Jurnal Linguistik Komputasional
Indonesian Association for Computational Linguistics

JLK
Vol 2, No 1
Maret 2019

Identifikasi Konten Kasar pada *Tweet* Bahasa Indonesia

Ahmad Fathan Hidayatullah[1], Aufa Aulia Fadila Yusuf[2], Kiki Purnama Juwairi[3], Royan Abida N. Nayoan[4]

Program Studi Sarjana Teknik Informatika, Universitas Islam Indonesia
Jl. Kaliurang, km 14,5, Sleman, Yogyakarta, Indonesia
[1]fathan@uii.ac.id, [2]15523070@students.uii.ac.id, [3]15523234@students.uii.ac.id, [4]15523084@students.uii.ac.id

Abstract*— This study aims to identify tweets containing abusive or offensive content. To do this, we performed five steps, such as, data collection, preprocessing, feature extraction, classification, and evaluation. We employed Multinomial Naïve Bayes and Support Vector Machine with linear kernel as our classification algorithm. Based on the experiment, it is known that the performance of the Support Vector Machine algorithm with linear kernel is superior overall compared to the Multinomial Naïve Bayes algorithm. It can be seen from the result of the values of accuracy, precision, recall, and F1-score for the SVM algorithm, respectively 0.9928; 0.9914; 0.9946; and 0.9930. Whereas the value of accuracy, precision, recall, and F1-score of the Multinomial Naïve Bayes algorithm are 0.9834; 0.9912; 0.9762; and 0.9836. However, it can be concluded that the Support Vector Machine and Multinomial Naïve Bayes algorithm have almost the same performance. This is evidenced by the difference in performance achievements that are not too striking from both algorithm.

***Keywords*— Twitter, classificaion, Multinomial Naïve Bayes, Support Vector Machine**

Abstrak*—Penelitian ini bertujuan untuk melakukan identifikasi *tweet* yang mengandung konten kasar atau ofensif. Untuk melakukan hal tersebut, ada lima tahap yang dilalui yaitu pengumpulan data, *preprocessing*, ekstraksi fitur, klasifikasi, dan evaluasi. Adapun algoritma klasifikasi yang digunakan adalah *Multinomial Naïve Bayes* dan *Support Vector Machine* dengan *linear kernel*. Berdasarkan eksperimen, diketahui bahwa performa algoritma *Support Vector Machine* dengan *linear kernel* lebih unggul secara keseluruhan dibandingkan dengan algoritma *Multinomial Naïve Bayes*. Hal tersebut dilihat dari perolehan nilai *accuracy, precision, recall, dan F1-score* untuk algoritma SVM berturut-turut adalah 0.9928; 0.9914; 0.9946; dan 0.9930. Sedangkan perolehan *accuracy, precision, recall, dan F1-score* algoritma *Multinomial Naïve Bayes* berturut-turut adalah 0.9834; 0.9912; 0.9762; dan 0.9836. Namun demikian, dapat disimpulkan bahwa algoritma *Support Vector Machine* dan *Multinomial Naïve Bayes* memiliki performa yang hampir sama baiknya. Hal tersebut dibuktikan dengan selisih capaian performa yang tidak terlalu mencolok dari keduanya.

***Kata kunci*— Twitter, klasifikasi, Multinomial Naïve Bayes, Support Vector Machine**

I. PENDAHULUAN

Twitter merupakan salah satu media jejaring sosial yang cukup populer di Indonesia. Keberadaan Twitter di Indonesia telah digunakan oleh berbagai macam instansi, organisasi, atau perseorangan. Institusi dan organisasi banyak memanfaatkan Twitter sebagai media untuk memberikan informasi penting kepada masyarakat luas. Adapun untuk perseorangan, seperti tokoh publik, politisi, artis, seniman, ataupun masyarakat biasa, Twitter merupakan sarana pribadi untuk berekspresi, mencurahkan pendapat, mengungkapkan isi hati, dan sebagainya.

Namun demikian, tidak semua individu pengguna Twitter bersikap bijak dalam memilih kata-kata dalam cuitannya. Tidak sedikit *netizen* menuliskan kata-kata yang berbau SARA(Suku, Agama, Ras, dan Antar golongan) atau bahkan mengungkapkan ekspresi dengan menuliskan kata-kata kasar dan bersifat ofensif. Kata-kata kasar dalam bahasa Indonesia biasanya diucapkan atau dituliskan untuk menyerang pihak tertentu, mengungkapkan kekesalah, kekecewaan, atau meluapkan emosi terhadap peristiwa tertentu.

Dalam bahasa Indonesia, kata kasar dapat diungkapkan salah satunya dengan menyebutkan jenis hewan tertentu, seperti anjing, monyet, dan sebagainya. Namun demikian, tidak semua kalimat yang memuat jenis hewan seperti contoh tersebut merupakan bentuk kalimat yang bersifat ofensif. Oleh karena itu, untuk mengidentifikasi apakah suatu kata dianggap sebagai kata yang kasar, maka perlu melihat konteks kalimat secara menyeluruh.

Berbagai penelitian terdahulu telah banyak membahas tentang identifikasi konten kasar dan sejenisnya pada data teks. Razavi, et al. [1] melakukan identifikasi bahasa yang bersifat ofensif dari data pesan berbahasa Inggris dengan pendekatan *multi-level classification*. Konten ofensif pada pesan tersebut didefinisikan sebagai konten yang bertujuan untuk menyerang pihak lain, mengandung kata, frasa, atau bahasa kasar yang mengajak kepada permusuhan. Bretschneider dan Peters [2] menggunakan

Submitted 20-01-2019; Revised 06-03-2019; Accepted 25-03-2019

pendekatan *machine learning* dengan *tools* RapidMiner untuk melakukan klasifikasi kalimat ofensif dalam Bahasa Jerman di sosial media. Chen, et al [3] juga meneliti kemunculan konten ofensif pada data sosial media. Arsitektur *Lexical Syntactic Feature* (LSF) digunakan pada penelitian tersebut untuk mendeteksi bahasa ofensif dalam Bahasa Inggris. Hasil penelitian menunjukkan bahwa metode LSF mengungguli metode *machine learning* tradisional lainnya dalam nilai *precision, recall,* dan *f-score.*

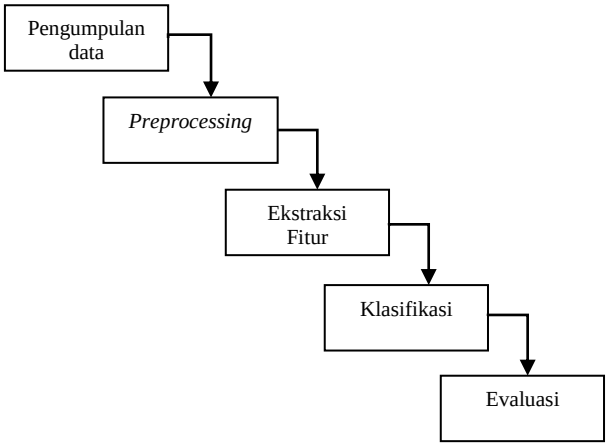
Mubarak et al. [4] melakukan identifikasi bahasa kasar pada sosial media Twitter berbahasa Arab dengan melakukan klasifikasi *tweet* ke dalam tiga kelas, yaitu *tweet* yang memuat konten cabul, ofensif, dan tidak memuat kedua konten cabul atau ofensif. Data Twitter juga digunakan oleh Malmasi dan Zampieri [5]. Penelitian tersebut melakukan pengenalan kalimat ujaran kebencian pada data Twitter dengan mengklasifikasikan *tweet* ke dalam tiga kelas, yaitu ujaran kebencian, kalimat ofensif, dan kalimat yang tidak memuat keduanya. Peneliti menerapkan metode *Support Vector Machine* dengan menggunakan tiga model ekstraksi fitur, di antaranya *n-grams, word skip-grams*, dan *Brown clusters*.

Berdasarkan beberapa penelitian di atas, belum ada penelitian yang secara spesifik melakukan identifikasi konten kasar berbahasa Indonesia pada Twitter. Oleh karena itu, penelitian ini bertujuan untuk melakukan identifikasi *tweet* berbahasa Indonesia yang memuat konten kata-kata kasar yang bersifat ofensif. Penelitian ini diharapkan memberikan kontribusi di dalam pembangunan *corpus* untuk identifikasi konten kasar pada data teks berbahasa Indonesia. Proses identifikasi konten kasar pada penelitian ini dilakukan dengan membangun model dari kumpulan *tweet* yang terdiri dari *tweet* bahasa Indonesia yang mengandung kata-kata kasar atau kotor serta *tweet* yang tidak memuat kata-kata kasar atau kotor. Selanjutnya dilakukan klasifikasi dengan *machine learning* yang dapat mengidentifikasi apakah suatu *tweet* yang mengandung konten kasar atau tidak. Penelitian ini membangun model dengan membandingkan antara dua buah metode *machine learning*, yaitu *Support Vector Machine* dan *Naïve Bayes* dengan fitur *unigram*.

Penulisan makalah ini terdiri dari lima bagian. Bagian pertama merupakan pendahuluan yang memuat latar belakang dari penelitian ini. Bagian kedua membahas tentang penelitian-penelitian sebelumnya yang terkait dan mendukung penelitian ini. Bagian ketiga berisi tentang metodologi dalam penelitian ini. Bagian keempat memaparkan hasil eksperimen dan pembahasan dari hasil yang diperoleh. Bagian kelima adalah bagian terakhir yang berisi kesimpulan dari penelitian.

II. METODOLOGI

Pada bagian ini, akan dibahas mengenai metodologi yang digunakan dalam penelitian. Proses identifikasi *tweet* yang memuat kata-kata kasar melalui beberapa tahap yang dijelaskan oleh gambar 1. Secara keseluruhan, ada lima tahap yang dilakukan yaitu pengumpulan data, *preprocessing*, ekstraksi fitur, klasifikasi, dan evaluasi. Masing-masing penjelasan secara rinci akan dijelaskan pada sub bab selanjutnya.



Gambar 1 Tahapan penelitian

A. Pengumpulan Data

Penelitian ini menggunakan data yang berasal dari Twitter dengan memanfaatkan Twitter API. Keseluruan *tweet* yang digunakan untuk membangun model pada penelitian ini adalah sebanyak 5462 *tweet* dengan cacah masing-masing *tweet* kasar dan tidak kasar yang berimbang, yaitu 2731 *tweet*. Untuk memperoleh data *tweet* yang mengandung kata-kata kasar, peneliti menentukan 15 kata kunci sebagai *query*, di antaranya anj*ng, b*ngs*t, bac*t, kuny*k, b*jing*n, bud*k, bol*t, kep*r*t, ta*, set*n, g*bl*k, tol*l, anj*r, br*ngs*k, dan sint*ng. Tabel I memperlihatkan beberapa contoh *tweet* kasar yang bersifat ofensif.

TABEL I
CONTOH *TWEET* KASAR

No	***Tweet***
1	@iKONICfess diemin aja , kalo makin bawel teriakin "bling bling bacot anjing bling bling bacot anjing"
2	@RajaPurwa Orang tua tailaso kau anjing, busyet, setan babi, berani nya ma anak kecil, banci kau babi
3	@fluidasae @budimandjatmiko @irvan_aria Hei kunyuk..loe yg bodoh...komen gak pake otak

Selain itu, penelitian ini juga mengumpulkan data *tweet* yang tidak memuat kata-kata kasar. *Tweet* yang tidak memuat konten kasar pada penelitian ini di antaranya diperoleh dengan mengambil *tweet* yang berisi berita dan informasi umum. Contoh *tweet* yang bebas dari kalimat yang bernada kasar diperlihatkan pada tabel II.

TABEL II
CONTOH *TWEET* TIDAK KASAR

No	*Tweet*
1	Makanan ringan apa yang nemenin kamu di liburan panjang ini? Ada info riset menarik nih☺
2	Kembangkan Sapi Madura Untuk Mendukung Swasembada Daging https://t.co/heak3OiSAK https://t.co/YQCC0CM4gG
3	3 Alasan Kebangkitan Paul Pogba di Manchester United https://t.co/kwrhv8OnNU

B. Preprocessing

Tahapan preprocessing yang dilakukan pada penelitian ini mengacu pada langkah *preprocessing* data Twitter yang telah dilakukan oleh Hidayatullah dan Ma'arif [6]. *Preprocessing* pada data teks bertujuan untuk membersihkan data yang tidak konsisten, tidak standar, atau data yang tidak sempurna. Adapun langkah *preprocessing* yang dilakukan di antaranya:

- Menghapus karakter non-ASCII.
- Menghapus URL.
- Menghapus karakter khusus yang ada pada Twitter diantaranya adalah *hashtag*, *username* dan RT.
- Menghapus simbol dan tanda baca.
- Menghapus angka.
- Menghapus kata yang hanya terdiri dari satu huruf.
- Penyeragaman huruf ke dalam bentuk *lowercase*.
- Menghapus spasi yang berlebihan.
- Menghapus *stopword*.

C. Ekstraksi Fitur

Penelitian ini menggunakan TF-IDF *(Term Frequency-Inverse Documen Frequency)* sebagai metode untuk melakukan ekstraksi fitur. TF-IDF merupakan kombinasi antara perhitungan *term frequency* dan IDF *(Inverse Document Frequency)* yang untuk menentukan bobot dari setiap token pada sekumpulan dokumen. Latar belakang dari konsep TF-IDF adalah untuk melihat seberapa penting keberadaan dari suatu token dalam suatu korpus. Adapun perhitungan nilai bobot kata dengan TF-IDF diperoleh dari persamaan (1) berikut:

$$w_{t,d} = tf_{t,d}\ x\ log\frac{N}{df_t} \qquad (1)$$

Di mana:

$tf_{t,d}$ = cacah kemunculan dari token t dalam dokumen d

df_t = cacah dokumen yang memuat token t

N = cacah total dokumen

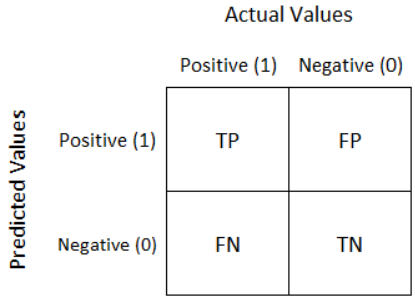
D. Model Klasifikasi

Ada dua buah pendekatan yang dipilih untuk melakukan klasifikasi pada penelitian ini, yaitu *Naïve Bayes* dan *Support Vector Machine*. Alasan pemilihan *Naïve Bayes* dan SVM adalah performa yang baik dari keduanya di dalam melakukan proses klasifikasi teks pada penelitian-penelitian terdahulu [7][8][9].

Selanjutnya, metode *Naïve Bayes* yang digunakan untuk membangun model klasifikasi pada penelitian ini adalah *Multinomial Naïve Bayes*. Metode *Multinomial Naïve Bayes* dipilih dengan alasan bahwa algoritma tersebut cocok diterapkan untuk klasifikasi teks atau dokumen [10]. Sedangkan untuk membangun model klasifikasi dengan algoritma SVM, dilakukan dengan pendekatan *linear kernel*. Pemilihan *linear kernel* pada SVM dilakukan atas dasar performa yang baik dari *linear kernel* untuk klasifikasi teks [11][12].

E. Evaluasi

Pada penelitian ini, *confusion matrix* digunakan sebagai alat ukur performa klasifikasi dari kedua metode yang digunakan. *Confusion matrix* adalah matriks yang cukup intuitif dan mudah untuk mengetahui tingkat ketepatan dan akurasi dari model yang dihasilkan. Gambar 2 menunjukkan ilustrasi dari *confusion matrix*.



Gambar 2 *Confusion Matrix*

Berdasarkan *confusion matrix*, banyaknya total prediksi yang benar ditunjukkan oleh variabel TP *(True Positive)* dan TN *(True Negative)*. Sedangkan banyaknya prediksi yang salah ditunjukkan oleh variabel FP (*False Positive)* dan FN *(False Negative)*. Adapun perhitungan indikator performa diperoleh dengan menghitung nilai *accuracy*, *precision*, *recall*, dan *F1-Score*. yang masing-masing ditunjukkan oleh persamaan (2), (3), (4), dan (5) berikut:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \qquad (2)$$

$$Precision = \frac{TP}{TP + FP} \qquad (3)$$

$$Recall = \frac{TP}{TP + FN} \qquad (4)$$

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \qquad (5)$$

TP = banyaknya data dari kelas *positive* (1) yang benar diprediksi sebagai kelas *positive* (1).

TN = banyaknya data dari kelas *negative* (0) yang benar diprediksi sebagai kelas *negative* (0).

FP = banyaknya data dari kelas *negative* (0) yang salah diprediksi sebagai kelas *positive* (1).

FN = banyaknya data dari kelas *positive* (1) yang salah diprediksi sebagai kelas *negative* (0).

Perhitungan akurasi dilakukan dengan membandingkan antara total data yang diklasifikasikan benar dengan keseluruhan data. *Precision* menunjukkan tingkat kepastian dengan membandingkan antara sampel kelas positif yang diklasifikasikan dengan benar terhadap keseluruhan sampel kelas positif. *Recall* mengukur rasio antara sampel kelas positif yang diklasifikasikan dengan benar terhadap sampel kelas positif yang salah diklasifikasikan ke dalam kelas negatif. Adapun *F1-Score* merupakan *harmonic mean* dari *precision* dan *recall*.

III. HASIL DAN PEMBAHASAN

Bagian ini menjelaskan tentang hasil uji coba berdasarkan skenario yang telah ditentukan sebelumnya. Proses pembagian data *training* dan *testing* dilakukan menggunakan Metode Holdout [13]. Penelitian ini membagi data *training* dan *testing* dengan perbandingan 2/3 data untuk *training* dan 1/3 data untuk *testing*.

Tabel III dan IV memperlihatkan *confusion matrix* yang diperoleh dari data *testing* menggunakan algoritma *Multinomial Naïve Bayes* dan *Support Vector Machine*. Pada tabel III dan IV, kelas yang dianggap sebagai sebagai *tweet* yang kasar dilabeli dengan 0 (nol) dan kelas *tweet* yang tidak kasar diberi label 1 (satu).

TABEL III
HASIL *CONFUSION MATRIX* DENGAN *MULTINOMIAL NAÏVE BAYES*

Multinomial Naïve Bayes		***Actual Values***	
		0	**1**
Predicted Values	**0**	871	8
	1	22	902

TABEL IV
HASIL *CONFUSION MATRIX* DENGAN *SUPPORT VECTOR MACHINE*

Support Vector Machine		***Actual Values***	
		0	**1**
Predicted Values	**0**	871	8
	1	5	919

Tabel III memperlihatkan nilai indikator performa yaitu *accuracy, precision, recall,* dan *F1-Score* yang diperoleh berdasarkan dari *confusion matrix* di atas. Secara keseluruhan, dapat dilihat bahwa algoritma *Support Vector Machine* dengan *kernel* linier mengungguli performa dari algoritma *Multinomial Naïve Bayes*. Selain itu, terlihat bahwa kedua algoritma, yaitu *Multinomial Naïve Bayes* dan *Support Vector Machine*, memiliki performa yang sangat baik di dalam melakukan klasifikasi teks.

TABEL III
HASIL *ACCURACY, PRECISION, RECALL, DAN F1-SCORE*

Algoritma	***Accuracy***	***Precision***	***Recall***	***F1-Score***
Multinomial Naïve Bayes	0.9834	0.9912	0.9762	0.9836
Support Vector Machine	0.9928	0.9914	0.9946	0.9930

Berdasarkan hasil akurasi, nilai akurasi dengan algoritma SVM mencapai 0.9928. Perolehan tersebut lebih unggul tipis dibandingkan dengan *Multinomial Naïve Bayes* dengan akurasi sebesar 0.9834. Hal ini menggambarkan bahwa SVM memiliki tingkat ketepatan yang sedikit lebih baik daripada *Multinomial Naïve Bayes* di dalam melakukan prediksi.

Perbandingan nilai presisi di antara kedua algoritma tidak terlalu mencolok. *Support Vector Machine* lebih unggul sebanyak 0.0002 dibandingkan nilai presisi yang diperoleh algoritma *Multinomial Naïve Bayes*. Hal ini menunjukkan bahwa kedua algoritma memiliki kemampuan yang sama baiknya di dalam ketepatan prediksi.

Berdasarkan *recall* yang diperoleh dari kedua algoritma, SVM kembali mengungguli *Multinomial Naïve Bayes*. Capaian *recall* untuk SVM adalah 0.9946, sedangkan nilai recall untuk *Multinomial Naïve Bayes* yaitu sebesar 0.9762. Hal tersebut menggambarkan bahwa SVM memiliki kemampuan lebih baik untuk menemukan data yang relevan.

Selanjutnya, perbandingan perolehan nilai *F1-Score* antara kedua algoritma juga tidak terlalu signifikan. Dalam hal ini, SVM masih lebih unggul dibandingkan dengan *Multinomial Naïve Bayes*, meskipun dengan selisih yang tidak terlalu jauh, yaitu sebesar 0.0094.

IV. KESIMPULAN

Penelitian ini telah berhasil melakukan identifikasi konten kotor yang bersifat ofensif pada data Twitter. Proses identifikasi dilakukan dengan mengklasifikasikan antara *tweet* yang mengandung kata-kata kasar yang ofensif dengan *tweet* yang tidak memuat konten kasar serta ofensif.

Berdasarkan eksperimen, diketahui bahwa performa algoritma *Support Vector Machine* dengan *linear kernel* lebih unggul secara keseluruhan dibandingkan dengan algoritma *Multinomial Naïve Bayes*. Hal tersebut dilihat dari perolehan nilai *accuracy, precision, recall, dan F1-score* untuk algoritma SVM berturut-turut adalah 0.9928; 0.9914; 0.9946; dan 0.9930. Sedangkan perolehan *accuracy, precision, recall, dan F1-score* algoritma *Multinomial Naïve Bayes* berturut-turut adalah 0.9834; 0.9912; 0.9762; dan 0.9836.

Namun demikian, dapat disimpulkan bahwa algoritma *Support Vector Machine* dan *Multinomial Naïve Bayes* memiliki performa yang hampir sama baiknya. Hal

tersebut dibuktikan dengan selisih capaian performa yang tidak terlalu mencolok dari keduanya. Selain itu, keduanya merupakan algoritma yang cukup menjanjikan untuk digunakan sebagai *machine learning* dalam melakukan klasifikasi teks.

Pada pengembangan selanjutnya, kata-kata kasar dapat dikategorikan secara lebih rinci sehingga diperoleh ragam informasi kalimat ofensif yang lebih bervariasi. Selain itu, dapat dikembangkan sebuah sistem yang dapat menghapus dan melakukan *filtering* terhadap kemunculan kata kotor, kata kasar, dan kata-kata ofensif yang telah berhasil teridentifikasi di dalam suatu teks.

REFERENSI

[1] A. H. Razavi, D. Inkpen, S. Uritsky, and S. Matwin, "Offensive Language Detection Using Multi-level Classification," in *Advances in Artificial Intelligence*, vol. 6085, A. Farzindar and V. Kešelj, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 16–27.

[2] U. Bretschneider and R. Peters, "Detecting Offensive Statements towards Foreigners in Social Media," p. 10.

[3] Y. Chen, Y. Zhou, S. Zhu, and H. Xu, "Detecting Offensive Language in Social Media to Protect Adolescent Online Safety," in *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, Amsterdam, Netherlands, 2012, pp. 71–80.

[4] H. Mubarak, K. Darwish, and W. Magdy, "Abusive Language Detection on Arabic Social Media," in *Proceedings of the First Workshop on Abusive Language Online*, Vancouver, BC, Canada, 2017, pp. 52–56.

[5] S. Malmasi and M. Zampieri, "Detecting Hate Speech in Social Media," *ArXiv171206427 Cs*, Dec. 2017.

[6] A. F. Hidayatullah and M. R. Ma'arif, "Pre-processing Tasks in Indonesian Twitter Messages," *J. Phys. Conf. Ser.*, vol. 801, p. 012072, Jan. 2017.

[7] P.-Y. Zhang, "A HowNet-Based Semantic Relatedness Kernel for Text Classification," *TELKOMNIKA Indones. J. Electr. Eng.*, vol. 11, no. 4, Apr. 2013.

[8] D. Li-guo, D. Peng, and L. Ai-ping, "A New Naive Bayes Text Classification Algorithm," *TELKOMNIKA Indones. J. Electr. Eng.*, vol. 12, no. 2, Feb. 2014.

[9] A. F. Hidayatullah and M. R. Ma'arif, "Penerapan Text Mining dalam Klasifikasi Judul Skripsi," p. 4, 2016.

[10] D. H. Kalokasari, I. M. Shofi, and A. H. Setyaningrum, "Implementasi Algoritma Multinomial Naive Bayes Classifier Pada Sistem Klasifikasi Surat Keluar (Studi Kasus : Diskominfo Kabupaten Tangerang)," *J. Tek. Inform.*, vol. 10, no. 2, Oct. 2017.

[11] I. M. Yulietha and S. A. Faraby, "Klasifikasi Sentimen Review Film Menggunakan Algoritma Support Vector Machine," p. 11.

[12] T. Joachims, "Text categorization with Support Vector Machines: Learning with many relevant features," in *Machine Learning: ECML-98*, vol. 1398, C. Nédellec and C. Rouveirol, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 137–142.

[13] A. F. Hidayatullah and A. Sn, "Analisis Sentimen Dan Klasifikasi Kategori Terhadap Tokoh Publik Pada Twitter," p. 8, 2014.