

# Perbandingan Tradisional dan *Ensemble Machine Learning* dalam Melakukan Klasifikasi Kalimat Ujaran Kebencian

Ridwan\*<sup>1</sup>, Riyan Latifahul Hasanah<sup>2</sup>, Eni Heni Hermaliani<sup>3</sup>

<sup>1</sup>Program Studi Ilmu Komputer, Universitas Nusa Mandiri, Jakarta

<sup>2,3</sup>Program Studi Sistem Informasi, Universitas Nusa Mandiri, Jakarta

e-mail: \*<sup>1</sup>14210121@nusamandiri.ac.id, <sup>2</sup>riyan.rlt@nusamandiri.ac.id,

<sup>3</sup>enie\_h@nusamandiri.ac.id

## Abstrak

Kalimat *Hate Speech* merupakan tindakan kejahatan yang diutarakan kepada individu atau kelompok berupa hinaan, fitnah, cacian yang berkaitan dengan ras, agama, budaya, dll. *Hate Speech* sering disampaikan melalui media sosial seperti Twitter. *Hate speech* mulai banyak tersebar di media sosial, sehingga penelitian ini bertujuan untuk melakukan analisis pengkategorian kalimat *hate speech* menggunakan *machine learning*. Pengkategorian dapat dilakukan dengan tahapan *pre-processing* yaitu *remove punctuations*, *lowercase*, *tokenizing*, *filtering* dan *stemming*. Dataset memiliki distribusi data yang tidak seimbang, sehingga metode *SMOTE* (*Synthetic Minority Over-sampling Technique*) sangat cocok untuk digunakan, dilanjutkan dengan menerapkan model *features engineering* yaitu *TF-IDF* (*Term Frequency-Inverse Document Frequency*) dan menggunakan algoritma *Logistic Regression*, *Decision Tree*, dan *Naïve Bayes*, kemudian dilakukan pengembangan algoritma *machine learning* menggunakan metode *ensemble*, yaitu *Adaptive Boosting* (*Adaboost*) dan *Random Forest*. Algoritma *Logistic Regression* mendapatkan nilai akurasi terbaik yaitu sebesar 91,40 dan mampu mengungguli algoritma lainnya.

**Kata kunci**—*hate speech*, *Logistic Regression*, *ensemble*, *Adaptive Boosting*

## 1. PENDAHULUAN

Media sosial merupakan sebuah alat untuk berkomunikasi secara *online* dengan tujuan untuk mempermudah seseorang dalam melakukan sosialisasi secara *intens* dan membutuhkan respon yang cepat. Media sosial menjadi kebutuhan di dalam masyarakat dan bukan hanya sekedar dijadikan sebagai alat komunikasi. Salah satu media sosial yang umum untuk digunakan oleh masyarakat Indonesia yaitu *Twitter*[1]. Masyarakat dapat menggunakan *Twitter* sebagai media komunikasi dan juga dapat memberikan opini kepada publik secara gratis melalui kalimat yang disebut dengan *Tweet*. *Tweet* dapat bersifat positif maupun negatif. *Tweet* yang bersifat positif seperti kalimat pujian, motivasi, saran positif atau kalimat yang mengundang semangat, sedangkan kalimat yang bersifat negatif seperti opini negatif terhadap seseorang atau kelompok, ujaran kebencian, fitnah atau celaan [2][4][5].

Saat ini media sosial mulai dijadikan alat untuk melakukan tindak kejahatan, salah satunya yaitu menyampaikan kalimat ujaran kebencian atau *hate speech*. Ujaran kebencian merupakan kalimat yang disampaikan untuk mengekspresikan rasa benci terhadap individu, kelompok, ras yang dilakukan dengan tujuan untuk merendahkan atau mempermalukan[6][8].

Pemerintah Indonesia telah berkerjasama dengan penegak hukum untuk menurunkan maraknya kalimat ujaran kebencian dengan cara memberikan teguran langsung atau pemanggilan kepada pihak berwenang. Pemerintah menerbitkan peraturan perundang undangan yaitu UU-ITE pasal 28 ayat 2 yang berisi bahwa setiap warga Indonesia dilarang untuk menyebar luaskan informasi yang dapat menimbulkan kebencian terhadap individu atau masyarakat tertentu.

Peraturan serupa pernah diperjelas dalam surat edaran (SE) Kapolri No. SE/06/X/2015 mengenai cakupan-cakupan dari beberapa jenis kalimat ujaran kebencian [8][9].

Etika dalam berinteraksi di media sosial perlu dipublikasikan dan diterapkan, sesuai karakter masyarakat Indonesia yang mengedepankan nilai kesopanan dan tata krama[4][10][11]. Namun dalam media sosial interaksi yang dilakukan tidak secara langsung yaitu secara online, dapat dijadikan kesempatan bagi seseorang yang memiliki niat jahat seperti ujaran kebencian, menghina, menghujat, melecehkan bahkan melakukan penipuan di media sosial[8][12].

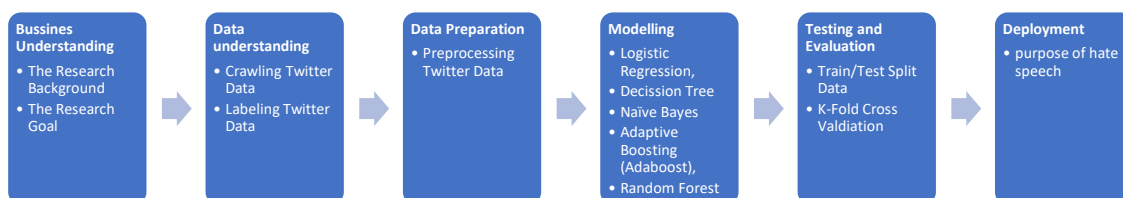
Beberapa penelitian yang sudah dilakukan untuk mendeteksi kalimat *hate speech* menggunakan *machine learning* oleh Akib Mohi Ud Din Khanday dengan menerapkan model *TF-IDF* (*Term Frequency-Inverse Document Frequency*) dan *Bag of Words*. Algoritma *machine learning* yang digunakan adalah *Logistic Regression (LR)*, *Multi-nomial Naïve Bayes (MNB)*, *Support Vector Machine (SVM)*, *Logistic Regression (LR)*, dan *Decision Tree* serta algoritma *ensemble* yaitu *Bagging*, *Adaboost*, *Random Forest* and *Gradient Stochastic Boosting Ensemble learning*. Hasil yang didapatkan algoritma *Decision Tree* yaitu 98% *precision*, 97% *recall*, 97% *F1 score* dan *accuracy* sebesar 97.9%, Sedangkan algoritma *ensemble* yaitu *Stochastic Gradient Boosting* menghasilkan 99% *precision*, 97% *recall*, 98% *F1 Score* dan *accuracy* sebesar 98.04% [7].

Penelitian selanjutnya yang dilakukan oleh Nabil Badria dengan menerapkan model *TF-IDF* (*Term Frequency-Inverse Document Frequency*) dan *Count Vectors*, algoritma yang digunakan yaitu *Multinomial Nave Bayes (MNB)*, *Logistic Regression (LR)*, *Random Forest (RF)*, *eXtreme Gradient Boosting machines (XGB)* dan *Support Vector Machines with RBF kernel (SVM)*. Hasil yang didapat yaitu pada *Not Offensive class* mendapatkan peningkatan akurasi yaitu sebesar 99% menggunakan algoritma Naïve Bayes, Sementara pada *Offensive class* mengalami peningkatan sebesar 89% dengan algoritma yang sama [8].

Penelitian ini memiliki tujuan untuk melakukan pengkategorian kalimat ujaran kebencian dengan menggunakan model *machine learning*. Tahapan *Features Engineering* yang digunakan yaitu *Term Frequency — Inverse Document Frequency (TF-IDF)* untuk memberikan bobot setiap kata. Untuk menangani tidak seimbangya distribusi data menggunakan metode *Synthetic Minority Over-sampling Technique (SMOTE)*. Untuk menjadikan distribusi data menjadi seimbang, algoritma *tradisional machine learning* yang digunakan yaitu *Logistic Regression*, *Decission Tree* dan *Naïve Bayes*, lalu dilakukan pengembangan algoritma *machine learning* menggunakan metode *ensemble*, yaitu *Adaptive Boosting (Adaboost)* dan *Random Forest*[11].

## 2. METODE PENELITIAN

Penelitian ini menggunakan metode *Cross-Industry Standard Process for Data Mining* (CRISP-DM), beberapa langkah yang harus dilakukan untuk mendapatkan hasil evaluasi terkait kalimat ujaran kebencian [15], berikut enam tahapan dari pengembangan *data mining* yang digunakan pada metodologi penelitian :



Gambar 1. Metode Penelitian

### 2.1 Bussines Understanding

Ujaran kebencian merupakan tindakan yang dilakukan dalam bentuk cacian, hinaan, provokasi, merendahkan salah satu pihak kepada individu atau kelompok[15], penelitian ini melakukan pengkategorian kalimat ujaran kebencian dengan menggunakan dua kategori yaitu

*hate speech* dan *non hate speech*. Tahapan yang dilakukan yaitu pengambilan *dataset twitter* menggunakan *Application Program Interface (API)*, tahap *pre-processing* data, *features engineering*, *splitting* data, evaluasi model menggunakan algoritma tradisional *machine learning* dan *ensemble machine learning*. Hasil akhir yaitu melihat algoritma apa yang memiliki *performance* yang paling baik untuk melakukan pengkategorian secara otomatis [7].

### 2.2 Data Understanding

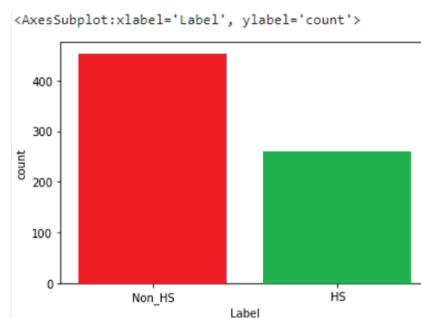
Dataset yang digunakan bersumber dari data publik dan berhasil terkumpul dikarenakan adanya peristiwa politik yaitu pemilihan gubernur dan wakil gubernur DKI Jakarta tahun 2017. Hal ini menarik dikarenakan salah satu calon gubernur dan wakil gubernur memiliki perbedaan jenis kelamin, rentang umur yang cukup dan salah satu calon berasal dari kelompok minoritas dari sisi agama. Data *twitter* dikumpulkan sejak bulan Februari 2017 yaitu pada fase pemilihan umum putaran pertama, selanjutnya pada bulan April 2017 pada fase pemilihan umum putaran kedua. Data *twitter* dikumpulkan berdasarkan hastag yaitu #DebatPilkadaDKI, #PilkadaJakarta2017 #SidangAhok, dll. Data yang berhasil dikumpulkan sebanyak 40.000 *tweet*. Setelah dilakukan proses pembersihan dan penghapusan *tweet* didapat sebanyak 1.100 data *tweet* yang sudah diberikan label secara manual yaitu *hate speech* dan *non hate speech* [14][4].

Tabel 1. Dataset Ujaran Kebencian Hasil Labeling Secara Manual

No	Tweet	Label
1	Kayak gini nih kampanye Tim AniesSandiUno ke kampung2 mengintimidasi masyarakat. Kampungan ye? #ASUtenan,"Like this, the Anienessandiuno team campaign to the villages intimidates the community. Kampungan Ye? #Asutenan"	<i>hate speech</i>
2	RT @prastow: Apapun itu, keberanian Ahok membongkar mafia anggaran menjadi tonggak penting menuju APBD yg kredibel dan akuntabel.", "RT @Prastow: Whatever it is, Ahok's courage to dismantle the budget mafia into an important milestone towards the credible and accountable APBD."	<i>non hate speech</i>

### 2.3 Data Preparation

Data *tweet* yang sudah terkumpul akan dilakukan pelabelan secara manual apakah termasuk kedalam kalimat ujaran kebencian atau tidak, jika ya maka akan diberikan label “*HS*” dan jika tidak akan diberikan label “*NonHS*”. Hasil dari pelabelan secara manual yaitu dari 1.100 *tweet* menjadi 713 *tweet* yang diberikan label dengan pembagian 260 *tweet* sebagai “*HS*” dan 453 *tweet* sebagai “*NonHS*”.



Gambar 1.. Perbedaan Distribusi data berdasarkan label "Non\_HS" dan "HS",

Dari barplot di atas didapati bahwa distribusi datanya tidak seimbang atau disebut dengan *unbalance* data, ketidak seimbangan kelas minoritas dan mayoritas dapat menyebabkan kelas mayoritas cenderung menghasilkan kinerja yang lebih baik hal ini dapat mengakibatkan *overfitting* pada saat pengujian menggunakan algoritma *machine learning*. Untuk menangani masalah tersebut penelitian ini menggunakan metode *Synthetic Minority Over-sampling Technique* (SMOTE) untuk menangani *unbalance data*, metode ini melakukan penambahan data sintetis pada kelas minoritas, dengan cara menambah data baru berdasarkan nilai ketetanggaan terdekat[16][17].

Sebelum dilakukan pemodelan menggunakan *machine learning* perlu dilakukan tahapan *Pre-Processing* data, dikarenakan data yang bersifat mentah atau memiliki banyak *noise* dapat mempersulit algoritma *machine learning* dalam melakukan pengkategorian. *Pre-Processing* ditujukan untuk mempersiapkan data sehingga dapat diolah oleh algoritma *machine learning* dengan melalui beberapa tahapan, yaitu :

### 2.3.1 Remove punctuations

Dataset yang sudah memiliki label *hate speech* dan *not hate speech* akan dilakukan proses pembersihan data dengan cara menghapus karakter yang tidak dibutuhkan seperti : *https:// (url)*, (koma), # (*hashtag*), (titik), @ (*mention*), dll[16].

### 2.3.2 Lowercase

Selanjutnya dataset yang sudah melalui tahap pembersihan akan dirubah secara otomatis kedalam huruf kecil dengan menggunakan `str.lower()` di dalam *python*, dengan tujuan untuk menyamaratakan bobot nilai pada masing masing kata [18].

### 2.3.3 Tokenizing

Tahap ini merupakan cara untuk memecah karakter menjadi beberapa kalimat atau bagian (kata/frasa) atau disebut dengan token.

### 2.3.4 Filtering

Tujuan dari tahap ini adalah untuk menghapus kata-kata yang sama sekali tidak memiliki arti pada saat melakukan pengkategorian otomatis. Penelitian ini menggunakan *stopword* Bahasa Indonesia yang didapatkan dari *library NLTK* untuk melakukan *filter* terhadap *Dataframe* dan menambahkan *kamusalay.csv* yang didalamnya mengandung bahasa gaul di Indonesia [8][19].

### 2.3.5 Stemming

Stemming yaitu proses untuk menghapus atau menghilangkan kata awalan dan akhiran untuk mengambil kata dasar atau akar kata. *Library* yang digunakan untuk melakukan proses ini yaitu menggunakan *library* Sastrawi untuk melakukan *stemming* Bahasa Indonesia [16].

## 2.4 Modeling

Pada tahap ini akan dilakukan proses pengkategorian menggunakan algoritma tradisional *machine learning* dan *ensemble machine learning*. Pengkategorian ini dilakukan berdasarkan kesamaan label dengan cara menemukan suatu pola yang dapat membedakan kelas-kelas tertentu berdasarkan kategori yang sudah ada yaitu *hate speech* dan *not hate speech* [2].

### 2.4.1 Logistic Regression

Algoritma klasifikasi yang digunakan untuk mencari apakah terdapat hubungan antara fitur diskrit/kontinu dengan probabilitas hasil output diskrit. *Logistic Regression* dikategorikan menjadi dua, yaitu *single logistic regression* dimana terdapat satu jenis variabel input, dan *multiple logistic regression* [14][2][14][20]. Berikut persamaan dari *Logistic Regression* :

$$\ln\left(\frac{p}{1-p}\right) = B_0 + B_1X \quad (1)$$

$\ln$  : Logaritma natural  
 $B_0 + B_1X$  : -persamaan OLS  
 $p$  : probabilitas logistik

### 2.4.2 Decision Tree (DT)

Perhitungan pada algoritma ini menggunakan struktur pohon keputusan dengan memodelkan beberapa kemungkinan sehingga didapati sebuah alternatif untuk memecahkan

---

masalah *Decision tree* mampu melakukan eliminasi pada sebuah data jika didalamnya tidak memiliki keterkaitan terhadap perhitungan [7][2][21]. Rumus dari algoritma *decision tree* adalah :

$$E(S) = \sum_{i=1}^C - P_i \log_2 P_i \tag{2}$$

Dimana **S** merupakan kondisi awal, **i** merupakan set kelas pada S yaitu seperti yes dan no, **P<sub>i</sub>** merupakan peluang suatu kejadian S.

#### 2.4.3 Multinomial Naïve Bayes (NB)

Algoritma klasifikasi untuk menghitung nilai probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai di dalam data. *Naïve bayes* akan mempelajari probabilitas sebuah kelas berdasarkan ciri-ciri yang ada di dalam data[2][21]. Algoritma ini merupakan klasifikasi probabilitas, karena itu disebut ‘*naïve*’ yang membuat asumsi kemunculan fitur tertentu tidak tergantung pada kemunculan fitur yang lain [7][4][11][14], berikut rumus *Naïve Bayes* (NB) :

$$P(x) = \frac{P(c) P(c)}{P(x)} \tag{3}$$

- P(c|x) : Probabilitas kelas (target) yang diberikan prediktor
- P(c) : Peluang pada kelas sebelumnya
- P(x|c) : Kemungkinan probabilitas kelas yang diberikan prediktor
- P(x) : Probabilitas sebelumnya dari predictor

#### 2.4.4 Random Forest (RF)

Struktur pohon yang dihasilkan dari algoritma ini sangatlah kompleks untuk sebuah data yang memiliki *variable* yang banyak. [9][22]. *Random forest* merupakan kumpulan *classifier* yang berbentuk pohon {h(x, θ<sub>k</sub>), k = 1, . . .} dimana θ<sub>k</sub> merupakan *vector* yang tersebar dan masing-masing pohon keputusan akan memilih kelompok yang paling populer pada input X[9][14].

#### 2.4.5 Ensemble Machine Learning

*Ensemble* merupakan cara melakukan klasifikasi dengan menggunakan kombinasi dari berbagai algoritma *machine learning* dengan tujuan untuk menghasilkan *performance* akurasi yang lebih baik dibandingkan dengan menggunakan satu algoritma saja. Terdapat beberapa jenis metode *ensemble* yaitu *bagging*, *boosting* dan *stacking* [7][22].

#### 2.4.6 Adaptive Boosting (Adaboost)

*Adaptive Boosting (Adaboost)* yaitu algoritma yang termasuk kedalam *ensemble machine learning* dengan menerapkan konsep *boosting*. Algoritma ini melakukan *boosting* kinerja *classification* dari sebuah algoritma *machine learning* untuk membuat sebuah *classifier* yang lebih kuat dan kemudian diberikan istilah *weak learner* [23][24].

#### 2.4.7 Term Frequency - Inverse Document Frequency (TF-IDF)

Langkah yang digunakan untuk melakukan ekstraksi fitur pada penelitian ini adalah menggunakan TF-IDF, metode ini bertujuan untuk menghitung bobot setiap kata dengan cara menghitung nilai *term frequency* (TF) dan *inverse document frequency* (IDF) untuk setiap token (*word*) pada setiap dokumen[1][2][11]. Sederhananya, metode TF-IDF menentukan berapa kali sebuah kata muncul dalam sebuah dokumen, berikut rumus dari TF-IDF :

$$W_{i,j} = \frac{n_{i,j}}{\sum_{j=1}^p n_{j,i}} \log_2 \frac{D}{d_j} \tag{4}$$

Keterangan :

- W<sub>i,j</sub> : Merupakan pembobotan Tf-Idf untuk *term* ke-j pada dokumen ke-i
- n<sub>i,j</sub> : Merupakan jumlah kemunculan *term* ke-j pada dokumen ke-i

- $p$  : Merupakan banyaknya *term* yang terbentuk  
 $\sum_{j=1}^p n_{j,i}$  : Merupakan jumlah kemunculan seluruh *term* pada dokumen ke- $i$   
 $D$  : Merupakan jumlah keseluruhan dokumen  
 $d_j$  : Merupakan banyaknya dokumen yang mengandung *term* ke- $j$

#### 2.4.8 Synthetic Minority Over-sampling Technique (SMOTE)

Merupakan salah satu metode untuk mengatasi ketidakseimbangan data atau biasa disebut dengan *oversampling*. Metode ini menyiapkan sampel pada kelas minoritas yang lebih banyak untuk dilakukan proses klasifikasi, sehingga saat proses klasifikasi dapat mencakup data yang lebih besar dalam mempelajari kelas minoritas [13].

#### 2.5 Testing dan Evaluasi

Evaluasi model yang digunakan yaitu, model train/test split data dan *K Fold Cross Validation*, train/test split melakukan perhitungan dengan cara membagi *dataset* menjadi dua bagian yaitu data latih dan data uji. Pembagian data sebesar 80:20 dimana 80% didistribusikan ke dalam data latih, dan 20% menjadi data uji. Cara kedua menggunakan model evaluasi *K Fold Cross Validation*, K-Fold akan menghasilkan satu kelompok data yang akan dijadikan data uji. Data inilah yang akan digunakan untuk proses pengujian sebanyak nilai  $K$ [6][9][25].

#### 2.6 Deployment

Pada tahap ini akan dilakukan pembuatan laporan atau proses implementasi *data mining* dengan tujuan untuk memberikan gambaran pada kesimpulan perhitungan algoritma *data mining*.

### 3. HASIL DAN PEMBAHASAN

Dataset yang sudah diberikan label "Non\_HS" dan "HS" akan dilakukan tahapan *Pre-Processing* menggunakan aplikasi *python*. Tahap ini menghasilkan bentuk data yang dapat diolah oleh algoritma *machine learning*, berikut ini pada tabel 2 merupakan tahapan *Pre-Processing* dan hasil yang didapatkan :

Tabel 2. Hasil Proses Pre-Processing Data Menggunakan *Python*

Proses	Data
Data Asli	@alayyubichoir: Sosok Anies Sandi harus diakui sgt mengayomi. Sangat kompeten utk membangun kerukukan & kebhinekaan di Jakarta,
Remove Punctuations	alayyubichoir Sosok Anies Sandi harus diakui sgt mengayomi Sangat kompeten utk membangun kerukukan kebhinekaan di Jakarta
Lowercase	alayyubichoir sosok anies sandi harus diakui sgt mengayomi sangat kompeten utk membangun kerukukan kebhinekaan di jakarta
Tokenizing	[alayyubichoir] [sosok] [anies] [sandi] [harus] [diakui] [sgt] [mengayomi] [sangat] [kompeten] [utk] [membangun] [kerukukan] [kebhinekaan] [di] [Jakarta]
Filtering	[alayyubichoir] [sosok] [anies] [sandi] [harus] [diakui] [mengayomi] [sangat] [kompeten] [membangun] [kebhinekaan] [Jakarta]
Stemming	[alayyubichoir] [sosok] [anies] [sandi] [harus] [akui] [ayom] [sangat] [kompeten] [bangun] [bhineka] [Jakarta]

Berikut ini pada tabel 3 hasil perhitungan menggunakan algoritma *machine learning* dengan model *splitting data* dengan pembagian 80% sebagai data latih dan 20% sebagai data uji. Selanjutnya dilakukan pengujian ulang menggunakan model *K-Fold Cross Validation* dengan nilai  $K=10$ . Hasil dari evaluasi model tersebut dilakukan perbandingan menggunakan metode *SMOTE* dan tidak menggunakan metode *SMOTE* :

Tabel 3. Perbandingan Nilai *Accuracy*, *Recall*, *Precision* dan *F1* dengan Metode *Splitting Data* Perbandingan 80:20

Algoritma	Splitting Data							
	With SMOTE				No SMOTE			
	Accuracy	Recall	Precision	F1	Accuracy	Recall	Precision	F1
Logistic Regression	86%	84,40%	83,70%	85,20%	81,10%	77,40%	85,70%	78,70%
Decision Tree (DT)	77,60%	76,90%	81,80%	76,70%	82,50%	81,60%	76,60%	81,70%
Multinomial Naïve Bayes (NB)	87,40%	76%	86,90%	87,10%	79,70%	88,30%	81,90%	77,10%
Random Forest (RF)	83,90%	81,30%	85,20%	82,30%	82,50%	79,50%	84,10%	80,60%
Adaptive Boosting (Adaboost)	84,60%	82,70%	84%	83,80%	83,90%	83,60%	83,40%	83%

Hasil perbandingan nilai akurasi yang didapat pada tabel 3 dengan menggunakan evaluasi model *Splitting Data* menghasilkan algoritma *Multinomial Naïve Bayes (NB)* menggunakan *SMOTE* mendapatkan nilai akurasi tertinggi yaitu sebesar 87,4% dibandingkan tanpa menggunakan *SMOTE*, begitupun juga nilai *Precision* sebesar 86,9% dan nilai *F1* sebesar 87,1%, akan tetapi nilai *recall* menggunakan *SMOTE* lebih rendah dibandingkan dengan tidak menggunakan *SMOTE* yaitu sebesar 76%, hal ini membuktikan bahwa algoritma *Multinomial Naïve Bayes (NB)* dengan menggunakan *SMOTE* memiliki *performance* yang cukup baik untuk melakukan pengkategorian kalimat *hate speech*.

Tabel 4. Perbandingan Nilai Accuracy, Recall, Precision dan F1 dengan Metode K-Fold Cross Validation Nilai K = 10

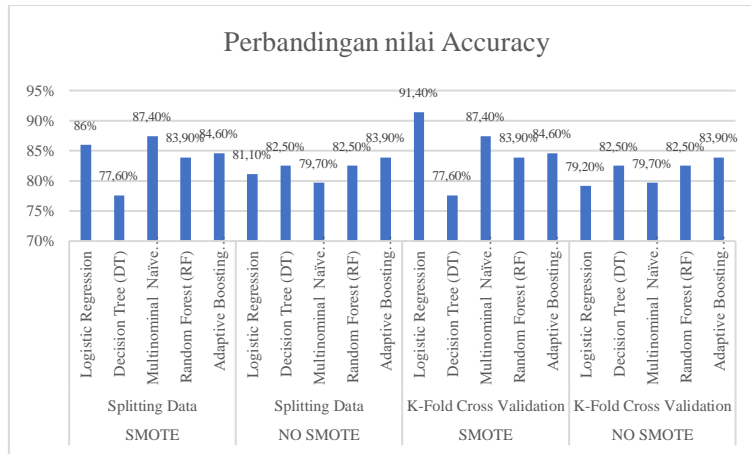
Algoritma	K-Fold Cross Validation							
	With SMOTE				No SMOTE			
	Accuracy	Recall	Precision	F1	Accuracy	Recall	Precision	F1
Logistic Regression	91,40%	84,40%	88,80%	97,10%	79,20%	77,40%	45,30%	92,6%
Decision Tree (DT)	77,60%	76,90%	85%	86,70%	82,50%	81,60%	68,50%	77,2%
Multinomial Naïve Bayes (NB)	87,40%	88,30%	95,60%	98%	79,70%	76%	56,10%	92,80%
Random Forest (RF)	83,90%	81,30%	84,10%	97,20%	82,50%	79,50%	62,50%	91,70%
Adaptive Boosting (Adaboost)	84,60%	83,60%	85,80%	93,30%	83,90%	82,70%	68,50%	85,30%

Hasil perbandingan nilai akurasi yang didapat pada tabel 4 dengan menggunakan evaluasi model *K-Fold Cross Validation* menghasilkan algoritma *Logistic Regression (LR)* menggunakan *SMOTE* mendapatkan nilai akurasi tertinggi yaitu sebesar 91,4%, akan tetapi nilai *Recall* lebih rendah dibandingkan dengan algoritma *Multinomial Naïve Bayes (NB)* dengan menggunakan *SMOTE* yaitu sebesar 88,3%, *Precision* 95,6% dan *F1* 98%.

Berikut ini pada tabel 5 merupakan pemodelan algoritma tradisional *machine learning* dan *ensemble machine learning* didalam aplikasi *python*. Pada algoritma tradisional *machine learning* penelitian ini mengisi parameter pada algoritma tersebut dengan *null* atau kosong parameter sedangkan pada algoritma *ensemble* kami mengisi parameter nilai pada masing-masing algoritma dengan tujuan untuk meningkatkan *performance* pada saat melakukan prediksi.

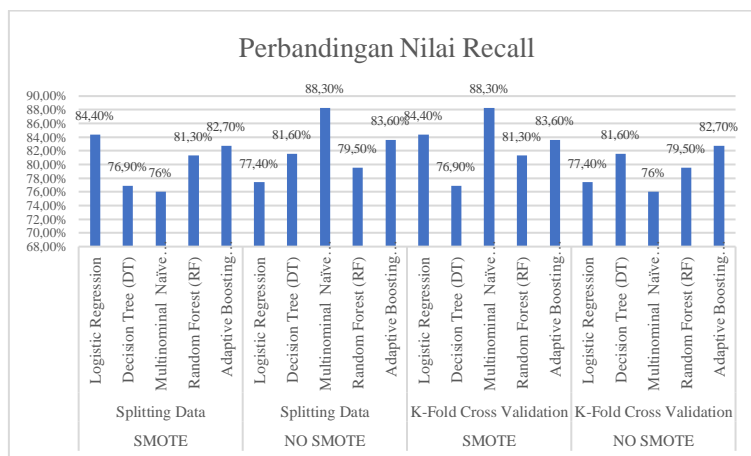
Tabel 5. Model Algoritma *Machine Learning* Tradisional dan *Ensemble*

LGREGG	LogisticRegression()
RF	RandomForestClassifier()
DT	DecisionTreeClassifier()
NB	MultinomialNB()
adaboost	AdaBoostClassifier(n_estimators=100)



Gambar 2. Perbandingan Nilai *Accuracy* antara Model *Splitting Data* dengan *K-Fold Cross Validation* serta dengan Menggunakan dan Tidak Menggunakan Metode *SMOTE*

Berdasarkan hasil perbandingan nilai akurasi pada gambar 2 menghasilkan dua algoritma terbaik pada evaluasi model menggunakan *K-Fold Cross Validation dengan SMOTE* yaitu *Logistic Regression (LR)* menghasilkan nilai akurasi sebesar 91,4% ,sedangkan algoritma *Multinomial Naïve Bayes (NB)* menggunakan dan tanpa menggunakan *SMOTE* nilai akurasi yang dihasilkan tetap sama yaitu sebesar 87,4%.

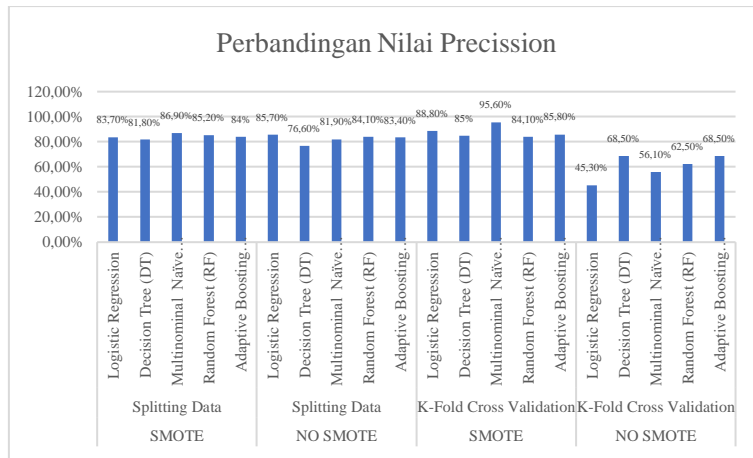


Gambar 3. Perbandingan Nilai *Recall* antara Model *Splitting Data* dengan *K-Fold Cross Validation* serta dengan Menggunakan dan Tidak Menggunakan Metode *SMOTE*

Berdasarkan hasil perbandingan nilai *Recall* pada gambar 3 menghasilkan dua algoritma terbaik pada evaluasi model menggunakan *Splitting data* tanpa *SMOTE* yaitu *Multinomial Naïve Bayes (NB)* sebesar 88,3%, jika menggunakan model *K-Fold Cross* menggunakan *SMOTE* nilai *Recall*

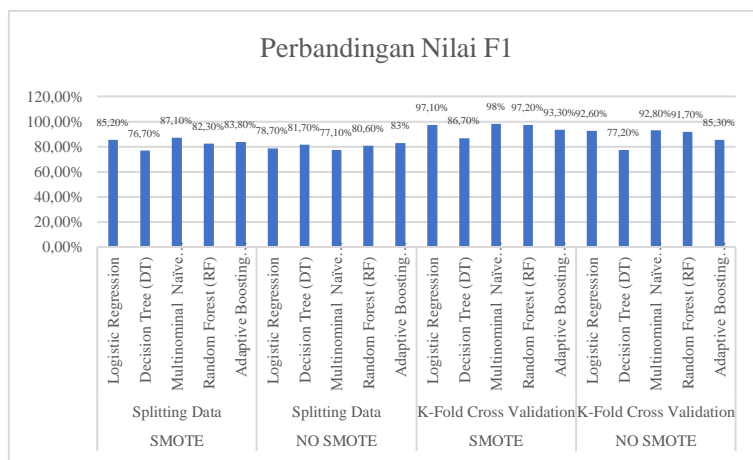


tetap sama, sedangkan pada algoritma *Logistic Regression (LR)* dengan menggunakan *Splitting data*, *K-Fold Cross Validation* dan *SMOTE* memiliki nilai yang sama yaitu sebesar 84,4%.



Gambar 4. Perbandingan Nilai *Precision* antara Model *Splitting Data* dengan *K-Fold Cross Validation* serta dengan Menggunakan dan Tidak Menggunakan Metode *SMOTE*

Berdasarkan hasil perbandingan nilai *Precision* pada gambar 4 menghasilkan algoritma *Multinomial Naïve Bayes (NB)* dengan menggunakan *K-Fold Cross Validation* dan *SMOTE* memiliki nilai yang sama yaitu sebesar 95,6%, mengungguli algoritma *Logistic Regression (LR)* yang hanya mendapatkan nilai *Precision* sebesar 88,8%.



Gambar 5. Perbandingan Nilai *F1* antara Model *Splitting Data* dengan *K-Fold Cross Validation* serta dengan Menggunakan dan Tidak Menggunakan Metode *SMOTE*

Berdasarkan hasil perbandingan nilai *F1* pada gambar 5 menghasilkan algoritma *Multinomial Naïve Bayes (NB)* dengan menggunakan *K-Fold Cross Validation* dan *SMOTE* memiliki nilai *F1* sebesar 98%, mengungguli algoritma *Logistic Regression (LR)* yang hanya mendapatkan nilai *F1* sebesar 97,1%.

#### 4. KESIMPULAN

Hasil penelitian yang didapat bahwa algoritma *Logistic Regression* dengan menggunakan metode *SMOTE (Synthetic Minority Over-sampling Technique)* mendapatkan nilai akurasi terbaik yaitu sebesar 91,40%, nilai *Recall* sebesar 84,49%, *Precision* sebesar 88,80% dan *F1*

sebesar 97,10%. Sedangkan algoritma *Multinomial Naïve Bayes (NB)* menggunakan evaluasi model *Splitting data* dan *K-Fold Cross Validation* serta menggunakan *SMOTE* menghasilkan akurasi yang sama yaitu sebesar 87,4%, jika menggunakan *K-Fold Cross Validation* nilai Recall meningkat menjadi 88,3%, Precision menjadi 95,6% dan F1 menjadi 98%. Jika ingin menggunakan nilai akurasi untuk melakukan pengkategorian kalimat *hate speech* dapat menggunakan algoritma *Logistic Regression* dengan menggunakan *SMOTE* dan evaluasi model *K-Fold Cross Validation*, dikarenakan algoritma tersebut menghasilkan nilai akurasi yang lebih baik dibandingkan dengan algoritma lainnya.

## 5. SARAN

Penelitian selanjutnya disarankan untuk berfokus kepada algoritma tradisional *machine learning* dan *ensemble machine learning* yang lain dan juga menerapkan metode *Hyperparameter Tuning* untuk melihat parameter terbaik yang akan digunakan oleh algoritma itu sendiri dengan harapan dapat meningkatkan efisiensi dan juga *performance* dari algoritma tersebut.

## DAFTAR PUSTAKA

- [1] D. A. N. Taradhita and I. K. G. D. Putra, "Hate speech classification in Indonesian language tweets by using convolutional neural network," *J. ICT Res. Appl.*, vol. 14, no. 3, pp. 225–239, 2021, doi: 10.5614/itbj.ict.res.appl.2021.14.3.2.
- [2] J. Patihullah and E. Winarko, "Hate Speech Detection for Indonesia Tweets Using Word Embedding And Gated Recurrent Unit," *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 13, no. 1, p. 43, 2019, doi: 10.22146/ijccs.40125.
- [3] A. Bayhaqy, S. Sfenrianto, K. Nainggolan, and E. R. Kaburuan, "Sentiment Analysis about E-Commerce from Tweets Using Decision Tree, K-Nearest Neighbor, and Naïve Bayes," *2018 Int. Conf. Orange Technol. ICOT 2018*, pp. 1–6, 2018, doi: 10.1109/ICOT.2018.8705796.
- [4] F. Wenando and E. Fuad, "Detection of Hate Speech in Indonesian Language on Twitter Using Machine Learning Algorithm," *Int. Conf. Recent Adv. Nat. Lang. Process. RANLP*, vol. 4, pp. 467–472, 2019.
- [5] S. Ahammed, M. Rahman, M. H. Niloy, and S. M. M. H. Chowdhury, "Implementation of Machine Learning to Detect Hate Speech in Bangla Language," *Proc. 2019 8th Int. Conf. Syst. Model. Adv. Res. Trends, SMART 2019*, pp. 317–320, 2020, doi: 10.1109/SMART46866.2019.9117214.
- [6] K. Sreelakshmi, B. Premjith, and K. P. Soman, "Detection of Hate Speech Text in Hindi-English Code-mixed Data," *Procedia Comput. Sci.*, vol. 171, no. 2019, pp. 737–744, 2020, doi: 10.1016/j.procs.2020.04.080.
- [7] A. M. U. D. Khanday, S. T. Rabani, Q. R. Khan, and S. H. Malik, "Detecting twitter hate speech in COVID-19 era using machine learning and ensemble learning techniques," *Int. J. Inf. Manag. Data Insights*, vol. 2, no. 2, p. 100120, 2022, doi: 10.1016/j.jjime.2022.100120.
- [8] N. Badri, F. Kboubi, and A. H. Chaibi, "Combining FastText and Glove Word Embedding for Offensive and Hate speech Text Detection," *Procedia Comput. Sci.*, vol. 207, no. Kes, pp. 769–778, 2022, doi: 10.1016/j.procs.2022.09.132.
- [9] Febiana Anistya and Erwin Budi Setiawan, "Hate Speech Detection on Twitter in Indonesia with Feature Expansion Using GloVe," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 6, pp. 1044–1051, 2021, doi: 10.29207/resti.v5i6.3521.
- [10] S. Abro, S. Shaikh, Z. Ali, S. Khan, G. Mujtaba, and Z. H. Khand, "Automatic hate speech detection using machine learning: A comparative study," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 8, pp. 484–491, 2020, doi: 10.14569/IJACSA.2020.0110861.
- [11] Bimrew Sendekie Belay, "Detecting Hate Speech In Twitter Using Long Short-Term Memory And Naïve Bayes Method," *γ787*, vol. 7, no. 8.5.2017, pp. 2003–2005, 2022.
- [12] S. Thaiparnit, N. Chumuang, and M. Ketcham, "A Comparative Study of Clasification Liver

- Dysfunction with Machine Learning,” *2018 Int. Jt. Symp. Artif. Intell. Nat. Lang. Process. iSAI-NLP 2018 - Proc.*, vol. 283, pp. 1–4, 2018, doi: 10.1109/iSAI-NLP.2018.8692808.
- [13] K. Ahammed, M. S. Satu, M. I. Khan, and M. Whaiduzzaman, “Predicting Infectious State of Hepatitis C Virus Affected Patient’s Applying Machine Learning Methods,” *2020 IEEE Reg. 10 Symp. TENSYP 2020*, no. June, pp. 1371–1374, 2020, doi: 10.1109/TENSYP50017.2020.9230464.
- [14] Y. E. Ika Alfina, Rio Mulia, Mohamad Ivan Fanany, “Hate Speech Detection in the Indonesian Language: A Dataset and Preliminary Study,” pp. 473–481, 1999, [Online]. Available: <https://support.twitter.com/articles/1>.
- [15] C. Schröer, F. Kruse, and J. M. Gómez, “A systematic literature review on applying CRISP-DM process model,” *Procedia Comput. Sci.*, vol. 181, no. 2019, pp. 526–534, 2021, doi: 10.1016/j.procs.2021.01.199.
- [16] Merinda Lestandy, Abdurrahim Abdurrahim, and Lailis Syafa’ah, “Analisis Sentimen Tweet Vaksin COVID-19 Menggunakan Recurrent Neural Network dan Naïve Bayes,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 4, pp. 802–808, 2021, doi: 10.29207/resti.v5i4.3308.
- [17] S. Visalakshi and V. Radha, “A literature review of feature selection techniques and applications: Review of feature selection in data mining,” *2014 IEEE Int. Conf. Comput. Intell. Comput. Res. IEEE ICCIC 2014*, no. 1997, 2015, doi: 10.1109/ICCIC.2014.7238499.
- [18] S. A. J. Zaidi, S. Tariq, and S. B. Belhaouari, “Future prediction of covid-19 vaccine trends using a voting classifier,” *Data*, vol. 6, no. 11, 2021, doi: 10.3390/data6110112.
- [19] L. M. Candanedo, V. Feldheim, and D. Deramaix, “Data driven prediction models of energy use of appliances in a low-energy house,” *Energy Build.*, vol. 140, pp. 81–97, 2017, doi: 10.1016/j.enbuild.2017.01.083.
- [20] Y. Wang, F. Wang, and H. Wang, “Influencing factors regression analysis of heating energy consumption of rural buildings in China,” *Procedia Eng.*, vol. 205, pp. 3585–3592, 2017, doi: 10.1016/j.proeng.2017.10.207.
- [21] I. C. Sari and Y. Ruldeviyani, “Sentiment Analysis of the Covid-19 Virus Infection in Indonesian Public Transportation on Twitter Data: A Case Study of Commuter Line Passengers,” *2020 Int. Work. Big Data Inf. Secur. IWBISS 2020*, pp. 23–28, 2020, doi: 10.1109/IWBISS50925.2020.9255531.
- [22] J. Singh, S. Bagga, and R. Kaur, “Software-based Prediction of Liver Disease with Feature Selection and Classification Techniques,” *Procedia Comput. Sci.*, vol. 167, no. 2019, pp. 1970–1980, 2020, doi: 10.1016/j.procs.2020.03.226.
- [23] G. S. Uttreshwar and A. A. Ghatol, “Hepatitis B diagnosis using logical inference and generalized regression neural networks,” *2009 IEEE Int. Adv. Comput. Conf. IACC 2009*, no. March, pp. 1587–1595, 2009, doi: 10.1109/IADCC.2009.4809255.
- [24] S. Sun and R. Huang, “An adaptive k-nearest neighbor algorithm,” *Proc. - 2010 7th Int. Conf. Fuzzy Syst. Knowl. Discov. FSKD 2010*, vol. 1, no. Fskd, pp. 91–94, 2010, doi: 10.1109/FSKD.2010.5569740.
- [25] F. Belaid, A. Ben Youssef, and N. Omrani, “Investigating the factors shaping residential energy consumption patterns in France: Evidence form quantile regression,” *Eur. J. Comp. Econ.*, vol. 17, no. 1, pp. 127–151, 2020, doi: 10.25428/1824-2979/202001-127-151.
- [26] Okfalisa, I. Gazalba, Mustakim, and N. G. I. Reza, “Comparative analysis of k-nearest neighbor and modified k-nearest neighbor algorithm for data classification,” *Proc. - 2017 2nd Int. Conf. Inf. Technol. Inf. Syst. Electr. Eng. ICITISEE 2017*, vol. 2018-Janua, pp. 294–298, 2018, doi: 10.1109/ICITISEE.2017.8285514.