Terakreditasi SINTA Peringkat 2
Surat Keputusan Dirjen Penguatan Riset dan Pengembangan Ristek Dikti No. 10/E/KPT/2019

Terbit online pada laman web jurnal: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 5 No. 5 (2021) 896 - 903 ISSN Media Elektronik: 2580-0760

Analisis Perbandingan SVM, XGBoost dan Neural Network pada Klasifikasi Ujaran Kebencian

Suwarno[1], Ryo Kusnadi[2]
[1,2]Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Internasional Batam
[1]suwarno.liang@uib.ac.id, [2]ryokusnadi@gmail.com

Abstract

In social media, it is found that hate speech is conveyed in the form of text, images and videos, as a result it can provoke certain people to do things that are against the law and harm other person. Therefore, it is necessary to make early detection of hate speech by utilizing machine learning algorithms. This study is to analyze the level of accuracy, precision, recall and F1-Score of 3 kinds of algorithms (SVM, XGBoost, and Neural Network) in the classification of hate speech, using datasets sourced from public hate speech on Twitter in Indonesian. The results of the analysis show that the SVM algorithm has a level of accuracy (83.2%), precision (83%), recall (83%) and F1-score (83%), SVM occupies the highest level compared to XGBoost and Neural Network, so the SVM algorithm can be considered for use in hate speech classification.

Keywords: Hate Speech Classification, Machine Learning, Speech Recognition, Sentiment Analysis

Abstrak

Dalam media sosial dijumpai adanya ujaran kebencian baik yang disampaikan dalam bentuk teks, gambar dan video, sebagai akibatnya dapat memancing sejumlah pihak tertentu melakukan hal yang bertentangan dengan hukum dan merugikan pihak lain. Oleh karenanya perlu upaya deteksi dini terhadap ujaran kebencian dengan memanfaatkan algoritma pembelajaran mesin (machine learning). Penelitian ini adalah untuk menganalisis tingkat accuracy, precision, recall dan F1-Score terhadap 3 macam algoritma (SVM, XGBoost, dan Neural Network) dalam klasifikasi ujaran kebencian, dengan menggunakan datasets yang bersumber pada ujaran kebencian publik di Twitter dalam bahasa Indonesia. Hasil analisis menunjukkan algoritma SVM memiliki tingkat accuracy (83.2%), precision (83%), recall (83%) dan F1-score (83%), SVM menduduki tertinggi dibanding XGBoost dan Neural Network, sehingga algoritma SVM dapat dipertimbangkan untuk digunakan dalam klasifikasi ujaran kebencian.

Kata kunci: Klasifikasi ujaran kebencian, pembelajaran mesin, SVM, XGBoost, Neural Network

1. Pendahuluan

Saat ini, banyak orang mengungkapkan pendapat di depan umum. Hal ini dikarenakan pemerintah Indonesia menjamin bahwa setiap warga negaranya memiliki hak kebebasan untuk menyuarakan pendapatnya [1]. Namun, beberapa orang menggunakan kebebasan berbicara untuk melakukan tindakan negatif. Salah satunya yaitu ujaran kebencian. Munculnya ujaran kebencian di suatu lingkungan seperti ucapan dan pendapat dapat merugikan anggota minoritas dari segi etnis, ras, dan agama [2]. Di era digitalisasi seperti sekarang, di mana akses informasi dapat diakses dengan cepat, seringkali terlalu sulit untuk membedakan mana informasi yang akurat atau salah [3]. Hal itu bisa berdampak cukup besar terhadap peningkatan kasus ujaran kebencian yang terjadi di masyarakat. Prosedur penanganan kasus ujaran kebencian pada saat ini tidak mudah [4]. Sebelum menangani suatu kasus, polisi perlu menentukan apakah suatu kasus dapat digolongkan sebagai kasus ujaran kebencian atau tidak. Dengan bertambahnya jumlah kasus, pemeriksaan laporan masyarakat secara manual pada umumnya memakan waktu dan biaya yang besar. Hal ini membuktikan bahwa perlu adanya model prediktif yang dapat mendeteksi ujaran kebencian untuk mengurangi sumber daya manusia.

Tujuan utama dari klasifikasi ujaran kebencian adalah untuk menganalisis kata-kata yang telah diucapkan dan menghitung skor probabilitas ujaran kebencian sehingga dapat mengkalisifkasi ucapan tersebut merupakan ujaran kebencian atau tidak. Namun, data audio yang berisi apa yang telah diucapkan tidak dapat langsung digunakan

Diterima Redaksi: 16-09-2021 | Selesai Revisi: 04-10-2021 | Diterbitkan Online: 25-10-2021

sebagai input untuk klasifikasi ujaran kebencian. Data audio perlu diubah menjadi data teks, yang kemudian digunakan untuk klasifikasi. Proses pengubahan data suara menjadi data teks sering disebut Speech-to-Text [5]. Penelitian ini akan menggunakan library SpeechRecognition untuk melakukan pengenalan suara. Ujaran kebencian diklasifikasikan menjadi dua kelompok berdasarkan polaritas kata-katanya: ujaran kebencian dan bukan ujaran kebencian. Dataset yang digunakan dalam penelitian ini dikompilasi dari beberapa set data sentimen Twitter publik dalam Bahasa Indonesia yang sebelumnya telah di-crawl, dan dari [6]. Data preparation diperlukan sebelum modeling karena kumpulan data berisi sejumlah besar data yang noisy dan tidak terstruktur yang dapat mengganggu accuracy model [7]. Tahapan data preparation yang digunakan dalam penelitian ini adalah data cleaning, case-folding, tokenizing, stemming, stop words dan slang words removal, encoding target value, text vectorization, dan balancing datasets.

Beberapa penelitian telah dilakukan untuk mengidentifikasi ujaran kebencian dengan menggunakan dataset media sosial seperti Twitter, namun sebagian besar berfokus pada bagaimana mengklasifikasikan ujaran kebencian dalam teks bahasa Inggris [8].

Penelitian [9] membahas tentang kepuasan pelanggan pembayaran digital menggunakan analisis sentimen. Mereka meng-crawl data dari tweet twitter dan melakukan pembersihan data sebelum menjadi model. Setelah itu mereka menggunakan validasi silang sebagai metode statistik untuk mengevaluasi kinerja model Naive Bayes dan KNN. Hasil penelitian menunjukkan bahwa KNN memiliki accuracy yang lebih baik dibandingkan dengan Naive Bayes.

Klasifikasi lain dengan menggunakan SVM dilakukan oleh [10] dengan menggunakan data Twitter dan fitur pembobotan TF-IDF pada model SVM, menunjukkan pentingnya fitur unigram sebagai atribut penting dan bersifat unik untuk memproduksi accuracy yang baik.

Penelitian lain yang telah dilakukan oleh [11] menunjukkan bahwa dalam penelitian mereka tentang melakukan analisis sentimen di Indonesia, perlu dilakukan stemming untuk teks-teks yang telah bersufiks dengan menggunakan Perpustakaan Sastrawi dan melakukan penghapusan stopwords.

Penelitian yang dilakukan oleh [12] menunjukkan bahwa dengan adanya persebaran distribusi data dalam setiap kelas akan mempengaruhi hasil accuracy model yang akan digunakan. Sehingga perlunya dilakukan pemerataan kelas, salah satunya dengan menggunakan SMOTE. accuracy tertinggi dari model yang digunakan setelah melalui pemerataan kelas yaitu 83.4%. Sedangkan tanpa dilakukan pemerataan kelas, accuracy tertinggi yang didapat yaitu 71.2%.

Dari penelitian yang telah disebutkan diatas, tinjauan penelitian terdahulu dapat disimpulkan ke dalam Tabel 1.

Tabel 1. Tinjauan Penelitian Terdahulu

Peneliti	Hasil Penelitian	Relevansi Penelitian
Wisnu, Afif, dan Ruldevyani, 2020	Perlu dilakukan pembersihan data sehingga performa model menjadi lebih baik. Dengan menggunakan validasi silang dapat mengetahui performa model terbaik.	Perlu dilakukan pembersihan data outlier, duplikat, dan data kosong. Melakukan validasi silang untuk evaluasi model terbaik.
Dhina dan Yuliant, 2020	Dengan menggunakan TF-IDF, pembobotan yang dihasilkan menujukkan pentingnya fitur unigram untuk balancing dataset sehingga menghasilkan accuracy yang lebih baik.	Perlu dilakukan vektorisasi sebagai input parameter pada model yang akan dipakai.
Astiko dan Khodar, 2020	Dengan menggunakan sastrawi sebagai library stemming, kata yang terkandung dalam suatu kalimat akan dapat lebih akurat dalam klasifikasi berbagai jenis pola kalimat.	Perlu dilakukan stemming untuk menghilangkan kata yang telah bersufiks dan kata-kata yang tidak memiliki makna.
Putri, Sriadhi, Sari, Rahmadani, dan Hutahaean, 2020	Dengan mengunakan SMOTE untuk pemerataan distribusi kelas, accuracy yang didapatkan dari model mengalami peningkatan.	Perlu dilakukan pemerataan kelas menggunakan SMOTE agar mendapatkan accuracy yang lebih baik.

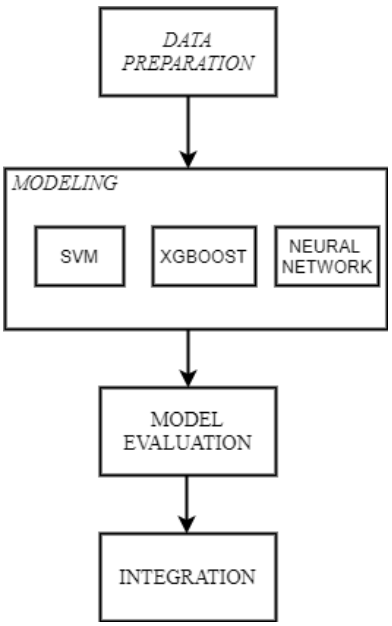
Penelitian ini menganalisis tiga algoritma yang optimal dalam melakukan klasifikasi ujaran kebencian. Algoritma yang telah dilatih akan digunakan dalam pengembangan website data input yang berupa ucapan akan diubah menjadi teks, penulis akan menggunakan salah satu model pembelajaran mesin (SVM, XGBOOST, atau Neural Network) yang memiliki hasil terbaik akan digunakan untuk mengklasifikasikan ujaran kebencian. Indikator accuracy, precision, recall dan F1-Score akan digunakan dalam pengujian hasil dalam penelitian ini. Penggabungan metode Feature Extraction TF-IDF dan Stemming para tahap data preparation dapat meningkatkan hasil pembelajaran mesin dalam

DOI: https://doi.org/10.29207/resti.v5i5.3506

mengklasifikasi ujaran kebencian. Dari hasil vektorisasi TF-IDF akan diperoleh data atribut penting dan atribut tidak penting. Atribut berpengaruh merupakan atribut yang bersifat unik dan memiliki dampak accuracy yang signifikan terhadap model, dimana atribut berpengaruh ini memiliki nilai TF-IDF yang tinggi. Sedangkan atribut yang tidak berpengaruh merupakan atribut yang memiliki dampak accuracy yang tidak signifikan terhadap model dan atribut ini merupakan kata-kata yang sering muncul dan memiliki nilai TF-IDF yang rendah [13].

2. Metode Penelitian

Berikut tahapan yang digunakan dalam penelitian ini: Data Preparation, Modeling, Evaluasi Model, dan Integrasi. Alur tersebut dapat dilihat pada Gambar 1.



Gambar 1. Alur metodologi penelitian

2.1. *Data Preparation*

Pada penelitian ini, langkah pertama yang dilakukan adalah menggabungkan dataset yang dikompilasi dari beberapa dataset sentimen publik Twitter dalam Bahasa Indonesia yang sebelumnya telah di-crawl, dan dari [6]. Selain itu, nilai duplikat pada dataset dihapus, sehingga total record dalam dataset berkurang dari 19109 menjadi 18793 baris dengan distribusi kelas 11282 baris data non-hate speech dan 7511 baris data hate speech. Dataset tersebut perlu melalui beberapa tahap data preparation sehingga dapat mendapatkan nilai hasil yang lebih bagus.

2.1.1. *Case-Folding*

Regular Expression digunakan dalam fase ini untuk menghilangkan beberapa bagian data, seperti @nama, tautan, baris baru, tanda baca, spasi awal dan akhir, dan spasi dengan spasi tunggal. Kemudian teks diubah menjadi huruf kecil.

2.*1.2. Tokenizing*

Tokenizing adalah proses pemecahan kalimat menjadi potongan-potongan kata, tanda baca, dan ungkapan bermakna lainnya berdasarkan ketentuan bahasa. Natural Language Toolkit (NLTK) digunakan dalam proses ini untuk fungsi tokenisasi kata. Contoh proses tokenizing dapat dilihat pada Tabel 2.

Tabel 2. Contoh Hasil Proses *Tokenizing*

Sebelum	Hasil *Tokenizing*
mari kita mengajarkan keluarga kita untuk menjauhi orang yang bukan seagama dengan kita.	'mari', 'kita', 'mengajarkan', 'keluarga', 'kita', 'untuk', 'menjauhi', 'orang', 'yang', 'bukan', 'seagama', 'dengan', 'kita'

2.1.3. *Stemming*

Stemming adalah proses menghilangkan kata imbuhan seperti awalan, akhiran, dan sisipan dari kata untuk sampai ke akar kata (bentuk awalan) dengan menggunakan fungsi *StemmerFactory* yang terdapat pada *library Sastrawi*. Contoh hasil *stemming* dapat dilihat pada Tabel 3.

Tabel 3. Contoh Hasil Proses *Stemming*

Sebelum	Hasil *Stemming*
'mari', 'kita', 'mengajarkan', 'keluarga', 'kita', 'untuk', 'menjauhi', 'orang', 'yang', 'bukan', 'seagama', 'dengan', 'kita'	'mari', 'kita', ajar, 'keluarga', 'kita', 'untuk', jauh, 'orang', 'yang', 'bukan', agama, 'dengan', 'kita'

2.1.4. *Stop Words Removal*

Stop words Removal adalah langkah untuk menghilangkan kata-kata yang kurang bermakna dalam Bahasa Indonesia dengan menggunakan *library Natural Language Tool Kit (NLTK).* Contoh proses *Stop Words Removal* dapat dilihat pada Tabel 4.

Tabel 4. Contoh Hasil Proses *Stop Words Removal*

Sebelum	Hasil *Stop Words Removal*
mari, 'kita', 'cipta', 'lingkungan', 'yang', 'harmonis'	mari, 'kita', 'cipta', 'lingkungan', 'harmonis'

2.1.5. *Slang Words Removal*

Slang words Removal adalah proses menghilangkan kata-kata bahasa tidak baku dan menggantinya dengan arti sebenarnya. Penelitian ini menggunakan dataset slangword publik Bahasa Indonesia yang tersedia di repositori publik Github. Contoh konversi slangword dapat dilihat pada Tabel 5.

Tabel 5. Contoh Hasil Proses *Slang Words Removal*

Sebelum	Hasil
adlh	adalah
km	kamu
otw	dalam perjalanan
gws	cepat sembuh

DOI: https://doi.org/10.29207/resti.v5i5.3506

kere	tidak punya uang
doku	uang
caper	cari perhatian

2.1.6. *Encoding* Nilai Target

Encoding Nilai Target adalah proses penggantian nilai kolom categorical target dengan nilai numerik. Langkah ini penting karena data harus numerik agar komputer dapat memproses data tersebut [14]. Karena nilai target hanya memiliki dua kelas, maka digunakan label encoding dalam penelitian ini.

2.1.7. *Text Vectorization*

Text Vectorization adalah teknik untuk menghitung bobot setiap kata yang paling umum untuk pencarian informasi. Metode Term Frequency-Inverse Document Frequency (TF-IDF) digunakan karena efisien, mudah dan akurat [15]. Untuk setiap token (kata) di setiap dokumen dalam korpus. Frekuensi kalimat yang muncul dihitung dalam *fw,d* dan IDF dihitung dalam *log(|D|/fw,D)*. |D| adalah nomor dalam dokumen. Setelah *fw,d* masing-masing dokumen diketahui maka dilakukan proses sortasi dimana semakin besar nilai *fw,d* maka nilai TF-IDF akan semakin kecil. Persamaan (1) merupakan rumus untuk TF-IDF. Data yang setelah ditransform melalui TF-IDF akan memperoleh nilai-nilai frekuensi untuk menentukan atribut tersebut berpengaruh atau tidak. Semakin besar nilai TF-IDF tersebut maka semakin berpengaruh dan penting kata tersebut. Jika TF-IDF memiliki nilai kecil maka kata tersebut merupakan kata-kata yang paling umum.

$$TFIDFwd = fw,d \times log(|D|/fw,D) \quad (1)$$

2.1.8. *Dataset Balancing*

Dalam penilitian ini, distribusi data antara ujaran kebencian dan bukan ujaran kebencian tidak seimbang, maka perlu menggunakan Teknik *Synthetic Minority Oversampling(SMOTE)* untuk mencegah kasus *over-fitting*. Untuk menyeimbangkan jumlah *instance*, pendekatan *SMOTE* menggunakan metode *oversampling* yang beroperasi pada level *feature* kelas.

2.2. *Modeling*

Dataset yang telah melalui tahap Data Preparation dimasukkan ke dalam model pembelajaran mesin seperti: SVM, XGBoost, dan Neural Network.

2.2.1. *Support Vector Machine (SVM)*

Support vector machine (SVM) adalah model pembelajaran mesin *supervised learning* yang menggunakan algoritma klasifikasi untuk menyelesaikan masalah klasifikasi dua kategori. Setelah memberikan set model SVM dari data pelatihan berlabel untuk setiap kategori, SVM dapat mengkategorikan teks baru. Berikut ini adalah cara kerja *SVM*. Asumsikan bahwa titik data ditunjukkan oleh {{(xi , yi)} di mana $xi = \{x1, x2, \ldots\ldots, xn\} \in Rn \quad , \quad yi \in \{+1, -1\}$.

Masalah optimasi berikut dapat diselesaikan dengan menggunakan Persamaan (2) yang merupakan persamaan *hard margin linear SVM*.

$$\min_{w,b} \ \frac{1}{2} ||w||^2 \quad (2)$$

Diikuti dengan persamaan (3)

$$s.t.\, yi(w^T x_i + b) \geq 1, \forall i, \quad (3)$$

Dimana $weight$(w) and $bias$(b) adalah $parameter$. Fungsi $weight$ bisa diperloleh melalui persamaan (4).

$$w = \sum_{i=1}^{N} a_i y_i x_i \quad (4)$$

Sedangkan untuk fungsi *bias* dapat diperoleh melalui persamaan (5).

$$b = -\frac{1}{2}\,(w.x^+ + w.x^-) \quad (5)$$

2.2.2. *Extreme Gradient Boosting (XGBoost)*

Extreme Gradient Boosting (XGBoost) adalah algoritma berbasis *decision tree* [16]. Model teresbut merupakan algoritma *tree ensemble* yang terdiri dari beberapa pohon klasifikasi dan regresi. Algoritma XGBoost melakukan optimasi lebih cepat dibandingkan implementasi *Gradient Boosting Method* lainnya baik dalam masalah klasifikasi maupun regresi [17]. Dalam algoritma berbasis pohon, *inner nodes* mewakili nilai untuk atribut pengujian dan *leaf nodes* dengan skor mewakili keputusan. Hasil prediksi adalah total skor yang diprediksi oleh pohon K seperti persamaan (6).

$$\widehat{y_l} = \sum_k^K f_k(x_i), f_k \in F \quad (6)$$

Dimana setiap paragraf dapat terdiri dari beberapa subparagraf, yang $\sum_{i-1}^{n} l\left(y_i, \widehat{y_l}\right)$ adalah fungsi kerugian yang dapat dibedakan untuk mengukur apakah model tersebut cocok untuk set data pelatihan dan $\sum_k^K \Omega(f_k)$ adalah item yang menentukan kompleksitas model yang dapat dilihat di persamaan (7).

$$obj(\theta) = \sum_{i-1}^{n} l\left(y_i, \widehat{y_l}\right) + \sum_k^K \Omega(f_k) \quad (7)$$

2.2.3. *Neural Network*

Neural Network merupakan paradigma pengolahan informasi yang terinspirasi dari sistem sel saraf dalam biologi [18]. Untuk menentukan output, setiap neuron menggunakan *activation function* yang didapatkan pada jumlah *input* yang diterima. Besarnya *output* ini kemudian dibandingkan dengan suatu *treshold*. *Neural Network* didefinisikan oleh tiga hal, Pertama adalah pola hubungan antar neuron yang disebut dengan arsitektur jaringan. Kedua adalah metode untuk menentukan *weight* dari *link*. Dan yang terakhir adalah *activation function*, yaitu fungsi yang digunakan untuk menentukan keluaran suatu neuron. Misalkan neuron Y menerima input dari neuron $X_1, X_2, \ldots, X_n$ dengan activation function $x_1, x_2, \ldots, x_n$ dan *weight*

DOI: https://doi.org/10.29207/resti.v5i5.3506

connection $w_1, w_2, \ldots, w_n$. Kemudian jaringan input pada neuron ke-j ditunjukkan pada Persamaan (8).

$$y_{inj} = \sum_{i=1}^{n} x_i w_{ij} \quad (8)$$

2.3. Evaluasi

Setelah melalui tahap Data Preparation dan Modeling, model yang sebelumnya telah dibangun akan dievaluasi berdasarkan metriks performa yang pada umumnya digunakan pada kasus klasifikasi. *Accuracy* digunakan untuk menganalisis ketepatan model dalam klasifikasi. *Precision* merupakan rasio antara True Positive dengan total seluruh positive. *Recall* merupakan rasio dalam menemukan semua instansi positif dengan data yang True Positive. Sedangkan *F1 Score* merupakan rata-rata harmonik tertimbang dari precision dan recall. Nilai-nilai tersebut dapat dihitung melalui persamaan berikut.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

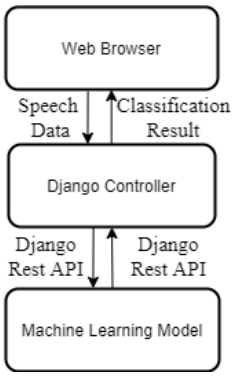
$$Recall = \frac{TP}{TP + FN} \quad (11)$$

$$F1\ Score = \frac{Precision \times Recall}{Precision + Recall} \quad (12)$$

Dimana True Positive (TP) merupakan hasil di mana model benar memprediksi kelas *positive*. *True Negative* (TN) adalah hasil di mana model benar memprediksi kelas *negative*. Sedangkan *False Positive* (FP) merupakan kesalahan dalam klasifikasi biner di mana hasil tes salah menunjukkan adanya suatu kondisi seperti teks tersebut merupakan ujaran kebencian ketika tidak mengandung hal tersebut. *False Negative* (FN) adalah kesalahan di mana hasil salah klasifikasi untuk menunjukkan adanya suatu kondisi ketika kondisi tersebut itu ada.

2.4. Integrasi

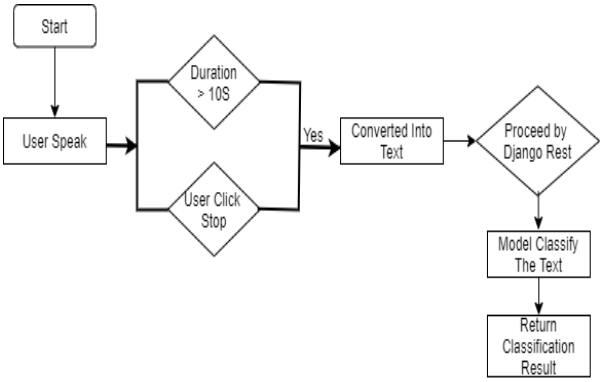
Aplikasi klasifikasi ujaran kebencian ini dibangun dengan menggunakan framework Django sebagai web framework karena framework tersebut memiliki *built-in template* yang memudahkan proses pengembangan [19]. Setiap 10 detik, atau ketika diklik berhenti merekam, ucapan yang dimasukkan ke dalam aplikasi diubah menjadi teks dengan menggunakan *library SpeechRecognition* dan data teks akan diproses dengan menggunakan *Django Rest Framework* ke dalam algoritma pembelajaran mesin untuk melakukan klasifikasi. Pembelajaran mesin yang telah dilatih sebelumnya disimpan dengan menggunakan format *.joblib*, yang bekerja dengan sejumlah besar data [20]. Algoritma mesin akan mengembalikan hasil dalam format biner, jika hasil pengembaliannya satu, situs web akan menampilkan "*hate speech!*" dan jika hasilnya nol, maka website akan kembali "*not hate speech*!" hasil. Rancangan sistem dapat ditunjukkan pada Gambar 2.



Gambar 2. Rancangan Sistem

3. Hasil dan Pembahasan

Tahap ini mencakup topik aplikasi sistem seperti evaluasi atribut, evaluasi pembelajaran mesin, dan implementasi sistem. Alur kerja sistem klasifikasi hate speech dapat dilihat pada Gambar 3.



Gambar 3. Alur kerja sistem

3.1. Evaluasi Atribut Berpengaruh dan Tidak

Dengan menggunakan data yang sebelumnya telah melalui data preparation, data tersebut ditransformasi dalam bentuk TF-IDF. Berikut adalah beberapa sampel atribut berpengaruh dan tidak berpengaruh yang ditunjukkan pada Tabel 6.

Tabel 6. Sampel Atribut Berpengaruh dan tidak

Atribut Berpengaruh	Atribut Tidak Berpengaruh
bajing	siapin
asing	diperingatin
viralkan	kurun
cantik	diperhatiin
berak	siapin
bebal	kumpeni
banci	purba

3.2. Evaluasi Pembelajaran Mesin

Semua pengujian dilakukan menggunakan bahasa pemrograman Python untuk pengembangan web, Scikit-learn untuk melatih model Extreme Gradient Boosting (XGBoost) dan Support Vector Machine (SVM), dan Tensorflow untuk model Neural Network.

DOI: https://doi.org/10.29207/resti.v5i5.3506

3.2.1. Evaluasi Support Vector Machine (SVM)

Dengan menggunakan daftar parameter pada Tabel 7 diperoleh hasil sebagai berikut.

Tabel 7. Evaluasi Support Vector Machine

Parameter	Accuracy	Precision	Recall	F1-Score
Kernel=Linear Gamma = Scale C = 1.0 tol = 1e-3 degree = 3	0.832	0.83	0.83	0.83
Kernel=Sigmoid Gamma = Scale C = 1.0 tol = 1e-3 degree = 3	0.79	0.79	0.78	0.79
Kernel = rbf Gamma = auto C = 1.0 tol = 1e-3 degree = 3	0.49	0.49	1.00	0.66

Dari hasil yang diperoleh dari Tabel 7, dapat dilihat kernel linear adalah kernel yang paling cocok didalam tugas klasifikasi ujaran menggunakan Support Vector Machine. Dari penilitian ini dapat dilihat Kernel yang digunakan sangat mempengaruhi accuracy yang didapatkan oleh model.

3.2.2. Evaluasi Extreme Gradient Boosting (XGBoost)

Dengan menggunakan beberapa parameter yang terdapat pada XGBoost, didapatkan hasil sebagai berikut.

Tabel 8. Evaluasi Extreme Gradient Boosting

Parameter	Accuracy	Precision	Recall	F1-Score
model = Randomized Search CV booster=gbtree learning_rate= 0.025,0.05,0.075, 0.1 n_estimators= 100, 250, 500 max_depth=6,7, 8	0.794	0.79	0.65	0.72
model= GridSearchCV C=0.001,0.01, 0.1, 1, 10,100 gamma=0.001, 0.01,0.1,1,10,100	0.796	0.78	0.72	0.75

Dari hasil yang didapatkan dari Tabel 8. Dapat dilihat dengan menggunakan GridSearchCV, model XGBoost memiliki hasil yang lebih baik. Parameter GridSearchCV merupakan cara dengan mencoba semua kemungkinan kombinasi parameter yang diinginkan dan menemukan yang terbaik.

3.2.3. Evaluasi Neural Network

Dengan menggunakan beberapa parameter yang terdapat pada Neural Network, didapatkan hasil sebagai berikut.

Tabel 9. Evaluasi Neural Network

Parameter	Accuracy	Precision	Recall	F1-Score
optimizer = adam loss = binary_ crossentropy epochs =10	0.828	0.80	0.86	0.81
optimizer = adam loss = categorical_ crossentropy epochs =10	0.491	0.49	0.99	0.64
optimizer = RMSProp loss = categorical_ crossentropy epochs =10	0.829	0.82	0.82	0.82

Dari hasil yang didapatkan dari Tabel 9. Dapat dilihat dengan menggunakan optimizer RMSProp, model Neural Network memiliki hasil yang lebih baik secara keseluruhan.

3.3. Evaluasi Secara Kesuluruhan

Setelah data divektorkan dengan menggunakan TF-IDF dan class balancing, data tersebut dimasukkan ke dalam masing-masing model machine learning, dan dievaluasi perfoma tersebut. Gambaran kinerja model dapat dilihat pada Tabel 10 dan Tabel 11. Kinerja model tersebut merupakan hasil dari performa yang terbaik yang telah diujikan menggunakan model yang telah ditentukan.

Tabel 10. Kinerja Model Sebelum Data Preparation

	SVM	XGBoost	Neural Network
Accuracy	**0.812**	0.794	0.782
Precision	**0.81**	0.79	0.76
Recall	**0.70**	0.65	0.65
F1-Score	**0.75**	0.72	0.67

Tabel 11. Kinerja Model Setelah Data Preparation

	SVM	XGBoost	Neural Network
Accuracy	**0.832**	0.796	0.829
Precision	**0.83**	0.78	0.82
Recall	**0.83**	0.72	0.82
F1-Score	**0.83**	0.75	0.82

Dari Tabel 10 dan Tabel 11 dapat dilihat terjadi peningkatan yang cukup banyak setelah melalui tahap data preparation, hal ini membuktikan perlunya tahap ini agar data yang digunakan dapat memberi hasil yang lebih baik terhadap model yang digunakan. Tabel 11 menunjukkan bahwa model SVM memiliki hasil terbaik pada *Accuracy*, *Precision,Recall* dan *F1-Score*. Dari

DOI: https://doi.org/10.29207/resti.v5i5.3506

penelitian ini dapat disimpulkan bahwa algoritma SVM lebih baik dalam melakukan tugas klasifikasi untuk dataset sentimen tidak terstruktur Bahasa Indonesia dari dataset publik twitter. Algoritma SVM memiliki *Error Rate* sekitar 16,8%. Hal ini disebabkan oleh kesulitan algoritma dalam membedakan antara atribut berpengaruh dan tidak berpengaruh dalam proses prediksi. Data yang telah digunakan juga sangat berpengaruh terhadap *accuracy*, algoritma SVM membutuhkan data yang dinormalisasi untuk mendapatkan *accuracy* yang baik.

$$\text{Error Rate} = \frac{FP + FN}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (13)$$

Error Rate = ~ 0.168

Dari hasil vektorisasi TF-IDF, diperoleh kata yang umum (nilai TF-IDF rendah) seperti kritik dan kata yang bersifat penting (nilai TF-IDF tinggi) seperti kata viralkan. Dimana semakin tinggi nilai TF-IDF akan berpengaruh pada model SVM yang digunakan. Model SVM dan TF-IDF akan diekspor ke format .joblib. TF-IDF akan digunakan sebagai vectorizer untuk mengubah parameter input.

3.4. Implementasi Sistem

Sistem ini menggunakan suara sebagai parameter input untuk melakukan klasifikasi. Data suara yang dikumpulkan selama sepuluh detik ataupun waktu user klik stop akan dikonversi menjadi teks dan teks tersebut akan ditransformasi menggunakan TF-IDF. Model akan melakukan klasifikasi hate speech berdasarkan teks tersebut dengan menggunakan model SVM yang sebelumnya dikonversi dalam bentuk .joblib. Setelah selesai melakukan klasifikasi data input yang berupa teks yang user bicarakan sebelumnya dan hasil klasifikasi akan dimasukkan ke dalam database. User Interface sistem dapat ditunjukkan pada Gambar 4.

Gambar 4. User Interface Sistem

4. Kesimpulan

Dari hasil penelitian yang telah dilakukan, algoritma SVM dengan menggunakan kernel linear beserta parameter Gamma Scale, C 1.0, tol 1e-3 dan degree 3 memiliki hasil terbaik daripada algoritma XGBoost dan Neural Network yang telah diuji dari segi accuracy, precision, recall dan F1-Score. Model SVM memiliki Error Rate sekitar 16,8%. Hal ini disebabkan oleh kesulitan algoritma dalam membedakan antara atribut berpengaruh dan tidak berpengaruh dalam proses prediksi. Algoritma SVM membutuhkan data yang dinormalisasi untuk mendapatkan accuracy yang lebih baik.

Saran yang dapat dipertimbangkan untuk penelitian selanjutnya adalah dataset yang digunakan dalam pelatihan model klasifikasi ujaran kebencian perlu memiliki jumlah data yang lebih seimbang dan menggunakan metode Indonesian Deep Learning seperti IndoBert untuk mendapatkan accuracy yang lebih baik.

Daftar Rujukan

[1] Marwadianto and H. A. Nasution, "Hak Atas Kebebasan Berpendapat Dan Berekspresi Dalam Koridor Penerapan Pasal 310 Dan 311 KUHP," J. HAM, vol. 11, no. 1, pp. 1–4, 2020, http://dx.doi.org/10.30641/ham.2020.11.1-25

[2] M. Bilewicz and W. Soral, "Hate Speech Epidemic. The Dynamic Effects of Derogatory Language on Intergroup Relations and Political Radicalization," Polit. Psychol., vol. 41, no. S1, pp. 3–33, 2020. https://doi.org/10.1111/pops.12670

[3] M. T. Palupi, "Hoax: Pemanfaatannya Sebagai Bahan Edukasi Di Era Literasi Digital Dalam Pembentukan Karakter Generasi Muda," J. Skripta, vol. 6, no. 1, pp. 1–12, 2020. https://doi.org/10.31316/skripta.v6i1.645

[4] A. Briliani, B. Irawan, and C. Setianingsih, "Hate speech detection in indonesian language on instagram comment section using K-nearest neighbor classification method," Proc. - 2019 IEEE Int. Conf. Internet Things Intell. Syst. IoTaIS 2019, pp. 98–104, 2019. https://doi.org/10.1109/IoTaIS47347.2019.8980398

[5] W. Singh, "Multilingual Speech to Text Conversion – A Review," Adv. Math. Sci. J., vol. 9, no. 6, pp. 3963–3970, 2020. http://dx.doi.org/10.37418/amsj.9.6.77

[6] R. Hendrawan, Adiwijaya, and S. Al Faraby, "Multilabel Classification of Hate Speech and Abusive Words on Indonesian Twitter Social Media," 2020 Int. Conf. Data Sci. Its Appl. ICoDSA 2020, 2020. https://doi.org/10.1109/ICoDSA50139.2020.9212962

[7] A. W. Pradana and M. Hayaty, "The Effect of Stemming and Removal of Stopwords on the Accuracy of Sentiment Analysis on Indonesian-language Texts," Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control, vol. 4, no. 3, pp. 375–380, 2019. http://dx.doi.org/10.22219/kinetik.v4i4.912

[8] P. Meel and D. K. Vishwakarma, "Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities," Expert Syst. Appl., vol. 153, p. 112986, 2020. https://doi.org/10.1016/j.eswa.2019.112986

[9] H. Wisnu, M. Afif, and Y. Ruldevyani, "Sentiment analysis on customer satisfaction of digital payment in Indonesia: A comparative study using KNN and Naïve Bayes," J. Phys. Conf. Ser., vol. 1444, no. 1, pp. 0–10, 2020. https://doi.org/10.1088/1742-6596/1444/1/012034

[10] N. F. Dhina and S. Yuliant, "Sentiment Analysis on KAI Twitter Post Using Multiclass Support Vector Machine (SVM)," J. RESTI (Rekayasa Sist. dan Teknol. Informasi), vol. 4, no. 5, pp. 846–853, 2020. https://doi.org/10.29207/resti.v4i5.2231

[11] F. Astiko and A. Khodar, "Membangun Model Machine Learning Untuk Meninjau Layanan Indosat Ooredoo Dari Twitter Menggunakan Naive Bayes Classifier," J. Appl.

DOI: https://doi.org/10.29207/resti.v5i5.3506

Comput. Sci. Technol. (JACOST), vol. 1, no. 2, pp. 61–66, 2020. https://doi.org/10.52158/ jacost.v1i2.79

[12] T. T. A. Putri, S. Sriadhi, R. D. Sari, R. Rahmadani, and H. D. Hutahaean, "A comparison of classification algorithms for hate speech detection," IOP Conf. Ser. Mater. Sci. Eng., vol. 830, no. 3, 2020. https://doi.org/10.1088/1757-899X/830/3/032006

[13] S. L. Bhutia, S. Borah, R. Pradhan, and B. Sharma, "An Experiment on Parameter Selection for Landslide Susceptibility Mapping using TF-IDF," J. Phys. Conf. Ser., vol. 1712, no. 1, 2020. https://doi.org/10.1088/1742-6596/1712/1/012029

[14] N. Sharma, H. V. Bhandari, N. S. Yadav, and H. V. J. Shroff, "Optimization of IDS using Filter-Based Feature Selection and Machine Learning Algorithms," Int. J. Innov. Technol. Explor. Eng., vol. 10, no. 2, pp. 96–102, 2020. https://doi.org/10.35940/ijitee.B8278.1210220

[15] S. Xiao and W. Tong, "Prediction of User Consumption Behavior Data Based on the Combined Model of TF-IDF and Logistic Regression," J. Phys. Conf. Ser., vol. 1757, no. 1, 2021. https://doi.org/10.1088/1742-6596/1757/1/012089

[16] Y. Jiang, G. Tong, H. Yin, and N. Xiong, "A Pedestrian Detection Method Based on Genetic Algorithm for Optimize XGBoost Training Parameters," IEEE Access, vol. 7, 2019. https://doi.org/10.1109/ACCESS.2019.2936454

[17] M. Guo, Z. Yuan, B. Janson, Y. Peng, Y. Yang, and W. Wang, "Older Pedestrian Traffic Crashes Severity Analysis Based on an Emerging Machine Learning Xgboost," Sustain., vol. 13, no. 2, pp. 1–26, 2021. https://doi.org/10.3390/su13020926

[18] Y. Li et al., "Oxide-Based Electrolyte-Gated Transistors for Spatiotemporal Information Processing," Adv. Mater., vol. 32, no. 47, pp. 1–12, 2020. https://doi.org/10.1002/adma.202003018

[19] A. F. Rochim, A. Rafi, A. Fauzi, and K. T. Martono, "As-RaD System as a Design Model of the Network Automation Configuration System Based on the REST-API and Django Framework," Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control, vol. 4, pp. 291–298, 2020. https://doi.org/10.22219/kinetik.v5i4.1093

[20] A. A. Gamova, A. A. Horoshiy, and V. G. Ivanenko, "Detection of Fake and Provokative Comments in Social Network Using Machine Learning," Proc. 2020 IEEE Conf. Russ. Young Res. Electr. Electron. Eng. EIConRus 2020, pp. 309–311, 2020. https://doi.org/10.1109/EIConRus49466.2020.9039423

DOI: https://doi.org/10.29207/resti.v5i5.3506