Klasifikasi Ujaran Kebencian pada Cuitan dalam Bahasa Indonesia

Kevin Antariksa[1], Y. Sigit Purnomo WP.[2], Dra. Ernawati[3]
[1,2,3]Program Studi Teknik Informatika, Fakultas Teknologi Industri
Universitas Atma Jaya Yogyakarta
Jl. Babarsari No 43, Yogyakarta, 55281, Daerah Istimewa Yogyakarta, Indonesia
Email: [1]kevinantariksa@gmail.com, [2]sigit.purnomo@uajy.ac.id, [3]ernawati@uajy.ac.id

Masuk: 2 September 2019; Direvisi: 12 Oktober; Diterima: 22 Oktober 2019

Abstract. *The sheer amount of hate speech in social media is making people nauseous. The amount of hate speech these days keeps increasing and yet, there was no preventive act to counter back the hate speech. Pre-existing hate speech detection is also not yet available in Bahasa Indonesia. A machine learning model that is able to recognize hate speech in Bahasa Indonesia will be explained in this article. The model will compare pre-existing methods in machine learning. Naive Bayes, SVM, and Logistics Regression are the methods that will be used for the model. Some of the parameters in the test will be altered to achieve the maximum value for detecting hate speech. The expectation is a machine learning model that is able to recognize hate speech in Bahasa Indonesia accurately. Expected accuracy is above 85%. After the experiment, the highest accuracy achieved was at 98%, while the lowest accuracy was only 80%.*
Keywords: *Hate speech detection, machine learning model, social media, Bahasa Indonesia, tweets*

Abstrak. *Banyaknya ujaran kebencian yang ada di media sosial sudah membuat jengah. Ujaran kebencian tersebut makin marak dijumpai namun masih belum ada upaya preventif dari media sosial untuk menangkalnya. Deteksi ujaran kebencian yang sudah dibuat juga belum tersedia dalam Bahasa Indonesia. Sebuah model pembelajaran mesin yang dapat mengenali ujaran kebencian dengan Bahasa Indonesia akan dibahas pada naskah ini. Dalam model tersebut dibandingkan beberapa metode pembelajaran mesin yang ada. Metode yang digunakan dalam pengujian adalah Naïve Bayes, SVM, dan Logistic Regression. Dalam pengujian, beberapa parameter akan diubah-ubah sehingga didapatkan nilai paling maksimal dalam deteksi ujaran kebencian. Hasil yang diharapkan adalah sebuah model pembelajaran mesin. Model tersebut diharapkan dapat mengenali ujaran kebencian berbahasa Indonesia secara akurat. Akurasi yang diharapkan adalah diatas 85%. Setelah percobaan, didapatkan nilai akurasi paling tinggi yaitu 98%, sedangkan nilai akurasi paling rendah yaitu 80%.*
Kata Kunci: *Deteksi ujaran kebencian, model pembelajaran mesin, media sosial, Bahasa Indonesia, cuitan*

1. Pendahuluan

Ujaran kebencian akhir-akhir ini banyak menyita perhatian. Salah satu kasus ujaran kebencian berbasis media sosial yang popular di Indonesia adalah kasus ujaran kebencian yang dilakukan oleh musisi Ahmad Dhani. Ahmad Dhani membuat cuitan di Twitter, pada Maret 2017 lalu. Cuitan tersebut secara tidak langsung merujuk pada Gubernur DKI nonaktif saat itu, Basuki Tjahaja Purnama. Kasus ujaran kebencian ini telah ditangani oleh kepolisian setelah pihak Basuki Tjahaja Purnama melapor ke Polisi.

Ujaran kebencian sendiri didefinisikan sebagai komunikasi yang bertujuan meremehkan orang, kelompok, atau golongan berdasarkan suku, agama, ras, etnik, golongan, kewarganegaraan dan karakteristik lain [1]. Bertambahnya pengguna internet setiap tahunnya, berdampak pada meningkatnya jumlah ujaran kebencian yang tersebar di media sosial [2].

Dengan banyaknya ujaran kebencian yang tersebar di internet, tentu saja membuat baik korban maupun pengguna media sosial yang melihat merasa tidak nyaman.

Hal yang membuat klasifikasi ujaran kebencian sangat sulit, karena tidak ada standar yang benar-benar baku untuk ujaran kebencian [3]. Beberapa merasa komentar/cuitan seseorang di media sosial sangat menyakitkan, namun mungkin bagi orang lain, hal tersebut bukan masalah. Tidak banyak korban ujaran kebencian yang melapor, baik karena takut atau merasa tidak peduli, membuat perilaku ujaran kebencian akan selalu tumbuh subur di Indonesia.

Indonesia sendiri, sudah memiliki Undang-Undang Informasi Transaksi Elektronik (UU ITE) yang dapat digunakan untuk menjerat para pelaku ujaran kebencian ke meja hijau. Salah satu cara lain untuk menghentikan ujaran kebencian adalah dengan melakukan penyaringan komentar/cuitan. Hal yang paling mudah untuk membatasi ujaran kebencian adalah melakukan pemblokiran cuitan untuk setiap kata/frasa yang dianggap sebagai bagian dari ujaran kebencian. Namun, karena tidak semua komentar maupun cuitan yang menggunakan kata/frasa tersebut. Sebagai contoh, menggunakan kata anjing dalam konteks hewan peliharaan akan sangat berbeda bila kata anjing digunakan untuk menyatakan umpatan/kebencian.

Pembelajaran mesin diperlukan untuk membantu media sosial dalam melacak ujaran kebencian, bahkan melakukan upaya preventif, yaitu menghindari ujaran kebencian tersebut tampil di media sosial. Alasan digunakannya pembelajaran mesin, karena melalui pembelajaran mesin, sistem dapat melakukan analisis untuk mengetahui apakah komentar/cuitan seseorang mengandung ujaran kebencian. Dataset ujaran kebencian juga diperlukan sehingga mesin dapat mempelajari mana saja yang termasuk ujaran kebencian. Untuk meningkatkan akurasi, maka dilakukan pengujian dengan menggunakan berbagai algoritma pembelajaran mesin, sehingga dapat ditemukan algoritma yang akurat dalam mendeteksi ujaran kebencian. Model pembelajaran mesin deteksi ujaran kebencian ini dapat digunakan dan dimanfaatkan siapa saja demi menciptakan internet yang bebas dari ujaran kebencian.

2. Tinjauan Pustaka

Ujaran kebencian sudah sangat marak di media sosial, tidak terkecuali Twitter. Banyak upaya yang telah dilakukan oleh pengembang, salah satunya mempekerjakan moderator konten. Moderator tersebut bertugas untuk menyaring banyak hal yang dilaporkan oleh pengguna, mulai dari gambar tidak senonoh, hingga ujaran kebencian.

Pekerjaan moderator konten akan bertambah banyak, seiring dengan perkembangan pengguna. Pertumbuhan pengguna juga berbanding lurus dengan konten yang dihasilkan. Apabila konten negatif dari situs sangat banyak, konfirmasi konten yang memuat ujaran kebencian akan memakan waktu sangat lama. Lamanya waktu konfirmasi ujaran kebencian akan sangat merugikan apabila sudah terjadi perselisihan dan menciptakan suasana tidak menyenangkan [4].

Dengan adanya pembelajaran mesin yang dapat mendeteksi ujaran kebencian, maka hal seperti moderator untuk menganalisa komentar atau konten menjadi tidak begitu diperlukan [5]. Sangat mungkin ujaran kebencian dapat lolos mengingat ragam dari ujaran kebencian sangat banyak. Disatu sisi, pekerjaan moderator konten akan lebih ringan dibanding saat tidak adanya sistem pengenalan ujaran kebencian.

Penelitian yang dijadikan rujukan adalah penelitian berjudul *Degree based Classification of Harmful Speech using Twitter Data* [6]. Dataset yang dibuat sebesar 9000 cuitan berbahasa Inggris. Pada *preprocessing*, cuitan dibersihkan dari URL dan *username*, normalisasi *hashtag* dengan memberikan spasi pada setiap suku kata, dan menghilangkan semua karakter khusus (*,&,%,_). Hal ini dianggap mempengaruhi performa dataset saat *preprocessing*. Ide tersebut akan diadaptasi pada penelitian ini. Algoritma yang digunakan dalam penelitian ini ada tiga, yaitu SVM, Naïve Bayes, dan Random Forest. Pada SVM dan Naïve Bayes, digunakan TF-IDF, sedangkan pada Random Forest digunakan Bag of Words. Akurasi tertinggi pada penelitian ini adalah Algoritma Random Forest, 76%, unggul 5% dibanding Naïve Bayes dan 3% dibanding SVM.

Penelitian berikutnya dilakukan oleh Arroyo [7] mengambil dataset TRAC-1. Dataset tersebut berisikan 12.041 data yang tersebar dalam beberapa kategori berdasarkan tingkat ujaran kebencian. Pada penelitian ini, digunakan beberapa algoritma seperti Naïve Bayes, Perceptron, SVM, Passive Aggressive. Penelitian tersebut menggunakan TF-IDF, Bag of Words, WISSE, dan N-gram. Pada penelitian tersebut, Perceptron dianggap sangat tidak stabil. Naïve Bayes dan SVM memiliki performa lebih baik dan stabil. Kenaikan akurasi sebesar kurang lebih 10% memiliki performa yang stabil saat TF-IDF dikombinasikan n-gram dibandingkan hanya TF-IDF. TF-IDF yang tidak digunakan bersama SVM dan Naïve Bayes mencatatkan performa yang lebih baik dibandingkan dengan Bag of Words. Secara menyeluruh, SVM dan Naïve Bayes mencatatkan performa paling baik dan akurasi paling tinggi dibandingkan Perceptron dan Passive Aggressive.

Referensi berikutnya adalah penelitian yang dilakukan oleh Samghabadi [8]. Penelitian ini mirip dengan penelitian sebelumnya karena menggunakan dataset TRAC-2018. Dataset tersebut diambil dari Facebook yang terdiri dari 12.000 ujaran kebencian berbahasa Inggris dan 12.000 ujaran kebencian berbahasa India. Pada penelitian kali ini digunakan algoritma Logistic Regression dan SVM. Pada akhirnya, karena dinilai lebih baik dibandingkan SVM, penelitian ini diteruskan dengan menggunakan algoritma Logistic Regression. Pada penelitian ini juga digunakan TF-IDF, N-grams, Sentinent, Word2Vec, LIWC, dan gender probability. Pada hasil percobaan, untuk dataset berbahasa Inggris diketahui TF-IDF memiliki akurasi paling tinggi, disusul dengan n-grams dan Word2Vec, unggul lebih dari 20% dibanding *sentiment*, LIWC dan gender probability. Kombinasi TF-IDF, Word2Vec, dan n-grams memiliki hasil 58,75%, unggul 0.71% dibanding saat hanya menggunakan TF-IDF.

Penelitian terakhir adalah penelitian Del Vigna [9]. Penelitian ini menggunakan dataset dari Facebook yang berisikan 17.567 komentar dengan Bahasa Italia. Penelitian ini menggunakan algoritma SVM dan *Long-Short-Term-Memory* (LSTM). Digunakan juga word2vec untuk melengkapi SVM dan LSTM. Dari hasil penelitian, dapat diketahui bahwa SVM memiliki akurasi lebih baik dibanding LSTM, yaitu akurasi lebih tinggi 4%.

Pada penelitian ini, dilakukan percobaan klasifikasi ujaran kebencian menggunakan beberapa algoritma yang dianggap baik. Tidak semua algoritma akan menghasilkan hasil yang sama dan sangat tergantung pada dataset. Pada penelitian Pistulis [10], LSTM memiliki hasil akurasi lebih dari 90%, Hasil yang bertolak belakang terjadi pada penelitian Del Vigna [9], dimana SVM memiliki hasil yang lebih baik dibandingkan dengan LSTM. Oleh karena itu akan dicoba menggunakan tiga algoritma yang sering digunakan seperti SVM, Logistic Regression, dan Naïve Bayes. Hal yang akan membedakan penelitian ini dengan beberapa penelitian sebelumnya adalah penelitian ini menggunakan *feature extraction* seperti TF-IDF, Word2Vec, dan n-grams. Ketiganya akan dicoba pada masing-masing algoritma untuk dicari mana algoritma yang menghasilkan nilai akurasi paling tinggi.

Berdasarkan tinjauan pustaka yang telah dilakukan, maka penelitian ini menggunakan tiga algoritma: SVM, Naïve Bayes, dan Logistic Regression. Algoritma tersebut merupakan algoritma *supervised learning* dan biasa digunakan sebagai klasifikasi. Algoritma-algoritma tersebut juga memilki performa yang baik dalam mendeteksi cuitan atau teks.

Algoritma Naïve Bayes bekerja dengan cara memberikan prediksi [11]. Prediksi tersebut diberikan setelah melakukan kalkulasi berdasarkan *input* yang telah diberikan. Kalkulasi melibatkan nilai hipotesa dan nilai probabilitas dari kejadian. SVM bekerja dengan cara mengelompokkan beberapa kategori dan memisahkan kategori tersebut dengan hyperplane [11]. Sedangkan support vector adalah data yang paling dekat letaknya dengan hyperplane. Logistic Regression bekerja dengan cara melakukan prediksi menggunakan metode-metode statistika, dimana pengguna dapat melakukan prediksi dari fungsi *sigmoid* [11].

3. Metodologi Penelitian

Penelitian ini merupakan pembuatan model pembelajaran mesin untuk mendeteksi ujaran kebencian berbahasa Indonesia. Dalam pembuatan model akan dilakukan beberapa pengujian

algoritma dan *preprocessing* sehingga model yang dibuat mampu mengenali ujaran kebencian dengan akurasi yang tinggi. Berikut ini adalah langkah-langkah yang akan dilakukan:

Pembuatan Dataset, melalui tahap ini, Dataset dibuat dengan cara mengumpulkan cuitan dari Twitter. Pengumpulan cuitan menggunakan *library* Tweepy. Cuitan yang dikumpulkan adalah cuitan dengan ujaran kebencian berbahasa Indonesia.

Pembersihan Dataset, dataset yang digunakan harus bersih dari elemen-elemen yang tidak dibutuhkan. Dataset juga harus bebas dari tulisan yang menggunakan karakter spesial atau huruf acak yang tidak memiliki makna. Dataset juga dibersihkan dari data yang tidak lengkap dengan cara menghapus data tersebut. Dalam pembersihan dataset, juga dilakukan pemisahan antara *hashtag*, nama pengguna, dan waktu cuitan dibuat.

Memberikan Label, dataset yang sudah bersih dapat diberikan label. Label yang diberikan berupa angka 0 dan 1, dimana 0 merepresentasikan cuitan yang bukan merupakan ujaran kebencian dan 1 adalah cuitan dengan ujaran kebencian. Cuitan yang dinilai sebagai ujaran kebencian harus memenuhi unsur provokasi, penghinaan, diskriminasi, ancaman kepada Suku, Agama, Ras, dan Antar golongan (SARA) yang bertujuan untuk memusuhi SARA tertentu. Dataset yang telah dikumpulkan kebanyakan mengarahkan kebencian kepada Presiden Jokowi, aparat penegak hukum, dan etnis Tionghoa.

Ujaran kebencian memiliki beberapa tahapan yang membedakannya dengan ujaran biasa [12]. Pada tahap pertama, adalah tahap ujaran biasa yang masih etis untuk disampaikan. Meskipun sudah mengandung unsur-unsur ujaran kebencian, ujaran ini masih etis karena tidak memunculkan unsur permusuhan. Sebagai contoh, kalimat "konglomerat di Indonesia umumnya orang keturunan Cina" [12] masih dianggap etis. Tahap kedua adalah tahap *stereotyping*, yang merupakan tahap menyamakan sifat atau karakteristik sebuah golongan dengan hal-hal informasi yang diperoleh dari pihak lain. Contoh kalimat yang merupakan *stereotype* adalah "semua orang Cina itu kaya, mereka yang bikin pribumi miskin" [12]. Ujaran tersebut belum dapat disebut sebagai ujaran kebenncian karena belum ada unsur untuk memusuhi. Kalimat tersebut baru mulai memunculkan kebencian. Cuitan yang diberikan label pada penelitain ini adalah cuitan yang masuk kategori terakhir. Dataset juga diberikan kolom untuk menampilkan sasaran ujaran kebencian dan keterangan rinci pada siapa ujaran tersebut ditujukan (subjek).

Tabel 1. Tabel Perbandingan Label

Label	Jumlah Cuitan
0 (bukan ujaran kebencian)	20.921
1 (ujaran kebencian)	384

presiden terbaik buat asing tidak baik buat pribumi membangun negeri untuk kejayaan china komunis pokoknya gw sudah tidak respek untuk dukung dia lagi tolak antek curang memilih aksi mahasiswa muslim mendukung capres yang cinta tanah air selamanya anti pki	1	pemerintah	presiden jokowi

Gambar 1. Contoh cuitan dengan ujaran kebencian

Tabel 1 menujukkan perbandingan komposisi label pada dataset setelah dilakukan pelabelan. Sedangkan pada Gambar 1, merupakan hasil tangkapan layar dari berkas dataset. Gambar 1 menampilkan contoh cuitan dengan keterangan: kolom pertama merupakan ujaran kebencian yang didapat dari Twitter, kolom kedua menunjukkan label (merujuk pada Tabel 1), kolom ketiga menunjukkan kepada lembaga apa cuitan tersebut ditunjukkan, dan kolom terakhir menunjukkan kepada siapa cuitan tersebut ditunjukkan (secara spesifik).

Feature Extraction, Model dibuat dengan memasukkan dataset, lalu untuk pembelajaran mesinnya menggunakan algoritma dan *feature extraction*. Disini model akan mulai diuji coba dengan melakukan implementasi *preprocessing*, *feature extraction*, dan penerapan algoritma.

Algoritma yang digunakan ada 3, yaitu Support Vector Machine, Logistic Regression, dan Naïve Bayes. *Feature extraction* yang digunakan, TF-IDF, n-grams, dan word2vec akan diimplementasikan pada setiap algoritma.

Term Frequency – Inverse Document Frequency (TF-IDF) bekerja dengan cara memberikan bobot pada setiap kata atau kalimat pada cuitan, berdasarkan frekuensi kemunculan kata atau kalimat tersebut [13]. N-gram bekerja dengan cara memotong-motong cuitan menjadi beberapa suku kata maupun kalimat [13]. Sedangkan word2vec bekerja dengan memberikan *vector* pada setiap kata atau kalimat [13].

TF-IDF pada penelitian ini, memindai *corpus* cuitan secara menyeluruh, lalu memberikan bobot pada setiap kata-kata tertentu yang sering muncul. Semakin sering sebuah kata muncul, semakin besar bobotnya. Pemberian bobot ini bisa dalam bentuk desimal maupun biner. Pada penelitian ini, digunakan pemotongan n-gram pada setiap kata dan kalimat. Terdapat dua percobaan, yaitu menggunakan dua hingga tiga kata dan dua hingga tiga kalimat. Percobaan yang menghasilkan akurasi tinggi yang akan dimasukan dalam tabel perbandingan. Sedangkan pada word2vec, menggabungkan seluruh cuitan menjadi sebuah *corpus*, lalu memberikan bobot dengan rentang nilai rata-rata diantara nol sampai satu.

Penerapan Algoritma, Alur pembuatan model adalah memanggil *library* yang ingin digunakan, melakukan *import* dataset yang sudah dibuat, *split* dataset untuk *training* dan *testing*, *feature extraction*, lalu baru diimplementasikan dengan algoritma. Pemisahan atau *split* dataset, digunakan K-Fold, yang mana K-Fold membantu untuk menyamakan rasio dari data *train* dan *test* yang akan diolah kemudian. K-Fold bekerja dengan cara memisahkan *train-test* lalu membuat rasio yang tepat. Rasio ini dapat diatur dalam pemanggilan fungsi K-Fold. *Splits* menunjukkan akan ada lima lipatan yang akan digunakan secara bergantian dalam *training* dan *testing*. Lipatan yang sudah ditentukan tadi, akan secara bergantian digunakan sebagai penampung untuk *training* atau *testing*. Jika splits yang digunakan lima, formulasi yang digunakan biasanya empat untuk *training* dan satu untuk *testing*

Data *train* yang digunakan sebanyak 16.737 data bukan ujaran kebencian dan 307 data ujaran kebencian. Data *test* sebanyak 4.184 data bukan ujaran kebencian dan 87 data ujaran kebencian. Sedangkan pada Word2Vec, digunakan *split* dataset yang berbeda. Algoritma yang dikombinasikan dengan Word2Vec digunakan sebanyak 16.747 data bukan ujaran kebencian dan 296 data ujaran kebencian. Data *test* sebanyak 4.174 data bukan ujaran kebencian dan 76 data ujaran kebencian. Model yang sudah selesai, bisa saja belum maksimal dalam deteksi ujaran kebencian. Dalam tahapan ini, model akan diuji coba dengan mengurangi atau menambahkan parameter yang mungkin akan meningkatkan akurasi dari dataset.

Penyesuaian Model, Model yang sudah jadi bisa saja belum maksimal dalam deteksi ujaran kebencian. Dalam tahapan ini, model akan diuji coba dengan mengurangi atau menambahkan parameter yang mungkin akan meningkatkan akurasi dari dataset. Pada tahap ini juga akan menyesuaikan model sehingga deteksi lebih akurat.

Evaluasi Model, Evaluasi model adalah menganalisa model yang sudah dibuat. Model yang sudah dibuat akan dilihat akurasinya dan dibandingkan antara algoritma satu dan algoritma lain. Evaluasi ini bertujuan untuk menguji seberapa akuratnya algoritma dan *preprocessing* yang digunakan. Pada tahap ini juga dilakukan penyesuaian ulang pada *feature extraction* maupun algoritma yang digunakan apabila dirasa masih kurang ideal.

4. Hasil dan Diskusi

Ujaran kebencian, untuk dapat dideteksi, diperlukan model pembelajaran mesin yang telah dilatih. Untuk melatih model pembelajaran mesin, diperlukan berbagai elemen yang akan digunakan. Berikut merupakan langkah-langkah penelitian yang telah dilakukan:

4.1. Pembuatan Dataset

Dataset merupakan elemen yang sangat diperlukan dalam membuat model pembelajaran mesin. Dataset yang dapat disebut baik, adalah dataset yang sudah tidak mengandung elemen-elemen yang tidak berguna, sebagai contoh data *redundant* dan data yang tidak dapat terbaca. Dataset dalam penelitian ini melewati beberapa langkah agar dapat dibaca dengan baik. Dataset

yang telah berhasil dikumpulkan sebanyak 21.384 cuitan dalam Bahasa Indonesia. Dataset telah melalui normalisasi dan pembersihan dataset, sehingga dapat diolah. Dataset juga telah mendapatkan label 0 atau 1 (bukan ujaran kebencian atau ujaran kebencian), sesuai dengan konten cuitan.

4.2. Pembuatan Model

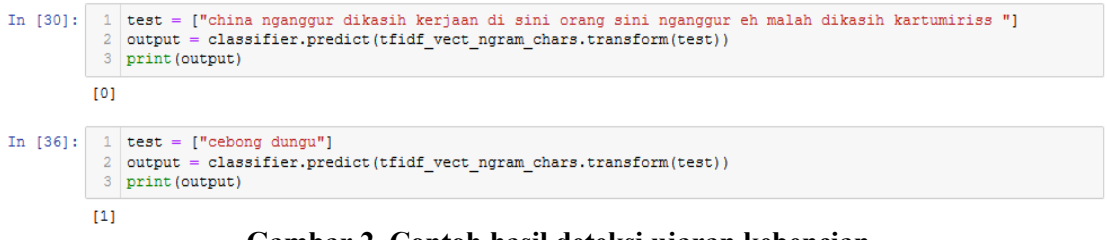
Dalam pembuatan model, Dataset akan diolah terlebih dahulu menggunakan *feature extraction* sebelum diproses menggunakan algoritma. Berikut merupakan rincian dari pembuatan model: (1) Memuat Dataset. Seperti yang telah dijelaskan sebelumnya, dataset memiliki peran yang vital sehingga perlu dimuat terlebih dahulu. Pemuatan dataset memiliki tujuan agar model pembelajaran mesin dapat bekerja dengan baik. Dataset dimuat dengan menggunakan *library* Pandas. (2) *Feature Extraction. Feature extraction* digunakan menampilkan hal-hal yang dianggap menarik, yang dinilai bisa mempengaruhi pembelajaran mesin. *Feature extraction* memiliki banyak jenis untuk berbagai kegunaan, beberapa diantaranya TF-IDF, n-gram, dan Word2Vec yang biasa digunakan untuk pembelajaran mesin. *Feature extraction* tersebut yang dapat membantu proses *training* pembelajaran mesin secara efektif. TF-IDF menggunakan *word* sebagai parameter untuk mencari frekuensinya. N-gram menggunakan *word* dan *char* sebagai parameter untuk melakukan pembobotan. N-gram menggunakan *range* dua hingga tiga huruf atau kata.. Word2Vec bekerja dengan cara memberi vektor pada setiap kata. Semua vektor kata tersebut digabungkan menjadi kata dan dijadikan rata-rata untuk kemudian diolah. (3) Penerapan Algoritma. Setelah didapatkan data *feature test* dan *feature train* untuk seluruh *feature extraction* yang telah dilakukan sebelumnya, barulah *feature* tersebut dimasukkan kedalam algoritma pembelajaran mesin. *Feature test* untuk setiap *feature extraction* akan diolah bersamaan dengan *feature train.* Pengolahan juga dilakukan secara bersama dengan label *test* dan label *train* yang telah dibuat sebelumnya. Untuk mempermudah, dibuat fungsi yang dapat memanggil algoritma untuk melakukan prediksi. Selain itu dibuat juga fungsi untuk menampilkan angka akurasi dan menampilkan *confusion matrix.*

Fungsi-fungsi tersebut kemudian dapat langsung digunakan dengan melakukan pemanggilan fungsi yang disertai dengan *feature* dan label untuk kemudian dapat diolah. Dalam pemanggilan fungsi juga langsung dapat diatur algoritma pembelajaran mesin yang akan digunakan. Pemanggilan prosedur untuk prediksi juga akan menjadi *trigger* untuk pemanggilan fungsi lain yang terkait seperti fungsi untuk menampilkan hasil prediksi dan menggambar *confusion matrix.* Terdapat sembilan kombinasi algoritma dan *feature extraction,* namun pada N-gram, karena terdapat N-gram untuk level *char* dan *word,* hanya diambil yang memiliki akurasi dan hasil prediksi yang lebih baik saja.

4.3. Evaluasi Model

Hasil penelitian disajikan dalam bentuk *confusion matrix* dan menampilkan akurasi model secara eksplisit. *Confusion matrix* ini berguna untuk memperkirakan seberapa banyak cuitan yang diduga sebagai ujaran kebencian (*false positive*) atau ujaran kebencian yang dianggap sebagai ujaran biasa (*false negative*). Secara umum, *confusion matrix* menunjukan seberapa akurat model yang dibuat dalam melakukan deteksi. Terdapat juga empat elemen yang digunakan sebagai parameter eksplisit untuk menentukan tingkat kepercayaan akan model yang telah dibuat yaitu *Accuracy, Precision, Recall,* dan *F-score.*

Data yang ditampilkan dalam tabel adalah data yang memiliki paling akurasi baik, atau dapat mengenali banyak cuitan ujaran kebencian dengan baik. N-gram yang menggunakan *char level* juga terbukti lebih unggul karena selalu mendapat nilai yang lebih baik dan atau sama dengan n-gram yang menggunakan *word level.* Bernoulli Naïve Bayes juga terlihat lebih unggul dalam melakukan deteksi ujaran kebencian disbanding Multinomial Naïve Bayes.



Gambar 2. Contoh hasil deteksi ujaran kebencian

Pada Gambar 2 dilakukan percobaan dengan menggunakan kalimat. Kalimat yang dimasukkan dalam *variable* test merupakan ujaran kebencian yang dibuat oleh penulis. Output 0 menunjukkan model tidak dapat mengenali ujaran kebencian dan output 1 menunjukkan kalimat yang dapat dikenali.

Tabel 2. Tabel Perbandingan Kombinasi *Feature Extraction* dengan Algoritma

No	Algoritma + *Feature extraction*	Akurasi	Presisi	*Recall*	F-Score	True Positive	False Positive	False Negative	True Negative
1	Bernoulli Naïve Bayes + TF-IDF	94%	97%	94%	95%	3.978	206	63	13
2	Bernoulli Naïve Bayes + N-gram Char Level	80%	97%	80%	88%	3.397	767	46	30
3	Multinomial Naïve Bayes + Word2Vec	98%	96%	98%	97%	4.174	0	87	0
4	Logistic Regression + TF- IDF	98%	98%	98%	97%	4.184	0	74	2
5	**Logistic Regression + N-gram Char Level**	**98%**	**98%**	**98%**	**98%**	**4.184**	**0**	**72**	**4**
6	Logistic Regression + Word2Vec	98%	96%	98%	97%	4.184	0	87	0
7	SVM + TF-IDF	98%	96%	98%	97%	4.184	0	76	0
8	SVM + N-gram Word Level	98%	96%	98%	97%	4.184	0	76	0
9	SVM + Word2Vec	98%	96%	98%	97%	4.174	0	87	0

Pada Tabel 2 diperoleh akurasi dan F-score yang tinggi (kolom yang dicetak tebal). Nilai akurasi dan F-score tersebut merupakan hasil perhitungan data bukan ujaran kebencian yang dideteksi oleh model, bukan akurasi ujaran kebencian dapat dideteksi. Ujaran kebencian masih belum dapat dikenali dengan baik terlihat pada beberapa algoritma. Algoritma Bernoulli Naïve Bayes + n-gram Char Level mampu menghasilkan deteksi yang tinggi, namun juga angka *false positive* yang tinggi. Perbedaan rasio yang besar antara data ujaran kebencian dan bukan ujaran kebencian (383 dibanding 20.921) diduga menjadi penyebab tingginya nilai akurasi namun hanya sedikit ujaran kebencian yang dapat dikenali. Semakin banyak data yang bukan ujaran kebencian, membuat model semakin hafal dengan data bukan ujaran kebencian. Hal ini berimplikasi pada kurang mampunya model dalam mengenali ujaran kebencian.

5. Kesimpulan dan Saran

Berdasarkan hasil penelitian, dataset telah dikumpulkan sebanyak 21.304 cuitan dengan menggunakan *library* Tweepy. Pembuatan model pembelajaran mesin untuk deteksi ujaran kebencian telah berhasil dibuat. Evaluasi model dengan melihat baik dari akurasi (F-score) dan *confusion matrix*. Logistic Regression, dengan n-gram yang menggunakan char level, mendapatkan skor akurasi sebesar 98%. Sedangkan berdasarkan *confusion matrix*, kombinasi Bernoulli Naïve Bayes dan n-gram yang menggunakan *char* level, mampu mengenali sebanyak 30 ujaran kebencian.

Dataset ujaran kebencian masih dapat dikembangkan dengan tidak hanya menggunakan tagar '2019gantipresiden'. Ujaran kebencian yang ada dalam dataset dinilai masih sangat sedikit, yaitu hanya sejumlah 383. Jumlah cuitan dalam dataset perlu ditambah untuk menghasilkan model yang lebih 'pintar' dan memiliki akurasi yang lebih tinggi. Model dalam penelitian ini juga masih dapat dikembangan sebagai *webservices.*

Referensi

[1] P. Fortuna, J. Ferreira, L. Pires, G. Routar, and S. Nunes, "Merging Datasets for Aggressive Text Identification," *Proc. First Work. Trolling, Aggress. Cyberbullying*, no. Section 2, pp. 39–50, 2018.

[2] A. Schmidt and M. Wiegand, "A Survey on Hate Speech Detection using Natural Language Processing," *Proc. Fifth Int. Work. Nat. Lang. Process. Soc. Media*, no. 2017, pp. 1–10, 2017.

[3] V. Golem, M. Karan, and J. Šnajder, "Combining Shallow and Deep Learning for Aggressive Text Detection," *Proc. First Work. Trolling, Aggress. Cyberbullying*, pp. 130–140, 2018.

[4] P. Parekh and Patel Hetal, "Toxic Comment Tools: A Case Study," *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 5, pp. 964–967, 2017.

[5] J. Pavlopoulos, P. Malakasiotis, and I. Androutsopoulos, "Deep Learning for User Comment Moderation," pp. 25–35, 2017.

[6] S. Sharma, S. Agrawal, and M. Shrivastava, "Degree based Classification of Harmful Speech using Twitter Data," pp. 106–112, 2018.

[7] I. Arroyo-Fernández, D. Forest, J.-M. Torres-Moreno, M. Carrasco-Ruiz, T. Legeleux, and K. Joannette, "Cyberbullying Detection Task: the EBSI-LIA-UNAM System (ELU) at COLING'18 TRAC-1," *Proc. First Work. Trolling, Aggress. Cyberbullying*, pp. 51–60, 2018.

[8] N. S. Samghabadi, D. Mave, S. Kar, and T. Solorio, "RiTUAL-UH at TRAC 2018 Shared Task: Aggression Identification," pp. 12–18, 2018.

[9] F. Del Vigna, A. Cimino, F. Dell'Orletta, M. Petrocchi, and M. Tesconi, "Hate me, hate me not: Hate speech detection on Facebook," *CEUR Workshop Proc.*, vol. 1816, pp. 86–95, 2017.

[10] G. K. Pitsilis, H. Ramampiaro, and H. Langseth, "Detecting Offensive Language in Tweets Using Deep Learning," pp. 1–17, 2018.

[11] J. Bell, *Machine Learning: Hands-On for Developers and Technical Professionals*. Wiley, 2014.

[12] Komisi Nasional HAM, *BUKU SAKU PENANGANAN UJARAN KEBENCIAN (HATE SPEECH)*. Jakarta, 2015.

[13] D. Sarkar, *Text Analytics with Python: A Practical Real-World Approach to Gaining Actionable Insights from Your Data*, 1st ed. California: Apress, 2016.