Komparasi Metode SVM, Naive Bayes dan Decision Tree Untuk Analisis Sentiment Terhadap Coretax Pada Platform X

Adit Fitra Yoga[1], Rangga Sanjaya[2] , Sari Susanti[3]
[1,2,3]Program Studi Sistem Informasi, Universitas Adhirajasa Reswara Sanjaya
e-mail: [1]aditfy06@gmail.com, [2]rangga@ars.ac.id, [3]sarisusanti@ars.ac.id

Abstrak

Coretax adalah sistem perpajakan digital yang digunakan pemerintah untuk membuat layanan pajak menjadi lebih efisien dan transparan. Sejak diluncurkan, muncul berbagai opini publik, terutama di Platform X. Penelitian ini bertujuan mengeksplorasi opini publik terkait Coretax melalui analisis sentimen, sekaligus membandingkan performa tiga algoritma klasifikasi teks, yakni Support Vector Machine (SVM), Naive Bayes (NB), dan Decision Tree (DT). Sebanyak 17.812 data dikumpulkan melalui proses crawling, kemudian dibersihkan dan difilter hingga tersisa 17.551 data. Data tersebut kemudian diberi label secara otomatis menggunakan metode lexicon-based, yang menghasilkan 13.567 data yang memenuhi kriteria sebagai data sentimen positif atau negatif. Data dianalisis menggunakan ketiga algoritma setelah melalui tahap preprocessing dan vektorisasi. Evaluasi dilakukan dengan memanfaatkan confusion matrix serta sejumlah metrik, di antaranya accuracy, precision, recall, dan F1-score. Berdasarkan hasil evaluasi tersebut, SVM terbukti menjadi model dengan performa paling optimal, ditunjukkan oleh tingkat accuracy yang mencapai 97,31%. Studi ini tidak hanya menunjukkan keunggulan SVM dalam klasifikasi teks, tetapi juga menekankan pentingnya media sosial sebagai sumber masukan untuk mengevaluasi kebijakan publik digital.

Kata kunci— Coretax, Analisis Sentimen, Platform X, Support Vector Machine, Naive Bayes, Decision Tree

Abstract

*Coretax is a digital taxation system implemented by the government to improve the efficiency and transparency of tax services. Since its launch, the platform has sparked a variety of public opinions, particularly on Platform X. This study aims to explore public opinion regarding Coretax through sentiment analysis, while also comparing the performance of three text classification algorithms: Support Vector Machine (*SVM*), Naive Bayes (NB), and Decision Tree (DT). A total of 17,812 data points were collected through a crawling process, then cleaned and filtered down to 17.551 entries. The data were automatically labeled using a lexicon-based method, resulting in 13.567 entries categorized as either positive or negative sentiment. The data were analyzed using the three algorithms after undergoing preprocessing and vectorization. Evaluation was carried out using a confusion matrix along with several metrics, including accuracy, precision, recall, and F1-score. Based on the evaluation results,* SVM *demonstrated the best performance, with an accuracy rate reaching 97.31%. This study not only highlights the superiority of* SVM *in text classification but also emphasizes the importance of social media as a source of public input for evaluating digital public policies*

Keywords— *Coretax, Sentiment Analyze, Platform X, Support Vector Machine*, *Naive Bayes*, *Decision Tree*

Corresponding Author:
Rangga Sanjaya,
Email: rangga@ars.ac.id

1. PENDAHULUAN

Transformasi digital telah mendorong lembaga pemerintahan untuk meningkatkan efisiensi layanan publik, termasuk dalam sektor perpajakan. Direktorat Jenderal Pajak meluncurkan sistem *Coretax*, sebuah sistem informasi perpajakan berbasis teknologi yang dirancang untuk menyederhanakan proses pelaporan dan meningkatkan transparansi [1]. Namun, dalam pelaksanaannya, sistem ini menimbulkan beragam tanggapan dari Masyarakat mulai dari apresiasi atas kemudahan akses hingga kekhawatiran terhadap hambatan teknis dan kesiapan pengguna [2].

Media sosial menjadi salah satu ruang publik yang paling aktif digunakan masyarakat untuk menyuarakan pendapat. *Platform X* (sebelumnya Twitter) memungkinkan penyebaran opini dalam bentuk teks singkat dan real-time. Menurut Meltwater dan Kepios [3], pengguna aktif *Platform X* di Indonesia mencapai lebih dari 25 juta orang, menjadikannya salah satu sumber utama dalam memetakan persepsi publik terhadap isu strategis.

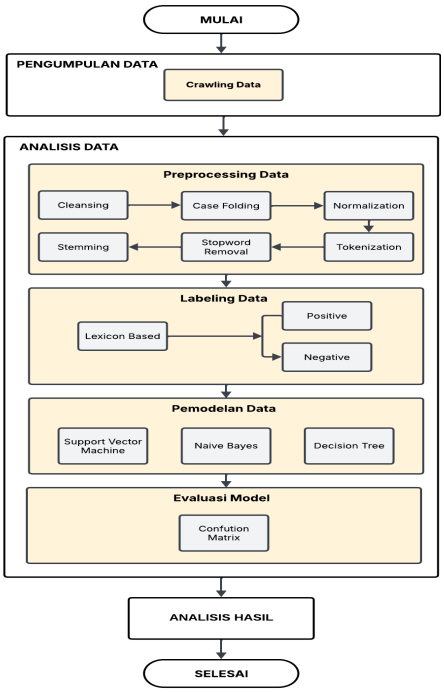
Dengan tingginya volume dan keragaman data teks di media sosial, analisis sentimen digunakan untuk mengidentifikasi kecenderungan sikap masyarakat terhadap suatu topik. Analisis ini menggabungkan teknik *natural language processing* dan pembelajaran mesin (*machine learning*) untuk mengklasifikasikan opini ke dalam sentimen positif maupun negatif [4]. Beberapa algoritma yang umum digunakan dalam *klasifikasi teks* adalah *Support Vector Machine* (SVM), *Naive Bayes* (NB), dan *Decision Tree* (DT), yang telah terbukti efektif dalam berbagai konteks analisis data tidak terstruktur [2].

Beberapa penelitian menunjukkan bahwa efektivitas algoritma dapat berbeda tergantung konteks dan karakteristik data. [5] membandingkan SVM dan *Decision Tree* dalam menganalisis sentimen program Merdeka Belajar Kampus Merdeka, dan hasilnya menunjukkan keunggulan SVM dalam hal *accuracy* dan *f1-score*. [2] juga melaporkan performa tinggi SVM dalam klasifikasi ulasan aplikasi saham dengan *accuracy* mencapai 88,70%. Di sisi lain, [6] menunjukkan bahwa *Naive Bayes* cukup efektif digunakan dalam analisis opini masyarakat mengenai kebijakan bantuan sosial.

Penelitian ini ditujukan untuk membandingkan kinerja tiga algoritma, yaitu SVM, *Naive Bayes*, dan *Decision Tree* dalam menganalisis sentimen masyarakat terhadap sistem Coretax berdasarkan data dari *Platform X*. Sebanyak 17.812 *tweet* dikumpulkan menggunakan teknik *crawling*, kemudian dilakukan *preprocessing* dan pelabelan otomatis menggunakan metode *lexicon-based* dengan kamus sentimen bahasa Indonesia InSet [7]. Hasil penelitian ini diharapkan mampu memberikan kontribusi atau masukan bagi pemilihan metode klasifikasi sentimen yang paling efektif untuk data media sosial berbahasa Indonesia.

2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif dengan metode eksperimen untuk menelaah sentimen masyarakat terhadap Coretax di *Platform X*. Data *tweet* yang diperoleh kemudian dilabeli menggunakan metode *lexicon-based* lalu dianalisis dengan tiga algoritma *klasifikasi*, yakni SVM, *Naive Bayes*, dan *Decision Tree*. Pengujian performa dilakukan melalui penghitungan *accuracy*, *precision*, *recall*, dan *F1-score* yang diperoleh dari *confusion matrix*. Rangkaian tahapan penelitian ditunjukkan pada Gambar 1.



Gambar 1. Alur Penelitian

2.1. Objek Penelitian

Data penelitian ini terdiri dari 17.812 *tweet* berbahasa Indonesia yang memuat pembahasan tentang Coretax, diperoleh dengan teknik *web crawling*. Data yang telah dikumpulkan dianalisis untuk menentukan polaritas sentimen ke dalam dua kategori, yaitu positif dan negatif. Proses pelabelan menggunakan pendekatan *lexicon-based* yang dinilai efisien untuk teks berbahasa Indonesia tanpa perlu anotasi manual [8]. Penelitian ini fokus membandingkan efektivitas tiga algoritma SVM, *Naive Bayes*, dan *Decision Tree* yang umum diterapkan dalam *klasifikasi teks* [9].

2.2. Pengumpulan Data

Pengumpulan data dilakukan menggunakan Tweet Harvester versi 2.6.1, sebuah tools berbasis Node.js yang dijalankan di Google Colab tanpa instalasi tambahan. Dengan kata kunci "Coretax" dan filter bahasa Indonesia, proses *crawling* dilakukan dalam tujuh sesi terpisah selama periode 31 Desember 2024 hingga 25 Mei 2025 untuk menghindari batasan teknis dan memastikan variasi data. Setiap sesi menghasilkan sebagian data hingga terkumpul total 17.812 *tweet*, yang disimpan dalam file dataset.csv. Tools ini menggunakan *token* autentikasi resmi dari Twitter agar proses pengambilan data dilakukan secara legal dan aman.

2.3. Analisis Data

2.3.1 Preprocessing Data

Tahap awal analisis ini adalah *preprocessing* untuk mempersiapkan teks sebelum dianalisis lebih lanjut. Prosesnya meliputi:

a. *Cleansing*, membersihkan teks dari URL, mention, emoji, angka, dan karakter tidak penting.
b. *Case folding*, mengonversi semua huruf menjadi kecil untuk menghindari perbedaan penafsiran pada kata yang sama tetapi berbeda kapitalisasi..
c. *Normalization*, mengganti kata tidak baku (misalnya "gk" jadi "tidak") dengan bantuan kamus CIL.
d. *Tokenization*, memecah kalimat menjadi unit kata atau token sehingga dapat dianalisis secara terpisah.
e. *Stopword removal*, menghapus kata-kata umum dalam Bahasa Indonesia yang tidak membawa makna penting dalam analisis sentimen.

f. *Stemming*, menggunakan library Sastrawi untuk mengembalikan setiap kata ke bentuk dasarnya, seperti "penjualan" menjadi "jual".

2.3.2 Labelling Data

Pada tahap ini, data yang sudah melalui *preprocessing* diberi label sentimen menggunakan pendekatan *lexicon-based*. Metode ini memanfaatkan kamus sentimen yang berisi daftar kata positif dan negatif. Setiap kata pada data dibandingkan dengan kamus tersebut; jika lebih banyak kata positif, maka teks diberi label positif, sedangkan jika kata negatif lebih dominan, diberi label negatif. Pendekatan ini dipilih karena bersifat otomatis, tidak memerlukan pelabelan manual (*unsupervised*), dan sangat efisien untuk menangani data dalam jumlah besar [10].

2.3.3 Pemodelan Data

Pada tahap ini, data yang telah dilabeli dibagi menjadi data latih dan data uji dengan perbandingan 80 banding 20 menggunakan *stratified sampling* agar distribusi sentimen tetap proporsional. Tujuan utamanya adalah supaya model dapat belajar dari sebagian data, lalu diuji untuk melihat kemampuannya memprediksi data baru yang belum pernah dilihat sebelumnya [11].

Setelah itu, teks diubah menjadi bentuk numerik melalui proses *vektorisasi* menggunakan metode *CountVectorizer*, yang menghitung frekuensi kata dan membentuk matriks fitur. Data yang telah di*vektorisasi* kemudian dimasukkan ke dalam tiga algoritma klasifikasi, yaitu *Support Vector Machine* (SVM), *Naive Bayes*, dan *Decision Tree* [5]. Masing-masing model dilatih menggunakan data latih dan diuji dengan data uji, untuk mengetahui metode mana yang memberikan hasil klasifikasi sentimen paling optimal pada *dataset* ini.

2.3.4 Evaluasi Model

Pada tahap ini, model yang telah dibangun dievaluasi untuk mengukur performanya dalam mengklasifikasikan sentimen. Evaluasi dilakukan menggunakan *confusion matrix*, yaitu sebuah matriks yang membandingkan kelas aktual dengan kelas prediksi, sehingga dapat diketahui jumlah prediksi yang benar maupun salah [12].

Confusion matrix memiliki empat komponen utama:

a. True Positive (TP): Data yang sebenarnya positif dan berhasil diklasifikasikan sebagai positif.
b. False Positive (FP): Data yang sebenarnya negatif tetapi salah diklasifikasikan menjadi positif.
c. False Negative (FN): Data yang sebenarnya positif tetapi salah diklasifikasikan menjadi negatif.
d. True Negative (TN): Data yang sebenarnya negatif dan berhasil dikenali sebagai negative.

Dari *confusion matrix*, diperoleh beberapa metrik evaluasi berikut:

a. Akurasi (*accuracy*): Mengukur proporsi prediksi yang benar dari seluruh data yang diuji.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

b. Presisi (*precision*): Menggambarkan ketepatan model saat memprediksi data positif, yakni seberapa besar prediksi positif yang benar-benar tepat.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

c. Daya tarik (*recall*): Menggambarkan ketepatan model saat memprediksi data positif, yakni seberapa besar prediksi positif yang benar-benar tepat.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

d. Skor harmonik (*F1-score*): Menghitung rata-rata harmonik antara *precision* dan *recall* sehingga diperoleh keseimbangan antara keduanya.

$$F1\text{-}Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

2.4. Analisis Hasil

Pada tahap ini, hasil evaluasi dari masing-masing model dianalisis berdasarkan *confusion matrix* untuk melihat bagaimana performanya dalam mengklasifikasikan sentimen publik terhadap Coretax. Dari *confusion matrix*, diperoleh nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN*)* yang menjadi dasar perhitungan metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-score* [13].

Perbandingan antar model dilakukan dengan memperhatikan beberapa aspek utama:

a. *Accuracy*, untuk melihat seberapa tepat model memprediksi data secara keseluruhan.
b. *Precision*, untuk mengukur seberapa akurat model saat memprediksi data positif.
c. *Recall*, untuk menilai kemampuan model menangkap seluruh data positif yang sebenarnya ada.
d. *F1-Score*, digunakan untuk mengevaluasi keseimbangan antara *precision* dan *recall*, yang sangat krusial apabila distribusi data tidak seimbang.

Selain itu, pola kesalahan juga diperhatikan melalui nilai FP dan FN. Sebagai contoh, jika sebuah model memiliki FN yang tinggi, berarti model tersebut sering gagal mendeteksi sentimen positif, yang bisa menjadi perhatian khusus jika sentimen positif lebih kritikal untuk diidentifikasi. Dengan menganalisis seluruh hasil ini, peneliti dapat memahami kelebihan maupun kelemahan masing-masing algoritma, sehingga dapat menentukan model yang paling cocok dengan karakteristik data opini pada platform media sosial.

3. HASIL DAN PEMBAHASAN

3.1. Crawling Data

Data penelitian diperoleh melalui proses *crawling* yang dilakukan secara bertahap menggunakan *Tweet Harvester* versi 2.6.1, yang dilakukan bertahap sebanyak tujuh kali untuk menghindari *error* teknis saat mengambil data besar. Pengambilan dilakukan dalam periode 31 Desember 2024 hingga 25 Mei 2025, sehingga terkumpul 17.812 *tweet* berbahasa Indonesia dengan kata kunci “Coretax”. *Dataset* ini kemudian disimpan dalam format .csv dan menjadi dasar proses *preprocessing* hingga analisis sentimen. Struktur kolom data bisa dilihat pada Tabel 1, sedangkan Tabel 2 menampilkan contoh *tweet* hasil *crawling*.

Tabel 1. Struktur Kolom Dataset Tweet

Nama Kolom	**Deksripsi**
conversation_id_str	ID percakapan *tweet* (thread)
created_at	Tanggal dan waktu *tweet* dibuat
favorite_count	Jumlah *like* atau favorit pada *tweet*
full_text	Isi lengkap dari *tweet*
id_str	ID unik dari *tweet*
image_url	URL gambar yang terlampir (jika ada)
in_reply_to_screen_name	Username pengguna yang dibalas oleh *tweet*
lang	Bahasa *tweet* (biasanya in untuk Indonesia)
location	Lokasi pengguna (jika tersedia)
quote_count	Jumlah quote *tweet*
reply_count	Jumlah balasan (reply)
re*tweet*_count	Jumlah re*tweet*

Nama Kolom	Deksripsi
*tweet*_url	URL yang mengarah langsung ke *tweet*
user_id_str	ID unik pengguna yang mengunggah *tweet*
username	Nama pengguna (handle) dari akun Twitter

Tabel 2. Data Hasil Crawling

created_at	full_text
Tue Dec 31 13:26:08 +0000 2024	@kumparan Jeng Sri itu otak nya seperti Gayus Tambunan tapi cara main seperti Edi Tansil coretax & Holiday tax to korporat patriarki
Thu Jan 02 09:19:54 +0000 2025	Siapa yang udah berhasil login coretax lalu permintaan sertel berhasil dan faktur pajak sudah upload? WOW SELAMAT YA. (sayanya mah masih nangis)

3.2. Preprocessing

3.2.1 Cleansing

Pada tahap awal *preprocessing*, dilakukan pembersihan (*cleansing*) pada kolom *full_text* untuk menghilangkan unsur-unsur yang tidak diperlukan seperti URL, tag HTML, emoji, simbol, tanda baca, serta angka. Proses ini juga mencakup penghapusan data duplikat agar setiap *tweet* yang dianalisis bersifat unik. Setelah tahap ini, jumlah data berkurang dari 17.812 menjadi 17.551 *tweet*. Ilustrasi perbedaan teks sebelum dan sesudah proses *cleansing* dapat dilihat pada Tabel 3.

Tabel 3. Hasil Cleansing

full_text	*Cleansing*
@kumparan Jeng Sri itu otak nya seperti Gayus Tambunan tapi cara main seperti Edi Tansil coretax & Holiday tax to korporat patriarki	kumparan Jeng Sri itu otak nya seperti Gayus Tambunan tapi cara main seperti Edi Tansil coretax amp Holiday tax to korporat patriarki
Siapa yang udah berhasil login coretax lalu permintaan sertel berhasil dan faktur pajak sudah upload? WOW SELAMAT YA. (sayanya mah masih nangis)	Siapa yang udah berhasil login coretax lalu permintaan sertel berhasil dan faktur pajak sudah upload WOW SELAMAT YA sayanya mah masih nangis

3.2.2 Case Folding

Setelah melalui tahap *cleansing*, data pada kolom *Cleansing* kemudian diproses dengan *case folding*. Tahap ini bertujuan untuk menyeragamkan semua huruf menjadi kecil (*lowercase*), agar perbedaan kapitalisasi tidak memengaruhi analisis misalnya kata “Pajak”, “PAJAK”, dan “pajak” akan dianggap sama. Hasil proses ini disimpan dalam kolom baru bernama *Case Folding*. Contoh sebelum dan sesudah case folding ditunjukkan pada Tabel 4.

Tabel 4. Hasil Case Folding

Cleansing	*Case Folding*
kumparan Jeng Sri itu otak nya seperti Gayus Tambunan tapi cara main seperti Edi Tansil coretax amp Holiday tax to korporat patriarki	kumparan jeng sri itu otak nya seperti gayus tambunan tapi cara main seperti edi tansil coretax amp holiday tax to korporat patriarki
Siapa yang udah berhasil login coretax lalu permintaan sertel berhasil dan faktur pajak sudah upload WOW SELAMAT YA sayanya mah masih nangis	siapa yang udah berhasil login coretax lalu permintaan sertel berhasil dan faktur pajak sudah upload wow selamat ya sayanya mah masih nangis

3.2.3 Normalization

Setelah tahap *case folding*, teks pada kolom *Case Folding* masuk ke proses *normalization*. Tujuan tahap ini adalah mengganti kata-kata tidak baku, slang, atau singkatan menjadi kata baku sesuai kaidah Bahasa Indonesia. Ini penting karena banyak pengguna media sosial menulis dengan ejaan tidak resmi, yang dapat memengaruhi hasil analisis jika tidak dinormalisasi lebih dulu. Proses ini dilakukan dengan mencocokkan setiap kata terhadap kamus kata tidak baku (*Colloquial Indonesian Lexicon*) yang telah disiapkan sebelumnya. Jika ditemukan, kata akan diganti dengan padanan bakunya, dan hasil akhirnya disimpan pada kolom baru bernama *Normalization*.

Tabel 5. Hasil Normalization

Case Folding	*Normalization*
kumparan jeng sri itu otak nya seperti gayus tambunan tapi cara main seperti edi tansil coretax amp holiday tax to korporat patriarki	kumparan jeng sri itu otak ya seperti gayus tambunan tapi cara main seperti edi tansil coretax amp holiday tax tapi korporat patriarki
siapa yang udah berhasil login coretax lalu permintaan sertel berhasil dan faktur pajak sudah upload wow selamat ya sayanya mah masih nangis	siapa yang sudah berhasil login coretax lalu permintaan sertel berhasil dan faktur pajak sudah upload wow selamat ya sayanya mah masih menangis

3.2.4 Tokenization

Tahap selanjutnya dalam proses *preprocessing* adalah *tokenization*, yang bertujuan memecah teks atau kalimat menjadi unit kata lebih kecil yang dikenal sebagai *token*. Pada penelitian ini, *tokenization* diterapkan pada data di kolom *Normalization* dengan memisahkan kata-kata berdasarkan spasi. Langkah ini penting agar setiap kata dapat dianalisis secara individu oleh algoritma *klasifikasi*. Hasil dari proses ini disimpan pada kolom baru bernama *Tokenization*, yang memuat daftar token dari masing-masing *tweet*. Perbedaan hasil sebelum dan sesudah *tokenization* dapat dilihat pada Tabel 6.

Tabel 6. Hasil Tokenization

Normalization	*Tokenization*
kumparan jeng sri itu otak ya seperti gayus tambunan tapi cara main seperti edi tansil coretax amp holiday tax tapi korporat patriarki	['kumparan', 'jeng', 'sri', 'itu', 'otak', 'ya', 'seperti', 'gayus', 'tambunan', 'tapi', 'cara', 'main', 'seperti', 'edi', 'tansil', 'coretax', 'amp', 'holiday', 'tax', 'tapi', 'korporat', 'patriarki']
siapa yang sudah berhasil login coretax lalu permintaan sertel berhasil dan faktur pajak sudah upload wow selamat ya sayanya mah masih menangis	['siapa', 'yang', 'sudah', 'berhasil', 'login', 'coretax', 'lalu', 'permintaan', 'sertel', 'berhasil', 'dan', 'faktur', 'pajak', 'sudah', 'upload', 'wow', 'selamat', 'ya', 'sayanya', 'mah', 'masih', 'menangis']

3.2.5 Stopword Removal

Setelah tahap *tokenization*, dilakukan proses *stopword removal* pada data di kolom *Tokenization*. *Stopword* merupakan kata-kata umum dalam Bahasa Indonesia yang sering muncul namun memiliki pengaruh sangat kecil terhadap analisis sentimen, misalnya “yang”, “dan”, “di”, “ke”, serta kata sejenis lainnya. Pada penelitian ini, daftar *stopword* diambil dari pustaka nltk untuk Bahasa Indonesia dan disesuaikan dengan menambahkan beberapa kata tidak bermakna yang lazim ditemukan pada percakapan di media sosial. Hasil akhirnya disimpan dalam kolom baru bernama *Stopword Removal*, yang memuat daftar token setelah kata-kata tidak penting tersebut dihapus. Perbedaan hasil sebelum dan sesudah penghapusan *stopword* dapat dilihat pada Tabel 7.

Tabel 7. Stopword Removal

Tokenization	*Stopword Removal*
['kumparan', 'jeng', 'sri', 'itu', 'otak', 'ya', 'seperti', 'gayus', 'tambunan', 'tapi', 'cara', 'main', 'seperti', 'edi', 'tansil', 'coretax', 'amp', 'holiday', 'tax', 'tapi', 'korporat', 'patriarki']	['kumparan', 'jeng', 'sri', 'otak', 'gayus', 'tambunan', 'main', 'edi', 'tansil', 'coretax', 'holiday', 'tax', 'korporat', 'patriarki']
['siapa', 'yang', 'sudah', 'berhasil', 'login', 'coretax', 'lalu', 'permintaan', 'sertel', 'berhasil', 'dan', 'faktur', 'pajak', 'sudah', 'upload', 'wow', 'selamat', 'ya', 'sayanya', 'mah', 'masih', 'menangis']	['berhasil', 'login', 'coretax', 'permintaan', 'sertel', 'berhasil', 'faktur', 'pajak', 'upload', 'wow', 'selamat', 'sayanya', 'mah', 'menangis']

3.2.6 Stemming

Tahap terakhir dalam proses *preprocessing* adalah *stemming*, yaitu mengembalikan setiap kata ke bentuk dasar atau akar katanya. Pada penelitian ini, *stemming* dilakukan pada token hasil *stopword removal* yang terdapat pada kolom *Stopword Removal*. Tujuan dari proses ini adalah agar kata-kata yang memiliki makna sama namun berbeda bentuk dapat disederhanakan menjadi satu bentuk dasar, sehingga jumlah fitur yang dianalisis menjadi lebih sedikit dan relevan.

Proses ini menggunakan library Sastrawi, sebuah *stemmer* Bahasa Indonesia berbasis algoritma Nazief & Adriani. Hasil akhirnya berupa string teks baru yang merupakan gabungan dari kata-kata hasil *stemming*, lalu disimpan dalam kolom baru bernama *Stemming*
Contoh hasil sebelum dan sesudah proses *stemming* dapat dilihat pada Tabel 8.

Tabel 8. Hasil Stemming

Stopword Removal	*Stemming*
['kumparan', 'jeng', 'sri', 'otak', 'gayus', 'tambunan', 'main', 'edi', 'tansil', 'coretax', 'holiday', 'tax', 'korporat', 'patriarki']	kumpar jeng sri otak gayus tambun main edi tansil coretax holiday tax korporat patriark
['berhasil', 'login', 'coretax', 'permintaan', 'sertel', 'berhasil', 'faktur', 'pajak', 'upload', 'wow', 'selamat', 'sayanya', 'mah', 'menangis']	hasil login coretax minta sertel hasil faktur pajak upload wow selamat saya mah menang

3.2.7 Visualisasi Before After *Preprocessing*

Dalam penelitian ini, visualisasi *wordcloud* digunakan untuk memeriksa perubahan distribusi kata sebelum dan setelah proses *preprocessing*. Hasil visualisasi ditampilkan pada Gambar 2 dan Gambar 3. Terlihat bahwa kata-kata tidak relevan seperti “https”, “co”, dan “rt” yang awalnya mendominasi data mentah berhasil dikurangi secara signifikan. Setelah melewati tahapan *cleansing* hingga *stemming*, *wordcloud* tampak lebih terpusat pada kata-kata utama seperti “coretax”, “pajak”, “lapor”, “kringpajak”, dan “sistem”, yang menandakan data sudah siap untuk dianalisis pada tahap selanjutnya.

Gambar 2. Wordcloud Sebelum Preprocessing

Gambar 3. Wordcloud Setelah Preprocessing

3.3. Labeling Data

Begitu semua tahap *preprocessing* selesai dilakukan, pelabelan sentimen pada data dilakukan secara otomatis memakai pendekatan *lexicon-based*. Setiap kata pada kolom *Stemming*

dibandingkan dengan daftar kata positif dan negatif dari kamus InSet (Indonesian Sentiment Lexicon) yang disusun oleh Fajri Koto.

3.3.1. Metode Pelabelan

Pelabelan dilakukan dengan menghitung selisih jumlah kata positif dan negatif dalam tiap teks. Penentuan label dilakukan dengan menghitung dominasi kata bernuansa positif atau negatif dalam tweet, jika terjadi keseimbangan data tersebut dikategorikan netral dan dikeluarkan dari *dataset*.

Data yang berlabel netral kemudian dihapus agar fokus analisis hanya pada *tweet* yang condong positif atau negatif. Dari total 17.551 *tweet* hasil *preprocessing*, terdapat 3.984 *tweet* (22,70%) yang netral sehingga disisihkan. Dengan begitu, tersisa 13.567 *tweet* (77,30%) yang siap dipakai untuk tahap klasifikasi berikutnya. Contoh hasil labeling dapat dilihat pada Tabel 9.

Tabel 9. Hasil Labeling

Stemming	**Score**	**Sentiment**
kumpar jeng sri otak gayus tambun main edi tansil coretax holiday tax korporat patriark	-1	Negatif
hasil login coretax minta sertel hasil faktur pajak upload wow selamat saya mah menang	1	Positif

Dari total 17.551 tweet yang telah melalui tahap *preprocessing* dan labeling, sebanyak 3.984 tweet (22,70%) berlabel netral dan dikeluarkan dari analisis lanjutan. Dengan demikian, dataset akhir memuat 13.567 tweet yang terdistribusi dalam dua kategori sentimen: positif dan negatif. Mayoritas tweet bernada negatif, menyoroti kendala teknis serta kompleksitas penggunaan Coretax, sedangkan tweet positif lebih banyak mengapresiasi percepatan layanan digital. Selanjutnya, visualisasi wordcloud pada Gambar 4 dan Gambar 5 memperlihatkan kata-kata dominan pada masing-masing sentimen, yang mengonfirmasi kecenderungan opini publik di media sosial yang lebih kritis terhadap penerapan sistem ini.

Gambar 4. Wordcloud Positive

Gambar 5. Wordcloud Negative

3.4. Pemodelan Data

Setelah data memiliki label sentimen, tahap berikutnya adalah membangun model klasifikasi untuk menguji performa beberapa algoritma *machine learning* dalam memprediksi sentimen publik terkait Coretax.

3.4.1. Pembagian Data

Dataset selanjutnya dipisahkan menjadi data *train* dan data *testing* dengan rasio 80:20 menggunakan fungsi *train_test_split* dari pustaka *scikit-learn*. Proses pembagian ini dilakukan pada kolom *Stemming* sebagai fitur dan kolom *Sentiment* sebagai label. Teknik *stratified sampling* diterapkan agar distribusi kelas sentimen (positif dan negatif) tetap proporsional pada kedua subset data tersebut

3.4.2. Ekstraksi Teks

Model klasifikasi tidak dapat memproses data dalam bentuk teks mentah secara langsung. Oleh karena itu, data teks terlebih dahulu dikonversi ke dalam bentuk numerik menggunakan teknik *CountVectorizer*. Teknik ini mengubah setiap dokumen menjadi representasi vektor berdasarkan frekuensi kemunculan kata. Hasil vektorisasi inilah yang kemudian digunakan sebagai input bagi model klasifikasi. Contoh hasil ekstraksi fitur ditunjukkan pada Gambar 6.

aaanjing	aadc	...	zonaintegritas	zonajajan	zonauang	zonk	zoom	zrahyck	zruisya	zxhrxd	zzsecretra
0	0	...	0	0	1	0	0	0	0	0	0
0	0	...	0	0	0	0	0	0	0	0	0
0	0	...	0	0	0	0	0	0	0	0	0
0	0	...	0	0	0	0	0	0	0	0	0
0	0	...	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...
0	0	...	0	0	0	0	0	0	0	0	0
0	0	...	0	0	0	0	0	0	0	0	0
0	0	...	0	0	0	0	0	0	0	0	0
0	0	...	0	0	0	0	0	0	0	0	0
0	0	...	0	0	0	0	0	0	0	0	0

Gambar 6. Hasil Ektraksi CountVectorizer

Setelah data dibagi dan diverktorisasi, dibangun tiga model klasifikasi menggunakan algoritma *Naive Bayes*, *Support Vector Machine* (SVM), dan *Decision Tree*. Masing-masing model dilatih pada data latih hasil *CountVectorizer* dan diuji pada data uji untuk memprediksi sentiment.

3.4.3. Support Vector Machine (SVM)

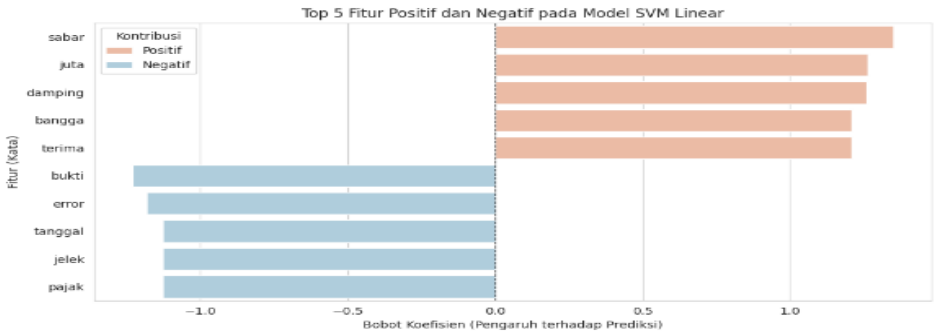
SVM digunakan deng0an kernel linear karena cocok untuk data teks berdimensi tinggi. Model dilatih lalu diuji untuk menghasilkan prediksi, seperti pada contoh potongan kode berikut:

```
# Membuat dan melatih model SVM
svm = SVC(kernel='linear')
svm.fit(X_train_vectorized, y_train)

# Prediksi data uji
y_pred_svm = svm.predict(X_test_vectorized)
```

Gambar 7. Pelatihan dan prediksi model SVM.

Selain memprediksi, juga dianalisis bobot tiap kata. Bobot positif menunjukkan kata lebih mendorong ke sentimen positif, sedangkan bobot negatif ke sentimen negatif. Tiga kata dengan bobot terbesar ditampilkan pada Gambar 8.



Gambar 8. Bobot Positif dan Negatif pada Model SVM Linear

3.4.4. Naive Bayes

Model yang digunakan adalah *Multinomial Naive Bayes* dari pustaka scikit-learn, yang memang dirancang untuk *klasifikasi teks* berbasis frekuensi kata. Model ini mengasumsikan tiap

kata muncul independen dan mengikuti distribusi *multinomial* terhadap labelnya. Berikut contoh potongan kode untuk membangun dan menguji model:

```
# Membuat dan melatih model Naive Bayes
nb = MultinomialNB()
nb.fit(X_train_vectorized, y_train)

# Prediksi data uji
y_pred_nb = nb.predict(X_test_vectorized)
```

Gambar 9. Pelatihan dan prediksi model Naive Bayes.

Naive Bayes juga menghitung probabilitas setiap kata pada masing-masing kelas untuk menentukan peluang sebuah dokumen termasuk ke kelas positif atau negatif. Contoh lima kata dengan nilai probabilitas tertinggi untuk tiap kelas ditampilkan pada Tabel 10.

Tabel 10. Kata Berdasarkan Probabilitas P(w|class) dari Model Naive Bayes

Kelas	Kata	P(w|class)
Positive	coretax	0.0570
Positive	lapor	0.0137
Positive	kringpajak	0.0132
Negative	coretax	0.0667
Negative	pajak	0.0298
Negative	sila	0.0119

3.4.5. Decision Tree

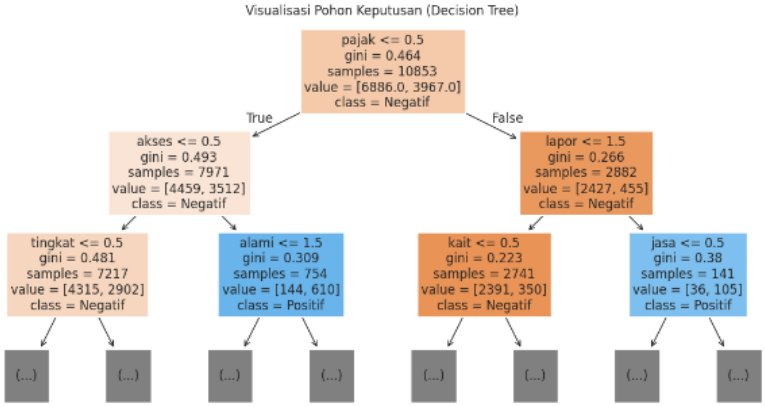
Model *Decision Tree* dibuat menggunakan DecisionTreeClassifier dari scikit-learn dengan pengaturan default. Model ini memecah data menjadi cabang-cabang berdasarkan kata yang paling banyak memberi informasi tentang kelas target. Berikut contoh potongan kode untuk membangun dan menguji model:

```
# Buat dan latih model Decision Tree
dt = DecisionTreeClassifier(random_state=42)
dt.fit(X_train_vectorized, y_train)

# Prediksi data uji
y_pred_dt = dt.predict(X_test_vectorized)
```

Gambar 10. Pelatihan dan prediksi model Decision Tree.

Visualisasi pohon keputusan ditampilkan pada Gambar 11, dibatasi hanya dua tingkat teratas agar lebih mudah dibaca. Pohon ini membantu melihat kata-kata mana yang pertama kali digunakan model untuk menentukan klasifikasi.



Gambar 11. Pohon Keputusan dari Model Decision Tree

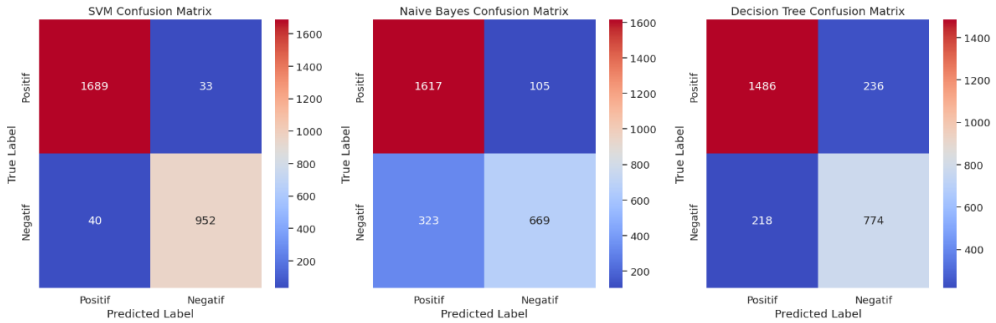
3.5. Evaluasi Model

Performa ketiga algoritma dalam mengklasifikasikan sentimen publik terhadap Coretax dievaluasi menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score* yang dihitung berdasarkan *confusion matrix*. Hasil evaluasi tersebut ditampilkan pada Tabel 11, yang menunjukkan bahwa SVM memiliki performa terbaik pada semua metrik, diikuti oleh *Naive Bayes* dan *Decision Tree*.

Tabel 11. Hasil Evaluasi Model

Model	***Accuracy* (%)**	***Precision* (%)**	***Recall* (%)**	***F1-Score* (%)**
SVM	97,31	97,69	98,08	97,78
NB	84,29	83,35	93,90	88,31
DT	83,27	90,14	86,29	86,72

Untuk memberikan gambaran yang lebih jelas terkait performa masing-masing model dalam mengklasifikasikan sentimen, *confusion matrix* divisualisasikan dalam bentuk heatmap seperti ditunjukkan pada Gambar 12 memperlihatkan heatmap perbandingan skor yang menggambarkan bagaimana distribusi prediksi tepat dan salah terjadi pada kelas positif serta negatif.



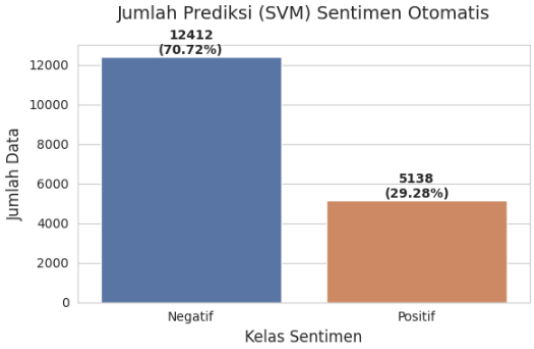
Gambar 12. Heatmap Confusion Matrix

3.6. Analisis Hasil

Berdasarkan hasil evaluasi yang ditampilkan pada Tabel 11, model *Support Vector Machine* (SVM) dipilih sebagai model terbaik untuk analisis sentimen dalam penelitian ini karena memperoleh nilai *accuracy*, *precision*, *recall*, dan *F1-score* tertinggi dibandingkan dua algoritma lainnya. Model ini kemudian digunakan untuk mengklasifikasikan seluruh data *tweet* terkait Coretax yang telah melewati tahap *preprocessing* dan labeling.

Dari hasil klasifikasi tersebut, diperoleh bahwa sekitar 12.412 *tweet* (70,72%) *tweet* diprediksi bernada negatif, sedangkan hanya 5.138 *tweet* (29,28%) yang bernada positif, sebagaimana ditunjukkan pada Gambar 13. Temuan ini mengindikasikan bahwa opini publik di *Platform X* terhadap penerapan sistem Coretax masih didominasi oleh sentimen negatif. Beberapa faktor yang mungkin memengaruhi hal ini antara lain adanya keluhan terkait kendala teknis, proses pelaporan yang dianggap belum cukup sederhana, serta kurangnya pemahaman masyarakat mengenai mekanisme layanan digital perpajakan.

Kondisi ini menunjukkan pentingnya bagi pihak terkait untuk terus melakukan penyempurnaan terhadap aspek teknis sistem, menyediakan panduan penggunaan yang lebih mudah diakses, serta aktif memonitor sentimen publik agar dapat merespons lebih cepat apabila muncul isu-isu yang berpotensi menurunkan tingkat kepuasan pengguna.



Gambar 13. Jumlah Prediksi (SVM) Sentimen terhadap Coretax.

4. KESIMPULAN

Penelitian ini melakukan perbandingan terhadap tiga algoritma *klasifikasi teks*, yakni *Support Vector Machine* (SVM), *Naive Bayes* (NB), dan *Decision Tree* (DT), dalam melakukan analisis sentimen terhadap sistem Coretax di *Platform X*. Berdasarkan evaluasi dengan metrik *accuracy*, *precision*, *recall*, dan *F1-score*, diperoleh bahwa SVM memberikan kinerja terbaik dengan tingkat *accuracy* mencapai 97,31%, diikuti oleh *Naive Bayes* sebesar 84,29%, serta *Decision Tree* yang memperoleh 83,27%.

Temuan ini menunjukkan bahwa SVM sangat cocok digunakan untuk klasifikasi opini publik pada data teks berbahasa Indonesia yang diperoleh dari media sosial. Analisis sentimen lebih lanjut mengungkapkan bahwa sekitar 70,72% opini publik mengenai Coretax bernada negatif, sedangkan hanya 29,28% yang positif. Hasil ini menjadi masukan penting bagi pihak terkait untuk meningkatkan kualitas layanan Coretax, baik melalui perbaikan teknis maupun pendekatan komunikasi publik yang lebih intensif.

Ke depan, penelitian dapat diperluas dengan menambahkan data dari platform lain, menggunakan pendekatan deep learning, atau memperkaya kamus lexicon agar klasifikasi sentimen semakin akurat dan representatif.

DAFTAR PUSTAKA

[1] Direktorat Jenderal Pajak, “Coretax,” Direktorat Jenderal Pajak. [Online]. Available: https://pajak.go.id/coretax

[2] S. Lestari and S. Saepudin, “Support Vector Machine: Analisis Sentimen Aplikasi Saham di Google Play Store,” *JUSIFO (Jurnal Sist. Informasi)*, vol. 7, no. 2, pp. 81–90, 2021, doi: 10.19109/jusifo.v7i2.9825.

[3] We Are Social; Meltwater; Kepios, “Digital 2025: Indonesia,” DataReportal. [Online]. Available: https://datareportal.com/reports/digital-2025-indonesia

[4] N. Leelawat *et al.*, “Twitter data sentiment analysis of tourism in Thailand during the COVID-

19 pandemic using machine learning," *Heliyon*, vol. 8, no. 10, p. e10894, 2022, doi: 10.1016/j.heliyon.2022.e10894.

[5] L. A. Pramesti and N. Pratiwi, "Analisis Sentimen Twitter Terhadap Program MBKM Menggunakan Decision Tree dan Support Vector Machine," *J. Inf. Syst. Res.*, vol. 4, no. 4, pp. 1145–1154, 2023, doi: 10.47065/josh.v4i4.3807.

[6] M. A. K. Fikri Alim Adiyatma, Syariful Alam, "ANALISIS SENTIMEN MASYARAKAT DI PLATFORM X TERHADAP PENGGUNAAN BANSOS UNTUK MEMENANGKAN SALAH SATU," vol. 8, no. 5, pp. 9941–9947, 2024, [Online]. Available: https://ejournal.itn.ac.id/index.php/jati/article/view/10836

[7] T. Widyanto, I. Ristiana, and A. Wibowo, "Komparasi Naïve Bayes dan SVM Analisis Sentimen RUU Kesehatan di Twitter," *SINTECH (Science Inf. Technol. J.*, vol. 6, no. 3, pp. 147–161, 2023, doi: 10.31598/sintechjournal.v6i3.1433.

[8] A. D. Adhi Putra, "Sentiment Analysis on User Reviews of the Bibit and Bareksa Application with the KNN Algorithm," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 8, no. 2, pp. 636–646, 2021.

[9] R. A. Husen, R. Astuti, L. Marlia, R. Rahmaddeni, and L. Efrizoni, "Analisis Sentimen Opini Publik pada Twitter Terhadap Bank BSI Menggunakan Algoritma Machine Learning," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 3, no. 2, pp. 211–218, 2023, doi: 10.57152/malcom.v3i2.901.

[10] S. A. Nugraha, "PENERAPAN LEXICON BASED UNTUK ANALISIS SENTIMEN MASYARAKAT INDONESIA TERHADAP DANANTARA," vol. 9, no. 3, pp. 4949–4957, 2025.

[11] S. R. I. Rezeki, Y. Restiviani, and R. Zahara, "PENGGUNAAN SOSIAL MEDIA TWITTER DALAM KOMUNIKASI ORGANISASI (Studi Kasus Pemerintah Provinsi DKI Jakarta Dalam Penanganan Covid-19)," *J. Islam. Law Stud.*, vol. 4, no. 2, pp. 63–78, 2020, [Online]. Available: https://jurnal.uin-antasari.ac.id/index.php/jils/article/download/3804/pdf/11870

[12] J. Erbani, P. É. Portier, E. Egyed-Zsigmond, and D. Nurbakova, "Confusion Matrices: A Unified Theory," *IEEE Access*, vol. 12, no. December, 2024, doi: 10.1109/ACCESS.2024.3507199.

[13] B. Farasqa Nauval Akbar, D. Arifianto, A. Maryam Zakiyyah, and A. Milu Susetyo, "Analisis Sentimen Terhadap Anies Baswedan Menggunakan Metode Support Vector Machine Studi Kasus Media Sosial Twitter Sentiment Analysis of Anies Baswedan Using the Support Vector Machine Method Case Study of Twitter Social Media," *J. Smart Teknol.*, vol. 4, no. 6, pp. 2774 1702, 2023, [Online]. Available: http://jurnal.unmuhjember.ac.id/index.php/JST