Indonesian Journal of Computer Science
ISSN 2302-4364 (*print*) dan 2549-7286 (*online*)
Jln. Khatib Sulaiman Dalam, No. 1, Padang, Indonesia, Telp. (0751) 7056199, 7058325
Website: ijcs.stmikindonesia.ac.id | E-mail: ijcs@stmikindonesia.ac.id

Deteksi *Hate Speech* pada Kolom Komentar Tiktok dengan menggunakan SVM

Amelia Ariska[1], Mia Kamayani[2]
ameliaariska12@gmail.com, mia.kamayani@uhamka.ac.id
Universitas Muhammadiyah Prof. Dr.Hamka

Informasi Artikel

Diterima : 8 Mei 2024
Direview : 15 Mei 2024
Disetujui : 30 Jun 2024

Kata Kunci

Hate speech, *lexicon, machine learning, support vector machine,* TikTok.

Abstrak

Aplikasi TikTok menyediakan banyak fitur termasuk komentar untuk berinteraksi antar pengguna. Pengguna dapat bertukar pendapatnya melalui kolom komentar secara terbuka. Namun, semakin banyak pengguna yang melakukan interaksi atau bertukar pendapat melalui aplikasi TikTok, secara sadar atau tidak penggunaan *hate speech* masih banyak digunakan. *Hate speech* merupakan suatu perbuatan yang dilakukan seseorang atau kelompok dapat memicu tindak kejahatan sehingga merugikan orang lain. Penelitian ini bertujuan untuk mengidentifikasi penggunaan *hate speech* pada kolom komentar TikTok menggunakan algoritma SVM serta melakukan perbandingan 2 *library* yang akan digunakan dalam proses pelabelan untuk melihat performa yang dilakukan oleh model algoritma SVM. Proses pelabelan dengan menggunakan pendekatan *lexicon-based*. Kamus yang digunakan pada penelitian ini adalah *Inset lexicon* dan *Vader Sentiment.* Algoritma SVM digunakan sebagai model untuk menguji hasil evaluasi. Hasil yang didapatkan dengan menggunakan pelabelan *Inset lexicon* akurasi sebesar 82% Sedangkan pelabelan kedua yaitu menggunakan *Vader Sentiment* mendapatkan hasil akurasi sebesar 96,21%.

Keywords

Hate speech, *lexicon, machine learning, support vector machine, TikTok.*

Abstract

The TikTok application provides numerous features, including the comment section for users to interact with each other. Users can exchange their opinions openly through the comment section. However, as the interaction or exchange of opinions among users increases, the use of hate speech, consciously or unconsciously, remains prevalent. Hate speech refers to actions by an individual or group that can incite criminal acts, thereby harming others. This study aims to identify the use of hate speech in TikTok comment sections using the SVM algorithm and to compare two libraries used in the labeling process to observe the performance of the SVM algorithm model. The labeling process employs a lexicon-based approach. The dictionaries used in this study are the Inset lexicon and Vader Sentiment. The SVM algorithm is used as the model to test the evaluation results. The results obtained using the Inset lexicon labeling show an accuracy of 82%, while the second labeling method using Vader Sentiment yields an accuracy of 96.21%.

A. Pendahuluan

Aplikasi TikTok menyediakan komentar dengan tujuan agar sesama pengguna dapat berinteraksi. Penulisan pada kolom komentar umumnya berisi tentang menyampaikan pendapat atas apa yang ada pada konten. Tidak sedikit dari pengguna menyampaikan pendapatnya dengan menggunakan perasaan [1]. Menyampaikan pendapat menggunakan perasaan pada kolom komentar memiliki dampak pada kalimat yang ditulis pada kolom komentar dapat berupa komentar baik atau buruk [2]. Emosi yang baik dapat menghasilkan kata atau komentar yang baik, namun jika emosi yang dikeluarkan tidak dapat dikendalikan dengan baik akan menghasilkan komentar atau kata yang buruk [3]. Penulisan pada komentar jika menggunakan perasaan yang baik menghasilkan komentar positif, namun jika tidak memiliki perasaan yang baik maka kalimat yang digunakan berupa kata kasar, kotor, cacian, bahkan bisa menyinggung antar pengguna dan memungkinkan pengguna lain dapat terprovokasi untuk membenci dengan menggunakan kata kebencian.

Menurut [4] *Hate speech* atau ujaran kebencian dapat disimpulkan bahwa, suatu tindakan berupa sebuah pesan melalui platform media sosial yang mengandung serangan secara langsung. *Hate speech* merupakan perbuatan yang dilakukan seseorang atau sekelompok orang dengan menuliskan pesan menggunakan kata kasar, kotor, dan cacian sehingga seseorang yang dituju merasakan intimidasi dari pengguna lain.

Pemerintah telah memberikan solusi terkait penggunaan *hate speech* dengan memberikan hukuman berupa tindak pidana UU No. 11 Tahun 2008 tentang ITE, UU No. 19 Tahun 2016 Tentang perubahan Atas UU No. 11 Tahun 2008 tentang Informasi dan transaksi elektronik, UU No. 40 Tahun 2008 tentang Penghapusan diskriminasi ras dan etnis, dan Surat Edaran Kapolri NOMOR SE/06/X/2015 [5].

Penelitian ini memiliki tujuan adalah untuk mengidentifikasi kata *hate speech* pada kolom komentar TikTok melakukan perbandingan terhadap pelabelan dengan menggunakan pendekatan *lexicon-based*. Perbandingan yang dilakukan adalah dengan menggunakan 2 *library* pada teknik *lexicon-based*. Model dari penelitian ini adalah dengan menggunakan algoritma SVM.

Penelitian sebelumnya telah dilakukan klasifikasi *hate speech* dengan menggunakan metode LSTM dataset yang diperoleh berasal dari *kaggle* adala *abusive* dan *hate speech* dengan jumkag 13.170 data dan memiliki 12 label yang telah disertakan pada dataset. Hasil dari penelitian tersebut akurasi data latih sebesar 86, 23% dan akurasi data validasi sebesar 87, 10% dengan epoch sebanyak 10 [6] .

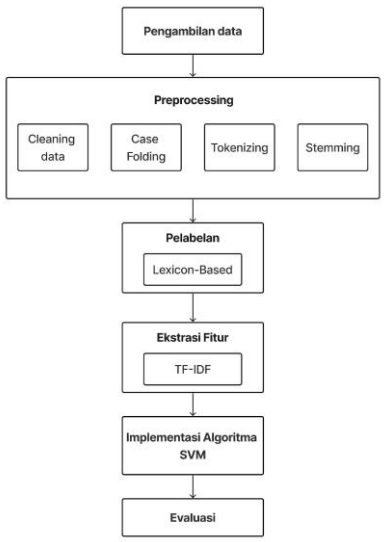
Mengidentifikasi *Hate speech* pada bahasa Indonesia dengan *lexicon based* dan *synonym-based* mengklasifikasi dengan menggunakan SVM data yang digunakan berasal dari twitter. Hasil yang didapatkan akurasi terbesar diperoleh dengan menggunakan metode SVM dan *synonym*. Namun jika digabungkan semua metode SVM, *lexicon*, dan Synonym mendapatkan akurasi sebesar 59,44%, presisi 71, 95%, *recall* 30.95%, dan F-*measure* 43,28% [7]. Penelitian [8] melakukan perbandingan hasil evaluasi penggunaan algoritma SVM dan *Naïve Bayes*. SVM merupakan algoritma terbaik dalam melakukan analisis teks, data yang didapatkan berasal dari *twitter* data yang diperoleh 331 berlabel positif dan 369 berlabel negatif. Penelitian sebelumnya juga dilakukan perbandingan dengan pelabelan

lexicon dengan menggunakan kamus kata *lexicon sentistrenght_id* dan *inset lexicon* dikombinasikan dengan fitur TF-IDF kemudian, menghasilkan kamus kata *lexicon sentistrenght_id* mendapatkan akurasi sedikit lebih tinggi dibandingkan dengan kamus *inset lexicon*. Sebanyak 64, 46% hasil akurasi dari kamus kata *lexicon sentistrenght_id,* dan kamus kata *inset lexicon* mendapatkan hasil akurasi sebesar 62, 65% [9].

Telah banyak dilakukan penelitian untuk mendeteksi *hate speech* menggunakan algoritma SVM, namun belum ada penelitian yang mendeteksi *hate speech* pada kolom komentar TikTok serta melakukan komparasi kamus kata pada proses pelabelan dengan menggunakan teknik *lexicon-based* untuk mendeteksi *hate speech* kemudian mengevaluasi dengan menggunakan algoritma SVM.

B. Metode Penelitian

Berikut merupakan alur dari penelitian yang digunakan untuk tahapan yang dilakukan proses penelitian ini. Tahapan yang akan dilakukan untuk penelitian ini terdiri dari pengambilan data, *preprocessing*, pelabelan, ekstraksi fitur, Implementasi algoritma SVM dan evaluasi.



Gambar 1. Alur Penelitian

1. Pengambilan data

Pengambilan data atau *scraping* data komentar pada aplikasi TikTok dilakukan dengan menggunakan *javascript* mendapatkan data sebanyak 1096 komentar. Proses untuk mendapatkan dataset, dengan cara aplikasi dibuka melalui web kemudian pengambilan dilakukan dengan menggunakan teknik *web scraping*. Data yang diambil berbentuk csv dan berbahasa Indonesia. Penelitian ini menggunakan komentar dari sebuah video yang sedang membahas tentang perdebatan konflik palestina dan israel kemudian dijadikan dataset.

2. *Preprocessing*

Preprocessing adalah tahapan yang digunakan untuk menentukan dokumen mana yang akan diambil memenuhi kebutuhan informasi untuk menentukan

pengguna. Tahapan yang umum dilakukan dalam proses *preprocessing* terdapat 4 langkah yaitu *case folding*, *tokenizing*, *filtering*, dan *stemming* [10]. Tahapan ini dilakukan dengan tujuan agar mempermudah mesin dalam melakukan proses membaca dan mengelola dataset. Proses yang dilakukan pada tahapan ini adalah dataset yang sebelumnya masih memiliki banyak karakter dan tanda baca yang tidak diperlukan. Tahapan yang dilakukan pada proses *preprocessing* adalah sebagai berikut:

Tabel 1. Cleaning data

Komentar	*Cleaning*
@fakta mba monic apaan banget	mba monic apaan banget
Salam sehat #ustadabdulsomat	Salam sehat
Monic lawak mulu 😆🤣	Monic lawak mulu

a. *Cleaning*, proses ini dilakukan dengan tujuan untuk pembersihan dataset dari sebuah karakter yang tidak diperlukan dalam mengklasifikasi teks. Pembersihan dilakukan dengan menghapus sebuah tanda baca, *emoticon*, karakter asing dan lain sebagainya. Hal ini bertujuan untuk mempermudah data dalam proses selanjutnya.

Tabel 2. *Case folding*

Komentar	*Case folding*
Monica Ibunya Siapa Woy	mba monic apaan banget
HoStnyA GaaDiL NitcH	hostnya gaadil nitch
PALESTINE MERDEKA	palestine merdeka

b. Selanjutnya tahapan kedua pada *preprocessing* adalah *case folding*, tahapan ini dilakukan perubahan pada kalimat atau kata dari dataset. Kata atau kalimat yang sebelumnya menggunakan *uppercase* akan diubah menjadi *lowercase*. Perubahan huruf kapital kemudian uppercase akan diubah keseluruhan dengan menggunakan huruf kecil.

Tabel 3. *Tokenizing*

Komentar	*Tokenizing*
Ibu monica sesuai realita sejarah	[‘ibu’, ’monica’, ‘sesuai’, ‘ realita’, ‘sejarah’]

c. Penelitian ini menggunakan kamus kata *Normalization KBBI*, adapun tujuan menggunakan *normalization* adalah penggunaan pada kata singkat dapat diubah kembali kedalam bentuk kata dasar hal ini sering digunakan dalam penulisan komentar.

Tabel 4. *Normalization*

Komentar	*Normalization*

ini orng rumahnya dmana si? meresakan sekali	ini orang rumahnya dimana si? meresahkan sekali

d. Tahapan ketiga pada proses *preprocessing* adalah *tokenizing*. Proses *tokenizing* dilakukan dengan memisahkan kalimat menjadi sebuah kata-kata. Tujuan dilakukannya *tokenizing* agar mempermudah dalam melakukan proses selanjutnya.

Tabel 5. *Stemming*

Komentar	***Stemming***
Pembelajaran untuk kalian, mengundang orang yang paham sejarah	[‘Pembelajaran’, ’untuk’, ‘kali’, ‘undang’, ‘orang’, ‘yang’, ‘paham’, ‘sejarah’]

e. Proses terakhir dari *preprocessing* adalah stemming, proses ini dilakukan dengan tujuan untuk melakukan perubahan kata menjadi bentuk dasar, kata sebelumnya memiliki imbuhan me-, ber-, kan-, dan sebagainya akan diubah menjadi kata asli. Perubahan kata dapat dilihat pada tabel 4.

3. Pelabelan

Penelitian ini melakukan pelabelan dengan menggunakan pendekatan *lexicon based*. *Lexicon based* merupakan sebuah teknik yang digunakan dengan cara menganalisis secara sentimen dengan menggunakan sebuah corpus yang sudah memiliki niat bobot atau *polarity score* dapat digunakan sebagai sumber data [7]. Pelabelan pada penelitian ini menggunakan 2 *library* untuk mengidentifikasi *hate speech* yang dilakukan dengan penilaian secara sentimen untuk menentukan 2 kelas pelabelan pada penelitian ini.

a. *Inset lexicon*

Fajri Koto dan Gemal Y. Rahmaningtyas telah melakukan penelitian yang berjudul “*Inset lexicon: Evaluation of a worls list for indonesian sentiment analysis in microblogs*” menghasilkan *Inset lexicon* data yang dikumpulkan berasal dari twitter [11]. Inset lexicon terdapat kata positif sebanyak 3.609 data dan kata negatif sebanyak 6.609 data, keduanya telah memiliki nilai polaritas atau *polarity score* -5 sampai dengan 5. Nilai minus akan dimasukan kedalam data negatif sedangkan nilai plus akan dimasukan ke data positif [12].

b. *Vader sentiment*

Kamus kata *Vader Sentiment* adalah kamus yang menggunakan bahasa inggris dengan menghitung *score* secara otomatis secara sentimen. Cara kerja dari kamus *Vader Sentiment* dengan memeriksa setiap kata pada kamus yang telah ada sebelumnya. *Vader Sentiment* memiliki 3 nilai sentimen yaitu positif, negatif dan netral [13]. Penilaian yang dilakukan jika polaritas bernilai lebih dari 0 dimasukan kedalam kelas positif, jika nilai sama dengan 0 maka akan dimasukan kedalam kelas netral dan jika nilai

kurang dari 0 maka akan dimasukan kedalam kamus negatif. Namun yang akan digunakan pada penelitian ini hanya 2 yaitu positif dan negatif seperti kamus kata *Inset lexicon.*

4. Ekstraksi Fitur

Untuk mengekstrak fitur penelitian ini menggunakan TF-IDF. Fitur ini paling umum digunakan untuk menghitung bobot [14]. Fitur ini menggunakan metode untuk mengukur frekuensi kemunculan pada sebuah kata dalam sebuah dokumen [15]. TF adalah mengukur seberapa sering kata yang muncul pada suatu dokumen, IDF digunakan untuk mengukur seberapa penting kata yang muncul pada suatu dokumen. TF-IDF adalah hasil perkalian dari skor TF dan skor IDF. Perhitungan untuk TF-IDF dapat diformulasikan sebagai berikut:

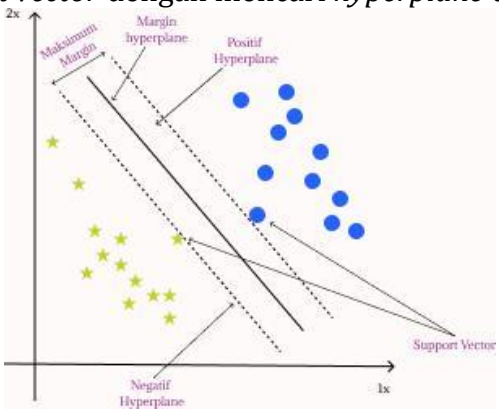
$$TF\ (i,x) = \frac{Jumlah\ kali\ kata(i)\ \ muncul\ dalam\ dokumen\ (x)}{Total\ \ jumlah\ kata\ dalam\ dokumen\ (x)} \quad (1)$$

$$IDF\ (i,y) = \frac{Total\ jumlah\ dokumen\ koleksi\ dokumen\ (y)}{Jumlah\ dokumen\ yang\ mengandung\ kata\ i + 1} \quad (2)$$

$$TF - IDF\ (i,x,y) = \frac{TF(i,x)}{IDF\ (i,y)} \quad (3)$$

5. Implementasi Algoritma SVM (*Support Vector Machine*)

Teknik klasifikasi yang beroperasi mencari *hyperplane* terbaik secara optimal adalah SVM (*Support Vector Machine*) [16]. Kinerja dari algoritma SVM adalah dengan menemukan hyperline terbaik dan mengoptimalkan jarak antar kelas, hyperline adalah sebuah model matematis yang berfungsi untuk membedakan antara 2 kelompok data berdasarkan kelas [17]. Berikut adalah ilustrasi sebuah *support vector* dengan mencari *hyperplane* terbaik.



Gambar 2. *Support Vector*

Ilustrasi pada gambar 1 merupakan sebuah *support vector*, dimana data dipisahkan pada suatu garis pembatas oleh maksimum maksimum margin. Pada ilustrasi diatas terdapat 2 kelas yang diberikan nila 1 dan -1 sehingga dapat diformulasikan sebagai berikut [8]:

$$P.xi + b \geq 1, yi = 1 \quad (4)$$

$$P.xi + b \leq 1, yi = -1 \quad (5)$$

Dari persamaan 1 dan 2, p adalah bobot *vector machine*, xi merupakan data ke-i, sedangkan yi adalah kelas ke-i dan b merupakan nilai bias.

6. Evaluasi

Tahapan akhir dalam penelitian ini adalah evaluasi. Evaluasi dilakukan dengan menghitung performa tertinggi diukur dari akurasi, presisi, *recall*, dan f1-*score*. Untuk menghitung performa tersebut maka dapat digunakan persamaan sebagai berikut .

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FN)} \quad (6)$$

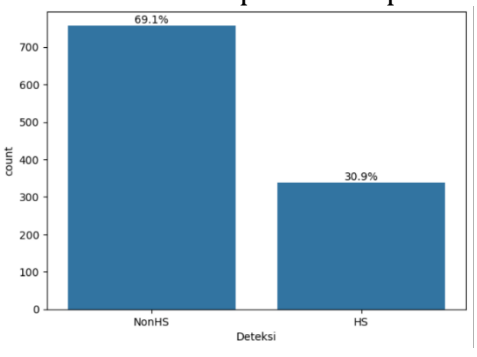
$$precision = \frac{TP}{(TP+FP)} \quad (7)$$

$$recall = \frac{TP}{(TP+FN)} \quad (8)$$

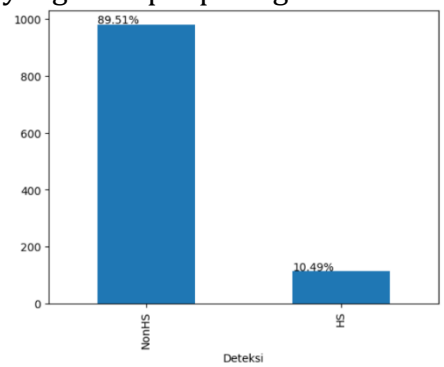
$$f1-score = \frac{(recall\ x\ presisi)}{(presisi+recall)} \quad (9)$$

C. Hasil dan Pembahasan

Pengambilan dataset menghasilkan data sebanyak 1906 data dari kolom komentar aplikasi TikTok dengan kasus perdebatan konflik palestina dan israel. Kemudian data diolah dengan tahapan *Preprocessing* hingga bersih dan dapat dilanjutkan ke proses pelabelan. Pada penelitian ini pelabelan menggunakan 2 kelas yaitu NonHS dan HS dengan pendekatan *lexicon based*. Pelabelan pertama dilakukan dengan menggunakan kamus *Inset lexicon*, sedangkan pelabelan kedua menggunakan *library Vader Sentiment*. Hasil dari implementasi kamus *Inset lexicon* didapatkan label NonHS sebanyak 757 data dan label HS sebanyak 339 data. Sedangkan, menggunakan *library Vader Sentiment* didapatkan label NonHS sebanyak 981 data dan label HS sebanyak 115 data. Perbandingan label antara NonHS dan HS dapat dilihat pada diagram yang terdapat pada gambar 3 dan 4.



Gambar 3. Pelabelan *Inset Lexicon*



Gambar 4. Pelabelan *SentimentIntensityAnalyzer*

Setelah dataset sudah diberikan label selanjutnya dataset diberikan bobot pada setiap kata. Hasil dari pelabelan diatas menunjukkan pelabelan NonHS memiliki nilai mayoritas dibandingkan dengan label HS. Hal ini diperlukannya oversampling data dengan menggunakan teknik SMOTE (*synethetic Minority over-sampling tehnique*). Hasil *oversampling* dapat dilihat dari gambar yang menunjukkan pada gambar 5 dan 6 yang telah dilakukan *oversampling.*

```
Counter({'NonHS': 757, 'HS': 339})
Counter({'NonHS': 757, 'HS': 757})
```

Gambar 5. Pelabelan *Inset Lexicon*

```
Counter({'NonHS': 981, 'HS': 115})
Counter({'NonHS': 981, 'HS': 691})
```

Gambar 6. Pelabelan *SentimentIntensityAnalyzer*

Dapat dilihat dari gambar 5 dan 6 data awal memiliki nilai mayoritas yang tinggi pada label NonHS dan memiliki nilai minoritas pada label HS. Setelah dilakukannya *oversampling* data memiliki kesetaraan antara nilai mayoritas dan minoritas. Selanjutnya, proses pembobotan kata dilakukan dengan mengektrak menggunakan fitur TF-IDF dibantu dengan modul *TFidfVectorizer* dan *library scikit-learn*. Penilaian yang dilakukan pada pembobotan fitur TF-IDF menunjukkan seberapa pentingnya kata dalam dokumen dibandingkan dengan kumpulan dokumen secara keseluruhan.

```
  (0, 1287)     0.5798338519870305
  (0, 1207)     0.6138981608124245
  (0, 913)      0.33700527262895946
  (0, 198)      0.4163522528596254
  (1, 1542)     0.6618864841586943
  (1, 1003)     0.6251593736329061
  (1, 940)      0.4136206470269044
  (2, 1283)     0.35874418271402914
  (2, 1123)     0.35874418271402914
  (2, 940)      0.2373538768346868
  (2, 834)      0.6875815487243194
  (2, 597)      0.35874418271402914
  (2, 139)      0.291208834192818 3
  (3, 1258)     0.6676774021636728
  (3, 912)      0.23344359847199675
  (3, 418)      0.7069023786722001
  (4, 912)      0.297896716268232
  (4, 846)      0.6453679714314148
  (4, 673)      0.7033901676076411
  (5, 912)      0.19479602959741169
  (5, 530)      0.5898717188204624
  (5, 278)      0.5898717188204624
  (5, 256)      0.5159042716674004
  (6, 1113)     0.428445781916737
  (6, 737)      0.5726605367476275
  :     :
  (1085, 582)   0.39296323262392221
  (1085, 266)   0.39296323262392221
  (1085, 238)   0.3556874182470376
  (1086, 929)   1.0
  (1088, 913)   0.4812183743817366
  (1088, 152)   0.8766007507168806
  (1089, 912)   0.3135783063209148
  (1089, 626)   0.9495623443486513
  (1090, 1579)  0.9495623443486513
  (1090, 912)   0.3135783063209148
  (1091, 1305)  0.4019088778803414
  (1091, 990)   0.8510407596914357
  (1091, 709)   0.3379332466993869
  (1092, 1477)  0.6073170916570445
  (1092, 1462)  0.6073170916570445
  (1092, 927)   0.30216957287894164
  (1092, 805)   0.4135522332047257
  (1093, 927)   0.5846596991497103
  (1093, 74)    0.8112786427548615
  (1094, 1057)  1.0
  (1095, 1450)  0.3424090301764513
  (1095, 751)   0.39913101415925073
  (1095, 570)   0.20116653664838216
  (1095, 142)   0.36958964112180553
  (1095, 138)   0.7391792822436111
```

Gambar 7. *TF-IDF Inset Lexicon*

```
  (0, 1175)     0.5236820605426539
  (0, 578)      0.5236820605426539
  (0, 100)      0.5236820605426539
  (0, 1389)     0.3185575444141244
  (0, 910)      0.27530417594770307
  (1, 291)      0.6552098989616985
  (1, 1544)     0.6552098989616985
  (1, 935)      0.37603188243179825
  (2, 1165)     0.3634921163130489
  (2, 1357)     0.3634921163130489
  (2, 416)      0.6966816030096779
  (2, 1092)     0.3634921163130489
  (2, 1176)     0.2635767414167899
  (2, 935)      0.22086758373911522
  (3, 751)      0.556926537339423
  (3, 375)      0.5227848818179404
  (3, 13)       0.6152920829588293
  (3, 909)      0.19479335571708936
  (4, 729)      0.7022801269119127
  (4, 600)      0.6523042383240948
  (4, 909)      0.2851347120380301
  (5, 613)      0.6042554972060716
  (5, 1243)     0.5469368630732188
  (5, 1455)     0.5469368630732188
  (5, 909)      0.19129931970474276
  :     :
  (1085, 1261)  0.38757843071574943
  (1085, 1053)  0.36607227187583574
  (1085, 565)   0.3389777160570432
  (1086, 919)   1.0
  (1088, 1100)  0.8737721102751803
  (1088, 910)   0.486335583013682
  (1089, 490)   0.9533641344967381
  (1089, 909)   0.30182250919254744
  (1090, 1584)  0.9533641344967381
  (1090, 909)   0.30182250919254744
  (1091, 1286)  0.4243502452423407
  (1091, 981)   0.8487004904846814
  (1091, 1530)  0.3156490880928162
  (1092, 1041)  0.5905880998886104
  (1092, 1469)  0.5905880998886104
  (1092, 924)   0.3806265295874843
  (1092, 470)   0.3969065853750847
  (1093, 1444)  0.760349325438247
  (1093, 935)   0.6495143595838376
  (1094, 295)   1.0
  (1095, 320)   0.39858027038975685
  (1095, 1127)  0.3585504367079487
  (1095, 968)   0.3585504367079487
  (1095, 144)   0.7381593204521373
  (1095, 661)   0.19783264229834716
```

Gambar 8. *TF-IDF SentimentIntensityAnalyzer*

Sebagai contoh pada gambar 5 terdapat implementasi ektaksi fitur TF-IDF, dengan memanfaatkan modul *TfidfVectorizer*. Diketahui (0, 1287) merupakan indeks kolom, setelah indeks merupakan nilai bobot kata yang telah diekstraksi menggunakan fitur TF-IDF. Sebagai contoh, (0, 1287) 0.5798338519870305 kata yang mewakili indeks 128 memiliki bobot TF-IDF sebesar 0.5798338519870305 dalam dokumen pertama.

Kemudian, sebelum dilakukan implementasi algoritma SVM, dataset terlebih dahulu melakukan pembagian data atau biasa disebut *splitting* data. Model algoritma memerlukan data *train* dan *test* sebagai pembelajaran model untuk mempelajari pola data dalam melakukan prediksi yang akurat.

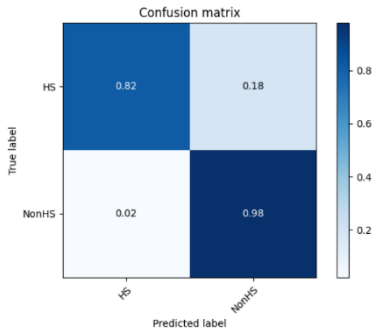
```
Jumlah seluruh data :  1514
Jumlah data train :  1211
Jumlah data test:  303
```

Gambar 9. *train* dan *test Inset Lexicon*

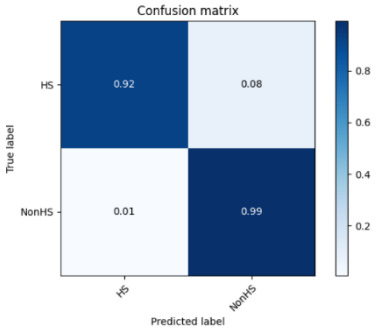
```
Jumlah seluruh data :  1672
Jumlah data train :  1170
Jumlah data test:  502
```

Gambar 10. *train* dan *test SentimentIntensityAnalyzer*

Pembagian data dilakukan sebanyak 30% untuk data *train* dan 70% untuk data *test.* Gambar 5 menunjukkan total pembagian data dari jumlah keseluruhan data sebanyak 1514, data *train* menjadi 1211 dan data *test* 303. Sedangkan pada gambar 6, total dari keseluruhan data sebanyak 1672, data *train* menjadi 1170 dan data *test* sebanyak 502. Selanjutnya, mengimplementasikan model algoritma SVM untuk mengetahui prediksi nilai model SVM. Gambar 7 dan 8 merupakan *confusion matrix* dari evaluasi model SVM.



Gambar 11.*Confusion Matrix Inset Lexicon*



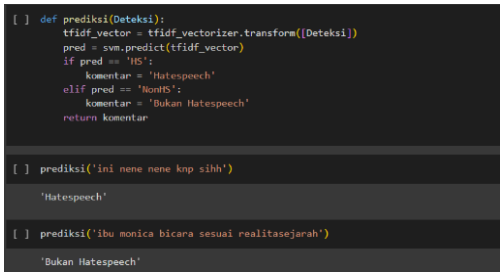
Gambar 12. *Confusion Matrix SentimentIntensityAnalyzer*

Menguji performa model untuk mendapatkan hasil evaluasi akurasi, presisi, *recall* dan f1-*score*. Hasil yang didapatkan pada tabel 6 merupakan hasil prediksi model algoritma SVM dari implementasi 2 *library* yang digunakan pada penelitian ini.

Tabel 6. Identifikasi kamus kata *Lexicon* dengan TF-IDF

	Inset Lexicon	***Vader Sentiment***
Akurasi	89%	96,21%
Presisi	81,64%	92,23%
Recall	97,72%	99%
F1-*score*	88,96%	95,50%

Dengan dilakukan mengidentifikasi dari hasil penelitian diatas mendapatkan pada tabel 6 *Inset lexicon* sudah cukup baik untuk menganalisis secara sentimen namun untuk mendeteksi *hate speech* lebih baik menggunakan *Vader Sentiment* karena akurasi yang didapatkan kamus *Vader Sentiment* sedikit lebih besar. *Vader Sentiment* mendapatkan akurasi 96, 21% sedangkan *Inset lexicon* 89%, presisi *Vader Sentiment* 92, 23% dan *Inset lexicon* mendapatkan 81, 64%, *recall Vader Sentiment* 99% sementara 97, 72% dan f1-*score Vader Sentiment* 95, 50% sedangkan *Inset lexicon* 88, 96%



Gambar 13. Uji prediksi label dan model

Terakhir, pengujian dari hasil prediksi label yang sudah mendapatkan akurasi terbaik dari performa model. Selanjutnya, pengujian dengan program sederhana untuk mendeteksi *hate speech*. Dari kedua kamus yang digunakan berhasil mendeteksi komentar *hate speech* pada penelitian ini.

D. Simpulan

Berdasarkan hasil dari penelitian diatas telah melakukan identifikasi terhadap *hate speech* pada komentar TikTok dapat simpulkan bahwa, penelitian ini menggunakan dataset 1096 data komentar diperoleh dari aplikasi TikTok. Klasifikasi yang dilakukan pada proses pelabelan otomatis dengan menggunakan kamus kata *Inset lexicon* menghasilkan 69, 1% data berlabel nonHS dan 30, 9% berlabel HS. Sedangkan *Vader Sentiment* menghasilkan 89, 51% data berlabel nonHS dan 10, 49% berlabel HS.

Hasil evaluasi yang telah dilakukan dengan memberikan bobot kata menggunakan fitur TF-IDF dan algoritma SVM didapati bahwa, penggunaan *Vader Sentiment* untuk *deteksi hate speech* lebih unggul mendapatkan akurasi sebesar 96, 21%, presisi 92, 23%, *recall* 99%, dan f1-*score* 95, 50%. Sedangkan, penggunaan *Inset lexicon* mendapatkan 89%, presisi 81, 64%, *recall* 97,72% dan f1-*score* 88, 96%.

Hasil evaluasi dapat disimpulkan bahwa, penggunaan kamus kata dengan pelabelan otomatis *inset lexicon* sudah baik dalam melakukan pelabelan otomatis secara sentimen. Meskipun demikian, *vader sentiment* lebih baik digunakan untuk mendeteksi *hate speech* karena memperoleh hasil akurasi sangat baik. Perolehan akurasi yang baik dipengaruhi pada pengambilan pada jumlah data dan memilih algoritma terbaik untuk melakukan klasifikasi.

E. Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada Universitas Muhammadiyah Prof. Dr. Hamka Jakarta atas seluruh dukungan dalam penelitian ini.

F. Referensi

[1] F. Mayasari, "Etnografi Virtual Fenomena Cancel Culture dan Partisipasi Pengguna Media terhadap Tokoh Publik di Media Sosial," *J. Commun. Soc.*, vol. 1, no. 01, pp. 27–44, 2022, doi: 10.55985/jocs.v1i01.15.

[2] U. Hasanah, "Analisis Penggunaan Gaya Bahasa Sarkasme Netizen di Media Sosial Instagram," *J. Onoma Pendidikan, Bahasa, dan Sastra*, vol. 7, no. 2, pp. 411–423, 2021, doi: 10.30605/onoma.v7i2.1255.

[3] Indah and K. Putri, "Komunikasi Online Dalam Penyebaran Hate Speech Di Media Sosial Tiktok," *Pros. Konf. Nas. Sos. dan Polit.*, vol. 1, no. 0, pp. 327–341, 2023, doi: https://doi.org/10.55357/sosek.v2i1.125.

[4] O. Yanto, "Pemindanaan atas kejahatan yang berhubungan dengan teknologi informasi," 1st ed., Alviana, Ed., Banguntapan Bantul DI Yogyakarta: Samudra Biru, 2021, pp. 57–65.

[5] I. M. Kardiyasa, A. A. S. L. Dewi, and N. M. S. Karma, "Sanksi Pidana Terhadap Ujaran Kebencian (Hate Speech)," *J. Analog. Huk.*, vol. 2, no. 1, pp. 78–82, 2020, doi: 10.22225/ah.2.1.1627.78-82.

[6] B. A. H. Kholifatullah and A. Prihanto, "Penerapan Metode Long Short Term Memory Untuk Klasifikasi Pada Hate Speech," *J. Informatics Comput. Sci.*, vol. 04, pp. 292–297, 2023, doi: 10.26740/jinacs.v4n03.p292-297.

[7] A. Wikandiputra, Afiahayati, and V. M. Sutanto, "Identifying Hate Speech in Bahasa Indonesia With Lexicon-Based Features and Synonym-Based Query Expansion," *ICIC Express Lett.*, vol. 16, no. 8, pp. 811–818, 2022, doi: 10.24507/icicel.16.08.811.

[8] R. Yunita and M. Kamayani, "Perbandingan Algoritma SVM Dan Naïve Bayes Pada Analisis Sentimen Penghapusan Kewajiban Skripsi," *Indones. J. Comput. Sci.*, vol. 12, no. 5, pp. 2879–2890, 2023, doi: 10.33022/ijcs.v12i5.3415.

[9] W. F. Abdillah, A. Premana, and R. M. H. Bhakti, "Analisis Sentimen Penanganan Covid-19 dengan Support Vector Machine: Evaluasi Leksikon dan Metode Ekstraksi Fitur," *J. Ilm. Intech Inf. Technol. J. UMUS*, vol. 3, no. 02, pp. 160–170, 2021, doi: 10.46772/intech.v3i02.556.

[10] L. Hermawan and M. Bellaniar Ismiati, "Pembelajaran Text Preprocessing berbasis Simulator Untuk Mata Kuliah Information Retrieval," *J. Transform.*, vol. 17, no. 2, p. 188, 2020, doi: 10.26623/transformatika.v17i2.1705.

[11] D. Musfiroh, U. Khaira, P. E. P. Utomo, and T. Suratno, "Analisis Sentimen terhadap Perkuliahan Daring di Indonesia dari Twitter Dataset Menggunakan InSet Lexicon," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 1, no. 1, pp. 24–33, 2021, doi: 10.57152/malcom.v1i1.20.

[12] S. Roiqoh, B. Zaman, and K. Kartono, "Analisis Sentimen Berbasis Aspek Ulasan Aplikasi Mobile JKN dengan Lexicon Based dan Naïve Bayes," *J. Media Inform. Budidarma*, vol. 7, no. 3, pp. 1582–1592, 2023, doi: 10.30865/mib.v7i3.6194.

[13] A. Faizal, A. Susilo, Y. Irawan, and D. Juardi, "Perbandingan Lexicon Based Dan Naïve Bayes Classifier Pada Analisis Sentimen Pengguna Twitter Terhadap Gempa Turki Comparison of Lexicon-Based and Naive Bayes Classifier Methods on Sentiment Analysis of Twitter Users To the Turkey Earthquake," *J. Inf. Technol. Comput. Sci.*, vol. 6, no. 2, 2023.

[14] C. S. Sriyano and E. B. Setiawan, "Pendeteksian Berita Hoax Menggunakan Naive Bayes Multinomial Pada Twitter dengan Fitur Pembobotan TF-IDF," *Repos. Telkom Univ.*, vol. 8, no. 2, p. 3396, 2021.

[15] S. A. Pratomo, S. Al Faraby, and M. D. Purbolaksono, "Analisis Sentimen Pengaruh Kombinasi Ekstraksi Fitur TF-IDF dan Lexicon Pada Ulasan Film Menggunakan Metode KNN," *e-Proceeding Eng.*, vol. 8, no. 5, pp. 10116–10126, 2021, [Online]. Available:

https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/15726

[16] I. Apriani, Y. Sibaroni, and I. Palupi, "Perbandingan Pembobotan Fitur TF-IDF dan TF-ABS Dalam Klasifikasi Berita Online Menggunakan Support Vector Machine (SVM)," *e-Proceeding Eng.* , vol. 10, no. 3, pp. 3652–3663, 2023, [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/20639

[17] Oryza Habibie Rahman, Gunawan Abdillah, and Agus Komarudin, "Klasifikasi Ujaran Kebencian pada Media Sosial Twitter Menggunakan Support Vector Machine," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 1, pp. 17–23, 2021, doi: 10.29207/resti.v5i1.2700.