

Aplikasi Pendeteksi Unsur Hinaan dalam Komentar di Media Sosial Berbahasa Indonesia

Yohanes Adam Sastrodikoro W.¹, Henry Novianus Palit², Justinus Andjarwirawan³

Program Studi Teknik Informatika, Fakultas Teknik Industri, Universitas Kristen Petra

Jl. Siwalankerto 121-131 Surabaya 60236

Telp. (031)-2983455, Fax. (031)-8417658

E-Mail: yohanesadams@gmail.com¹, hnpalit@petra.ac.id², justin@petra.ac.id³

ABSTRAK

Komunikasi di media sosial sudah menjadi bagian dari gaya hidup masyarakat Indonesia. Sebagian besar bentuk dari komunikasi antar pengguna adalah berbentuk teks komentar. Seringkali terdapat komentar yang mengandung unsur hinaan yang ditujukan pada orang lain. Perlu pendeteksian unsur hinaan dalam teks komentar di media sosial.

Langkah pertama dalam metode tersebut adalah melakukan normalisasi terhadap teks sehingga teks komentar lebih teratur dan memiliki standar. Penggantian variasi kata hinaan menjadi kata hinaan aslinya juga dilakukan. Kemudian dilakukan proses *stemming* untuk mengambil kata dasar tiap kata. Dengan melakukan *feature extraction* terhadap teks tersebut, didapatkan setiap fitur dari teks tersebut kemudian diprediksi menggunakan metode SVM, *Logistic Regression*, dan *Naive Bayes*.

Aplikasi yang digunakan pada karya tulis ilmiah ini berhasil mendeteksi unsur hinaan dalam teks komentar dari media sosial menggunakan ketiga metode klasifikasi SVM, *Logistic Regression*, dan *Naive Bayes*.

Kata Kunci: teks, komentar media sosial, unsur hinaan

ABSTRACT

Communication in social media has become a lifestyle of Indonesian people. Most of the kind of communication between users are commentary text. Often there are comments that contain elements of insults directed at others. Detection of insult elements in comment text in social media is needed.

The first step in the method is to normalize the text so that the comment text is more organized and standardized. The replacement of the insult verbal variation into the original insult word is also done. Then do the stemming process to take the basic word of each word. By performing feature extraction of the text, obtained every feature of the text is then predicted using the SVM method.

The application used in this paper successfully detected insulting elements in the text of comments from social media using the three methods of SVM classification, Logistic Regression, and Naive Bayes.

Keywords: texts, social media comments, insult elements

1. PENDAHULUAN

Media sosial telah menjadi bagian dari gaya hidup masyarakat Indonesia. Dua dari tiga situs teratas yang paling sering dikunjungi oleh masyarakat Indonesia adalah media sosial. Media sosial menjadi sarana yang baik bagi masyarakat untuk berkomunikasi

dengan orang lain dengan cara yang beragam dan tidak terbatas akan jarak.

Pada kenyataannya, interaksi dalam media sosial seringkali mengandung komentar yang mengandung hinaan atau nistaan untuk pihak lain. Komentar hinaan tersebut mengandung kata-kata yang tidak pantas dan mengakibatkan interaksi di media sosial tidak sehat. Komentar hinaan tersebut seringkali berdampak buruk bagi pihak yang dihina maupun pihak yang memberi hinaan. Sebagian besar komentar pada media sosial adalah berbentuk teks.

Untuk menjawab permasalahan yang ada, diperlukan aplikasi yang dapat mengkategorikan komentar dari media sosial dalam bentuk teks untuk menentukan bahwa komentar tersebut mengandung unsur hinaan atau tidak. Komentar yang hendak disimpan pada sistem perlu dikategorikan sebagai komentar mengandung hinaan atau tidak terlebih dahulu sebagai tindakan preventif. Diharapkan dengan adanya aplikasi pendeteksi komentar hinaan dapat membantu instansi ataupun perusahaan yang memiliki website / aplikasi yang melibatkan interaksi banyak user dalam bentuk teks komentar sehingga interaksi berjalan dengan sehat tanpa hinaan.

2. LANDASAN TEORI

2.1 Support Vector Machine

Support Vector Machine (SVM) adalah teknik pembelajaran yang diawasi dari bidang machine learning yang berlaku baik untuk klasifikasi maupun regresi. SVM berakar pada Statistical Learning Theory yang dikembangkan oleh Vladimir Vapnik dan rekan kerja di AT & T Bell Laboratories pada tahun 1995. Konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari hyperplane terbaik yang berfungsi sebagai pemisah dua buah class yang menjadi input. Hyperplane merupakan sebuah bidang yang membagi ruang vector dalam dua bagian, yang masing – masing berkorespondensi pada class yang berbeda. SVM didasarkan pada prinsip Structural Risk Minimization (SRM). Prinsip dasar SVM adalah linear classifier, dan selanjutnya dikembangkan agar dapat bekerja pada problem non-linear [9].

Pendekatan menggunakan jaringan syaraf tradisional mengalami kesulitan dalam generalisasi, menghasilkan model yang cenderung overfit terhadap data tertentu. Ini adalah konsekuensi dari algoritma optimasi yang digunakan untuk pemilihan parameter dan ukuran statistik yang digunakan untuk memilih model 'terbaik'. Dasar-dasar Support Vector Machine telah dikembangkan oleh Vapnik [10] dan mendapatkan popularitas karena banyak fitur menarik dan menjanjikan kinerja empiris. Perumusan ini mewujudkan prinsip Structural Risk Minimization (SRM) yang telah terbukti lebih unggul [2] terhadap prinsip Empirical Risk Minimization (ERM) tradisional yang dipekerjakan oleh jaringan syaraf konvensional. SRM meminimalkan batas atas pada risiko yang diharapkan,

berlawanan dengan ERM yang meminimalkan kesalahan pada data pelatihan. Perbedaan inilah yang melengkapi SVM dengan kemampuan yang lebih baik untuk pembelajaran yang bersifat general yang merupakan tujuan dalam pembelajaran statistik.

2.2 Logistic Regression

Logistic Regression adalah metode statistik untuk menganalisa dataset dimana ada satu atau lebih variabel independen yang menentukan suatu hasil [6]. Hasilnya diukur dengan variabel dikotomis (hanya ada dua kemungkinan hasil). Tujuan dari logistic regression adalah untuk mencari model yang paling cocok untuk mendeskripsikan hubungan antara variabel dependen (respon atau hasil) dengan variabel independen (prediksi). Logistic regression menghasilkan koefisien dari sebuah rumus untuk memprediksi logit transformation dari probabilitas karakteristik yang diminati.

2.3 Naive Bayes

Naive Bayes (NB) adalah algoritma pembelajaran sederhana menggunakan aturan Bayes bersama dengan asumsi bahwa tiap atribut bersifat kondisional independen. Meski asumsi independensi ini dalam praktiknya sering dilanggar, Naive Bayes tetap mampu memberikan akurasi klasifikasi yang kompetitif. Dengan kemampuan Naive Bayes untuk memberikan efisiensi komputasi dan banyak fitur lain yang diinginkan, Naive Bayes diterapkan secara luas dalam praktik dunia nyata [9].

Naive Bayes menyediakan mekanisme untuk menggunakan informasi dalam data sampel untuk memperkirakan probabilitas posterior $P(y | x)$ pada setiap kelas y , dengan sebuah objek x . Dengan probabilitas tersebut, NB dapat digunakan untuk klasifikasi atau aplikasi pendukung keputusan lainnya. Untuk data sampel yang kecil, Naive Bayes dapat mengungguli metode lain yang lebih kompleks [1].

2.4 Feature Extraction

Data teks mentah tidak dapat langsung diteruskan ke algoritma klasifikasi untuk diproses karena kebanyakan dari algoritma klasifikasi teks mengharapkan input berupa fitur numerical vector dengan ukuran yang baku ketimbang teks mentah dengan panjang yang beragam.

- *tokenizing* string dan memberikan identitas integer untuk setiap token yang ada
- *counting* atau menghitung kemunculan token di setiap dokumen
- normalisasi dan pembobotan token di setiap dokumen

Sebuah korpus dokumen dapat ditunjukkan oleh matriks dengan satu baris per teks dan satu kolom per token yang terdapat dalam teks. Vectorization dilakukan untuk mengubah kumpulan dokumen teks menjadi vektor fitur numerik. Proses ini (tokenizing, counting dan normalisasi) disebut sebagai Bag of Words atau "Bag of n-gram".

2.5 tf-idf Weighting

Tf adalah singkatan dari term frequency atau frekuensi istilah. Nilai tf didapatkan dengan cara membagi berapa kali sebuah istilah ditemukan dalam dokumen dengan jumlah total istilah dalam dokumen. Pembagian ini dengan panjang dokumen dapat mencegah bias terhadap dokumen yang lebih panjang dengan menormalisasi frekuensi dari istilah tersebut menjadi skala yang sebanding [4].

Idf merupakan singkatan dari inverse document frequency atau frekuensi dokumen invers. Hal ini dihitung dengan mengambil nilai logaritmik dari jumlah total dokumen di dalam korpus dibagi

dengan jumlah dokumen dimana istilah tersebut terjadi. Normalisasi ini adalah untuk menimbang-nimbang persyaratan langka di korpus.

Penggabungan kedua nilai tf dan idf menghasilkan nilai tf-idf untuk tiap kata atau istilah dalam suatu dokumen. Perhitungan tf-idf weighting untuk term atau istilah t yang terjadi dalam dokumen dapat dilihat dalam Rumus 1.

$$tfidf_{t,d} = tf_{t,d} \times idf_t \quad (1)$$

2.6 K-fold Cross Validation

Cross validation adalah sebuah prosedur untuk melakukan pengujian untuk memperoleh estimasi performa tidak hanya berdasarkan berdasarkan performa secara umum, namun juga rata-rata dan varian sehingga dapat diketahui bagaimana performa terhadap berbagai dataset [8]. *Cross validation* juga membuat penggunaan dataset terbatas lebih baik. Tidak seperti membelah data menjadi data testing dan data training, cross validation menghitung estimasi atas semua data dengan melakukan beberapa pemecahan dan secara sistematis menukar sampel untuk pengujian.

2.7 F-measure

F-measure biasa digunakan dalam information retrieval dimana recall adalah frekuensi dimana dokumen yang relevan diambil atau 'diingat' oleh sebuah sistem, namun diketahui di tempat lain sebagai Sensitivity atau True Positive Rate (TPR). Precision adalah frekuensi dimana dokumen atau prediksi yang diambil relevan atau 'benar', dan benar merupakan bentuk akurasi, juga dikenal sebagai Positive Predictive Value (PPV) atau True Positive Accuracy (TPA). F dimaksudkan untuk menggabungkannya menjadi satu ukuran 'keefektifan' pencarian [7]. Rumus perhitungan F-measure dapat dilihat pada Rumus 2.

$$F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (2)$$

3. METODE

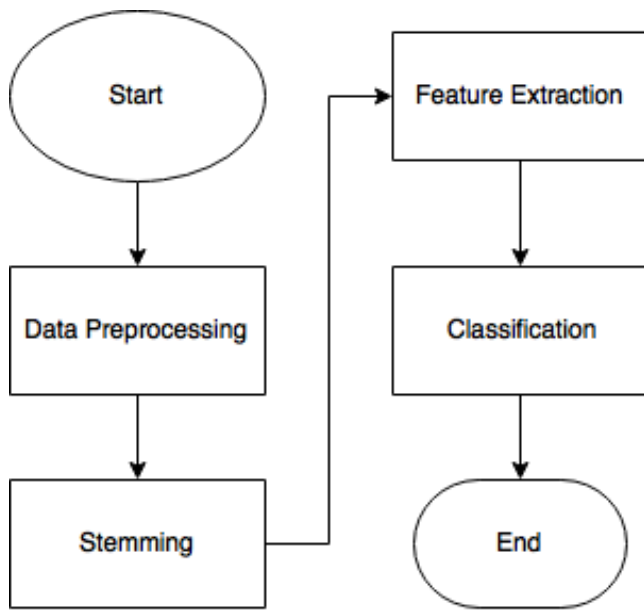
Garis besar proses yang dilakukan dapat dilihat pada Gambar 1. Terdapat empat proses utama yang terdiri dari *data preprocessing*, *stemming*, *feature extraction*, dan *classification*.

3.1 Data Preprocessing

Pada tahap ini, pertama dilakukan penghapusan tanda baca terhadap data teks yang menjadi input. Dilakukan perubahan setiap huruf kapital menjadi huruf kecil. Data komentar dari media sosial yang identik dengan kata-kata yang tidak baku diubah menjadi kata yang baku berdasarkan list *abbreviation* yang telah dibuat sebelumnya. Selanjutnya dilakukan proses *badwords replacement* dimana setiap variasi dari kata hinaan yang ditemukan berdasarkan list *badwords* yang telah dibuat sebelumnya dengan kata hinaan aslinya.

3.2 Stemming

Stemming digunakan untuk mengubah kata-kata ke bentuk akar (stem). Proses ini digunakan dalam tahap *pre processing* sistem pengambilan informasi. *Stemming* mempengaruhi waktu pengindeksan dengan mengurangi ukuran file indeks dan meningkatkan kinerja proses pengambilan [3].



Gambar 1. Diagram Alur Pendeteksian Unsur Hinaan dalam Komentar Berbahasa Indonesia

3.3 Feature Extraction

Proses feature extraction melakukan tokenizing, counting, dan normalisasi pada data komentar karena data komentar mentah tidak dapat langsung diteruskan ke algoritma klasifikasi karena kebanyakan dari algoritma klasifikasi teks mengharapkan input berupa fitur numerical vector dengan ukuran yang baku ketimbang teks mentah dengan panjang yang beragam. Proses ini menghasilkan nilai numerical vector dari tiap teks yang diteruskan ke proses training tiap algoritma klasifikasi. Teknik *feature extraction* yang digunakan adalah *unigram-bigram feature extraction* dimana setiap fitur berupa satu kata atau pasangan kata sehingga pola dalam kalimat dapat dikenali oleh *classifier*.

3.4 Classification

Proses *classification* dilakukan dengan menggunakan tiga metode klasifikasi yaitu SVM, *Logistic Regression*, dan *Naive Bayes*. Hasil prediksi berupa 1 (positif mengandung unsur hinaan) dan 0 (negatif mengandung unsur hinaan). Dilakukan *voting* terhadap hasil klasifikasi tiap-tiap metode dengan cara *best-of-three* untuk mendapatkan kesimpulan akhir bahwa kalimat input mengandung unsur hinaan atau tidak.

4. IMPLEMENTASI

Program yang digunakan untuk menguji metode diimplementasikan dengan menggunakan bahasa pemrograman Python 2.7 dan PHP. *Library* yang digunakan adalah *scikit-learn* (klasifikasi dan *feature extraction*), *pandas* (pembacaan file) *matplotlib* (visualisasi data), *pickle* (menyimpan obyek sebagai file), dan *pySastrawi* (*stemming*).

Percobaan dilakukan terhadap empat data yaitu komentar yang memiliki kata hinaan dan positif memiliki unsur hinaan dan komentar yang memiliki kata hinaan dan negatif memiliki unsur hinaan. Setiap komentar diinputkan dan diproses melalui service yang mengembalikan data dalam bentuk JSON.

Data yang dikembalikan oleh JSON terdiri dari lima data, yaitu prediksi SVM, prediksi Logistic Regression, prediksi Naive Bayes,

kesimpulan prediksi, dan waktu eksekusi yang diperlukan untuk melakukan deteksi pada suatu komentar tersebut.

Hasil implementasi program terhadap komentar yang memiliki kata hinaan dan positif memiliki unsur hinaan dapat dilihat pada Tabel 1, sedangkan hasil implementasi program terhadap komentar yang memiliki kata hinaan dan negatif memiliki unsur hinaan dapat dilihat pada Tabel 2.

Tabel 1. Hasil Implementasi Terhadap Komentar yang Memiliki Kata Hinaan dan Positif Memiliki Unsur Hinaan

| No | Teks Komentar | Hasil |
|----|--|---|
| 1 | emang babi lu | SVM : 1 Logistic Regression : 1 Naive Bayes : 1 Kesimpulan : 1 (Positif) |
| 2 | dasar loe anjing keparat | SVM : 1 Logistic Regression : 1 Naive Bayes : 1 Kesimpulan : 1 (Positif) |
| 3 | bapak anak sekeluarga sama2 kek anjing | SVM : 0 Logistic Regression : 0 Naive Bayes : 1 Kesimpulan : 0 (Negatif) |
| 4 | sok tau lo kntl | SVM : 1 Logistic Regression : 1 Naive Bayes : 1 Kesimpulan : 1 (Positif) |

Tabel 2. Hasil Implementasi Terhadap Komentar yang Memiliki Kata Hinaan dan Negatif Memiliki Unsur Hinaan

| No | Teks Komentar | Hasil |
|----|--|---|
| 1 | lucu banget jadi pengen plihara anjing | SVM : 0 Logistic Regression : 0 Naive Bayes : 1 Kesimpulan : 0 (Negatif) |
| 2 | angen babi kecap masakan mama gw | SVM : 0 Logistic Regression : 0 Naive Bayes : 0 Kesimpulan : 0 (Negatif) |
| 3 | manteb bener neh nasi goreng jancuk | SVM : 0 Logistic Regression : 0 Naive Bayes : 1 Kesimpulan : 0 (Negatif) |
| 4 | monyet, anjing, babi, dan jangkrik adalah nama-nama binatang | SVM : 0 Logistic Regression : 0 Naive Bayes : 0 Kesimpulan : 0 (Negatif) |

5. KESIMPULAN

Dari percobaan yang telah dilakukan dapat disimpulkan bahwa metode yang digunakan pada artikel ilmiah ini mampu mendeteksi unsur hinaan dalam komentar dari media sosial berbahasa Indonesia.

Metode ini bergantung pada data *training* dan list *badwords* yang dapat ditambah dan diperbaiki untuk mendapatkan akurasi yang lebih tinggi dan membuat program mengenali kata hinaan yang lebih beragam.

6. REFERENCES

- [1] Domingos, P. dan Pazzani, M. 1997. *On the optimality of the simple bayesian classifier under zero-one loss*. *Machine Learning*, 29 (2-3), 103-130.
- [2] Gunn, S., Brown, M., dan Bossley, K. 1997. *Network performance assessment for neuro- fuzzy data modelling*. *Advances in Intelligent Data Analysis Reasoning about Data* (pp. 313-323). Berlin: Springer.
- [3] Hajeer, S.I., Ismail, R.M., Badr, N.L., dan Tolba, M.F. 2017. *A New Stemming Algorithm for Efficient Information Retrieval Systems and Web Search Engines*. In: Hassanien A., Mostafa Fouad M., Manaf A., Zamani M., Ahmad R., Kacprzyk J. (eds) *Multimedia Forensics and Security. Intelligent Systems Reference Library*, vol 115. Springer, Cham.
- [4] Manning, D. M., Raghavan, P., dan Schütze, H. 2008. *Introduction to Information Retrieval*. Cambridge, United Kingdom: Cambridge University Press.
- [5] Nugroho, A., Witarto, A., dan Handoko, D. 2003. *Support Vector Machine Teori dan Aplikasinya dalam Bioinformatika*. Kuliah Umum IlmuKomputer.Com. URI=<http://asnugroho.net/papers/ikcsvm.pdf>.
- [6] Osborne, J. W. 2015. *Best Practices in Logistic Regression*. Thousand Oaks: SAGE Publications.
- [7] Powers, D.M.W. 2011. *Evaluation from precisions, recall, and F-measure to ROC, informedness, markedness and correlation*. *International Journal of Machine Learning Technology*, vol 2, 37-63.
- [8] Provost, F., & Fawcett, T. 2013. *Data Science for Business*. Sebastopol: O'Reilly Media, Inc.
- [9] Sammut, C., dan Webb, G. I. 2017. *Encyclopedia of Machine Learning and Data Mining*. New York: Springer US.
- [10] Vapnik, V. 1995. *The Nature of Statistical Learning Theory*. New York: Springer