', 'nasution', 'pilgub', 'sumut']

Data Asli: TNI Siapkan 133.416 Personel Bantu Pengamanan Pilkada 2024
Hasil Stopwords Removal: ['tni', 'siapkan', '133.416', 'personel', 'bantu', 'pengamanan', 'pilkada', '2024']

Data Asli: KPU Resmi Umumkan Prabowo-Gibran Pemenang Pilpres 2024!
Hasil Stopwords Removal: ['kpu', 'resmi', 'umumkan', 'prabowo-gibran', 'pemenang', 'pilpres', '2024', '!']

Data Asli: Golkar Resmi Usung Khofifah-Emil di Pilkada Jatim
Hasil Stopwords Removal: ['golkar', 'resmi', 'usung', 'khofifah-emil', 'pilkada', 'jatim']
    
```

Gambar 12. Hasil Pemrosesan Stopword Removal

3.2.4 Lemmatization

Lemmatization bertujuan untuk mengubah kata dalam kalimat menjadi bentuk dasarnya atau bentuk yang tercantum dalam kamus, yang dikenal sebagai *lemma*. Proses ini dilakukan dengan cara menghilangkan akhiran dan awalan kata. Tabel 4 menunjukkan hasil dari *lemmatization*.

Tabel 4. Lemmatization

Hasil Stopword Removal	Hasil Lemmatization
“kpu”, ”jatuhkan”, ”denda”, ”pidana”, ”gibran”, ”dicoret”, ”cawapres”	“kpu”, ”jatuh”, ”denda”, ”pidana”, ”gibran”, ”coret”, ”cawapres”
“prabowo”, ”20”, ”juta”, ”pemilihnya”, ”pilpres”, ”2024”	“prabowo”, ”20”, ”juta”, ”pilih”, ”pilpres”, ”2024”
“golkar”, ”resmi”, ”dukung”, ”bobby”, ”nasution”, ”pilgub”, ”sumut”	“golkar”, ”resmi”, ”dukung”, ”bobby”, ”nasution”, ”pilgub”, ”sumut”

Dengan mengurangi variasi kata yang berbeda, *lemmatization* membantu mengurangi kompleksitas teks yang sedang dianalisis. Setelah proses *lemmatization* selesai, langkah selanjutnya adalah melanjutkan ke

tahapan proses TF-IDF.

Hasil dari proses *lemmatization* yang dilakukan di *Google Colab* sebagai berikut :

```

Lemmatization:
Data Asli: KPU Jatuhkan Denda dan Pidana Kepada Gibran Serta Dicoret dari Cawapres
Hasil Lemmatization: ['kpu', 'jatuh', 'denda', 'pidana', 'gibran', 'coret', 'cawapres']

Data Asli: Prabowo beri 20 juta bagi pemilihnya di Pilpres 2024
Hasil Lemmatization: ['prabowo', '20', 'juta', 'pilih', 'pilpres', '2024']

Data Asli: Golkar Resmi Dukung Bobby Nasution di Pilgub Sumut
Hasil Lemmatization: ['golkar', 'resmi', 'dukung', 'bobby', 'nasution', 'pilgub', 'sumut']

Data Asli: TNI Siapkan 133.416 Personel Bantu Pengamanan Pilkada 2024
Hasil Lemmatization: ['tni', 'siap', '133 416', 'personel', 'bantu', 'aman', 'pilkada', '2024']

Data Asli: KPU Resmi Umumkan Prabowo-Gibran Pemenang Pilpres 2024!
Hasil Lemmatization: ['kpu', 'resmi', 'umum', 'prabowo-gibran', 'menang', 'pilpres', '2024', '']

Data Asli: Golkar Resmi Usung Khofifah-Emil di Pilkada Jatim
Hasil Lemmatization: ['golkar', 'resmi', 'usung', 'khofifah-emil', 'pilkada', 'jatim']

```

Gambar 13. Hasil Pemrosesan *Lemmatization*

3.2.5 TF-IDF

TF-IDF (*Term Frequency-Inverse Document Frequency*), seperti yang ditunjukkan pada Tabel 5, dirancang untuk memberikan bobot pada frekuensi kemunculan kata dalam dokumen. Kata-kata yang sering muncul dalam satu dokumen tetapi jarang terdapat di dokumen lain akan memperoleh bobot yang lebih tinggi. Sebaliknya, kata-kata yang sering ditemukan di banyak dokumen akan memiliki bobot yang lebih rendah. Bobot kata dalam keseluruhan kumpulan dokumen akan meningkat seiring dengan frekuensi kemunculan kata tersebut dalam dokumen.

Tabel 5. TF-IDF

Data	Token
D1	kpu jatuh denda pidana kepada gibran coret cawapres
D2	prabowo beri 20 juta bagi pemilih pilpres 2024
D3	kpu resmi umum prabowo gibran menang pilpres 2024

Tabel 6. Perhitungan TF-IDF

Token	tf			df	D/df	IDF= Log(D/df)	W= tf * IDF		
	D1	D2	D3				D1	D2	D3
kpu	1	0	1	2	1.5	0.176	0.176	0	0.176
jatuh	1	0	0	1	3	0.477	0.477	0	0
denda	1	0	0	1	3	0.477	0.477	0	0
kepada	1	0	0	1	3	0.477	0.477	0	0
gibran	1	0	1	2	1.5	0.176	0.176	0	0.176
coret	1	0	0	1	3	0.477	0.477	0	0
cawapres	1	0	0	1	3	0.477	0.477	0	0
prabowo	0	1	1	2	1.5	0.176	0	0.176	0.176
beri	0	1	0	1	3	0.477	0	0.477	0
20	0	1	0	1	3	0.477	0	0.477	0
juta	0	1	0	1	3	0.477	0	0.477	0
bagi	0	1	0	1	3	0.477	0	0.477	0
pemilih	0	1	0	1	3	0.477	0	0.477	0
pilpres	0	1	1	2	1.5	0.176	0	0.176	0.176
2024	0	1	1	2	1.5	0.176	0	0.176	0.176
resmi	0	0	1	1	3	0.477	0	0	0.477
umum	0	0	1	1	3	0.477	0	0	0.477
menang	0	0	1	1	3	0.477	0	0	0.477
Total							2.737	2.913	2.311

Kata-kata yang sering muncul dalam satu dokumen namun jarang ditemukan di dokumen lain akan mendapatkan bobot yang lebih tinggi, sementara kata-kata yang sering muncul di berbagai dokumen akan memiliki bobot yang lebih rendah. Setelah proses TF-IDF selesai, langkah selanjutnya adalah melanjutkan ke tahap perancangan algoritma.

3.3 Perancangan Algoritma

Dalam proses perancangan algoritma terdapat beberapa tahapan, berikut tahapan dalam perancangan sistem:

3.3.1 Pemisahan Data

Pada tahap ini, data dibagi menjadi dua bagian: data pelatihan dan data pengujian dengan perbandingan 70:30. Rasio data pelatihan harus lebih besar dibandingkan dengan rasio data pengujian untuk memastikan bahwa model dapat mencapai akurasi yang lebih baik selama proses pelatihan dan pembelajaran. Setiap sesi pelatihan akan dievaluasi untuk memverifikasi hasilnya, kemudian diuji kembali dengan data pengujian untuk menilai keakuratan model yang telah dikembangkan. Hasil dari pemisahan data ditampilkan dalam Tabel 7.

Tabel 7. Pemisahan Data

Banyak Data	
Training Data	2998
Testing Data	1285
Jumlah	4283

3.3.2 Penerapan Model KNN

Penerapan metode *K-Nearest Neighbors* (KNN) melibatkan inisialisasi model dengan memilih jumlah tetangga terdekat (*k*). Proses pembelajaran dilakukan dengan menghitung jarak antara data menggunakan

Euclidean Distance, dan prediksi kelas data dilakukan dengan memilih mayoritas kelas dari tetangga terdekat. Pada tahap ini, analisis akan dilakukan dengan menggunakan algoritma *K-Nearest Neighbor*, dengan nilai k yang diterapkan dalam proses klasifikasi adalah 5.

Data akan dipisahkan dengan perbandingan 70:30 antara data pelatihan dan data pengujian. Pemisahan data ini dilakukan menggunakan metode *train-test split* dari *library scikit-learn* dengan parameter *random state=42*. Pelatihan model KNN dilakukan dalam *google colab* sebagai berikut.

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report,
accuracy_score
```

Gambar 14. Kode Penerapan Model KNN

Source code di atas berfungsi untuk mengimpor beberapa modul dan pustaka yang diperlukan untuk melakukan analisis data dan klasifikasi teks menggunakan algoritma *K-Nearest Neighbors* (KNN). Berikut penjelasannya :

1. *import pandas as pd* Mengimpor pustaka *pandas*, yang digunakan untuk analisis data.
2. *from sklearn.feature_extraction.text import TfidfVectorizer* Mengimpor *TfidfVectorizer* dari pustaka *scikit-learn*, yang berfungsi untuk mengkonversi teks menjadi fitur numerik dengan menggunakan metode TF-IDF
3. *from sklearn.model_selection import train_test_split* Mengimpor fungsi yang digunakan untuk membagi dataset: data pelatihan dan data pengujian.
4. *from sklearn.neighbors import KNeighborsClassifier* Mengimpor kelas *KNeighborsClassifier* dari *scikit-learn*. Kelas ini digunakan untuk membuat model *K-Nearest Neighbors* (KNN).
5. *from sklearn.metrics import classification_report, accuracy_score* Mengimpor fungsi yang digunakan untuk menilai kinerja model klasifikasi dengan mengukur tingkat akurasi dari model tersebut.

```
# Pembagian data menjadi data latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)
```

Gambar 15. Pemisahan Data KNN

Source code di atas digunakan untuk membagi dataset menjadi data pelatihan (*training*) dan data pengujian (*testing*).

1. *train_test_split(X, y, test_size=0.3, random_state=42)*:

- X : Merupakan fitur (data *input*) dari dataset.
- y : Merupakan label atau target (data *output*) dari dataset.
- *test_size=0.3* : Menentukan proporsi data yang akan digunakan sebagai data pengujian. Dalam hal ini, 30% dari data akan digunakan untuk pengujian dan 70% untuk pelatihan.
- *random_state=42* : Menetapkan nilai acak untuk memastikan hasil pembagian data konsisten setiap kali kode dijalankan.

2. X_{train} , X_{test} , y_{train} , y_{test} :

- X_{train} : Data fitur untuk pelatihan.
- X_{test} : Data fitur untuk pengujian.
- y_{train} : Label/target untuk pelatihan.

- `y_test` : Label/target untuk pengujian.

Dengan menggunakan `train_test_split`, dataset dapat dipisahkan menjadi dua bagian: satu untuk melatih model dan yang lainnya untuk menguji kinerja model. Dengan menggunakan `source code` tersebut, dapat melihat secara detail performa dari model KNN yang telah dilatih, termasuk akurasi secara keseluruhan dan evaluasi klasifikasi untuk setiap kelas target.

Berikut hasil dari detail performa dari model KNN yang telah dilatih :

```
KNN Accuracy: 0.9073929961089494
```

```
KNN Classification Report:
```

	precision	recall	f1-score	support
0	0.92	0.98	0.95	1123
1	0.74	0.41	0.53	162
accuracy			0.91	1285
macro avg	0.83	0.69	0.74	1285
weighted avg	0.90	0.91	0.90	1285

Gambar 16. Hasil Akurasi Model KNN

Dari Gambar 16 yang menunjukkan hasil akurasi dari model KNN dapat dijelaskan bahwa model KNN berhasil mengklasifikasikan dengan baik sebesar 90.73% dari total data uji. Rata-rata tertimbang (*weighted avg*) yang memperhitungkan jumlah contoh di setiap kelas menunjukkan bahwa model memiliki presisi, *recall*, dan *F1-score* masing-masing sebesar 0.90, 0.91, dan 0.90.

3.3.3 Penerapan Model SVM

Penerapan *Support Vector Machine* (SVM) melibatkan inisialisasi model dengan memilih *kernel* yang sesuai (*linear*, *polynomial*, atau *radial basis function*). Model SVM membangun *hyperplane* terbaik untuk memisahkan kelas. Proses pembelajaran melibatkan penentuan *hyperplane* yang optimal. Evaluasi model SVM menggunakan metrik evaluasi yang sama dengan KNN. Dalam penelitian ini, *kernel* yang digunakan adalah *kernel linear*. Model SVM akan mencari *hyperplane* terbaik dalam ruang *input* dengan menentukan batas kelas yang memiliki jarak terbesar antara titik-titik data yang paling dekat. Jarak maksimum ini dihitung dengan menentukan margin *hyperplane*, yang merupakan jarak antara titik-titik terdekat dari *hyperplane* untuk setiap kelas. Margin ini mengukur seberapa jauh *hyperplane* berada dari titik-titik terdekat di masing-masing kelas. Pelatihan model SVM dilakukan dalam *google colab* sebagai berikut.

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.pipeline import Pipeline
from sklearn.metrics import classification_report,
accuracy_score
```

Gambar 17. Kode Penerapan SVM

```
# Inisialisasi model SVM dengan parameter terbaik
svm_model = SVC(kernel='linear', C=10, gamma=0.1,
random_state=42)
svm_model.fit(X_train_tfidf, y_train)
```

Gambar 18. Kode Inisialisasi SVM

`svm_model = SVC(kernel='linear', C=10, gamma=0.1, random_state=42)` menginisialisasi model SVM dengan *kernel linear* dan parameter regulasi *C* yang telah disesuaikan untuk menghasilkan performa terbaik berdasarkan eksperimen atau validasi sebelumnya. `svm_model.fit(X_train_tfidf, y_train)`: melatih model

SVM dengan data *training* yang telah ditransformasikan ke dalam bentuk fitur TF-IDF, dan label yang sesuai untuk mengklasifikasikan data berdasarkan fitur yang diberikan. Berikut hasil dari detail performa dari model SVM yang telah dilatih :

```
SVM Accuracy: 0.9198443579766536

SVM Classification Report:
              precision    recall  f1-score   support

     0         0.92         0.99         0.96         1123
     1         0.89         0.41         0.57          162

 accuracy          0.92         0.92         0.91         1285
 macro avg         0.91         0.70         0.76         1285
 weighted avg         0.92         0.92         0.91         1285
```

Gambar 19. Hasil Akurasi Model SVM

Dari Gambar 19 yang menunjukkan hasil akurasi dari model SVM dapat dijelaskan bahwa model SVM berhasil mengklasifikasikan dengan baik sebesar 91.98% dari total data uji. Rata-rata tertimbang (*weighted avg*) yang memperhitungkan jumlah contoh di setiap kelas menunjukkan bahwa model memiliki presisi, *recall*, dan *F1-score* masing-masing sebesar 0.92, 0.92, dan 0.91.

4. Pembahasan

4.1 Penerapan Model Gabungan

Pada tahap ini, dilakukan penerapan model gabungan menggunakan pendekatan *stacking* yang mengintegrasikan prediksi dari model *K-Nearest Neighbors* (KNN) dan *Support Vector Machine* (SVM). Langkah-langkah implementasi mencakup pelatihan model KNN dan SVM secara terpisah menggunakan data *training*, diikuti dengan pengumpulan hasil prediksi keduanya sebagai *input* untuk model meta atau model tingkat kedua.

Hasil prediksi dari model KNN dan SVM digunakan sebagai fitur tambahan untuk melatih model *stacking*. *Model stacking* merupakan sebuah pendekatan di dalam *machine learning* untuk menggabungkan beberapa model pembelajaran mesin untuk meningkatkan kinerja prediksi yang bertujuan untuk memanfaatkan kelebihan masing-masing model dasar guna meningkatkan kinerja prediksi secara keseluruhan. Penerapan *metode stacking* ini tidak hanya bertujuan untuk meningkatkan akurasi prediksi, tetapi juga untuk mengurangi variabilitas hasil prediksi dengan menggabungkan kekuatan dari dua pendekatan klasifikasi yang berbeda, yaitu KNN dan SVM.

Pelatihan model dalam *metode stacking* ini dilakukan menggunakan *Google Collab* untuk memfasilitasi komputasi yang skalabel dan memanfaatkan sumber daya komputasi yang tersedia secara *cloud*.

```
# Membuat model stacking dengan KNN dan SVM sebagai base
estimator
stacking_model = StackingClassifier(
    estimators=[('knn', knn_model), ('svm', svm_model)],
    final_estimator=LogisticRegression())
```

Gambar 20. Model Stacking Penggabungan KNN Dan SVM

Source code di atas berfungsi membangun sebuah *model stacking* yang mengkombinasikan dua model dasar, yaitu *K-Nearest Neighbors* (KNN) dan *Support Vector Machine* (SVM), untuk menghasilkan prediksi yang lebih akurat.

Prediksi dari kedua model dasar ini kemudian digabungkan menggunakan *Logistic Regression* sebagai *final estimator*. Dengan pendekatan ini, diharapkan *model stacking* dapat menangkap kekuatan dari masing-

masing *base estimator* dan mengurangi kelemahan mereka, sehingga menghasilkan kinerja prediktif yang lebih baik dibandingkan dengan menggunakan satu model saja.

Mengevaluasi kinerja *model stacking* dengan melakukan prediksi pada data uji dan menghitung akurasi-nya. Selain itu, juga menghasilkan dan mencetak laporan klasifikasi yang memberikan metrik evaluasi yang lebih mendetail seperti *precision*, *recall*, dan *f1-score* untuk setiap kelas.

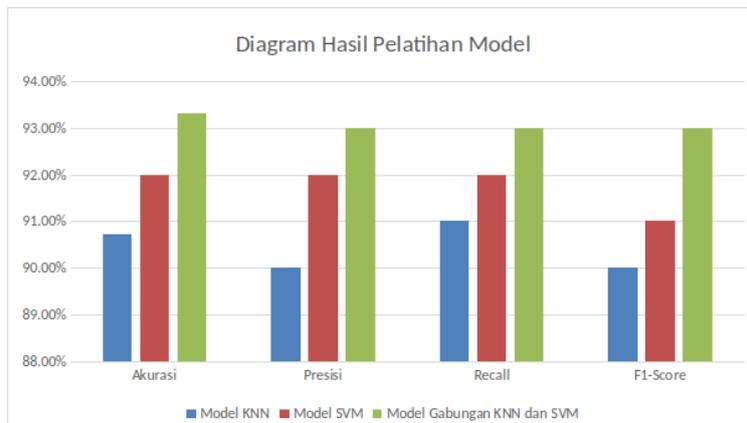
Stacking Model KNN and SVM Accuracy: 0.9330739299610895

Stacking Model Classification Report:

	precision	recall	f1-score	support
0	0.95	0.98	0.96	1123
1	0.80	0.62	0.70	162
accuracy			0.93	1285
macro avg	0.87	0.80	0.83	1285
weighted avg	0.93	0.93	0.93	1285

Gambar 21. Hasil Akurasi Model Gabungan KNN Dan SVM

Dari Gambar 21 di atas hasil *model stacking* yang menggabungkan KNN dan SVM menunjukkan kinerja yang sangat baik dengan akurasi sekitar 93.31% menandakan bahwa model mampu membuat prediksi yang benar pada sebagian besar data uji. Selain itu, penggunaan *model stacking* dengan KNN dan SVM sebagai *base estimators*, dikombinasikan dengan *Logistic Regression* yang digunakan untuk memodelkan probabilitas sebagai *final estimator*.



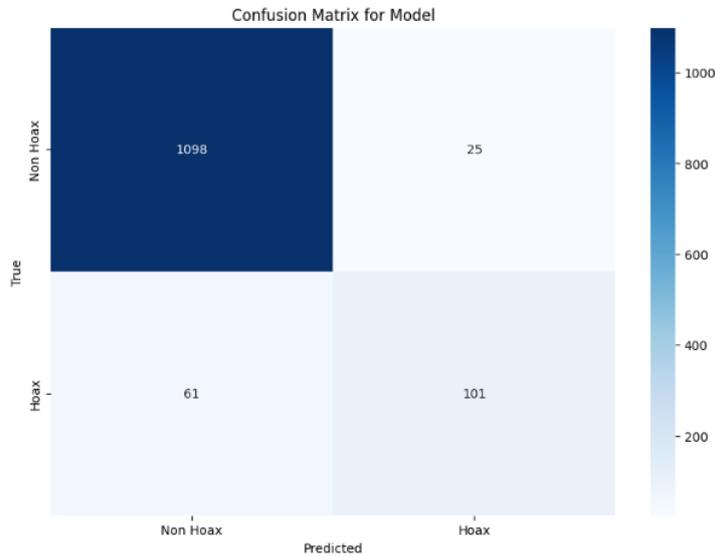
Gambar 22. Diagram Hasil Model

Berdasarkan diagram hasil pelatihan model, terdapat peningkatan yang signifikan pada semua metrik evaluasi ketika menggunakan Model Gabungan KNN dan SVM dibandingkan dengan model KNN dan SVM secara terpisah. Akurasi model gabungan mencapai lebih dari 93%, yang menunjukkan peningkatan sekitar 3% dibandingkan dengan Model KNN dan sedikit lebih tinggi dari Model SVM yang mencapai 92%. Presisi juga mengalami peningkatan lebih dari 3% dibandingkan dengan Model KNN, serta sedikit lebih baik daripada Model SVM. Selain itu, *recall* pada Model Gabungan hampir mencapai 93%, mengungguli kedua model individu. F1-Score, sebagai gabungan antara presisi dan *recall*, juga menunjukkan peningkatan yang konsisten pada Model Gabungan dibandingkan dengan KNN dan SVM. Secara keseluruhan, Model Gabungan KNN dan SVM memberikan peningkatan performa yang signifikan pada semua aspek, menjadikannya pilihan yang lebih baik dalam hal akurasi, presisi, *recall*, dan *F1-Score* dibandingkan dengan penggunaan model KNN atau SVM secara terpisah.

Penelitian yang telah dilakukan sebelumnya dalam mengembangkan sistem deteksi berita *hoax* dalam situs web yang menerapkan metode *Term Frequency-Inverse Document Frequency* dan *Support Vector Machine*. Dengan menggunakan *10-fold cross validation*, menunjukkan bahwa algoritma SVM memiliki nilai

F-Measure dan akurasi yang tinggi, masing-masing sebesar 0.804 dan 80.48% [11]. Sedangkan dari hasil penelitian ini yang dilakukan dengan metode gabungan KNN dan SVM mendapatkan akurasi yang lebih tinggi sebesar 93,31%, menandakan bahwa metode gabungan ini lebih baik digunakan dalam pengembangan sistem pendeteksi *hoax* daripada metode yang digunakan dalam penelitian sebelumnya.

4.2 Evaluasi Sistem



Gambar 23. Confusion Matrix

Confusion matrix diatas merupakan hasil dari model gabungan antara KNN dan SVM yang melalui tahap evaluasi. Tahapan selanjutnya adalah menghitung kinerja model seperti akurasi, presisi, *recall*, dan *F1 Score*. Berikut interpretasi dari *confusion matrix*:

Tabel 8. Hasil Confusion Matrix

	Non-Hoax(Prediksi)	Hoax(Prediksi)
True Non-Hoax	1098	25
True Hoax	61	101

1. Untuk Kelas *Non Hoax* (0)

- True Positives (TP) : 1098
- False Positives (FP) : 61
- False Negatives (FN) : 25
- True Negatives (TN) : 101

$$\begin{aligned}
 \text{Presisi} &= \frac{TP}{TP + FP} \\
 &= \frac{1098}{1098 + 61} \\
 &= \frac{1098}{1159} \\
 &= 0.947
 \end{aligned}$$

$$\begin{aligned}
 \text{Recall} &= \frac{TP}{TP + FN} \\
 &= \frac{1098}{1098 + 25} \\
 &= \frac{1098}{1123} \\
 &= 0.978
 \end{aligned}$$

$$\begin{aligned}
 F1 \text{ Score} &= 2 \times \frac{\text{presisi} \times \text{recall}}{\text{presisi} + \text{recall}} \\
 &= 2 \times \frac{0.947 \times 0.987}{0.947 + 0.987} \\
 &= 2 \times \frac{0.926}{1.925} \\
 &= 0.962
 \end{aligned}$$

2. Untuk Kelas *Hoax* (1)

True Positives (TP)	:	101
False Positives (FP)	:	25
False Negatives (FN)	:	61
True Negatives (TN)	:	1098

$$\begin{aligned}
 \text{Presisi} &= \frac{TP}{TP + FP} \\
 &= \frac{101}{101 + 25} \\
 &= \frac{101}{126} \\
 &= 0.802
 \end{aligned}$$

$$\begin{aligned}
 \text{Recall} &= \frac{TP}{TP + FN} \\
 &= \frac{101}{101 + 61} \\
 &= \frac{101}{162} \\
 &= 0.623
 \end{aligned}$$

$$\begin{aligned}
 F1 \text{ Score} &= 2 \times \frac{\text{presisi} \times \text{recall}}{\text{presisi} + \text{recall}} \\
 &= 2 \times \frac{0.802 \times 0.603}{0.802 + 0.603} \\
 &= 2 \times \frac{0.499}{1.425} \\
 &= 0.702
 \end{aligned}$$

3.

$$\begin{aligned}
 \text{Akurasi} &= \frac{TP + TN}{TP + FP + FN + TN} \\
 &= \frac{1098 + 101}{1098 + 25 + 61 + 101} \\
 &= \frac{1199}{1285} \\
 &= 0.933
 \end{aligned}$$

Model yang menggabungkan KNN dan SVM menunjukkan kinerja yang sangat baik dalam mendeteksi berita yang bukan *hoaks* dengan presisi dan *recall* yang tinggi. Namun, kinerja model dalam mendeteksi berita *hoax* masih bisa ditingkatkan. Dengan akurasi keseluruhan 93.3%, model ini efektif untuk deteksi *hoaks*.

Dikarenakan tujuan dari penelitian ini adalah menguji performa algoritma KNN dan SVM dalam membedakan mengklasifikasikan berita yang mengandung *hoax* atau pun *non-hoax*, maka model ini masih layak untuk diimplementasikan pada sistem.

5. Simpulan

Penelitian ini menggunakan 4283 berita, terdiri dari 3040 berita *non-hoax* dan 1243 berita *hoax*, dari periode Agustus 2023 hingga Juni 2024. Algoritma yang digunakan adalah gabungan KNN dan SVM. Model KNN berhasil mengklasifikasikan 90.73% dengan presisi, *recall*, dan *F1-score* masing-masing 0.90, 0.91, dan 0.90. Model SVM berhasil mengklasifikasikan 91.98% dengan presisi, *recall*, dan *F1-score* masing-masing 0.92, 0.92, dan 0.91.

Model Gabungan KNN dan SVM menunjukkan performa terbaik di semua metrik evaluasi. Dari segi akurasi, model ini mencapai nilai tertinggi di atas 93%, yang menunjukkan kemampuan yang sangat baik dalam mengklasifikasikan berita *hoax* dan *non-hoax* secara keseluruhan. Dalam hal presisi, model ini juga unggul dengan tingkat presisi sekitar 93%, yang berarti sebagian besar prediksi *hoax* yang dihasilkan oleh model adalah benar. *Recall* model ini juga menunjukkan hasil yang sama impresifnya, dengan sekitar 93%, yang mengindikasikan bahwa model mampu mendeteksi sebagian besar berita *hoax* yang ada. Selain itu, *F1-Score* model gabungan ini, yang merupakan keseimbangan antara presisi dan *recall*, juga berada di angka tertinggi sekitar 93%. Hal ini menegaskan bahwa Model Gabungan KNN dan SVM memberikan kinerja yang optimal dalam mendeteksi *hoax* dan *non-hoax* dengan tingkat akurasi, presisi, *recall*, dan *F1-Score* yang seimbang dan sangat baik.

Dari hasil penelitian performa algoritma KNN dan SVM dapat dengan baik dalam mengklasifikasikan berita yang mengandung *hoax* atau pun *non-hoax*.

Pustaka

- [1] F. Rahutomo, I. Y. R. Pratiwi, and D. M. Ramadhani, "Eksperimen Naïve Bayes Pada Deteksi Berita Hoax Berbahasa Indonesia," *J. Penelit. Komun. Dan Opini Publik*, vol. 23, no. 1, pp. 1–15, 2019, doi: 10.33299/jpkop.23.1.1805.
- [2] F. Tchakounté, K. A. Calvin, A. A. A. Ari, and D. J. F. Mbogne, "A smart contract logic to reduce hoax propagation across social media," *J. King Saud Univ. Inf. Sci.*, vol. 34, no. 6, pp. 3070–3078, 2022, doi: 10.1016/j.jksuci.2020.09.001.
- [3] R. Arif, "Internet as a hope or a hoax for emerging democracies: Revisiting the concept of citizenship in the digital age," *Procedia-Social Behav. Sci.*, vol. 236, pp. 4–8, 2016, doi: 10.1016/j.sbspro.2016.12.002.
- [4] B. Mudjiyanto and A. Dunan, "Media mainstream jadi rujukan media sosial," *Maj. Semi Ilm. Pop. Komun. Massa*, vol. 1, no. 1, pp. 21–34, 2020.
- [5] Kominfo, "Mengkominfo: Isu Hoax Pemilu Meningkatkan Hampir 10 Kali Lipat," 2023. https://www.kominfo.go.id/content/detail/52570/siaran-pers-no-422hmkominfo102023-tentang-menkominfo-isu-hoaks-pemilu-meningkat-hampir-10-kali-lipat/0/siaran_pers (accessed Feb. 20, 2024).
- [6] P. Utami, "Hoax in modern politics: The meaning of hoax in Indonesian politics and democracy," *J. Ilmu Sos. dan Ilmu Polit.*, vol. 22, no. 2, pp. 85–97, 2018, doi: 10.22146/jsp.34614.
- [7] J. Y. Hui, "Social media and the 2019 Indonesian elections: Hoax takes the centre stage," *Southeast Asian Aff.*, pp. 155–171, 2020, doi: 10.1355/9789814881319-010.
- [8] F. Putra and H. Patra, "Analisis Hoax pada Pemilu: Tinjauan dari Perspektif Pendidikan Politik," *Naradidik J. Educ. Pedagog.*, vol. 2, no. 1, pp. 95–102, 2023, doi: 10.24036/nara.v2i1.119.
- [9] J. H. T. Purba, "Identifikasi dan Klasifikasi SMS Hoax dengan Kombinasi Metode N-Gram dan Naïve Bayes." Universitas Sumatera Utara, 2019.
- [10] A. A. Permana et al., *Artificial Intelligence Marketing*. GET PRESS INDONESIA, 2023.
- [11] I. Ahmad, M. Basher, M. J. Iqbal, and A. Rahim, "Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection," *IEEE access*, vol. 6, pp. 33789–33795, 2018, doi: 10.1109/ACCESS.2018.2841987.

- [12] I. P. Santoso, "Sistem Pendeteksi Berita Hoax pada Situs Web Menggunakan Pendekatan Term Frequency-Inverse Document Frequency dan Support Vector Machine," Universitas Telkom, 2019.
- [13] F. S. Jumeilah, "Penerapan Support Vector Machine (SVM) untuk Pengkategorian Penelitian," *J. RESTI (Rekayasa Sist. Dan Teknol. Informasi)*, vol. 1, no. 1, pp. 19–25, 2017, doi: 10.29207/resti.v1i1.11.
- [14] V. Gurusamy and S. Kannan, "Preprocessing Techniques for Text Mining Preprocessing Techniques for Text Mining. 5," 2014.
- [15] P. D. Turney and M. L. Littman, "Measuring praise and criticism: Inference of semantic orientation from association," *ACM Trans. Inf. Syst. ...*, 2003, doi: 10.1145/944012.944013.
- [16] W. Chou and B. H. Juang, *Pattern recognition in speech and language processing*. books.google.com, 2003.
- [17] D. F. N. Anisa, I. Mukhlash, and M. Iqbal, "Deteksi Berita Online Hoax Covid-19 Di Indonesia Menggunakan Metode Hybrid Long Short Term Memory dan Support Vector Machine," *J. Sains dan Seni ITS*, vol. 11, no. 3, pp. A101–A108, 2023, doi: 10.12962/j23373520.v11i3.83227.
- [18] O. I. Gifari, M. Adha, I. R. Hendrawan, and F. F. S. Durrand, "Analisis Sentimen Review Film Menggunakan TF-IDF dan Support Vector Machine," *J. Inf. Technol.*, vol. 2, no. 1, pp. 36–40, 2022, doi: 10.46229/jifotech.v2i1.330.
- [19] A. Afriza and J. Adisantoso, "Metode Klasifikasi Rocchio untuk Analisis Hoax," *J. Ilmu Komput. dan Agri-Informatika*, vol. 5, no. 1, pp. 1–10, 2018, doi: 10.29244/jika.5.1.1-10.