

Penerapan *Naïve Bayes* & *Support Vector Machine* untuk Analisis Sentimen Pilpres Pada *Platform X*

Implementation of Naïve Bayes & Support Vector Machine for Election Sentiment Analysis on Platform X

Alamsyah Nur Alif^{1*}, Imelda Imelda², Wahyu Pramusinto³, Mardi Hardjianto⁴

^{1,2,3,4}Fakultas Teknologi Informasi
Universitas Budi Luhur

Email: ¹*2011501216@student.budiluhur.ac.id, ²imelda@budiluhur.ac.id, ³wahyu.pramusinto@budiluhur.ac.id,

⁴mardi.hardjianto@gmail.com

(* : corresponding author)

Abstrak

Pemilihan Presiden (Pilpres) 2024 menjadi topik yang sangat hangat diperbincangkan di media sosial, terutama di *Platform X*, yang banyak digunakan oleh masyarakat untuk menyuarakan opini mereka. Namun, dengan banyaknya opini yang tersebar, menjadi tantangan tersendiri untuk memahami sentimen masyarakat secara keseluruhan. Penelitian ini bertujuan untuk menganalisis sentimen masyarakat terhadap Pilpres 2024 di *Platform X* dengan membandingkan dua algoritma pembelajaran mesin, yaitu *Naïve Bayes* dan *Support Vector Machine* (SVM). Data yang digunakan dalam penelitian ini terdiri dari 503 tweet, yang setelah melalui proses *preprocessing* termasuk penghapusan data duplikat, tersisa 445 *tweet*. Data tersebut dibagi menjadi dua bagian, yaitu 80% untuk data latih dan 20% untuk data uji. Metode analisis yang digunakan mencakup perhitungan *Term Frequency-Inverse Document Frequency* (TF-IDF) pada kedua algoritma tersebut untuk mengoptimalkan representasi teks. Hasil penelitian menunjukkan bahwa algoritma SVM memiliki akurasi keseluruhan yang lebih tinggi, yaitu 91,01%, dibandingkan dengan *Naïve Bayes* yang hanya mencapai 42,70%. Namun, performa kedua algoritma bervariasi berdasarkan kelas sentimen. SVM menunjukkan kinerja sangat baik dalam memprediksi kelas netral dengan presisi sebesar 0,9101, *recall* sebesar 1,0, dan *F1-Score* sebesar 0,9529. Sebaliknya, *Naïve Bayes* menunjukkan presisi sebesar 0,9211 pada kelas netral tetapi memiliki *recall* yang rendah sebesar 0,4321, menghasilkan *F1-Score* sebesar 0,5882. Kedua algoritma menunjukkan keterbatasan signifikan dalam memprediksi kelas positif dan negatif, dengan SVM gagal memprediksi kedua kelas tersebut, sedangkan *Naïve Bayes* memiliki presisi dan *recall* yang sangat rendah. Temuan ini menunjukkan bahwa meskipun SVM lebih andal dalam memprediksi sentimen netral, diperlukan peningkatan metode atau eksplorasi algoritma lain untuk mencapai hasil yang lebih baik dalam prediksi semua kelas sentimen.

Kata Kunci: Analisis Sentimen, *Naïve Bayes*, *Support Vector Machine*, Pilpres 2024, *Platform X*, TF-IDF

Abstract

The 2024 presidential election has become a very hot topic of discussion on social media, especially on *Platform X*, which is widely used by people to voice their opinions. However, with so many opinions spread, it is a challenge to understand the sentiment of the community as a whole. This research aims to analyze public sentiment towards the 2024 presidential election on *Platform X* by comparing two machine learning algorithms, namely *Naïve Bayes* and *Support Vector Machine* (SVM). The data used in this study consisted of 503 tweets, which after going through a *preprocessing* process including the removal of duplicate data, 445 tweets remained. The data was divided into two parts, 80% for training data and 20% for test data. The analysis method used includes the calculation of *Term Frequency-Inverse Document Frequency* (TF-IDF) in both algorithms to optimize text representation. The results showed that the SVM algorithm had a higher overall accuracy of 91.01%, compared to *Naïve Bayes* which only reached 42.70%. However, the performance of both algorithms varied based on sentiment class. SVM showed excellent performance in predicting the neutral class with a precision of 0.9101, *recall* of 1.0, and *F1-Score* of 0.9529. In contrast, *Naïve Bayes* showed a precision of 0.9211 on the neutral class but had a low *recall* of 0.4321, resulting in an *F1-Score* of 0.5882. Both algorithms showed

significant limitations in predicting the positive and negative classes, with SVM failing to predict both classes, while Naïve Bayes had very low precision and recall. These findings suggest that while SVM is more reliable in predicting neutral sentiment, improvement of the method or exploration of other algorithms is required to achieve better results in the prediction of all sentiment classes.

Keywords: Analisis Sentimen, Naïve Bayes, Support Vector Machine, Pilpres 2024, Platform X, TF-IDF

1. PENDAHULUAN

Sentimen analisis adalah bagian dari *Natural Language Processing* (NLP) yang digunakan untuk mengenali opini atau pandangan dalam bentuk teks terkait isu atau kejadian, baik positif, negatif, maupun netral. Bidang ini cukup populer dalam penelitian karena dianggap bermanfaat di berbagai sektor, seperti dalam prediksi harga saham, isu politik, kepuasan produk atau layanan, dan analisis reputasi. Sebagai contoh, penelitian yang dilakukan menunjukkan perbandingan antara metode *Naïve Bayes* dan *Support Vector Machine* (SVM) dalam analisis sentimen di Twitter [1].

Pemilihan Presiden 2024 di Indonesia merupakan momen penting yang akan sangat berpengaruh dalam menentukan kebijakan dan transformasi sosial. Dalam konteks ini, media sosial, khususnya *Platform X*, menjadi sarana utama bagi masyarakat untuk berinteraksi, menyampaikan pandangan politik, dan mendapatkan informasi. Penelitian sebelumnya, seperti yang dilakukan, menunjukkan bahwa SVM dapat digunakan untuk menganalisis sentimen masyarakat terhadap program pemerintah di media sosial [2].

Platform X menyediakan platform dinamis bagi pengguna untuk berbagi pendapat, menciptakan diskusi, dan mempengaruhi opini publik. Volume besar data teks yang dihasilkan oleh pengguna *Platform X* memerlukan pendekatan analisis yang canggih untuk mendapatkan wawasan yang berharga. Sebagai contoh, menemukan bahwa metode *Naïve Bayes* dan SVM mampu mengidentifikasi sentimen netral yang dominan dalam percakapan terkait Pilpres Amerika 2020 di Twitter [3].

Juga, menggunakan SVM dalam studi mereka tentang opini masyarakat terkait pemilihan gubernur menunjukkan efektivitas metode ini dalam menganalisis sentimen dari data teks yang kompleks [4]. Selain itu, *text mining*, sebuah teknik yang memungkinkan ekstraksi informasi penting dari teks yang tidak terstruktur, juga memainkan peran penting dalam analisis sentimen terkait pemilihan presiden. Sebagai contoh, menerapkan algoritma *Naïve Bayes* dalam analisis sentimen terkait Pilpres Indonesia 2019, mencapai akurasi yang tinggi dengan dominasi sentimen positif [5].

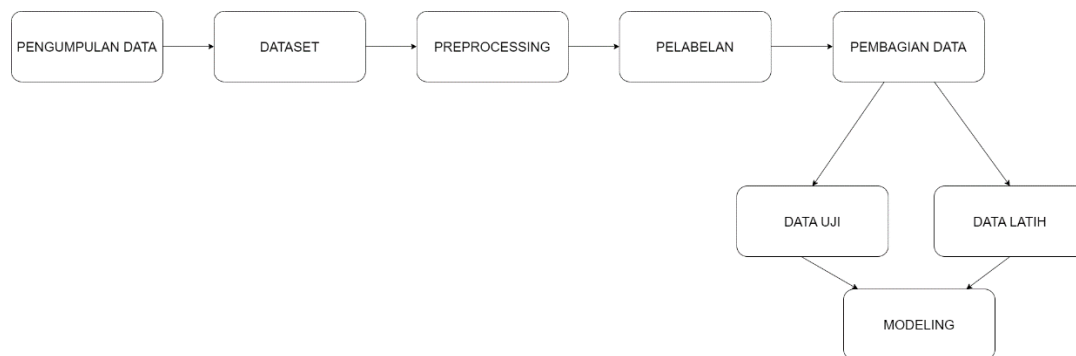
Pada penelitian lainnya, menemukan bahwa SVM mampu mengungkap dominasi sentimen negatif dalam diskusi mengenai Pilpres Indonesia 2019, yang menunjukkan kemampuan SVM dalam menangkap sentimen negatif yang mungkin lebih kompleks [6]. Pemilihan metode *Naïve Bayes* dan SVM didasarkan pada kemampuannya menghasilkan analisis yang konsisten dan akurat, terutama saat diterapkan pada data dengan volume besar dan kompleksitas tinggi. Sebagai contoh, penelitian menemukan bahwa kombinasi ekstraksi fitur TF-IDF dan *Lexicon* dapat meningkatkan performa analisis sentimen dalam ulasan film [7].

Penelitian lain menunjukkan bahwa algoritma *Naïve Bayes* memiliki performa yang kompetitif dalam klasifikasi data, yang juga relevan dalam analisis sentimen terkait Pilpres [8]. Sementara itu, mengaplikasikan *Naïve Bayes* dan SVM untuk menganalisis sentimen terkait boy band BTS di Twitter memberikan contoh bagaimana metode ini dapat diterapkan dalam konteks yang berbeda tetapi tetap relevan dalam analisis sentimen [9]. Penelitian juga menunjukkan bahwa kombinasi *Support Vector Machine* dengan teknik TF-IDF dapat digunakan untuk klasifikasi topik dengan akurasi tinggi, yang dapat diterapkan dalam analisis sentimen terkait Pemilihan Presiden 2024 di Indonesia [10].

2. METODE PENELITIAN

Gambar 1 menunjukkan tahapan-tahapan dalam metode penelitian yang dilakukan. Tahapan ini dimulai dengan pengumpulan data, di mana data yang diperlukan untuk penelitian dikumpulkan. Setelah data terkumpul, data tersebut diolah menjadi dataset. Tahapan selanjutnya adalah *preprocessing*, di mana data dalam dataset diproses untuk memastikan kualitas dan konsistensinya sebelum masuk ke tahap pelabelan. Setelah pelabelan, data kemudian dibagi menjadi dua bagian, yaitu data uji dan data

latih dalam tahap pembagian data. Data latih digunakan untuk *modeling*, sedangkan data uji digunakan untuk menguji model yang telah dibuat.



Gambar 1. Tahapan Metode

2.1 Pengumpulan Data

Data yang digunakan dalam penelitian ini berasal dari media sosial di *Platform X*, yang terdiri dari 503 *tweet* yang dikumpulkan dari 23 April 2024 hingga 24 April 2024. Proses *crawling* digunakan untuk mengumpulkan dataset ini berdasarkan satu parameter kata kunci yang terkait dengan pilpres 2024, yaitu “Pilpres2024”.

2.2 Preprocessing

Dalam tahap *preprocessing*, data *tweet* diolah melalui berbagai langkah penting untuk mempermudah analisis lanjutan. Langkah-langkah ini mencakup pembersihan data (*filtering*), konversi ke huruf kecil (*case folding*), *cleansing*, pemenggalan teks (*tokenizing*), normalisasi, penghapusan kata kunci (*stopword*), serta *stemming*. Ilustrasi dari proses *preprocessing* data dapat dilihat pada Gambar 2.



Gambar 2. Tahap *Preprocessing*

a. Pembersihan Data (*Filtering*)

Proses *filtering*, yang menghapus kata-kata dan simbol yang tidak relevan, jumlah *tweet* yang siap untuk analisis sentimen turun dari 503 menjadi 445. Ini memastikan bahwa data yang digunakan untuk analisis adalah data yang bersih dan relevan, yang memungkinkan hasil pemodelan sentimen yang lebih akurat dan signifikan.

b. Konversi ke huruf kecil (*Case Folding*)

Case folding adalah proses mengubah semua teks menjadi huruf kecil untuk konsistensi. Misalnya, kata "Komisi" akan diubah menjadi "komisi", sehingga menghilangkan perbedaan yang disebabkan oleh penggunaan huruf kapital. Proses ini penting dalam analisis teks untuk memastikan bahwa semua kata diperlakukan secara seragam, memudahkan algoritma dalam mengidentifikasi dan menganalisis kata-kata tanpa mempertimbangkan variasi kapitalisasi.

c. *Cleansing*

Proses *cleansing* melibatkan pembersihan dan penghapusan teks serta simbol yang tidak diperlukan untuk memastikan bahwa data yang akan dianalisis bersih dan relevan. Misalnya, dalam kasus teks yang mengandung emoji, tautan, *mention*, karakter khusus, dan angka seperti "@Portal_PWT 2024-2029 Komisi #Pilpres <https://t.co/7161kHyxR4>", proses *cleansing* akan menghapus elemen-elemen tersebut sehingga menghasilkan teks yang lebih bersih seperti "komisi". Dengan menghapus elemen-elemen yang tidak relevan, proses ini membantu meningkatkan akurasi dan efektivitas analisis teks, memudahkan algoritma dalam fokus pada informasi yang relevan.

d. Pemenggalan Text (*Tokenizing*)

Pada tahap ini, setiap kalimat dipecah menjadi kata-kata individu. Tujuan dari pemisahan ini adalah untuk memungkinkan pengolahan setiap kata secara terpisah pada tahap berikutnya, yang memungkinkan analisis yang lebih mendalam dan efektif. Sebagai contoh, kalimat "komisi pemilihan

umum" akan dipecah menjadi token-token individu yaitu "komisi, pemilihan, dan umum". Dengan memisahkan kalimat menjadi kata-kata yang lebih kecil, proses *tokenizing* memfasilitasi analisis kata per kata yang lebih terperinci dan akurat.

e. Normalisasi

Pada tahap ini, setiap kalimat yang telah dipecah menjadi kata-kata individu diperiksa dengan cermat untuk mendeteksi adanya singkatan atau kata-kata *informal*. Jika ditemukan, kata-kata tersebut akan diproses dan diperbaiki. Peneliti juga telah menambahkan beberapa kata ke dalam dataset kamus formal dan *informal* yang diambil dari *repository* yang tersedia di: <https://github.com/louisowen6/NLP-bahasa-resources?tab=readme-ov-file#stop-words>. Misalnya, kata "ri" akan diubah menjadi "republik indonesia".

f. Penghapusan Kata Kunci (*Stopword*)

Pada tahap penghapusan *stopword*, kata-kata yang kurang penting dan sering muncul dalam kalimat akan dihilangkan. Kata-kata yang dihapus ditandai dengan huruf tebal. Sebagai contoh, dalam kalimat "prabowo subianto dan gibran rakabuming raka", kata "dan" yang merupakan *stopword* akan dihapus, sehingga menghasilkan kalimat "prabowo subianto gibran rakabuming raka". Proses ini dilakukan dengan menggunakan pustaka sastrawi untuk memastikan bahwa hanya kata-kata yang relevan yang tersisa, yang dapat meningkatkan kualitas analisis teks.

g. *Stemming*

Tahap *stemming* mengubah kata-kata dengan imbuhan menjadi bentuk dasarnya. Sebagai contoh, kata "pemilihan" akan diganti menjadi "pilih", dan kata "menetapkan" akan diganti menjadi "tetap". Untuk menjamin akurasi dan konsistensi perubahan kata, proses *stemming* ini dilakukan dengan menggunakan pustaka sastrawi.

2.3 Pelebelan Data

Pada langkah ini, setiap data diberi label atau kategori sesuai dengan karakteristik kalimat tersebut ada dalam file. *Tweet* yang sudah melewati tahap *Preprocessing* hendak diberikan label positif jika mengandung kalimat positif tentang pilpres 2024, label negatif jika mengandung kalimat negatif tentang pilpres 2024, dan label netral jika tidak mengandung sentimen yang jelas.

2.4 Pembagian Data (*Split Data*)

Di tahap ini, data *tweet* yang sudah diberi label positif, negatif, dan netral segera dibagi menjadi data latih dan data uji, dengan perbandingan 80% data latih dan 20% data uji. Pembagian ini bertujuan untuk memastikan bahwa model yang dikembangkan dapat dilatih dengan jumlah data yang memadai dan diuji untuk mengevaluasi kinerjanya secara akurat. Proses pembagian ini dilakukan secara acak untuk menghindari bias dan memastikan representativitas data dalam kedua set tersebut.

2.5 TF-IDF

Term Frequency - Inverse Document Frequency (TF-IDF) merupakan metode yang sering digunakan untuk mengukur pentingnya sebuah kata dalam sebuah dokumen. TF-IDF membantu memberikan identifikasi khusus pada dokumen. Dalam penghitungan, metode ini mengevaluasi frekuensi kemunculan sebuah kata dalam sebuah dokumen. *Term Frequency* (TF) dihitung dengan membagi jumlah kemunculan kata tersebut dengan total kata dalam dokumen [7]

Oleh karena itu, nilai TF-IDF diperoleh dengan mengalikan TF dan IDF. Berikut adalah perhitungan TF-IDF:

$$W_{ij} = tf_{ij} \times \log \frac{D}{idf_i} \quad (1)$$

Keterangan

W_{ij} = Bobot TF-IDF

idf_i = *Inverse Dokumen Frequency*

tf_{ij} = Frekuensi Suatu Kata

2.6 Confusion Matrix

Metode *Confusion Matrix* digunakan untuk melakukan evaluasi klasifikasi penelitian ini. Ini dilakukan dengan membandingkan matriks prediksi dengan kelas asli, yang memiliki informasi nyata dan prediksi nilai klasifikasi. Setelah sistem berhasil mengklasifikasikan *tweet*, diperlukan ukuran untuk mengevaluasi ketepatan dan akurasi klasifikasi. Untuk memastikan keakuratan hasil, proses pengujian klasifikasi menggunakan metode *Confusion Matrix* ini sangat penting [2]. Penelitian ini [4] mendukung penggunaan metode tersebut dalam mengevaluasi kinerja model klasifikasi.

Tabel 1. *Confusion Matrix*

		Kelas Prediksi		
		Positif	Negatif	Netral
Kelas Aktual	Positif	TP	FP ₁	FP ₂
	Negatif	FN _{g1}	TNg	FN _{g2}
	Netral	FN _{t1}	FN _{t2}	TNt

Terdapat 9 istilah dalam *confusion matrix* di Tabel 1 untuk klasifikasi, yaitu TP: Jumlah data yang sebenarnya Positif dan diprediksi sebagai Positif; FP₁: Jumlah data yang sebenarnya Positif tetapi diprediksi sebagai Negatif; FP₂: Jumlah data yang sebenarnya Positif tetapi diprediksi sebagai Netral; TNg: Jumlah data yang sebenarnya Negatif dan diprediksi sebagai Negatif; FN_{g1}: Jumlah data yang sebenarnya Negatif tetapi diprediksi sebagai Positif; FN_{g2}: Jumlah data yang sebenarnya Negatif tetapi diprediksi sebagai Netral; TNt: Jumlah data yang sebenarnya Netral dan diprediksi sebagai Netral; FN_{t1}: Jumlah data yang sebenarnya Netral tetapi diprediksi sebagai Positif; FN_{t2}: Jumlah data yang sebenarnya Netral tetapi diprediksi sebagai Negatif. Perhitungan tiga parameter, yaitu akurasi, presisi, dan *recall*, akan dilakukan menggunakan rumus *Confusion Matrix* untuk mengukur ketiga metrik tersebut.

$$akurasi = \frac{TP + TNg + TNt}{TP + FP_1 + \dots + FN_{g2} + TNg} \quad (2)$$

$$presisi\ positif = \frac{TP}{TP + FP_1 + FP_2} \quad (3)$$

$$presisi\ negatif = \frac{TNg}{TNg + FN_{g1} + FN_{g2}} \quad (4)$$

$$presisi\ netral = \frac{TNt}{TNt + FN_{t1} + FN_{t2}} \quad (5)$$

$$recall\ positif = \frac{TP}{TP + FN_{g1} + FN_{t1}} \quad (6)$$

$$recall\ negatif = \frac{TNg}{TNg + FN_{t2} + FP_1} \quad (7)$$

$$recall\ netral = \frac{TNt}{FP_2 + FN_{g2} + TNt} \quad (8)$$

2.7 Naïve Bayes Classifier

Algoritma *Naive bayes* adalah metode klasifikasi probabilistik yang menghitung probabilitas dengan menganalisis frekuensi dan kombinasi nilai dalam suatu kumpulan data. Algoritma ini didasarkan pada teorema *Bayes* dan mengasumsikan bahwa semua variabel bersifat independen dengan mempertimbangkan nilai variabel kelas. Asumsi independensi bersyarat ini jarang berlaku dalam aplikasi dunia nyata, sehingga disebut "*Naive*", namun algoritma ini cenderung belajar dengan cepat dalam berbagai masalah klasifikasi terkontrol. Teorema *Bayes* adalah rumus matematika yang digunakan untuk menentukan probabilitas bersyarat (Persamaan 1), dinamai sesuai dengan ahli matematika Inggris abad ke-18, Thomas Bayes [8].

$$P(A|B) = \frac{P(A)P(B|A)}{P(A)} \quad (9)$$

Keterangan

$P(A|B)$ = Peluang terjadinya kejadian A ketika kejadian B terjadi

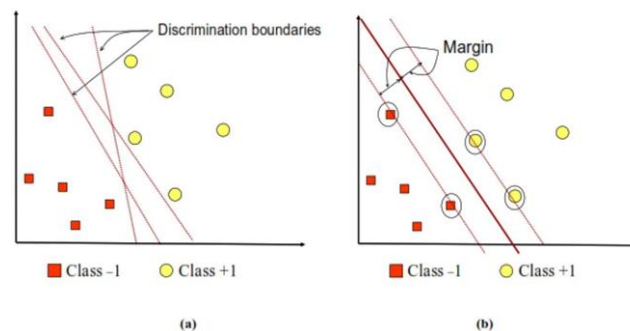
$P(A)$ = Peluang terjadinya A

$P(B|A)$ = Peluang terjadinya kejadian B bila kejadian A terjadi

$P(B)$ = Peluang terjadinya B

2.8 Support Vector Machine

Algoritma *Support Vector Machine* (SVM) adalah salah satu teknik yang digunakan dalam pengklasifikasian data. Prinsip utama SVM adalah untuk menemukan *hyperplane* yang memaksimalkan jarak antara dua kelas data dalam ruang fitur. SVM menggunakan ruang fitur berdimensi tinggi dan berfokus pada fungsi linear sebagai hipotesis untuk memisahkan kelas data noviana [9]



Gambar 3. Penentuan *Hyperlane* Terbaik

Menampilkan beberapa pola milik dua kelas, yaitu -1 dan +1 pada Gambar 3. Pola pada kelas -1 diwakili oleh kotak merah, sedangkan pola pada kelas +1 diwakili oleh kotak kuning berbentuk bulat. Untuk menyelesaikan masalah klasifikasi, coba tentukan sebuah garis (*hyperlane*) yang memisahkan kedua kelas tersebut. Gambar a menunjukkan batas pemisah dari berbagai opsi yang tersedia. *Hyperlane* merupakan garis pemisah terbaik antara kedua kelompok. Menemukan *hyperlane* ini bisa dilakukan dengan mencari *margin hyperplane* dan titik maksimal. Garis solid yang terlihat pada gambar b menunjukkan *hyperlane* yang optimal, karena terletak tepat di antara kedua kelas, sementara *support vector* diwakili oleh titik merah dan kuning yang berada di dalam lingkaran hitam [10].

Hyperlane klasifikasi linear pada *Support Vector Machine* dinyatakan sebagai:

$$f(x) = w \cdot x + b \quad (10)$$

Dari persamaan tersebut, diperoleh pertidaksamaan untuk kelas +1 (positif) sebagai berikut:

$$w \cdot x + b \leq +1 \quad (11)$$

Sementara itu, pertidaksamaan untuk kelas -1 adalah:

$$w \cdot x + b \leq -1 \quad (12)$$

Di sini, w adalah bobot, b adalah bias, dan x adalah jumlah data.

2.9 Gradient Descend

Gradient Descent adalah proses pembelajaran iteratif yang bertujuan untuk meminimalkan fungsi tujuan dengan mengikuti arah penurunan paling curam, sehingga koefisien terbaik untuk pemodelan dapat ditemukan. *Gradient Descent* digunakan untuk meminimalkan fungsi biaya dalam berbagai algoritma *machine learning*. Rumus yang diterapkan pada *Support Vector Machine* adalah sebagai berikut:

$$w_2 = w_1 - L \cdot \frac{dj}{dw} \quad (13)$$

$$b_2 = b_1 - L \cdot \frac{dj}{db} \quad (14)$$

Keterangan

- w = Bobot
- b = Bias
- L = Kecepatan *Learning* (*hyperparameter*)
- $\frac{dj}{dw}$ = Turunan parsial dari fungsi terhadap bobot
- $\frac{dj}{db}$ = Turunan parsial dari fungsi terhadap bias

2.10 MultiClass Support Vector Machine

Support Vector Machine pertama kali diperkenalkan oleh Vapnik pada tahun 1992. Pada awalnya, *Support Vector Machine* hanya mampu mengklasifikasikan data ke dalam dua kelas. Namun, seiring perkembangan waktu, *Support Vector Machine* kini mampu melakukan klasifikasi ke dalam banyak kelas (*multiclass*). Terdapat dua Pendekatan untuk menerapkan *MultiClass Support Vector Machine*. Pendekatan yang umum digunakan adalah *One Versus Rest* dan *One Versus One* [9].

a. One Versus Rest

Metode *One Versus Rest* atau *One-Against-All* membangun k model *Support Vector Machine* biner, di mana k adalah jumlah kelas. Model *Support Vector Machine* ke-m dilatih dengan semua contoh kelas m diberi label positif, sementara yang lainnya diberi label negatif. Berdasarkan data latih $l(x_1, y_1), \dots, (x_l, y_l)$ di mana $x_i \in \mathbb{R}^d, y_i \in \{1, 2, \dots, k\}$ merupakan kelas dari permasalahan tersebut dapat diselesaikan dengan:

$$(w_m)^T \phi(x_m) + b_m \geq 1 - \xi_m \rightarrow y_i = m \quad (15)$$

$$(w_m)^T \phi(x_m) + b_m \geq 1 - \xi_m \rightarrow y_i \neq m \quad (16)$$

$$\xi_m \geq 0, i = 1, \dots, l \quad (17)$$

b. One Versus One

Metode *One Versus One* atau *One-Against-One* membangun $\frac{k(k-1)}{2}$ model klasifikasi biner, di mana k adalah jumlah kelas. Setiap model klasifikasi dilatih menggunakan data dari dua kelas. Data latih dari kelas m dan n diselesaikan dengan:

$$(w_{mn})^T \phi(x_i) + b_{mn} \geq 1 - \xi_{mn} \rightarrow y_i = m \quad (18)$$

$$(w_{mn})^T \phi(x_i) + b_{mn} \leq 1 - \xi_{mn} \rightarrow y_i = n \quad (19)$$

$$\xi_{mn} \geq 0 \quad (20)$$

3. HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan nilai akurasi setiap algoritma dalam memprediksi label (kelas) pada data uji, yang diperoleh setelah mengikuti langkah-langkah yang digambarkan pada Gambar 1, mulai dari pengumpulan data dari Twitter. Data dikumpulkan melalui proses *crawling* menggunakan pustaka *tweet harvest* dari 23 April 2024 hingga 24 April 2024 dengan kata kunci "Pilpres2024". Hasil *crawling* kemudian disimpan dalam format file excel (.CSV). Data tersebut akan diimpor ke sistem *database* untuk pemrosesan lebih lanjut. Contoh sampel dari data pilpres 2024 dapat dilihat pada Tabel 2 berikut ini.

Tabel 2. *Dataset*

Tweet	Username	Created At
Wapres Maruf Apresiasi Prabowo Rangkul Semua Pihak Usai Pilpres 2024.	IDNTimes	Wed Apr 24 07:58:06 +0000 2024
Tak Hadiri Penetapan Prabowo-Gibran Pemenang Pilpres 2024, Ganjar: Tak Dapat Undangan	finindonesia	Wed Apr 24 07:54:54 +0000 2024

Yang menarik dari akhir perjalanan sengketa Pilpres 2024 ini, adalah terkait respon semua pihak, bahwa nampak sekali sikap kenegaraan mereka. Misalnya masing-masing capres dan cawapres dari 01 dan 03 begitu legowo dengan putusan MK ini	Perju4ng4n	Wed Apr 24 07:59:54 +0000 2024
---	------------	--------------------------------

Data yang baru dikumpulkan dari *Platform X* sering kali masih mengandung simbol, tautan, dan elemen lainnya yang tidak relevan. Oleh sebab itu, proses *preprocessing* diperlukan untuk membersihkan data dan memastikan konsistensinya sebelum analisis lebih lanjut. Hasil dari *preprocessing* ini dapat dilihat pada Gambar 4, yang memperlihatkan data setelah dibersihkan dan siap untuk dianalisis.

Tweet	Tweet Bersih
Penetapan Hasil Pilpres 2024, 4.266 Personel Gabungan Amankan Kantor KPU https://t.co/3tVa88HJvZ	tetap hasil pilpres personel gabung aman kantor komisi pilih umum
Prabowo Subianto akhirnya resmi memenangkan Pilpres 2024 setelah tiga kali kalah dalam kontestasi pemilihan presiden pada periode sebelumnya. Pada pidato penetapan sebagai presiden terpilih dalam Pilpres 2024, Ketua Umum Partai Gerindra itu bilang ke Paskon 01, Anies https://t.co/YbbbKuOxWz	prabowo subianto akhir resmi menang pilpres tiga kali kalah kontestasi pilih presiden periode belum pidato tetap presiden pilih pilpres ketua umum partai gerindra bilang paskon anies
Terkait Putusan MK Soal, Sengketa Pilpres 2024, Ini Pernyataan PBNU https://t.co/0ulpKmk5FF	kait putus mahkamah konstitusi soal sengketa pilpres nyata urus besar nahdlatul ulama

Gambar 4. Hasil Preprocessing

Setelah proses *preprocessing*, data yang telah dibersihkan siap untuk tahap berikutnya, yaitu pelabelan. Terdapat dua metode untuk menentukan kelas sentimen dari teks yang telah diproses: metode pertama menggunakan skor dari kamus sentimen, sementara metode kedua melibatkan penilaian oleh ahli atau pakar. Hasil dari pelabelan ini dapat dilihat pada Gambar 5, yang menunjukkan data yang telah diberi label berdasarkan metode yang diterapkan.

Tweet Bersih	Sentimen Automation	Sentimen Pakar
tetap hasil pilpres personel gabung aman kantor komisi pilih umum	Netral	Netral
prabowo subianto akhir resmi menang pilpres tiga kali kalah kontestasi pilih presiden periode belum pidato tetap presiden pilih pilpres ketua umum partai gerindra bilang paskon anies	Netral	Negatif
kait putus mahkamah konstitusi soal sengketa pilpres nyata urus besar nahdlatul ulama	Netral	Netral
tarik akhir jalan sengketa pilpres adalah kait respon semua pihak nampak sekali sikap kenegaraan misal masingmasing capres cawapres dan legowo putus mahkamah konstitusi	Netral	Positif

Gambar 5. Hasil Labelling

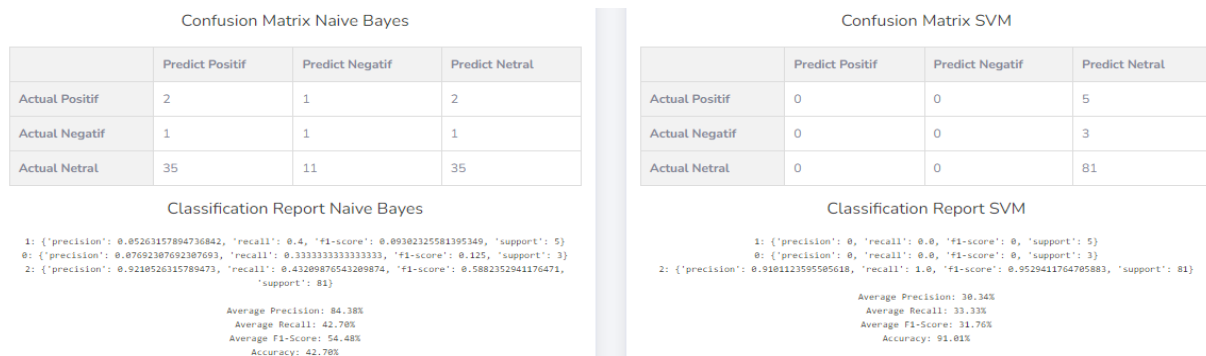
Setelah tahap *labelling* selesai, proses selanjutnya adalah *split* data. Dari total 445 *tweet* yang dilabeli, data ini dibagi dengan rasio 80:20, menghasilkan 356 *tweet* sebagai data *training* dan 89 *tweet* sebagai data *testing*. Sebagai ilustrasi, Tabel 3 di bawah ini menyajikan contoh data dari data *training* dan data *testing* yang telah dihasilkan.

Tabel 3. Data Training dan Data Testing

Data Training	Data Testing
prabowo pilpres selesai ajak satu bangun bangsa	prabowo pilpres selesai ajak satu bangun bangsa
konyol anjing babi bodoh nyata orang anda hakim busuk putus mahkamah konstitusi anies gibran pilpres	
kita hormat putus mahkamah konstitusi kait sengketa pilpres	

Dataset ini akan dianalisis menggunakan algoritma *Naïve Bayes Classifier* dan *Support Vector Machine* untuk mengklasifikasikan sentimen, dengan data pelatihan yang disimpan menggunakan metode TF-IDF sebagai bobot untuk setiap kata. Hasil klasifikasi dari kedua algoritma tersebut kemudian dianalisis dalam sebuah *Confusion Matrix* untuk mengukur tingkat akurasi. Gambar 6

memperlihatkan hasil dalam bentuk *Confusion Matrix*, yang menunjukkan jumlah prediksi benar dan salah dari masing-masing algoritma serta memberikan gambaran jelas tentang kinerja dan akurasi keduanya.



Gambar 6. Hasil *Confusion Matrix* 2 Algoritma

Berdasarkan *confusion matrix* yang ditampilkan, terlihat jelas perbedaan mencolok dalam tingkat akurasi antara model yang diuji menggunakan *Naïve Bayes Classifier* dan *Support Vector Machine*. *Support Vector Machine* berhasil mencapai akurasi sebesar 91.01%, dihitung dari total prediksi yang benar dibagi total prediksi yaitu $(0+0+82) / (0+0+82+0+0+3+0+0+0) = 91.01\%$. Sementara itu, *Naïve Bayes Classifier* hanya berhasil mencapai akurasi sebesar 42.70%, dihitung dari total prediksi yang benar dibagi total prediksi yaitu $(2+1+35) / (2+1+2+1+1+1+35+11+35) = 42.70\%$.

4. KESIMPULAN

Evaluasi yang dilakukan mengungkapkan bahwa metode *Naïve Bayes* dan *Support Vector Machine* (SVM) dapat diterapkan secara efektif untuk menganalisis opini masyarakat di *Platform X* mengenai Pilpres 2024 di Indonesia. Dari hasil pengujian, *Support Vector Machine* berhasil mencapai tingkat akurasi yang tinggi, yaitu 91,01%, menunjukkan keunggulannya dalam mengklasifikasikan sentimen dengan lebih akurat. Di sisi lain, *Naïve Bayes* hanya mencapai tingkat akurasi sebesar 42,70%, yang menunjukkan performa yang jauh lebih rendah dibandingkan dengan SVM dalam konteks analisis sentimen ini.

DAFTAR PUSTAKA

- [1] M. I. Fikri, T. S. Sabrila, and Y. Azhar, ‘Perbandingan Metode Naïve Bayes dan Support Vector Machine pada Analisis Sentimen Twitter’, *Smatika Jurnal*, vol. 10, no. 02, pp. 71–76, 2020.
- [2] N. Hendrastuty, A. R. Isnain, and A. Y. Rahmadhani, “Analisis Sentimen Masyarakat Terhadap Program Kartu Prakerja Pada Twitter Dengan Metode Support Vector Machine”, vol. 6, no. 3, pp. 150-155, 2021.
- [3] G. Nugroho, D. T. Murdiansyah, and K. M. Lhaksana, “Analisis Sentimen Pemilihan Presiden Amerika 2020 di Twitter Menggunakan Naïve Bayes dan Support Vector Machine”, 2021.
- [4] A. E. S. Saputro, K. A. Notodiputro, and I. Indahwati, “Study of Sentiment of Governor’s Election Opinion in 2018”, *IJSRSET: International journal of Scientific Research in Science, Engineering and Technology*, vol. 4, no. 11, pp. 231–238, 2018.
- [5] F. A. Wenando, R. Hayami, and A. J. Anggrawan, “Analisis Sentimen Pada Pemerintahan Terpilih Pada Pilpres 2019 Ditwitter Menggunakan Algoritme Naïvebayes”, *JURTEKSI (Jurnal Teknologi dan Sistem Informasi)*, vol. 7, no. 1, pp. 101–106, 2020.
- [6] R. I. Oetama, “Analisis Sentimen Warganet Twitter Terhadap Pemilihan Presiden Indonesia”, *Skripsi*, Prodi Studi Matematika, UIN Syarif Hidayatullah, Jakarta, 2023.
- [7] S. A. Pratomo, S. Al Faraby, and M. D. Purbolaksono, “Analisis Sentimen Pengaruh Kombinasi Ekstraksi Fitur TF-IDF dan Lexicon Pada Ulasan Film Menggunakan Metode KNN”, 2021.
- [8] M. M. Saritas, and A. Yasar, “Performance Analysis of ANN and Naive Bayes Classification Algorithm for Data Classification”, *IJISAE: International Journal of Intelligent Systems and Applications in Engineering*, vol. 7, no. 2, pp. 88–91, 2019.

- [9] R. Noviana and I. Rasal, “Penerapan Algoritma Naive Bayes Dan Svm Untuk Analisis Sentimen Boy Band Bts Pada Media Sosial Twitter”, *JTS: Jurnal Teknik dan Science*, vol. 2, no. 2, pp. 51-60, 2023.
- [10] A. Dewan, D. Wibiyanto, and A. Wibowo, “Penerapan Algoritma Multiclass Support Vector Machine dan TF-IDF Untuk Klasifikasi Topik Tugas Akhir”, *SKANIKA: Sistem Komputer dan Teknik Informatika*, vol. 6, no. 1, pp. 42–51, 2023.