



Perbandingan Akurasi Metode Deteksi Ujaran Kebencian dalam Postingan Twitter Menggunakan Metode SVM dan Decision Trees yang Dioptimalkan dengan Adaboost

Yuda Septiawan¹, Chairani*²

^{1,2}Fakultas Ilmu Komputer, Informatics & Business Institute Darmajaya, Jl. 2.A. Pagar Alam No. 93, Bandar Lampung - Indonesia 35142; Telp. (0721) 787214 Fax. (0721) 700261

*Email Penulis Korespondensi: chairani@darmajaya.ac.id

Abstrak

Pada era digital saat ini, kemudahan akses informasi dan transaksi online telah memberikan kemudahan dan efisiensi bagi manusia. Namun, peningkatan penggunaan media sosial juga telah memunculkan masalah baru, seperti ujaran kebencian. Ujaran kebencian di media sosial memiliki dampak negatif yang signifikan, termasuk ketegangan sosial, konflik, dan pelanggaran hak asasi manusia. Untuk mengatasi masalah ini, penelitian terbaru telah mengarah pada pengembangan algoritma deteksi ujaran kebencian berbasis machine learning. Dalam penelitian ini, kami membandingkan akurasi deteksi ujaran kebencian menggunakan metode Support Vector Machine (SVM) dan metode Decision Trees sebelum dan sesudah ditingkatkan dengan metode Adaboost. Kami menggunakan dataset Twitter yang terdiri dari tweet-tweet yang mengandung ujaran kebencian dan tweet-tweet netral. Melalui perbandingan akurasi algoritma, kami berharap dapat memberikan pemahaman yang lebih baik tentang kinerja metode deteksi ujaran kebencian pada media sosial Twitter. Hasil penelitian ini didapat digunakan algoritma SVM setelah dioptimalkan Adaboosts dan Decision Trees setelah dioptimalkan Adaboosts, dari perbandingan berikut ini dapat dilihat bahwa akurasi yang didapatkan dengan algoritma SVM setelah dioptimalkan Adaboosts adalah sebesar 90,03%. Sedangkan, akurasi yang diperoleh algoritma Decision Trees setelah dioptimalkan Adaboosts adalah sebesar 89,53%.

Kata kunci— Ujaran kebencian, Support Vector Machine (SVM), Decision Trees, Adaboost

Abstract

In the current digital era, the ease of access to information and online transactions has provided convenience and efficiency for humans. However, the increasing use of social media has also brought about new problems, such as hate speech. Hate speech on social media has significant negative impacts, including social tension, conflicts, and human rights violations. To address this issue, recent research has focused on developing machine learning-based algorithms for hate speech detection. In this study, we compared the accuracy of hate speech detection using Support Vector Machine (SVM) and Decision Trees methods before and after being enhanced with Adaboost. We used a Twitter dataset consisting of tweets containing hate

speech and neutral tweets. By comparing the accuracy of the algorithms, we hope to gain a better understanding of the performance of hate speech detection methods on the Twitter social media platform. The results of this study showed that the optimized SVM algorithm with Adaboost achieved an accuracy of 90.03%. Meanwhile, the optimized Decision Trees algorithm with Adaboost obtained an accuracy of 89.53%.

Keywords— *Hate speech, Support Vector Machine (SVM), Decision Trees, Adaboost*

1. PENDAHULUAN

Pada era kini, manusia dimudahkan dalam mengakses informasi dan melakukan transaksi secara efektif dan efisien secara *online*. Di Indonesia, jumlah penduduk yang terkoneksi internet semakin meningkat dari 64,8% penduduk pada tahun 2018, lalu menjadi 73,7% penduduk tahun 2019-2020 dan kemudian meningkat lagi menjadi 77,02% penduduk pada tahun 2021-2022 [1]. Namun, peningkatan penggunaan media sosial juga telah memunculkan masalah baru, seperti ujaran kebencian. Ujaran kebencian di media sosial dapat memiliki dampak negatif yang signifikan, termasuk menyebabkan ketegangan sosial, memicu konflik, dan melanggar hak asasi manusia. Kementerian Komunikasi dan Informatika Republik Indonesia merilis konferensi pers tentang penggunaan media sosial untuk menyebarkan konten negatif [2].

Konten negatif mencakup ujaran kebencian yang melanggar hukum. Menurut Undang-Undang Informasi dan Transaksi Elektronik (UU ITE) dan Komisi Nasional Hak Asasi Manusia, ujaran kebencian adalah tindakan sengaja dan ilegal yang menyebarkan informasi dengan tujuan menimbulkan kebencian dan permusuhan terhadap individu atau kelompok tertentu berdasarkan suku, agama, dan ras [3]. Media sosial dapat menimbulkan efek samping dari kebebasan berpendapat yang didapat hingga kebebasan untuk membenci orang lain. Hal yang dianggap aktivitas kebencian diantaranya berusaha untuk membungkam orang lain; menyudutkan dan menindas orang; atau menimbulkan rasa benci, malu, bersalah, dan berbagai perasaan atau kondisi tidak nyaman lainnya. Kebebasan untuk membenci juga memberikan lahan subur bagi berkembangnya narasi sektarian dan rasis [4].

Banyak peneliti yang melakukan pendeteksian ujaran kebencian dalam beberapa tahun terakhir untuk mengatasi masalah klasifikasi ujaran dengan menggunakan berbagai algoritma. Algoritma-algoritma yang populer dan sering digunakan dalam klasifikasi tersebut antara lain *Decision Tree*, *Naive Bayes Classifier* (NBC), dan *k-Nearest Neighbor* (k-NN) [5]. Sebagai contoh, Luqyana dan rekan-rekannya [6] juga telah melakukan analisis sentimen terkait *cyberbullying* di komentar *Instagram* menggunakan metode klasifikasi *Support Vector Machine* (SVM) dan mencapai akurasi tertinggi sebesar 90%. Selain itu, Hakiem dan Fauzi [7] melakukan klasifikasi ujaran kebencian di *Twitter* menggunakan NBC berbasis *N-Gram* dengan seleksi fitur *information gain* dan berhasil mencapai akurasi tertinggi sebesar 84%. Dan juga, Fauzi dan rekan-rekannya [8] menggunakan rekayasa fitur dan pengaturan parameter pada Algoritma *Decision Tree* untuk klasifikasi yang menghasilkan Rata-rata akurasi dari ketiga kelas meningkat dari 69,77 % menjadi 70,48 % untuk komposisi data latih-uji 90:10, dan dari 69,35 % menjadi 69,54 % untuk komposisi 80:20.

Sebagian besar algoritma yang digunakan dalam penelitian tersebut masih merupakan algoritma standar (*baseline*) yang belum dioptimalkan untuk meningkatkan akurasinya. Ada banyak metode yang dapat dilakukan untuk meningkatkan akurasi algoritma *machine learning* dalam tugas klasifikasi, dan salah satunya adalah dengan menggunakan teknik rekayasa fitur (*feature engineering*). Untuk mengatasi masalah ini, penelitian terbaru telah mengarah pada pengembangan algoritma deteksi ujaran kebencian berbasis *machine learning*. Dalam jurnal ini, kami memfokuskan pada deteksi ujaran kebencian pada platform media sosial *Twitter*. Kami membandingkan dua metode *machine learning* populer, yaitu metode *Support Vector Machine*

(SVM) dan metode *Decision Trees*, yang ditingkatkan dengan metode *Adaboost*.

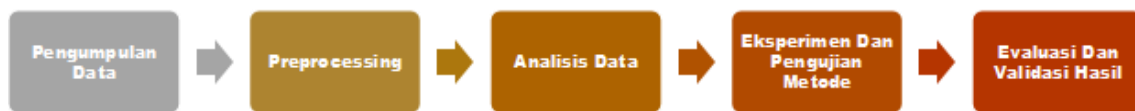
Metode SVM telah terbukti efektif dalam mempelajari pola kompleks dalam data, sedangkan metode *Decision Trees* memiliki keunggulan dalam interpretabilitas dan kemampuan menghasilkan aturan keputusan yang jelas. Namun, keduanya memiliki kelemahan, seperti masalah *overfitting* dan *underfitting*. Oleh karena itu, kami mengadopsi metode *Adaboost* untuk meningkatkan kinerja kedua metode tersebut.

Penelitian ini bertujuan untuk membandingkan akurasi deteksi ujaran kebencian antara metode SVM dan metode *Decision Trees* yang ditingkatkan dengan metode *Adaboost*. Kami menggunakan *dataset Twitter* yang terdiri dari *tweet-tweet* yang mengandung konten ujaran kebencian dan *tweet-tweet* netral sebagai data pelatihan. Kami membagi *dataset* menjadi subset pelatihan dan subset pengujian untuk melakukan evaluasi kinerja algoritma. Dengan melakukan perbandingan akurasi antara metode *Support Vector Machine (SVM)* dan metode *Decision Trees* sebelum dan sesudah ditingkatkan dengan metode *Adaboost*, kami berharap dapat memberikan pemahaman yang lebih baik tentang kinerja algoritma dalam mendeteksi ujaran kebencian pada media sosial *Twitter*. Hasil penelitian ini dapat digunakan sebagai landasan untuk pengembangan sistem deteksi ujaran kebencian yang lebih efektif dan akurat.

2. METODE PENELITIAN

2.1 Jenis Penelitian

Penelitian ini menggunakan jenis penelitian eksperimen, dengan tahapan penelitian sebagai berikut:



Gambar 1 Alur Penelitian

Seperti yang ditunjukkan pada Gambar 1. Setelah melakukan pengumpulan data dan menentukan metode selanjutnya adalah tinggal melakukan uji coba dari hasil data yang sudah dikumpulkan. Selanjutnya data dari hasil penelitian akan dianalisis dengan cara melakukan verifikasi dan validasi data hingga akan menghasilkan output analisis yang memiliki tingkat akurasi yang tinggi.

2.2 Pengumpulan Data

Tahap ini dilakukan sebagai awal dari suatu penelitian. Studi literatur dilakukan pada literatur – literatur yang sesuai dengan penelitian ini. *Dataset* yang digunakan pada penelitian ini menggunakan *dataset* berupa postingan/unggah Twitter yang diperoleh melalui internet. Dalam penelitian ini metode pengumpulan data untuk mendapatkan sumber data yang akan digunakan adalah dengan metode pengumpulan data skunder. Data dari penelitian ini diambil dari situs *kaggle dataset repository* (<https://www.kaggle.com/datasets/ankitkumar2635/sentiment-and-emotions-of-tweets>) yang terdiri dari 24.970 dengan 9 label data, namun terdapat data noise sebanyak 2.092 data sehingga jumlah data yang digunakan pada penelitian ini adalah 22.878. Label yang digunakan terdiri dari dua yaitu *Text*, dan *Sentiment*, dimana label *Sentiment* dibagi menjadi tiga yaitu Positif, Netral, dan Negatif. Contoh distribusi label *Sentiment* pada *dataset* dapat dilihat pada Tabel 1. Sedangkan untuk data pendukung didapatkan dari buku jurnal dan publikasi lainnya.

Tabel 1 Contoh Kata Positif, Netral dan Negatif

Positif	Netral	Negatif
Happy	Okay	Sad
Excited	Fine	Angry

Wonderful	Neutral	Disappointed
Great	Indifferent	Frustrated
Love	Fair	Terrible
Excellent	Average	Bad
Fantastic	Ordinary	Awful
Delighted	Acceptable	Unpleasant
Amazing	Balanced	Upset
Awesome	Moderate	Hate

2.3 Preprocessing

Preprocessing merupakan suatu tahap untuk mempersiapkan data yang telah diperoleh dari tahap pengumpulan data sebelumnya sebelum data tersebut digunakan untuk tahap selanjutnya. Persiapan data yang dilakukan berupa membersihkan data dengan menghilangkan *noise*, menghapus data duplikat, memeriksa data untuk ketidakkonsistenan, dan memperbaiki kesalahan dalam data, seperti kesalahan ketik.

Pada tahap ini, data akan melalui beberapa proses pembersihan dan mengubah teks mentah menjadi representasi yang lebih terstruktur dan siap digunakan dalam pemrosesan lanjutan. Berikut adalah tahapan dalam proses *preprocessing* data tersebut:

- Lowercasing*: Mengubah seluruh teks menjadi huruf kecil. Tujuannya adalah untuk mengurangi variasi dan menyamakan teks yang memiliki kapitalisasi berbeda, seperti "Hello" dan "hello".
- Tokenisasi: Memisahkan teks menjadi unit-unit yang lebih kecil yang disebut token. Token bisa berupa kata, frasa, atau karakter tergantung pada kebutuhan. Misalnya, kalimat "Saya sedang belajar pemrosesan teks" dapat ditokenisasi menjadi ["Saya", "sedang", "belajar", "pemrosesan", "teks"].
- Menghapus karakter yang tidak penting: Menghapus karakter yang tidak memberikan informasi penting dalam teks, seperti tanda baca, simbol, angka, atau karakter khusus tertentu. Hal ini bergantung pada kebutuhan dan konteks analisis Anda. Contoh proses penghapusan data atau *Cleansing Data* menggunakan *Python* seperti pada gambar 2.

```
def clean_tweet(tweet):
    if type(tweet) == np.float:
        return ""
    temp = tweet.lower()
    temp = re.sub(" ", "", temp)
    temp = re.sub("@[A-Za-z0-9_]+", "", temp)
    temp = re.sub("#[A-Za-z0-9_]+", "", temp)
    temp = re.sub(r'http\S+', '', temp)
    temp = re.sub('(!|?)', ' ', temp)
    temp = re.sub('\. *\?', ' ', temp)
    temp = re.sub("^\a-z0-9", " ", temp)
    temp = temp.split()
    temp = [w for w in temp if not w in stopwords]
    temp = " ".join(word for word in temp)
    return temp

tw['clean_Text'] = tw['Text'].apply(clean_tweet)
tw.head(20)
```

Gambar 2 *Cleansing Data* dengan *Python*

- Menghapus *stopword*: Menghapus kata-kata yang umum dan tidak memberikan banyak informasi seperti "dan", "atau", "juga", "dari", dll. Anda dapat menggunakan daftar *stopword* yang telah disediakan oleh pustaka atau membuat daftar *stopword* sendiri.

- e) *Stemming* atau *Lemmatisasi*: Memperkecil kata-kata menjadi akar kata (*stemming*) atau mengubah kata-kata ke bentuk dasar (*lemmatisasi*). Contohnya, kata-kata "berlari", "berlari", dan "berlari" dapat distem menjadi "lar".
- f) *Normalisasi kata*: Mengubah kata-kata ke bentuk yang lebih umum atau standar. Misalnya, mengubah "gimana" menjadi "bagaimana" atau "gak" menjadi "tidak".
- g) *Pembentukan vektor representasi*: Mengubah teks menjadi representasi numerik seperti vektor menggunakan metode TF-IDF untuk memahami teks dalam bentuk numerik untuk pemrosesan lanjutan.

Setelah tahapan *preprocessing* selesai, teks akan siap digunakan untuk analisis lebih lanjut. Dan berikut ini, contoh hasil data yang telah melalui tahap *preprocessing* data pada Tabel 2.

Tabel 2 Hasil Sebelum dan sesudah di *Cleansing*

No	Data Awal	Cleansing data
1.	As @CRN celebrates its 40th anniversary, Bob Faletra and @stevenjburke spoke with me about the milestones, companies and personalities that helped build the channel. https://t.co/stiuBObPIO #CRN40 #podcast #internationalpodcastday @Cisco @Microsoft @HPE @hp @Dell @intel	celebrates 40th anniversary bob faletra spoke milestones companies personalities helped build channel
2.	@dell your customer service is horrible especially agent syedfaisal who has made this experience of purchasing a new computer downright awful and I'll reconsider ever buying a Dell in the future @DellTech"	customer service horrible especially agent syedfaisal made experience purchasing new computer downright awful reconsider ever buying dell future

2. 4 Analisa Data

Tahap analisis adalah prosedur sistematis untuk menentukan data yang sesuai dan cara terbaik untuk memanfaatkannya. Tahap ini melibatkan kombinasi metode atau teknik dalam mengolah data. Proses analisis data dimulai dengan membandingkan hasil eksperimen. Data yang telah dikumpulkan kemudian dibersihkan dan dinormalisasi sebelum dilakukan transformasi data pengklusteran. Hasil transformasi ini akan dibagi menjadi beberapa kelompok atau grup data.

Setelah pengelompokan data selesai, tahap selanjutnya adalah pemodelan data. Data akan diubah menjadi nilai-nilai terpisah dengan menggunakan teknik pelebelan data atau dalam format *Comma Separated Value* (CSV) untuk digunakan sebagai input dalam *Machine Learning*. Setelah pemodelan data selesai, langkah berikutnya adalah menerapkan model algoritma *Support Vector Machine* (SVM), *Decision Tree* dan *Adaboost*.

Evaluasi dilakukan untuk mengukur performa model algoritma yang digunakan. Pengukuran ini melibatkan akurasi, presisi, recall, dan waktu. Hasil evaluasi tersebut digunakan untuk membuat kesimpulan dalam klasifikasi pada penelitian ini. Kesimpulan ini akan menjadi acuan atau pedoman dalam pengukuran yang dilakukan.

2. 5 Eksperimen dan Pengujian Metode

Tahap ini akan menjelaskan eksperimen dan teknik pengujian yang akan digunakan untuk menguji dan mengevaluasi suatu metode atau teknik dengan menggunakan data yang telah dikumpulkan dari observasi atau percobaan. Eksperimen dan pengujian metode merupakan

bagian penting dari proses penelitian, karena dengan melakukannya maka akan tercipta suatu metode yang dapat digunakan secara efektif dan efisien untuk menyelesaikan suatu masalah.

Eksperimen dan pengujian metode juga bertujuan untuk mengetahui kelebihan dan kekurangan suatu metode atau teknik, serta untuk menemukan cara-cara baru yang lebih baik dalam menyelesaikan masalah. Oleh karena itu, eksperimen dan pengujian metode sangat penting dilakukan agar dapat memperoleh hasil yang bermutu dan dapat dipertanggungjawabkan.

2. 5. 1 Support Vector Machine (SVM)

Support Vector Machine (SVM) adalah salah satu metode klasifikasi yang banyak dikembangkan saat ini. Konsep dasar metode ini adalah memaksimalkan batas *hyperplane* yang memisahkan suatu set data[9]. *Support Vector Machine* (SVM) adalah metode klasifikasi yang menggunakan konsep mencari *hyperplane* (bidang pemisah) yang optimal dalam suatu ruang *feature* untuk memisahkan dua kelas. *Hyperplane* yang dicari adalah yang memberikan jarak paling jauh dari setiap titik data. SVM dapat digunakan untuk klasifikasi *linearly separable* dan *non-linearly separable*.

1. SVM linear

Mencari *hyperplane* yang memisahkan dua kelas dengan cara mencari garis yang paling baik memisahkan antara dua kelas. Rumus dasar SVM linear adalah:

$$f(x) = wx + b \quad (1)$$

di mana x adalah input (vektor fitur), w adalah vektor bobot, dan b adalah bias. Untuk membuat klasifikasi, maka ditambahkan dengan fungsi kondisional seperti:

$$y = \{1 \text{ if } f(x) \geq 0, -1 \text{ otherwise}\} \quad (2)$$

2. SVM non-linear

Jika data tidak *linearly separable*, maka akan digunakan kernel trick untuk mengubah data ke dalam dimensi yang lebih tinggi sehingga dapat di separasi dengan linear. Rumus dasar SVM non-linear adalah:

$$f(x) = \sum (a_i y_i K(x_i, x)) + b \quad (3)$$

di mana x adalah input (vektor fitur), a adalah vektor bobot, y adalah label, K adalah kernel yang digunakan (misalnya *radial basis function*, *polynomial*, atau *sigmoid*), b adalah bias.

SVM mencari *hyperplane* yang memberikan jarak paling jauh dari setiap titik data. Untuk menemukan nilai a_i dan b yang optimal digunakan metode *Quadratic Programming* (QP) yang mencari solusi yang meminimalkan fungsi *objective* dengan memperhatikan kendala yang ditentukan. Secara umum, SVM mencari solusi optimal dari persamaan *quadratic programming* (QP) yang di dalamnya termasuk kendala dari persamaan yang akan di optimalkan.

2. 5. 2 Decision Tree

Decision Tree C4.5 merupakan salah satu algoritma klasifikasi yang banyak digunakan untuk memperoleh hasil klasifikasi *non biner*. Dibanding algoritma sejenis, *Decision Tree C4.5* memiliki kelebihan pada kemampuan untuk mengelola data dalam berbagai format [10]. Struktur algoritma ini mirip *flowchart* dimana masing-masing node mewakili nilai atribut dan masing masing cabang merepresentasikan hasil pengujian, dan masing-masing daun merepresentasikan kelas atau distribusi kelas. *Decision Tree C4.5* merupakan bentuk

pengembangan dari algoritma ID3 yang mengadopsi pendekatan *greedy* dengan pengambilan keputusan berdasarkan tree yang terbentuk menggunakan pendekatan rekursif *top down* dan sistem bagi serang.

Algoritma *Decision Tree* C4.5 memiliki kelebihan dibanding ID3 dan CART karena kemampuannya untuk tidak membatasi cabang dalam bentuk biner. Selain itu, C4.5 secara *default* juga memungkinkan cabang untuk masing-masing nilai ke dalam atribut kategori sehingga klasifikasi yang dihasilkan lebih terkelompokkan dibanding ekspektasi. Berdasarkan kondisi tersebut dapat dilihat bahwa metode *Decision Tree* C4.5 merupakan metode yang dapat memberikan tingkat akurasi tinggi pada kasus-kasus prediktif dengan beberapa atribut kategori.

Secara umum, langkah-langkah pembentukan algoritma C4.5 adalah sebagai berikut:

- a) Langkah 1: tentukan atribut root node
- b) Langkah 2: tentukan cabang untuk masing-masing nilai atribut
- c) Langkah 3: pisahkan kasus sesuai cabang
- d) Langkah 4: Ulangi proses hingga semua kasus dalam cabang memiliki kelas yang sama

Untuk menentukan atribut yang menjadi root, dilakukan penghitungan nilai gain untuk masing-masing atribut menggunakan Rumus (4) berikut:

$$Gain(S, A) = Entropy(s) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropy(S_i) \quad (4)$$

dimana: $\{S_1, S_2, S_3, \dots, S_i, \dots, S_n\}$ = partisi S sejumlah nilai atribut A
 n = jumlah atribut A
 $|S_i|$ = Jumlah kasus dalam partisi S_i
 $|S|$ = Total kasus S

Sementara untuk memperoleh nilai entropi digunakan Rumus (5) berikut.

$$Entropy(S) = \sum_{i=1}^n -p_i * \log_2 p_i \quad (5)$$

dimana:

S = jumlah kasus

n = jumlah kasus dalam partisi S

p_i = proporsi S_i terhadap S

2. 5. 3 Adaboost

Algoritma *Adaboost* (*Adaptive Boosting*) adalah algoritma *ensemble learning* yang digunakan untuk meningkatkan kinerja model prediktif dengan menggabungkan beberapa model prediksi yang relatif lemah menjadi satu model yang lebih kuat. *Adaboost* bertujuan untuk mengurangi bias dan varians dari model prediksi yang lemah dengan memberikan bobot yang berbeda pada setiap model.

Adaptive boosting juga merupakan algoritma klasifikasi yang membentuk sebuah tim dari klasifier yang lemah dan meningkatkan performa algoritma *Machine Learning* dengan menggabungkannya menjadi sebuah klasifier yang lebih kuat. Tujuan utama dari klasifier ini adalah meningkatkan bobot pada titik-titik yang belum terklasifikasi dan mengurangi bobot pada titik-titik yang sudah terklasifikasi sebelumnya. Ada beberapa keuntungan yang dimiliki oleh *AdaBoost*, seperti kemudahan penggunaan dan tingkat akurasi yang tinggi dari klasifier yang lemah. Namun, terdapat juga beberapa kelemahan, seperti kebutuhan akan data yang berkualitas serta optimisasi *hiperparameter*. Selain itu, pohon yang dihasilkan oleh algoritma ini juga dikenal dengan sebutan *Decision Stump*. Model akan terus dilatih hingga mencapai tingkat kesalahan yang lebih rendah [11].

Berikut adalah penjelasan langkah-langkah algoritma *Adaboost*:

- a) Inisialisasi bobot: Setiap sampel dalam *dataset* diberi bobot awal yang sama.
- b) Iterasi pembelajaran:
 - 1) Pelatihan model: Model prediksi yang relatif lemah, seperti *Decision Trees*, dibangun pada *dataset* latih dengan menggunakan bobot sampel yang diberikan.
 - 2) Menghitung kesalahan: Kesalahan model pada *dataset* latih dihitung dengan membandingkan prediksi model dengan label sebenarnya.
 - 3) Menghitung bobot model: Bobot model dihitung berdasarkan kesalahan yang terjadi. Bobot yang lebih besar diberikan pada sampel yang dianggap sulit untuk diprediksi dengan benar oleh model sebelumnya.
 - 4) Mengupdate bobot sampel: Bobot sampel dalam *dataset* latih diperbarui berdasarkan kesalahan prediksi model. Sampel yang dianggap sulit untuk diprediksi akan memiliki bobot yang lebih tinggi untuk memberikan penekanan yang lebih besar dalam iterasi selanjutnya.
 - 5) Normalisasi bobot: Bobot sampel dinormalisasi agar jumlah total bobot menjadi 1.
- c) Gabungan model: Model-model yang diperoleh dari iterasi pembelajaran digabungkan dengan bobot yang diberikan pada masing-masing model. Model-model tersebut memiliki bobot yang sebanding dengan kinerja prediksi mereka.
- d) Prediksi: Model gabungan digunakan untuk membuat prediksi pada *dataset* uji.

2. 6 Evaluasi dan Validasi Hasil

Tahap ini adalah kesimpulan dari evaluasi eksperimen yang dilakukan menggunakan *dataset* yang dikumpulkan. Evaluasi dan validasi hasil merupakan proses untuk mengevaluasi dan memvalidasi hasil suatu proyek atau kegiatan. Evaluasi adalah penilaian keberhasilan proyek atau kegiatan dengan mengukur sejauh mana tujuan yang ditetapkan tercapai. Validasi adalah pengujian kebenaran atau keabsahan hasil.

Evaluasi dan validasi hasil sangat penting untuk memastikan bahwa proyek atau kegiatan sesuai dengan tujuan yang ditetapkan dan hasilnya dapat dipertanggungjawabkan. Dengan melakukan evaluasi dan validasi hasil, proyek atau kegiatan dapat berkualitas dan memberikan manfaat yang optimal.

Dalam kasus ini, terdapat dua metode evaluasi yang diterapkan:

1. Evaluasi Opini: Metode ini digunakan untuk mengevaluasi opini yang diberikan dalam sebuah teks. Tujuannya adalah mengidentifikasi pandangan atau pendapat yang dinyatakan dalam teks.
2. Evaluasi Performa: Metode ini digunakan untuk mengevaluasi kinerja model analisis sentimen. Hal ini melibatkan pengukuran akurasi, presisi, recall, dan F-measure. F-measure adalah metode evaluasi performa yang menggabungkan presisi dan recall menjadi satu nilai yang mencerminkan performa keseluruhan model.

Presisi mengukur seberapa banyak data yang diidentifikasi sebagai positif oleh model benar-benar positif, sedangkan recall mengukur seberapa banyak data positif yang diidentifikasi oleh model. *F-measure* menggabungkan presisi dan recall untuk memberikan satu nilai yang memperhitungkan kekurangan masing-masing metrik tersebut.

F-measure adalah *harmonic* mean dari precision dan recall, dinyatakan pada formula (6) dan formula (7).

$$F_{-measure} = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \quad (6)$$

$$F1_{avg} = \frac{\sum_{i=1}^n \frac{2 \times P_i \times R_i}{P_i + R_i}}{n} \quad (7)$$

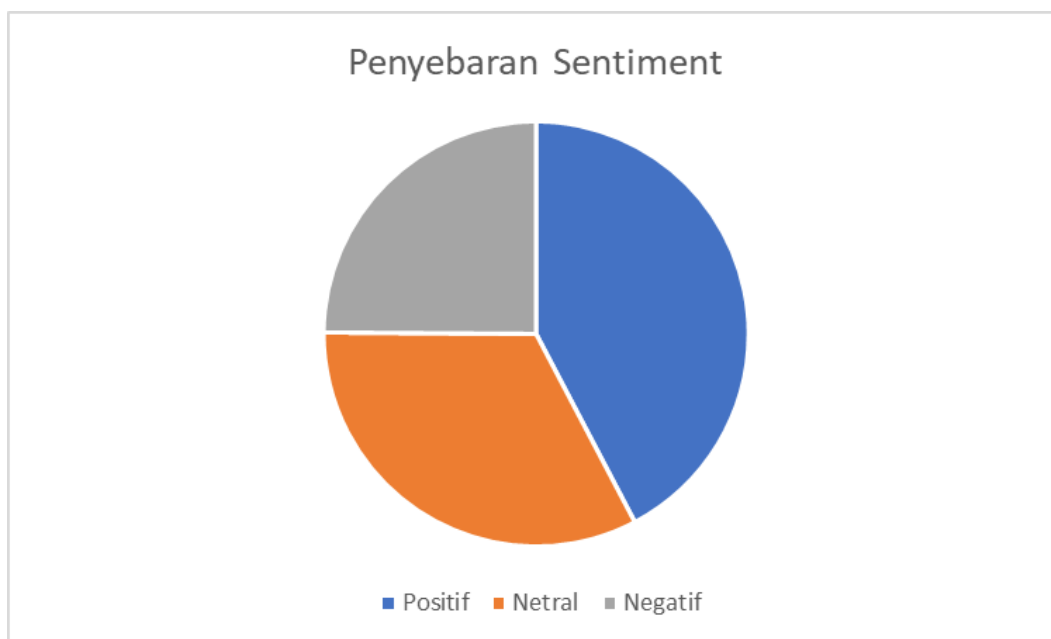
Nilai *F-measure* berkisar dari 0 hingga 1, di mana nilai 1 menunjukkan performa yang sempurna. Metode ini sering digunakan dalam penelitian sentimen analisis untuk mengevaluasi kinerja model dalam mengidentifikasi tipe sentimen dari sebuah teks [12].

3. HASIL DAN PEMBAHASAN

Penelitian ini bertujuan untuk menerapkan algoritma SVM dan *Decision Tree* sebelum dan sesudah dioptimalkan dengan *Adaboost* untuk membandingkan tingkat akurasi klasifikasi ujaran kebencian. Hasil dari penelitian ini berupa pengolahan data kualitatif dan data kuantitatif dengan perhitungan yang dilakukan pada sebuah *dataset* yang telah diperoleh. Yang sudah melalui proses preprocessing data dan analisis data.

3.1 Dataset

Dalam penelitian ini, digunakan *dataset* yang berasal dari situs kaggle *dataset repository* (<https://www.kaggle.com/datasets/anvikumar2635/sentiment-and-emotions-of-tweets>) yang terdiri dari 24.970 data yang kemudian mengalami proses labelisasi dengan menggunakan *stemming* dan penghapusan *stopword* sehingga jumlah data yang digunakan pada penelitian ini adalah 22.878. Dari proses tersebut, diperoleh hasil bahwa terdapat 9.680 tweet yang termasuk dalam kelas positif, 7.504 termasuk dalam kelas netral dan 5.694 tweet yang termasuk dalam kelas negatif. Seperti terlihat pada gambar 3 penyebaran *sentiment* yang ada dalam *dataset*.



Gambar 3 Data Penyebaran Sentiment

3.2 Hasil

Pada bagian hasil, penulis akan menjelaskan hasil dari pengujian terhadap model yang diusulkan. Pengujian dilakukan sesuai dengan tahapan yang telah dijelaskan dalam bagian metode penelitian. Setelah melalui proses *preprocessing*, dilakukan analisis perbandingan menggunakan algoritma SVM dan *Decision Tree* sebelum dan sesudah dioptimalkan dengan *Adaboost*.

a) Algoritma SVM

```

SVM Accuracy: 88.75%
      precision    recall  f1-score   support

 Negative         0.89      0.80      0.84      1146
  Neutral         0.86      0.96      0.91      1511
  Positive         0.91      0.89      0.90      1919

 accuracy                   0.89      4576
 macro avg         0.89      0.88      0.88      4576
 weighted avg      0.89      0.89      0.89      4576

```

Gambar 4 Akurasi SVM Sebelum Dioptimalkan Dengan *Adaboost*

Pada penggunaan metode SVM Sebelum Dioptimalkan Dengan *Adaboost*, nilai akurasinya berada di angka 88,75% seperti pada gambar 4.

```

Akurasi SVM dengan Adaboost : 90.03%
      precision    recall  f1-score   support

 Negative         0.89      0.80      0.84      1146
  Neutral         0.86      0.96      0.91      1511
  Positive         0.91      0.93      0.92      1919

 accuracy                   0.90      4576
 macro avg         0.89      0.89      0.89      4576
 weighted avg      0.89      0.90      0.89      4576

```

Gambar 5 Akurasi SVM Setelah Dioptimalkan Dengan *Adaboost*

Lalu, pada penggunaan SVM Setelah Dioptimalkan Dengan *Adaboost*, tingkat akurasi yang didapat berada di angka 90,03% seperti gambar 5. Dengan menggunakan metode SVM sebelum dioptimalkan dengan *Adaboost*, nilai akurasinya adalah sebesar 88,75% seperti yang terlihat pada gambar di atas. Namun, setelah menggunakan SVM yang telah dioptimalkan dengan *Adaboost*, tingkat akurasi meningkat menjadi 90,03% seperti yang juga terlihat dalam gambar yang sama. Dengan demikian, penggunaan *Adaboost* sebagai metode pengoptimalan pada SVM berhasil meningkatkan tingkat akurasi dalam klasifikasi data tersebut.

b) Algoritma *Decision Tree*

```

Decision Trees Accuracy: 89.40%
      precision    recall  f1-score   support

 Negative         0.83      0.81      0.82      1146
  Neutral         0.95      0.97      0.96      1511
  Positive         0.89      0.89      0.89      1919

 accuracy                   0.89      4576
 macro avg         0.89      0.89      0.89      4576
 weighted avg      0.89      0.89      0.89      4576

```

Gambar 6 Akurasi *Decision Trees* Sebelum Dioptimalkan Dengan *Adaboost*

Pada penggunaan metode *Decision Trees* Sebelum Dioptimalkan Dengan *Adaboosts*, nilai akurasi berada di angka 89,40% seperti pada gambar 6.

Akurasi *Decision Trees* dengan *Adaboost* : 89.53%

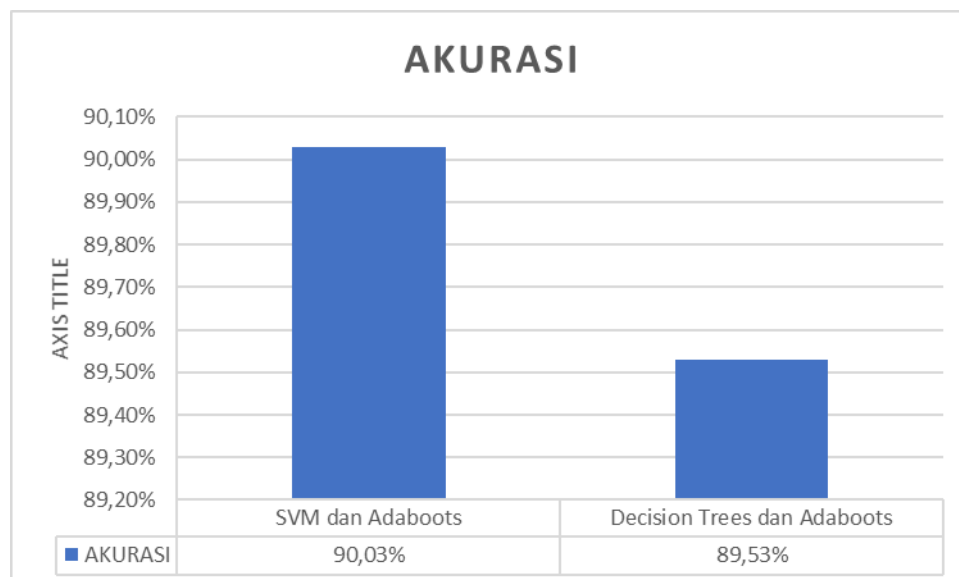
	precision	recall	f1-score	support
Negative	0.84	0.80	0.82	1146
Neutral	0.95	0.97	0.96	1511
Positive	0.89	0.89	0.89	1919
accuracy			0.90	4576
macro avg	0.89	0.89	0.89	4576
weighted avg	0.89	0.90	0.89	4576

Gambar 7 Akurasi *Decision Trees* Setelah Dioptimalkan Dengan *Adaboosts*

Lalu, pada penggunaan *Decision Trees* Setelah Dioptimalkan Dengan *Adaboosts*, tingkat akurasi yang didapat berada di angka 89,53% seperti gambar 7. Dapat dilihat, dengan adanya pengoptimalan menggunakan *Adaboost*, kinerja model *Decision Trees* meningkat sedikit dan mencapai akurasi yang lebih baik. Meskipun peningkatan akurasi tidak signifikan, penggunaan *Adaboost* membantu meningkatkan performa model dan memberikan hasil yang lebih baik dibandingkan dengan model *Decision Trees* tanpa dioptimalkan.

c) Perbandingan optimasi akurasi dengan *Adaboosts*

Berikut ini adalah hasil perbandingan algoritma SVM setelah dioptimalkan *Adaboosts* dan *Decision Trees* setelah dioptimalkan *Adaboosts*, dari perbandingan berikut ini dapat dilihat bahwa akurasi yang didapatkan dengan algoritma SVM setelah dioptimalkan *Adaboosts* adalah sebesar 90,03%. Sedangkan, akurasi yang diperoleh algoritma *Decision Trees* setelah dioptimalkan *Adaboosts* adalah sebesar 89,53%. Perbandingan akurasi ini dapat dilihat dari gambar 8.



Gambar 8 Perbandingan akurasi SVM dan *Decision Trees* (Setelah Dioptimalkan *Adaboosts*)

4. KESIMPULAN

Dari penjelasan diatas ini dapat diambil kesimpulan penelitian dengan menggunakan SVM setelah dioptimalkan *Adaboost* dan *Decision Trees* setelah dioptimalkan *Adaboost* dilakukan agar dapat mengetahui perbandingan nilai akurasi yang dihasilkan.

Pengujian dalam penelitian ini menggunakan *Accuracy* untuk melihat akurasi prediksi yang didapatkan dari hasil pengujian. Dari nilai akurasi yang terdapat diatas, dapat diketahui algoritma yang lebih baik digunakan pada penerapan sentiment deteksi ujaran kebencina sehingga dapat disimpulkan algoritma SVM dan *Decision Tree* setelah dioptimalkan dengan *Adaboost* pada penggunaan *dataset* yang ada.

Kesimpulan akhir dari penerapan algoritma *Decision Tree* dan *Adaboost*, menghasilkan tingkat akurasi sentiment deteksi ujaran kebencina yang baik. Namun, masih lebih baik algoritma SVM dan *Adaboost* dari pada algoritma *Decision Tree* dan *Adaboost*.

5. SARAN

Pada *dataset*, tidak terdapat *error* sehingga *dataset* asli yang digunakan. Namun, *dataset* tersebut masih bisa diolah kembali agar dapat menghasilkan hasil yang lebih baik. Serta belum adanya perbandingan analisis akurasi dengan algoritma yang lain dengan menggunakan *dataset* yang sama. Untuk mengimprove akurasi dapat mengubah parameter *sampling linier* disesuaikan dengan *dataset*.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tim Redaksi Jurnal Teknika Politeknik Negeri Sriwijaya yang telah memberi kesempatan, sehingga artikel ilmiah ini dapat diterbitkan.

DAFTAR PUSTAKA

- [1] APJII, *Profil Internet Indonesia 2022*. Jakarta: Pustaka Pelajar Asosiasi Penyelenggara Jasa Internet Indonesia, 2022. [Online]. Available: <https://apjii.or.id/survei2022x>
- [2] Ferdinandus Setu, "Warganet Paling Banyak Laporkan Akun Twitter SIARAN PERS NO. 08/HM/KOMINFO/01/2019," *Kementerian Komunikasi dan Informatika Republik Indonesia*, 2019. [Online]. Available: https://www.kominfo.go.id/index.php/content/detail/15852/siaran-pers-no-08hmkominfo012019-tentang-warganet-paling-banyak-laporkan-akun-twitter/0/siaran_pers
- [3] *Buku saku penanganan ujaran kebencian (hate speech)*. Komisi Nasional Hak Asasi Manusia Republik Indonesia, 2015. [Online]. Available: <https://books.google.co.id/books?id=F6U0xQEACAAJ>
- [4] Veronika and A. E. Raharjo, "Profiling news consumption on social media," *JKP*, vol. 5, no. 4, pp. 320–334, Jul. 2021, doi: 10.25139/jkp.v5i4.3794.
- [5] G. Florin, *Data mining: concepts, models and techniques*. in Intelligent systems reference library, no. 12. Berlin, Heilelberg: Springer-Verlag, 2011.
- [6] W. A. Luqyana, I. Cholissodin, and R. S. Perdana, "Analisis Sentimen Cyberbullying pada Komentar Instagram dengan Metode Klasifikasi Support Vector Machine".
- [7] R. M. Yazid, F. R. Umbara, and P. N. Sabrina, "Deteksi Ujaran Kebencian dengan Metode Klasifikasi Naïve Bayes dan Metode N-Gram pada Dataset Multi-Label Twitter Berbahasa Indonesia," *INDEX*, vol. 4, no. 2, pp. 46–52, Jan. 2023, doi: 10.36423/index.v4i2.894.

-
- [8] F. Ihsan, I. Iskandar, N. S. Harahap, and S. Agustian, "Decision tree algorithm for multi-label hate speech and abusive language detection in Indonesian Twitter," *J. Teknol. dan Sist. Komput*, vol. 9, no. 4, pp. 199–204, Oct. 2021, doi: 10.14710/jtsiskom.2021.13907.
- [9] M. R. A. Nasution and M. Hayaty, "Perbandingan Akurasi dan Waktu Proses Algoritma K-NN dan SVM dalam Analisis Sentimen Twitter," *Jl. Jurnal. Informatika*, vol. 6, no. 2, pp. 226–235, Sep. 2019, doi: 10.31311/ji.v6i2.5129.
- [10] S. Supangat, A. R. Amna, and T. Rahmawati, "Implementasi Decision Tree C4.5 Untuk Menentukan Status Berat Badan dan Kebutuhan Energi Pada Anak Usia 7-12 Tahun," *teknika*, vol. 7, no. 2, pp. 73–78, Nov. 2018, doi: 10.34148/teknika.v7i2.90.
- [11] E. O. Ogunseye, C. A. Adenusi, A. C. Nwanakwaugwu, S. A. Ajagbe, and S. O. Akinola, "Predictive Analysis of Mental Health Conditions Using AdaBoost Algorithm," *paradigmplus*, vol. 3, no. 2, pp. 11–26, Aug. 2022, doi: 10.55969/paradigmplus.v3n2a2.
- [12] X. Fang and J. Zhan, "Sentiment analysis using product review data," *Journal of Big Data*, vol. 2, no. 1, p. 5, Dec. 2015, doi: 10.1186/s40537-015-0015-2.