ANALISIS SENTIMEN UNTUK DETEKSI UJARAN KEBENCIAN PADA MEDIA SOSIAL TERKAIT PEMILU 2024 MENGGUNAKAN METODE SUPPORT VECTOR MACHINE

Raflizar Deswandi Yahya, Suryo Adi Wibowo, Nurlaily Vendyansyah
Teknik Informatika, Institut Teknologi Nasional Malang
Jalan Raya Karanglo km 2 Malang, Indonesia
2018112@scholar.itn.ac.id

ABSTRAK

Analisis sentimen adalah teknik pemrosesan bahasa alami yang digunakan untuk mengukur sentimen atau opini tentang teks. *Support Vector Machine* (SVM) adalah teknik *machine learing* yang dapat digunakan untuk menyelesaikan masalah klasifikasi dan regresi. Mendekati pemilihan umum (pemilu) 2024, perdebatan politik akan menjadi topik utama pemberitaan dan media sosial. Pengguna media sosial bebas memperoleh informasi, mengemukakan pendapat, dan mengkritik dengan berbagai cara. Namun karena tingginya kebebasan berekspresi di media sosial, tanpa disadari pengguna dapat membuat konten yang mengandung unsur ujaran kebencian. Pada penelitian ini mengembangkan sebuah sitem berbasis *website* yang mampu mendeteksi ujaran kebencian dalam kalimat yang berbahasa indonesia. Metode klasifikasi sentimen yang akan diimplementasikan pada *website* ini adalah *Support Vector Machine* (SVM). Data yang digunakan untuk melatih dan menguji sistem ini berasal dari komentar dan tweet terkait pemilu 2024. Hasil evaluasi pengujian dengan menggunakan 100 data kalimat teratas dari data training untuk dibandingkan hasilnya dengan hasil klasifikasi SVM, hasil evaluasi dari pengujian tersebut menunjukkan keefektifan SVM dengan nilai recall 76%, presisi 96%, dan akurasi 81%. Hasil dari klasifikasi SVM sangat dipengaruhi dari jumlah data *training* dan proses yang dilakukan pada tahap *text preprocessing.*

Kata kunci : *Analisis sentimen, Support Vector Machine, Pemilu, Ujaran Kebencian, Media Sosial*

1. PENDAHULUAN

Analisis sentimen adalah metode dalam pemrosesan bahasa alami yang dipakai untuk mengekstrak dan mengukur sentimen atau opini dari teks. Banyak penelitian yang sudah dikembangkan untuk mengembangkan metode analisis sentimen, termasuk metode Support Vector Machine (SVM) dan pendekatan berbasis kosakata, serta pemanfaatan arsitektur analisis sentimen berbasis deep learning. [1].

Pada masa kini, media sosial sudah menjadi kebutuhan dalam kehidupan sehari-hari, selain sebagai sarana untuk kebutuhan pribadi, media sosial juga sebagai alat untuk keperluan bisnis dan berbagai aspek lainnya. Media sosial populer seperti Facebook, Instagram, Whatsapp, dan Twitter menyediakan wadah di mana individu dapat terhubung dan berbagi sudut pandang mereka. Dengan ragam konten yang tersedia di platform ini, termasuk gambar, komentar, emotikon, video, dan lain sebagainya, masyarakat dapat dengan leluasa menyampaikan pendapat mereka. [2].

Mendekati pemilihan umum (pemilu) 2024, perdebatan politik akan menjadi topik utama pemberitaan dan media sosial. Pengguna media sosial bebas memperoleh informasi, mengemukakan pendapat, dan mengkritik dengan berbagai cara. Namun karena tingginya kebebasan berekspresi di media sosial, tanpa disadari pengguna dapat membuat konten yang mengandung unsur ujaran kebencian. Fenomena ini bisa menimbulkan permasalahan serius, mengingat ujaran kebencian diatur dalam UU ITE [3]. Pada penelitian sebelumnya yang dilakukan oleh [4]. Riset ini difokuskan pada keyword pencarian "pilpres 2019" terkait dengan beberapa kota di Indonesia dan mengumpulkan total data sebanyak 5055. Hasil klasifikasi menunjukkan bahwa sentimen yang dianggap tidak relevan sebanyak 11 . 3 %, yang setara dengan 573 data. Sementara itu, sentimen negatif mencapai 35 . 4 % dengan 1786 data, sentimen netral mencapai 26 . 7 % dengan 1350 data, dan sentimen positif mencapai 26 . 6 % dengan 1343 data. Di kelima kota yang diteliti, sentimen negatif mencatatkan persentase tertinggi, yaitu sebesar 35 . 4 %.

Ujaran kebencian merujuk pada tindakan kejahatan yang melibatkan penggunaan kata-kata kasar dan penghinaan terhadap individu atau kelompok berdasarkan kriteria seperti ras, jenis kelamin, orientasi seksual, etnis, atau agama. [5]. Penelitian yang dilakukan oleh [6] dengan judul Analisis Jenis dan Makna Pragmatis Ujaran Kebencian di Media Sosial Twitter Tahun 2023 bertujuan untuk memberikan pemahaman kepada masyarakat mengenai ujaran kebencian yang muncul di media sosial. Hasil penelitian ini menyajikan solusi untuk membantu masyarakat memahami jenis-jenis ujaran kebencian, seperti penghinaan, pencemaran nama baik, penodaan agama, gangguan, berita palsu, dan provokasi/hasutan. Selain itu, penelitian ini juga mengungkapkan makna pragmatis, seperti makna sarkasme, gambaran pemimpin, penggambaran menyombongkan diri, pertanyaan, kekecewaan, dan ajakan. Kehadiran ujaran kebencian ini memiliki

potensi mengancam integritas pemilu, merugikan debat publik, dan bahkan dapat memicu ketegangan sosial.

Support Vector Machine (SVM) adalah teknik *machine learing* yang digunakan untuk memecahkan masalah klasifikasi dan regresi. SVM bekerja dengan mencari *hyperplane* atau fungsi diskriminan yang optimal untuk memisahkan kelas-kelas dalam ruang berdimensi N, dimana N adalah jumlah data. Dalam penelitian Primadani A. dan Retno W. berjudul "Analisis Sentimen Wacana Pemindahan Ibu Kota Indonesia Menggunakan Algoritma Support Vector Machine (SVM)" hasil pengujian menggunakan SVM terhadap 1236 tweet (404 positif dan 832 negatif) menunjukkan akurasi sebanyak 96.68 %, presisi sebanyak 95.82 %, recall sebanyak 94.04 %, dan AUC sebanyak 0 . 979. Dari hasil pengujian, disimpulkan bahwa SVM lebih unggul dibandingkan dengan metode sebelumnya, yaitu BM25 + KNN dan Naive Bayes. Keunggulan utama SVM terletak pada kemampuannya menangani masalah data berdimensi tinggi dan sampel data yang besar. Selain itu, SVM sering digunakan karena kompleksitas komputasinya yang efisien dibandingkan dengan beberapa algoritma lainnya. Kelebihan lain SVM adalah kemampuannya menentukan jarak menggunakan Support Vector, sehingga dapat memisahkan dua kelompok data yang berbeda dengan baik [7].

Penelitian ini bertujuan untuk mengembangkan sistem berbasis website yang menggunakan metode Support Vector Machine (SVM) untuk mendeteksi ujaran kebencian dalam kalimat berbahasa Indonesia. Data yang digunakan berasal dari komentar dan tweet yang terkait dengan pemilu tahun 2024. Proses pengembangan sistem mencakup pengumpulan data dari berbagai sumber, preprocessing data untuk membersihkan dan mengubah teks menjadi representasi numerik, pembagian data menjadi data latih dan data uji, pelatihan model SVM, evaluasi performa model menggunakan metrik yang relevan, dan pengembangan website untuk implementasi sistem deteksi ujaran kebencian. Dengan demikian, penelitian ini bertujuan untuk menyediakan solusi teknologi yang efektif dalam mengatasi masalah ujaran kebencian dalam konteks bahasa Indonesia, terutama sehubungan dengan pemilu tahun 2024.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Menurut penelitian yang dilakukan oleh Maria Mega Mala O. dkk "Analisis sikap pengguna Twitter terhadap Covid-19 di Indonesia menggunakan metode *Naïve Bayes Classifier* (NBC)". Tujuan dari penelitian ini adalah untuk mengembangkan sistem yang mampu melakukan analisis Bayesian naif terhadap perasaan pengguna Twitter terhadap Covid-19. Ketika diuji dengan 75 tweet, hasil pengukuran menunjukkan nilai recall sebesar 32%, presisi sebesar 80%, F-measure sebesar 45%, dan presisi sebesar 36% [8].

Menurut [9], dalam penelitiannya yang berjudul "Analisis Sentimen *Cyberbullying* di Media Sosial Twitter Menggunakan Metode *Support Vector Machine*", mereka menciptakan sebuah platform web yang dapat mengklasifikasikan sentimen *cyberbullying* di media sosial Twitter dengan menggunakan Metode *Support Vector Machine*. Uji coba sistem dengan 100 tweet memungkinkan sistem melakukan klasifikasi dengan waktu pemrosesan rata-rata 101.100,2 milidetik dan kecepatan pemrosesan sekitar 0,000989 data per milidetik. Evaluasi klasifikasi dengan menggunakan matriks konfusi menghasilkan nilai *recall* sekitar 64%, presisi sekitar 58%, dan akurasi sekitar 70%.

Menurut [10] dalam penelitiannya yang berjudul "Perancangan Model Sentimen *Tweet* Pilkada DKI Jakarta Tahun 2017 Menggunakan Algoritma *Naïve Bayes*" tujuan penelitian ini adalah untuk menganalisis sentimen dari populasi terkait dengan jawaban Regional E DKI Jakarta pada tahun 2017 di Twitter. Penelitian ini juga mencoba mengklasifikasikan tanggapan positif dan negatif terhadap kalimat menggunakan teknik analitik. Berdasarkan hasil pengujian dengan menggunakan dataset yang terdiri dari 1129 tweet, akurasi yang diperoleh adalah sebesar 50,54%.

Penelitian oleh [11] dengan judul "Analisis Sentimen Layanan Indihome Berbasis Twitter Menggunakan Metode Klasifikasi *Support Vector Machine* (SVM)", bertujuan untuk mengembangkan model klasifikasi sentimen menggunakan SVM dan mengevaluasi keakuratan metode SVM dalam analisis mood dan kepuasan pengguna. dengan layanan Indihome berbasis tweet. Hasil pengujian didapatkan tingkat akurasi sebanyak 87%, presisi sebanyak 86%, recall sebanyak 95%, tingkat error sebanyak 13%, dan f1-score sebanyak 90%.

Menurut [12] dalam penelitiannya "Penerapan klasifikasi *Naïve Bayes* dan support vector machine pada analisis emosi akibat virus Corona di Twitter", tujuannya adalah untuk menganalisis pendapat masyarakat mengenai dampak virus corona dengan memperhatikan sisi positifnya, emosi negatif dan netral diungkapkan di Twitter. Dari hasil pengujian metode Naïve Bayes mampu mengklasifikasikan sentimen dengan akurasi sebesar 81,07% meskipun tanpa penambahan fitur. Sebagai konfirmasi terhadap hasil penelitian, peneliti juga melakukan pengujian menggunakan metode SVM diperoleh akurasi sebesar 79,96%.

2.2. Preprocessing Text

Preprocessing adalah tahap persiapan data sebelum masuk dalam proses selanjutnya. Secara umum, preprocessing data melibatkan penghapusan atau transformasi data yang tidak sesuai sehingga dapat diproses dengan lebih mudah di sistem. Proses preprocessing berperan sangat penting dalam menciptakan analisis sentimen, terutama ketika data berasal dari media sosial yang cenderung banyak

memuat kalimat-kalimat informal, tidak terstruktur, dan memiliki noise. [8].

2.3. Pembobotan TF-lDF

Teknik pembobotan Term Frekuensi-Invers Dokumen Frekuensi (TF-lDF) merupakan metode pemrosesan bahasa alami yang digunakan untuk mengekstrak fitur dari teks dokumen. [9]. Metode ini mengukur nilai bobot kata dalam suatu dokumen dengan mempertimbangkan frekuensi munculnya kata tersebut dalam kalimat dan sejauh mana kata tersebut unik di seluruh korpus kalimat. Hasil pembobotan kata yang dihasilkan dari teknik ini dapat diterapkan untuk tujuan seperti klasifikasi, clustering, dan pengambilan informasi dokumen.

2.3.1. Term Frequency (TF)

Term Frequency (TF) mengindikasikan seberapa sering suatu kata muncul dalam sebuah dokumen tertentu. Semakin tinggi frekuensi suatu kata (TF tinggi) dalam dokumen muncul, maka bobot dari kata tesebut akan semakin besar. Rumus dari perhitungan Term Frequency dapat dilihat pada persamaan (1).

$$TF = 1 + \log(F_{t,d}),\ F_{t,d} > 0 \quad (1)$$

Keterangan:
TF = *Term Frequency*
$F_{t,d}$ = *Frequency* term pada dokumen

2.3.2. Inverse Document Frequency (lDF)

Inverse Document Frequency (lDF) adalah penghitungan yang mencerminkan sejauh mana suatu kata tersebar luas dalam koleksi dokumen tertentu. LDF menggambarkan tingkat ketersediaan suatu kata dalam seluruh dokumen. Perhitungan Inverse Document Frequency (lDF) dapat dilakukan menggunakan persamaan (2).

$$IDF = \log({}^{D}/_{df}) \quad (2)$$

Keterangan:
lDF = *Inverse Document Frequency*
D = Jumlah keseluruhan dokumen
df = Jumlah dokumen yang mengandung term

Berikutnya, untuk memperoleh nilai dari TF-lDF, dilakukan perkalian antara hasil nilai TF dan nilai lDF. Rumus pada persamaan (3) digunakan untuk menghitung nilai TF-IDF.

$$W = tf\ x\ idf \quad (3)$$

Keterangan:
W = Bobot TF-lDF
tf = Hasil perhitungan dari nilai TF
lDF = Hasil perhitungan dari nilai lDF

2.4. Support Vector Machine

Support Vector Machine (SVM) adalah algoritma *machine learning* yang digunakan untuk tugas klasifikasi dan regresi. SVM bekerja dengan mencari *hyperplane* atau fungsi pemisah paling opimal untuk memisah kelas dalam ruang N-dimensi (dengan N sebagai jumlah fitur), sehingga dapat dengan jelas mengklasifikasikan setiap titik data. Dalam SVM, titik yang berada sangat dekat dengan hyperplane disebut sebagai support vector. [13].

Data direpresentasikan sebagai $x_i = \in R^d$ sedangkan label diindikasikan dengan $y_i \in \{-1, +1\}$ untuk i = 1, 2,…,n. Di sini, n merupakan jumlah total data. Dengan asumsi bahwa kedua kelas, yaitu -1 dan +1, dapat dipisahkan secara linear, maka persamaan bidang pembatasnya didefinisikan oleh persamaan 4:

$$w * x_i + b = 0 \quad (4)$$

Data x_i yang dibagi menjadi du akelas, termasuk kelas -1 yang didefinisikan sebagai vector yang memenuhi pertidaksamaan (4). Pada saat yang sama vector tersebut memenuhi pertidaksamaan (5) untuk vector yang termasuk dalam kategori +1.

$$w * x_i + b < 0\ untuk\ y_i = -1 \quad (4)$$
$$w * x_i + b > 0\ untuk\ y_i = +1 \quad (5)$$

Keterangan :
x_i = data input
y_i = label kelas
w = nilai dari vektor normal bidang
b = posisi bidang relatif terhadap pusat koordinat

Parameter w dan b merupakan parameter yang nilainya diperkirakan. Ketika label y_i = -1, maka persamaan pembatasnya menjadi (6), sedangkan jika label data y_i = +1, maka persamaan pembatasnya menjadi (7).

$$w * x_i + b \leq -1 \quad (6)$$
$$w * x_i + b \leq -1 \quad (7)$$

Untuk memperoleh nilai a_i, transformasikan terlebih dahulu tiap kalimat menjadi support vector = $\begin{bmatrix} X \\ Y \end{bmatrix}$. Kemudian nilai vektor masing-masing kalimat diganti dengan persamaan (8).

$$D_i = \emptyset \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \sqrt{x_n^2 + y_n^2} - x + (x-y)^2 \\ \sqrt{x_n^2 + y_n^2} - y + (x-y)^2 \end{bmatrix} \quad (8)$$

Nilai x diperoleh pada persamaan (9) kernel linear untuk x berikut :

$$\sum_{i=1,j=1}^{1} x_i x_j^T\ ,(i,j = 1,\dots,n) \quad (9)$$

Nilai y diperoleh pada persamaan (10) kernel linear untuk y berikut :

$$\sum_{i=1,j=1}^{1} y_i y_j^T\ ,(i,j = 1,\dots,n) \quad (10)$$

Untuk mendapatkan jarak yang optimal dengan mempertimbangkan vektor positif, hasil perhitungan substitusi nilai x dan y pada Persamaan (8) menghasilkan perpindahan sebesar 1. Selanjutnya, parameter a_i dicari dengan menghitung nilai fungsinya untuk setiap pernyataan, menggunakan Persamaan (11). Selanjutnya, nilai persamaan linier a_i dicari dengan Persamaan (12) dengan memperhatikan indeks i dan j yang berkisar dari 1 hingga n.

$$\sum_{i=1,j=1}^{1} a_i D_i^T D_j \quad (11)$$
$$\sum_{i=1,j=1}^{1} a_i D_i^T D_j = y_i \quad (12)$$

Setelah memperoleh parameter a_i, kemudian disubtitusikan pada persamaan (13) berikut :

$$\widetilde{W} = \sum_{i=1}^{n} a_i D_i \quad (13)$$

Hasil dari perhitungan dari persamaan (13), akan digunakan pada persamaan (14) untuk meperoleh nilai w dan b :

$$y = w.x + b \qquad (14)$$

2.5. Evaluasi Klasifikasi

Pada fase ini, dilakukan evaluasi hasil eksperimen, perbandingan, dan analisis kinerja klasifikasi teks. Metrik evaluasi yang umum digunakan melibatkan perhitungan Recall, Precision, dan akurasi. Nilai-nilai dalam semua perhitungan ini diperoleh dari data confusion matrix, seperti yang terlihat pada Tabel 1, yang dihasilkan setelah menguji model menggunakan data uji [9].

Tabel 1. *Confusion Matrix*

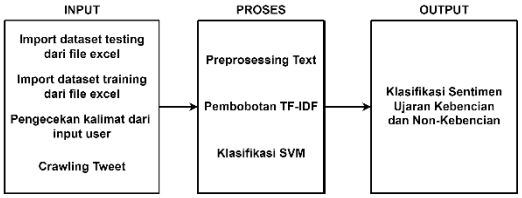
Predicted Class	**True Class**	
	Positif	Negatif
Positif	True Positif (TP)	False Positif (FP)
Negatif	False Negatif (FN)	True Negatif (TN)

Keempat parameter pada Tabel 1. digunakan dalam perhitungan tiga peringkat klasifikasi [9], yaitu:

1. *Recall* adalah rasio antara total dokumen yang terdeteksi sebagai relevan dengan total keseluruhan dokumen yang memang relevan. Formula untuk menghitung Recall adalah sebagai berikut:
$$Recall = \frac{TP}{TP+FP} \; X \; 100\% \qquad (15)$$
2. *Precision* adalah rasio antara total dokumen yang terdeteksi sebagai relevan dengan total dokumen yang terdeteksi. Formula untuk menghitung Precision adalah sebagai berikut:
$$Precision = \frac{TP}{TP+FN} \; X \; 100\% \qquad (16)$$
3. *Accuracy* adalah sejauh mana sistem mampu mengklasifikasikan data secara benar. Formula untuk menghitung Accuracy adalah sebagai berikut:
$$Accuracy = \frac{TN+TP}{TN+TP+FN+FP} \; X \; 100 \qquad (17)$$

3. METODE PENELITIAN

3.1. Diagram Blok Sistem

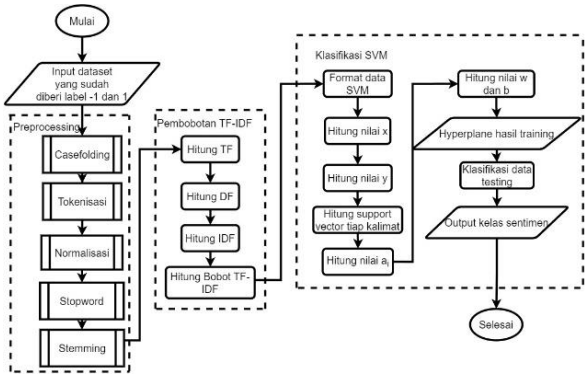


Gambar 1. Blok diagram sistem

Dalam Gambar 1. digambarkan blok diagram sistem yang terdiri dari proses *input* yaitu impor dataset exel untuk *testing*, impor *dataset* exel untuk *training*, dan pengecekan kalimat oleh pengguna. Proses selanjutnya yaitu *text preprocessing*, kemudian pembobotan TF-lDF, dan terakhir proses klasifikasi menggunakan *support vector machine*. Setelah proses terdapat *output* pada sistem yang dikembangkan keluarannya berupa kalimat sentimen mengandung kebencian atau tidak mengandung kebencian.

3.2. Flowchart Metode

Secara umum *website* yang akan dikembangkan digambarkan dalam *flowchart* berikut:



Gambar 2. Flowchart metode

Gambar 2. menunjukkan proses analisis sentimen mengunakan SVM. Proses melibatkan *preprocessing* dataset (labeling, *casefolding*, tokenisasi, normalisasi, *stopword*, dan *stemming*). Kemudian hasil dari proses *preprocessing* diolah dalam pembobotan TF LDF. Hasilnya digunakan dalam klasifikasi SVM dengan pencarian parameter dan *hyperplane* pada data training. Pengujian dilakukan pada data testing untuk menentukan sentimen dari data tersebut.

3.3. Pengumpulan Data

Langkah pertama adalah mengumpulkan data pemilu 2024 menggunakan kata kunci #pemilu2024, #pemiluserentak2024 dan #pilpres2024. Data diambil dari Twitter dengan menggunakan *library scrapper tweet-harvets*. Data dari *Facebook* dan Instagram diperoleh dengan memasukkan *link* pada postingan yang dipublikasikan dengan *hashtag* yang digunakan pada *website* https://exportcomments.com/, sehingga menghasilkan file CSV yang berisi 100 data komentar. Kemudian data *tweet* diperoleh dengan memasukkan kata kunci yang digunakan ke dalam *library scrapper tweet-harvets*, dan hasil yang diperoleh juga diperoleh dalam bentuk file CSV.

3.4. Pelabelan Data

Langkah selanjutnya setelah mendapatkan *dataset* dari *tweet* dan komentar adalah memberi label pada *dataset* data *training*. Data training yang dipakai, sebanyak 1.200 kalimat yang berasal dari postingan dan *tweet Facebook*, *Twitter*, dan *Instagram*. Pada tahap ini, data diberi label secara manual oleh annotator, yang memberi label pada setiap data dengan kalimat yang mengandung kebencian atau non-kebencian. Kalimat akan diberi label -1 untuk kebencian dan 1 untuk non-kebencian.

3.5. Preprocessing Data

Sebelum memulai proses klasifikasi, langkah awal yang diambil adalah melakukan preprocessing teks. Proses ini melibatkan serangkaian tahapan, termasuk casefolding, pembersihan (cleaning), normalisasi, eliminasi kata-kata umum (stopword), stemming, dan tokenisasi.

3.6. Casefolding

Tabel 2. Contoh Tahap *Casefolding*

Input proses	Output proses
@firdaus_frs ciri2 buzzer tolol kayak lu suka banget ketampar fakta, emang rata2 SDM pendukung Anis rendah banget ini fakta yang GK bisa dihindarkan	@firdaus_frs ciri2 buzzer tolol kayak lu suka banget ketampar fakta, emang rata2 sdm pendukung anis rendah banget ini fakta yang gk bisa dihindarkan

Dalam Tabel 2, terlihat hasil dari proses *casefolding* dimana teks sudah seragam menggunakan huruf kecil semua. Perubahan dapat dilihat pada kata SDM, Anis, dan GK yang awalnya huruf kapital diubah menjadi huruf kecil.

3.7. Cleansing

Tabel 3. Contoh Tahap *Cleansing*

Input proses	Output proses
@firdaus_frs ciri2 buzzer tolol kayak lu suka banget ketampar fakta, emang rata2 sdm pendukung anis rendah banget ini fakta yang gk bisa dihindarkan	ciri buzzer tolol kayak lu suka banget ketampar fakta emang rata sdm pendukung anis rendah banget ini fakta yang gk bisa dihindarkan

Pada Tabel 3. merupakan tahap *cleansing* pada teks *input*. Perubahan dapat dilihat dari dihapusnya mention @firdaus_frs, angka 2, dan tanda baca koma. Hasil dari tahapan ini teks mejadi lebih bersih dan sederhana.

3.8. Tokenisasi

Tabel 4. Contoh Tahap *Tokenisasi*

Input proses	Output proses
ciri buzzer tolol kayak lu suka banget ketampar fakta emang rata sdm pendukung anis rendah banget ini fakta yang gk bisa dihindarkan	‘ciri, ‘buzzer’, ‘tolol’, ‘kayak’, ‘lu’, ‘suka’, ‘banget’, ‘ketampar’, ‘fakta’, ‘emang’, ‘rata’, ‘sdm’, ‘pendukung’, ‘anis’, ‘rendah’, ‘banget’, ‘ini’, ‘fakta’, ‘yang’, ‘gk’, ‘bisa’, ‘dihindarkan’

Pada Tabel 4. merupakan contoh tahap tokenisasi yang telah diterapkan pada teks *input*. Hasilnya yaitu berupa array yang berisikan kata kata dalam kalimat yang dilakukan proses tokenisasi.

3.9. Normalisasi

Tabel 5. Contoh Tahap Normalisasi

Input proses	Output proses
ciri buzzer tolol kayak lu suka banget ketampar fakta emang rata sdm pendukung anis rendah banget ini fakta yang gk bisa dihindarkan	ciri buzzer tolol kayak kamu suka banget tertampar fakta memang rata sdm pendukung anis rendah banget ini fakta yang tidak bisa dihindarkan

Pada Tabel 5. merupakan contoh tahap normalisasi pada teks *input*. Teks *input* yang awalnya kata lu diubah menjadi kamu, kata gk diubah menjadi tidak, kata ketampar diubah menjadi tertampar, dan kata emang diubah menjadi memang.

3.10. Stopword Removal

Tabel 6. Contoh Tahap *Stopword*

Input proses	Output proses
ciri buzzer tolol kayak kamu suka banget tertampar fakta memang rata sdm pendukung anis rendah banget ini fakta yang tidak bisa dihindarkan	ciri buzzer tolol kayak kamu suka banget tertampar fakta memang rata pendukung anis rendah banget fakta dihindarkan

Pada Tabel 6. merupakan contoh tahap *stopward removal* pada teks *input*. Pada teks input kata sdm, ini, yang, tidak, dan bisa akan dihapus karena termasuk dalam kriteria *stopword*.

3.11. Steaming

Tabel 7. Contoh Tahap *Steaming*

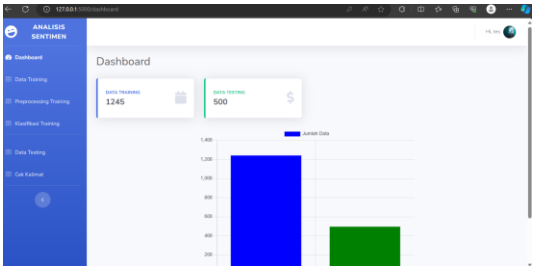
Input proses	Output proses
ciri buzzer tolol kayak kamu suka banget tertampar fakta memang rata pendukung anis rendah banget fakta dihindarkan	ciri buzzer tolol kayak kamu suka banget tampar fakta memang rata dukung anis rendah banget fakta hindar

Pada Tabel 7. merupakan contoh tahap *steaming* dimana pada teks *input* kata tertampar dibah menjadi tampar, kata pendukung diubah menjadi dukung, dan kata dihindarkan diubah menjadi hindar.

4. HASIL DAN PEMBAHASAN

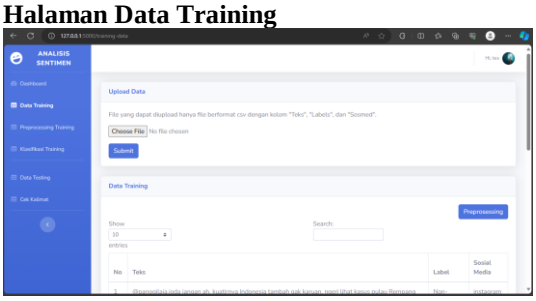
4.1. Halaman Dasboard

Pada Gambar 3. yaitu merupakan hasil implementasi halaman dashboard. Pada menu ini menampilkan informasi berapa banyak data training dan testingnya. Selain ditulis hanya angka, jumlah datanya juga di visualisasikan dengan sebuah grafik.



Gambar 3. Tampilan Dashboard

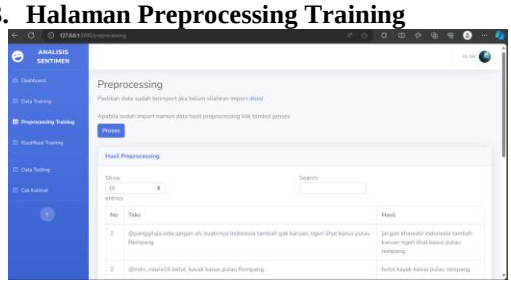
4.2. Halaman Data Training



Gambar 4. Tampilan Menu Data *Training*

Pada Gambar 4. yaitu hasil implementasi dari halaman data training. Menu ini digunakan untuk menampilkan data training yang sudah diimport. Namun apabila baru pertama kali dibuka dan data masih kosong maka juga disediakan tempat untuk melakukan import data berupa file csv.

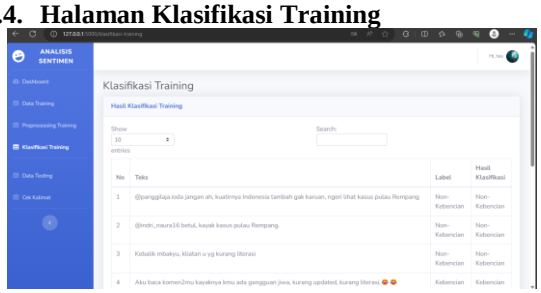
4.3. Halaman Preprocessing Training



Gambar 5. Tampilan Menu *Preprocessing Training*

Pada Gambar 5. merupakan hasil implementasi dari halaman *preprocessing* untuk data training. Tujuan dari halaman ini adalah untuk menampilkan bagaimana hasil kalimat kalimat pada dataset training yang telah dilakukan proses *preprocessing*.

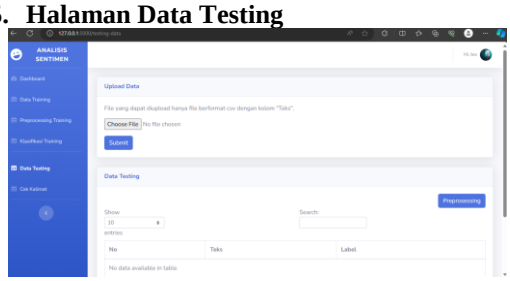
4.4. Halaman Klasifikasi Training



Gambar 6. Tampilan Menu Klasifikasi *Training*

Pada Gambar 6. merupakan hasil implementasi dari halaman klasifikasi untuk data training. Pada halaman ini akan ditampilkan data training beserta label asli yang diberikan oleh annotator.

4.5. Halaman Data Testing



Gambar 7. Tampilan Menu Data Testing

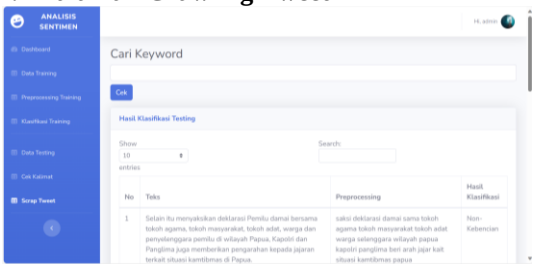
Pada Gambar 7. yaitu merupakan hasil implementasi dari halaman data testing. Halaman ini mempunyai fungsi yang sama seperti pada halaman data training. Namun pada halaman ini dataset yang diimport hanya berupa dataset berisi kalimat kalimat tanpa label.

4.6. Halaman Cek Kalimat

Gambar 8. Tampilan Menu Cek Kalimat

Pada Gambar 8. adalah implementasi halaman untuk cek kalimat. Pada halaman ini pengguna dapat mengisikan sebuah kalimat pada form, kemudian menekan tombol cek untuk mengetahui hasilnya. Hasil akan keluar dibawah form tersebut akan menampilkan hasil preprocessing lalu hasil sentimen nya.

4.7. Halaman Crawling Tweet



Gambar 9. Tampilan Menu *Crawling Tweet*

Pada Gambar 9. merupakan hasil implementasi halaman untuk melakukan cek *keyword*. Pada halaman ini pengguna dapat mengisikan sebuah *keyword* pada *form* yang disediakan, kemudian menekan tombol cek untuk mengetahui hasilnya. Hasil akan keluar dibawah form dengan menampilkan hasil *preprocessing* lalu hasil sentimen nya.

4.8. Pengujian Klasifikasi Evaluasi

Untuk menguji performa model klasifikasi, *dataset* ini dibagi dengan menggunakan skema membagi 80% train dan 20% test. 80% dari 1245 data akan digunakan sebagai data test untuk melatih model, sementara 20% sisanya digunakan sebagai data test. Evaluasi dilakukan dengan menghitung hasil klasifikasi berdasarkan tabel confusion matrix yang merepresentasikan perbandingan antara label yang diberikan oleh anotator dengan label yang dihasilkan oleh sistem. Tabel perhitungan confusion matrix disajikan pada Tabel 8.

Tabel 8. Perhitungan Confusion *Matrix*

Predicted Class	True Class		Total
	Positif	Negatif	
Positif	953	15	968
Negatif	70	207	277
Total	1023	222	1245

Setelah menghitung confusion matrix, nilai tersebut akan digunakan untuk menghitung recall menggunakan persamaan (15), precision dengan persamaan (16), dan accuracy dengan persamaan (17). Hasil perhitungan evaluasi sebagai berikut:

$$Recall = \frac{TP}{TP+FP} x10 = \frac{953}{953+70} x100 = 93\%$$

$$Precision = \frac{TP}{TP+FN} x100\% = \frac{953}{953+15} x100\% = 98\%$$

$$Accuracy = \frac{TN+TP}{TN+TP+FN+FP} \text{ x } 100\% = \frac{207+953}{207+953+15+70} x100\% = 93\%$$

Dari hasil perhitungan evaluasi diatas dengan banyak data sebanyak 1245 kalimat diperoleh hasil *recall* 93%, *precission* 98%, dan *accuracy* sebesar 93%. Selain menjalankan pengujian melalui pembagian data selama pelatihan, juga dilakukan pengujian dengan menggunakan 100 data kalimat teratas. Data yang dipakai ini diambil dari dataset *training* dan akan dibandingkan label yang diberikan oleh anotator dengan label yang dihasilkan oleh sistem.

Dari hasil perbandingan 100 kalimat yang diuji didapatkan nilai untuk melakukan perhitungan *confusion matrix*.

Tabel 9. Perhitungan Confusion Matrix

Predicted Class	True Class		Total
	Positif	Negatif	
Positif	54	27	81
Negatif	17	2	19
Total	71	29	100

Setelah menghitung confusion matrix, nilai tersebut akan digunakan untuk menghitung recall menggunakan persamaan (15), precision dengan persamaan (16), dan accuracy dengan persamaan (17). Hasil dari perhitungan confusion matrix yaitu:

$$Recall = \frac{TP}{TP+FP} x100\% = \frac{54}{54+17} x100\% = 76\%$$

$$Precision = \frac{TP}{TP+FN} x100\% = \frac{54}{54+2} x100\% = 96\%$$

$$Accuracy = \frac{TN+TP}{TN+TP+FN+FP} \text{ x } 100\% = \frac{27+54}{27+54+2+17} x100\% = 81\%$$

Berdasarkan Tabel 9. terdapat 54 kalimat yang diprediksi sebagai positif (non-kebencian) dan memang faktanya positif (non-kebencian). Sementara itu, terdapat 27 kalimat yang diprediksi sebagai positif (non-kebencian), tetapi faktanya negatif (kebencian). Sebanyak 17 kalimat diprediksi sebagai negatif (kebencian) dan memang faktanya negatif (kebencian), sedangkan 2 kalimat diprediksi sebagai negatif (kebencian), tetapi faktanya positif (non-kebencian). Evaluasi hasil perhitungan menunjukkan rata-rata akurasi dengan dataset berjumlah 100 kalimat, yakni recall sebesar 76%, presisi sebesar 96%, dan tingkat akurasi sebesar 81%.

5. KESIMPULAN DAN SARAN

Dari hasil penelitian ini, sistem berhasil melakukan klasifikasi pada kalimat yang mengandung unsur kebencian dan yang tidak mengandung kebencian. Keberhasilan klasifikasi dengan 100 data dokumen tweet, mencapai nilai recall 76%, precision 96%, dan tingkat akurasi 81%. Tingkat akurasi dan kinerja klasifikasi sentimen dipengaruhi oleh jumlah data pelatihan dan keakuratan proses text preprocessing. Hasil penelitian ini dapat dijadikan sumber edukasi untuk masyarakat,khususnya pengguna media sosial, dengan saran untuk menambah jumlah data training, meningkatkan tahap text preprocessing, dan mengembangkan sistem untuk analisis sentimen topik lain guna menciptakan lingkungan yang lebih positif.

DAFTAR PUSTAKA

[1] A. Fatihin, A. Susanto and E. Fetrina, Analisis Sentimen Terhadap Ulasan Aplikasi Mobile Menggunakan Metode Support Vector Machine (Svm) Dan Pendekatan Lexicon Based, Jakarta: Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta, 2021.

[2] P. A. Permatasari, L. and L. Jasa, "Survei Tentang Analisis Sentimen Pada Media Sosial," *Majalah Ilmiah Teknologi Elektro,* vol. 20, no. 2, pp. 177-186, 2021.

[3] M. Setyani, "Warning! Waspada Politik Identitas Menjelang Pemilu 2024," Bawaslu Kabupaten Pekalongan, 11 Februari 2022. [Online]. Available: https://pekalongankab.bawaslu.go.id/berita/detail/warning-waspada-politik-identitas-menjelang-pemilu-2024. [Accessed 3 Oktober 2023].

[4] M. Chalida and M. R. Wahyudi, "Analisis sentimen ujaran kebencian pemilihanpresiden 2019 menggunakan algoritma NaïveBayes," *JNANALOKA,* 2020.

[5] Y. M. Rohani, "Ujaran Kebencian Menurut Ali Bin Abi Thalib," *Jurnal Al-'Adl,* vol. 11, no. 1, pp. 85-99, 2018.

[6] K. N. Widyatnyana, I. W. Rasna and . Putrayasa, "ANALISIS JENIS DAN MAKNA PRAGMATIK UJARAN KEBENCIAN DI DALAM MEDIA SOSIAL TWITTER," *Jurnal Pendidikan dan Pembelajaran Bahasa Indonesia.*

[7] K. A. Rohman, B. and P. Arsi, "Perbandingan Metode Support Vector Machine Dan Decision Tree," *JOISM : Jurnal Of Information System Management*, vol. 2, no. 2, pp. 1-7, 2021.

[8] M. M. M. Olhang, S. Achmadi and F. Ariwibisono, "Analisis Sentimen Pengguna Twitter Terhadap Covid-19 Di Indonesia Menggunakan Metode Naïve Bayes Classifier(NBC)," *Jurnal Mahasiswa Teknik Informatika*, vol. 4, no. 2, pp. 214-221, 2020.

[9] M. F. Rizki, K. Auliasari and R. P. Prasetya, "Analisis Sentiment Cyberbullying Pada Sosial Media Twitter Menggunakan Metode Support Vector Machine," *Jurnal Mahasiswa Teknik Informatika*, vol. 5, no. 2, pp. 548-556, 2021.

[10] M. T. Lazuardi, T. Suprapti and Y. A. Wijaya, "Perancangan Model Sentimen Tweet Terhadap Pilkada DKI Jakarta Tahun 2017 Menggunakan Algoritma Naive Bayes," *Jurnal Mahasiswa Teknik Informatika*, vol. 7, no. 1, pp. 308-312, 2023.

[11] R. Tineges, A. Triayudi and I. D. Sholihati, "Analisis Sentimen Terhadap Layanan Indihome Berdasarkan Twitter," *Jurnal Media Informatika Budidarma*, vol. 4, no. 3, pp. 650-658, 2020.

[12] C. F. Hasri and D. Alita, "Penerapan Metode Naïve Bayes Classifier Dan Support Vector Machine Pada Analisis Sentimen Terhadap Dampak Virus Corona Di Twitter," *Jurnal Informatika dan Rekayasa Perangkat Lunak (JATIKA)*, vol. 3, no. 2, pp. 145-160, 2022.

[13] K. R. Sulaeman, C. Setianingsih and R. E. Saputra, "Analisis Algoritma Support Vector Machine Dalam Klasifikasi Penyakit Stroke," *e-Proceeding of Engineering*, vol. 9, no. 3, pp. 922-928, 2022.