

Pengaruh *Phrase Detection* dengan *POS-Tagger* terhadap Akurasi Klasifikasi Sentimen menggunakan SVM

Hermawan Arief Putranto¹, Onny Setyawati², Wijono³

Abstract--Sentiment analysis or opinion mining, which is one of the application of Natural Language Processing (NLP), aims to find a method to facilitate human in communicating with a computer using their common language. To simplify the process of understanding human language, there are three important stages that must be carried out by a computer, which are tokenizing, stemming and filtering. The tokenizing that breaks down the sentence into a single word will make the computer assume all words (token) are the same. If there is a phrase formed from one of unimportant words, which is happened to be in the stoplist, the phrase will be deleted. Solution for the aforementioned problem is tokenizing based on phrase detection using Hidden Markov Model (HMM) POS-Tagger to improve classification performance using Support Vector Machine (SVM). With this approach, computer will be able to distinguish a phrase from others, then store the phrase into a single entity. There is an increase in accuracy by approximately 6% on Dataset I and 3% on Dataset II in the classification process using phrase detection, due to reduction of missing features that usually occurs in the filtering process. In addition, the detection of the phrase-based approach also produces the most optimal classification model, as seen from the ROC value that reaches 0.897.

Intisari--Analisis sentimen atau *opinion mining*, yang merupakan penerapan dari cabang ilmu Pemrosesan Bahasa Alamiah (PBA), bertujuan untuk menemukan cara agar manusia bisa berkomunikasi dengan komputer menggunakan bahasa sehari-hari. Untuk memudahkan proses pemahaman bahasa manusia, ada tiga tahapan penting yang harus dilakukan oleh komputer, yaitu *tokenizing*, *stemming*, dan *filtering*. Proses *tokenizing* yang memecah kalimat menjadi kata tunggal membuat komputer menganggap semua kata sama. Masalah yang timbul adalah apabila ada frasa yang dibentuk dari salah satu kata yang tidak penting yang masuk dalam daftar *stoplist*, maka frasa tersebut akan terhapus. Solusi yang ditawarkan dalam makalah ini adalah proses tokenisasi berbasis deteksi frasa menggunakan *Hidden Markov Model* (HMM) *POS-Tagger* untuk meningkatkan unjuk kerja klasifikasi menggunakan *Support Vector Machine* (SVM). Dengan pendekatan ini, komputer diharapkan mampu membedakan frasa dan bukan frasa, kemudian menyimpan frasa tersebut sebagai satu entitas. Terdapat peningkatan akurasi pada proses klasifikasi menggunakan pendekatan berbasis deteksi frasa, yaitu sekitar 6% pada *Dataset I* dan sekitar 3% pada *Dataset II*. Hal ini dibuktikan dengan banyaknya kalimat dan paragraf yang terklasifikasi dengan benar, sesuai dengan kelas sentimennya, baik pada *Dataset I* maupun pada *Dataset II*. Selain itu,

pendekatan berbasis deteksi frasa juga menghasilkan model klasifikasi yang paling optimal, terlihat dari ROC yang mencapai nilai 0,897.

Kata Kunci— analisis sentimen, deteksi frasa, *HMM POS-Tagger*, *ROC*, *Support Vector Machine*, tokenisasi.

I. PENDAHULUAN

A. Latar Belakang

Di era informasi seperti saat ini, kemampuan untuk mengolah data menjadi informasi merupakan komoditas yang sangat berharga bagi sebuah organisasi. Partai politik bisa mengetahui popularitas calon presiden yang diusungnya melalui kicauan para pengguna Twitter [1]. Selain itu, dengan banyaknya aplikasi *microblogging* dan media sosial, hasil analisis dari tulisan atau komentar seorang konsumen bisa digunakan sebagai pendukung manajemen merek dan *corporate reputation* [2]. *Online shop* yang berkembang dengan pesat, baik dalam hal jumlah maupun jenis barang yang dijual, juga menyediakan banyak informasi yang bisa digunakan oleh para pelaku bisnis jual beli *online* [3]. Hal ini menyebabkan munculnya gagasan untuk menilai tanggapan atau ulasan publik terhadap sebuah produk atau kebijakan dari organisasi tertentu yang kemudian digunakan sebagai tolok ukur keberhasilan produk atau kebijaksanaan tersebut. Namun, pekerjaan tersebut akan menjadi sangat berat apabila dilakukan secara manual, karena data yang didapatkan dari aplikasi *microblogging* dan media sosial mayoritas berupa data tekstual yang tidak terstruktur, sehingga akan memakan banyak waktu dan tenaga bila dilakukan oleh manusia. Oleh sebab itu, dibutuhkan sebuah sistem yang secara otomatis bisa melakukan analisis terhadap data tekstual, yaitu analisis sentimen atau *opinion mining*.

Analisis sentimen atau *opinion mining* merupakan sebuah kegiatan untuk memahami, mengekstrak, dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat opini [4]. Sentimen itu sendiri merupakan cerminan dari *attitude* pembicara atau penulis berkenaan dengan topik tertentu. *Attitude* di sini bisa berupa penilaian atau evaluasi mereka, pernyataan afektif mereka (pernyataan emosional penulis saat menulis), atau komunikasi emosional yang diinginkan penulis (efek emosional dari pembaca yang diinginkan oleh penulis) [5]. Salah satu aplikasi dalam analisis sentimen adalah mengelompokkan polaritas dari teks yang ada dalam dokumen dan kalimat, atau menilai ciri/tingkat aspek pendapat yang dikemukakan dalam dokumen. Kegiatan ini disebut sebagai klasifikasi sentimen. Hal ini dilakukan dengan cara menilai setiap kata yang terdapat dalam kalimat opini sehingga bisa dicari pola yang menunjukkan kelas sentimen tertentu.

¹Mahasiswa, Program Magister Teknik Elektro Universitas Brawijaya, Jln. Veteran Malang 65145, INDONESIA (e-mail: hermawan_arief_putranto@yahoo.com)

^{2, 3}Dosen, Teknik Elektro Universitas Brawijaya, Jln. Veteran Malang 65145, INDONESIA (telp: 0341-551611 email: osetyawati@ub.ac.id²; email: wijono@ub.ac.id³)

Sejak diperkenalkannya metode *machine learning* untuk analisis sentimen, banyak algoritme klasifikasi yang mulai diaplikasikan dalam pengelompokan sentimen, di antaranya adalah *Support Vector Machine (SVM)*. SVM adalah metode *machine learning* yang bekerja atas prinsip *Structural Risk Minimization (SRM)* dengan tujuan menemukan *hyperplane* terbaik yang memisahkan dua buah kelas pada *input space* [6]. Pendekatan SRM pada SVM memberikan *error* generalisasi yang lebih kecil daripada yang diperoleh dari strategi *Empirical Risk Management (ERM)* pada *neural network* maupun metode yang lain. Akan tetapi, data yang tidak terstruktur seperti data tekstual cenderung menghasilkan banyak atribut saat proses klasifikasi. Hal ini menyebabkan kinerja *classifier* menjadi berat sehingga akurasi menurun sampai 20% [7].

Salah satu proses yang umumnya digunakan untuk mengurangi atribut pada data tekstual adalah *filtering*. Proses *filtering* berfungsi untuk menghapus kata tidak penting yang dianggap tidak mempengaruhi hasil klasifikasi sentimen. Proses ini menggunakan bantuan daftar kata yang sering muncul dalam kalimat yang disebut *stoplist*. Pada proses *filtering*, kata tunggal hasil proses tokenisasi dibandingkan dengan kata yang ada di dalam *stoplist*. Apabila ada kata yang sama, maka kata di dalam *array* akan terhapus secara otomatis. Namun, proses *filtering* konvensional ini bisa menyebabkan terhapusnya ciri penting yang dapat mempengaruhi hasil klasifikasi sentimen, khususnya pada dokumen berbahasa Indonesia. Hal ini terjadi karena pada saat proses tokenisasi, kalimat dalam dokumen dipecah menjadi kata tunggal, sehingga apabila ada gabungan kata yang membentuk frasa, komputer tidak akan mengenali frasa tersebut. Hal ini menyebabkan terhapusnya kata dalam kalimat secara otomatis karena kata tersebut juga berada dalam *stoplist*, walaupun merupakan bagian dari frasa.

Analisis sentimen, yang merupakan penerapan dari cabang ilmu Pemrosesan Bahasa Alami (PBA) bertujuan untuk menemukan cara agar manusia bisa berkomunikasi dengan komputer menggunakan bahasa sehari-hari [8]. Akan tetapi, komputer tidak dibekali kemampuan untuk memahami bahasa manusia secara langsung, sehingga komputer tidak bisa membedakan antara gabungan kata yang membentuk frasa dan yang bukan. Salah satu cara untuk mengetahui perbedaan frasa dan bukan frasa, sebelumnya komputer harus mengetahui ciri-ciri frasa. Supaya komputer mengetahui ciri-ciri frasa, komputer harus mampu memberi label tiap kata sesuai kelas katanya. Proses pemberian kelas kata ini dikenal dengan nama *POS-Tagger*. Selain pengelompokan kata berdasarkan kelas kata, dengan *POS-Tagger* juga bisa diketahui sifat-sifat apa yang cenderung melekat pada kata tersebut, sehingga bisa ditentukan makna suatu kata atau menentukan aturan tata bahasa suatu kalimat. Ada beberapa pendekatan yang bisa digunakan untuk melakukan *POS-Tagger*, yaitu pendekatan berdasar aturan (*rule based*), pendekatan probabilistik, dan pendekatan berbasis transformasi (*transformational based*). Untuk *POS-Tagger* yang menggunakan metode probabilistik, metode yang biasa digunakan yaitu *Hidden Markov Model (HMM)*. Hal ini

karena proses *POS-Tagger* bisa dipandang sebagai proses klasifikasi suatu rangkaian atau urutan *tag* untuk tiap kata dalam suatu kalimat. Kelebihan dari proses *POS-Tagger* menggunakan pendekatan probabilistik adalah adanya proses *training* dalam pemberian kelas kata, sehingga tidak tergantung pada aturan kelas kata [9].

Dalam makalah ini, kelas kata digunakan untuk memudahkan komputer dalam memahami ciri-ciri sebuah frasa. Dengan kelas kata, bisa dibuat aturan atau *rule* yang berisi kombinasi kelas kata yang kemungkinan besar bila digabungkan akan membentuk frasa. Selain itu, tidak semua kalimat dalam data tekstual berupa kalimat sederhana yang berpola S-P-O-K, tetapi ada juga yang merupakan gabungan pola kalimat yang disebut kalimat majemuk. Hal ini juga dapat menyebabkan permasalahan identifikasi pada proses tokenisasi. Oleh karena itu, kelas kata juga digunakan dalam pemenggalan kalimat majemuk menjadi klausa pembentuknya atau dikenal dengan sebutan *chunking* [10]. Proses ini menggunakan *chunk indicator* dalam pelaksanaannya. *Chunking* digunakan untuk menghindari kesalahan identifikasi dari sebuah frasa.

Dengan mengacu pada permasalahan di atas, solusi yang akan diajukan dalam makalah ini adalah proses tokenisasi menggunakan pendekatan berbasis deteksi frasa. Pada pendekatan ini, kalimat tidak langsung dipecah menjadi kata tunggal, tetapi apabila ditemukan dua kata yang membentuk frasa, maka dua kata tersebut akan disimpan sebagai satu entitas. Dengan pendekatan berbasis deteksi frasa, diharapkan komputer mampu mengenali ciri-ciri gabungan dua kata yang berupa frasa dan yang bukan frasa, sehingga komputer secara otomatis dapat membedakan dan kemudian menyimpan frasa tersebut sebagai satu entitas. Dengan demikian, kejadian terhapusnya ciri penting saat proses *filtering* dengan *stoplist* dapat dihindari, sehingga penggunaan tokenisasi berbasis deteksi frasa ini diharapkan bisa meningkatkan akurasi pada klasifikasi sentimen.

B. Implementasi Support Vector Machine dan POS-Tagger

Aplikasi media sosial yang semakin beragam menyebabkan penambahan data semakin pesat, baik kecepatan maupun ukurannya, mulai dari yang berupa pesan singkat sampai berupa media gambar atau video. Jumlah data yang melimpah ini bisa dimanfaatkan untuk dicari informasi yang berharga di dalamnya bagi organisasi tertentu. Karena itulah, penelitian tentang pengolahan informasi dari media sosial banyak dilakukan, di antaranya yaitu penelitian tentang pemanfaatan media Twitter untuk menilai popularitas presiden Indonesia Joko Widodo [1]. Pada penelitian tersebut, digunakan normalisasi dan *stemming* untuk *preprocessing* data serta membandingkan metode klasifikasi *Naive Bayes* dan SVM. Normalisasi digunakan untuk mengubah kata tidak baku menjadi kata baku yang sesuai dengan Kamus Besar Bahasa Indonesia. Proses *stemming* digunakan untuk mengubah kata berimbuhan menjadi kata dasar. Hasil yang didapatkan adalah terjadinya peningkatan akurasi sebesar 89,26 % pada *preprocessing* menggunakan normalisasi dan *stemming* bila dibandingkan dengan nilai 88,70 % yang didapat dari *preprocessing* tanpa *stemming*. Nilai akurasi tersebut

didapatkan dengan menggunakan metode klasifikasi SVM. Sayangnya, dalam penelitian tersebut tidak dilakukan filterisasi untuk memperbaiki tingkat akurasi proses klasifikasi.

Terdapat penelitian lain tentang penggunaan analisis sentimen yang dimanfaatkan sebagai sistem pendukung manajemen merek dan *corporate reputation* [2]. Penelitian tersebut juga menggunakan *microblog* Twitter sebagai sumber kalimat opini. Penelitian tersebut bertujuan untuk memisahkan *tweet* yang mengandung kalimat opini dan *tweet* yang tidak. Hanya *tweet* yang memiliki kata kunci merek *provider* tertentu yang diambil. Setelah memisahkan kalimat opini dan bukan opini, sistem menggolongkan opini tersebut ke kelas negatif atau positif dan mengekstrak kata kunci untuk tiap kelas. Hasil yang didapatkan dari penelitian tersebut menunjukkan bahwa ciri *keyword extraction* masih kurang memuaskan karena hanya 34,48% *keyword* yang dapat diambil.

Banyaknya ciri dan atribut yang menjadi masukan dapat menurunkan tingkat akurasi sebuah *classifier*. Masalah inilah yang diangkat oleh sebuah penelitian yang memfokuskan kegiatannya untuk mencari kombinasi algoritme seleksi ciri dan metode klasifikasi yang terbaik untuk mengelompokkan sentimen [11]. Data yang digunakan adalah data *Movie Review* yang berisi 1000 *review* film yang sudah dikelompokkan ke dalam *review* positif dan negatif. Dari penelitian tersebut, SVM mendapatkan hasil yang terbaik dengan akurasi 81,10% dan *Area Under Curve* (AUC) 0,904. Sedangkan untuk komparasi seleksi ciri, *information gain* mendapatkan hasil yang paling baik dengan akurasi rata-rata 84,57% dan AUC rata-rata 0,899. Selain itu juga didapatkan integrasi algoritme klasifikasi dan algoritme seleksi ciri terbaik adalah SVM dan *information gain* yang menghasilkan akurasi 81,50 % dan AUC 0,929. Namun, data yang digunakan dalam penelitian tersebut berupa data *review* film yang masih berbahasa Inggris, sehingga untuk dokumen berbahasa Indonesia hasil yang didapatkan bisa jauh berbeda.

Kemudian ada juga penelitian tentang cara mengekstrak data opini masyarakat umum tentang perguruan tinggi [12]. Penelitian tersebut bertujuan mengembangkan sistem *opinion mining* untuk mengolah data opini berbahasa Indonesia pada suatu perguruan tinggi. Selain menggunakan metode *machine learning* untuk proses *opinion orientation*, digunakan juga POS-Tagger untuk memisahkan kalimat opini dan kalimat fakta serta menentukan kata-kata yang menjadi target opini. Metode yang digunakan adalah HMM based POS-Tagger. Hasil yang didapatkan adalah nilai *precision* dan *recall* untuk proses *document subjectivity* adalah 0,99 dan 0,88, sedangkan nilai *precision* dan *recall* untuk proses *target detection* adalah 0,92 dan 0,93. Sayangnya nilai akurasi yang didapatkan dari *Naive Bayes Classifier* masih kurang memuaskan.

II. METODOLOGI

A. Persiapan Data

Data yang akan digunakan adalah *tagged data set*, yaitu data yang dihasilkan dari proses POS-Tagger; *chunked data set*, yaitu data yang dihasilkan dari proses pemenggalan kalimat menjadi klausa; *tag set*, yaitu kumpulan kelas kata

yang digunakan dalam bahasa Indonesia; data komentar positif dan negatif; dan data *review* film. Data komentar didapatkan dari *microblogging* Twitter yang sebelumnya juga digunakan pada penelitian tentang *opinion mining* [12]. Data komentar ini kemudian akan disebut sebagai *Dataset I*. Data *review* film berisi *review* dan sinopsis film yang didapatkan dari *database* film IMDb (*Internet Movie Database*) [13] dan sudah diterjemahkan ke dalam bahasa Indonesia. Data *review* film ini kemudian akan disebut sebagai *Dataset II*.

B. Variabel dan Parameter

Variabel yang digunakan dalam pembobotan teks adalah *term frekuensi* (Tf), yaitu banyaknya jumlah *term* (t) yang sama yang muncul dalam sebuah dokumen (d); *dokumen frekuensi* (Df), yaitu jumlah dokumen yang memuat *term* (t) dalam seluruh dokumen (N); dan Idf , yang merupakan *inverse* nilai dari Df .

Variabel yang digunakan dalam proses klasifikasi dan pengujian adalah *True positive* (TP) yang mewakili jumlah komentar positif yang berhasil diklasifikasikan di kelas positif, *False positive* (FP) yang mewakili jumlah komentar negatif yang diklasifikasikan di kelas positif, *True negative* (TN) yang mewakili jumlah komentar negatif yang diklasifikasikan dalam kelas negatif, dan *False negative* (FN) yang mewakili komentar positif yang diklasifikasikan dalam kelas negatif.

Variabel yang digunakan dalam proses validasi adalah k , yaitu jumlah segmentasi pada proses *k-fold cross validation*. Parameter yang digunakan dalam proses klasifikasi adalah parameter regulasi dalam pencarian *soft-margin* terbaik (C) dan parameter fungsi *kernel* (γ).

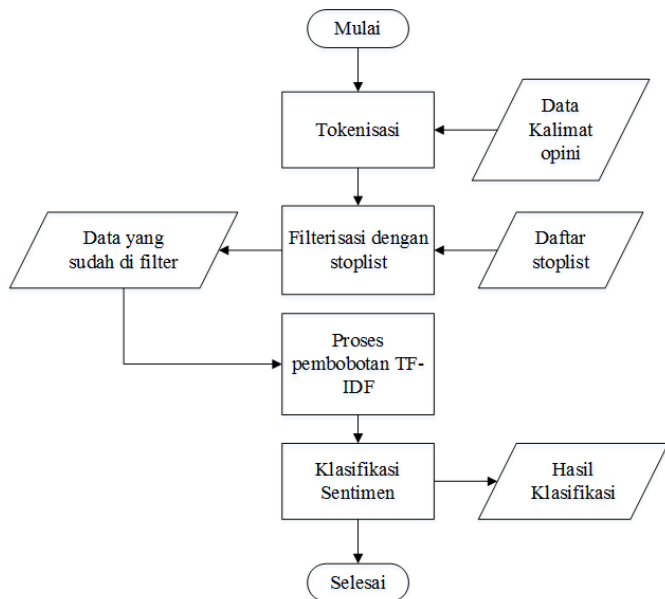
C. Klasifikasi Sentimen

Sebelum data dimasukkan ke dalam proses klasifikasi, data mengalami proses *cleansing* untuk menghilangkan kode *html* dan karakter lain yang bukan bagian dari data tekstual. Untuk mengetahui pengaruh dari pendekatan berbasis deteksi frasa terhadap akurasi klasifikasi menggunakan SVM, digunakan dua perlakuan yang berbeda untuk *dataset* yang sama. Pada perlakuan pertama, proses klasifikasi dilakukan tanpa menggunakan pendekatan berbasis deteksi frasa. Pada perlakuan yang kedua, proses klasifikasi dilakukan menggunakan pendekatan berbasis deteksi frasa.

Gbr. 1 menunjukkan alur langkah perlakuan pertama, yaitu proses klasifikasi tanpa pendekatan berbasis deteksi frasa. Setelah proses *cleansing*, data langsung dipecah menjadi *token* dan disimpan ke dalam *array* sementara. Kemudian, *token* tersebut diambil satu per satu untuk dibandingkan dengan kata yang berada dalam daftar *stoplist*. Setiap *token* yang sama dengan kata dalam daftar *stoplist* akan terhapus secara otomatis.

Token yang tersisa disimpan ke dalam *file* teks. Kemudian *file* teks tersebut masuk ke dalam proses pembobotan TF-IDF. Proses ini dilakukan untuk mengubah *file* teks menjadi bentuk vektor dokumen yang dikenali oleh *classifier*. Kemudian vektor dokumen tadi masuk ke dalam proses klasifikasi dan diperoleh hasil klasifikasi yang berupa nilai *precision*, *recall*, *F-Measure*, *Receiver Operating Characteristic* (ROC), dan Akurasi [14]. Nilai *precision* didapatkan dengan cara

menghitung jumlah komentar yang diklasifikasikan dengan benar dibagi dengan jumlah total komentar yang berhasil diklasifikasikan dalam satu kelas yang sama. Nilai *recall* didapatkan dengan cara menghitung jumlah komentar yang diklasifikasikan dengan benar dibagi jumlah total komentar dalam kelas sebenarnya. *F-Measure* dicari dengan cara menghitung rata-rata harmonik dari *precision* dan *recall*. ROC didapatkan dari hasil klasifikasi yang digambarkan dalam bentuk dua dimensi. Area di bawah kurva ROC inilah yang disebut dengan AUC, yang digunakan untuk melihat apakah model klasifikasi tersebut sudah optimal atau belum. Apabila nilai ROC < 0,5 maka model klasifikasi belum optimal. Nilai Akurasi didapatkan dari pembagian antara jumlah data yang terklasifikasi dengan benar dengan jumlah data seluruhnya.

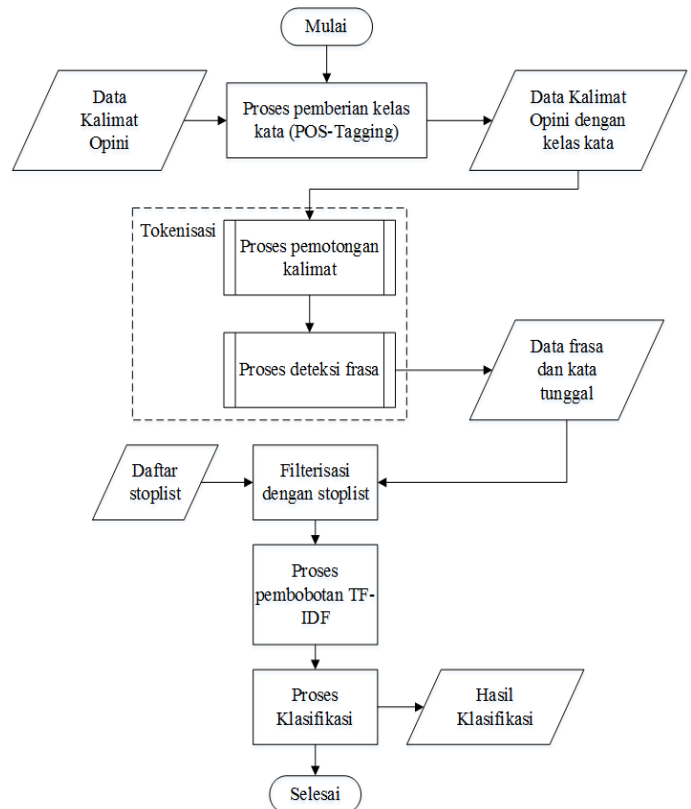


Gbr. 1 Diagram alir proses klasifikasi tanpa pendekatan berbasis deteksi frasa.

Pada perlakuan yang kedua, yaitu proses klasifikasi menggunakan pendekatan berbasis deteksi frasa, data juga mengalami proses *cleansing* terlebih dahulu. Akan tetapi, terdapat tiga proses tambahan yang harus dilakukan terlebih dahulu sebelum proses filterisasi. Ketiga proses ini ditunjukkan pada Gbr. 2.

Proses yang pertama yaitu proses *POS-Tagging*, yang dilakukan menggunakan *software* Ipostagger [15]. Setelah tiap kata memiliki kelas kata, data masuk ke dalam proses kedua yaitu *chunking*. Proses yang menghasilkan klausa ini dilakukan dengan bantuan sebuah aturan yang disusun berdasarkan sifat-sifat *chunk indicator* [10]. Kemudian, setiap klausa dimasukkan ke dalam proses deteksi frasa. Pada proses ini, gabungan dua kata yang sesuai dengan aturan frasa akan disimpan sebagai satu entitas. Gabungan dua kata yang tidak sesuai dengan aturan frasa tetap disimpan sebagai dua entitas yang berbeda. Hasil dari proses deteksi frasa ini disimpan ke dalam *file* teks yang kemudian dimasukkan ke dalam proses filterisasi dengan *stoplist*. Kemudian hasil dari proses

filterisasi tersebut dimasukkan ke dalam proses pembobotan dan klasifikasi.



Gbr. 2 Diagram alir proses klasifikasi menggunakan pendekatan berbasis deteksi frasa.

D. Deteksi Frasa

Agar komputer dapat mengetahui apakah gabungan dua kata dalam kalimat termasuk frasa atau bukan, diperlukan sebuah aturan yang berisi kombinasi kelas kata yang berpeluang besar membentuk frasa. Kombinasi kelas kata ini ditunjukkan pada Tabel I. Dengan aturan tersebut, komputer diharapkan mampu mendeteksi ciri-ciri dari gabungan dua kata yang membentuk frasa.

Alur langkah untuk proses deteksi frasa ditunjukkan pada Gbr. 3. Dapat dilihat bahwa setelah kalimat dipecah menjadi kata tunggal, komputer mengambil dua kata pertama dan melihat kombinasi kelas katanya. Apabila kombinasi kelas kata tersebut sesuai dengan aturan yang berada dalam Tabel I, gabungan dua kata tersebut terdeteksi sebagai frasa. Frasa itu kemudian disimpan sebagai satu entitas, sedangkan kombinasi kelas kata yang tidak berada di dalam Tabel I dianggap bukan frasa, sehingga disimpan sebagai dua entitas yang berbeda.

E. Pengujian dan Validasi Klasifikasi

Pada makalah ini digunakan pendekatan *k-fold cross validation* sebagai metode pengujiannya. Dalam pendekatan *cross validation*, setiap *record* digunakan beberapa kali dalam jumlah yang sama untuk *training* dan hanya satu kali untuk *testing*. Pendekatan ini mempartisi data ke dalam dua *subset* data yang berukuran sama. Pertama-tama *subset* data pertama digunakan sebagai data *training* dan *subset* data kedua

digunakan untuk *testing*. Setelah proses *training* dan *testing* selesai, dilakukan pertukaran fungsi *subset*, sehingga *subset* yang sebelumnya sebagai *training set* menjadi *testing set*, dan sebaliknya. Total *error* diperoleh dengan menjumlahkan *error* dari kedua proses tersebut.

TABEL I
RULE/ATURAN DETEKSI FRASA [14]

No.	Jenis Frasa	Kombinasi Kelas Kata	Contoh
1	Frasa Verbal	VB JJ	Kerja keras
		VB NN	Banting tulang
		VB RB	Pemeriksaan harian
2	Frasa Nominal	CD NN	Dua biji, sebuah apel
		NN NN	Jam malam, dunia malam
		NN RB	Buku harian
		NN JJ	Rumah megah, baju bagus
		RB JJ	Begitu indah, lebih mulia
		RB NN	Keras kepala, panjang tangan
3	Frasa Adverbial	IN RB	Dengan keras
		PRP PR	Saya ini, mereka itu
		IN PRP	Dengan saya
4	Frasa Pronominal	PRP NN	Dia wanita
		PRP NNP	Dia Pak Ali
		PRP NNP	Dia Pak Ali
5	Frasa Adjektival	JJ RB	Bagus sekali, tepat benar
		IN JJ	Dengan manis, dengan panas
		JJ JJ	Panas dingin, hitam legam

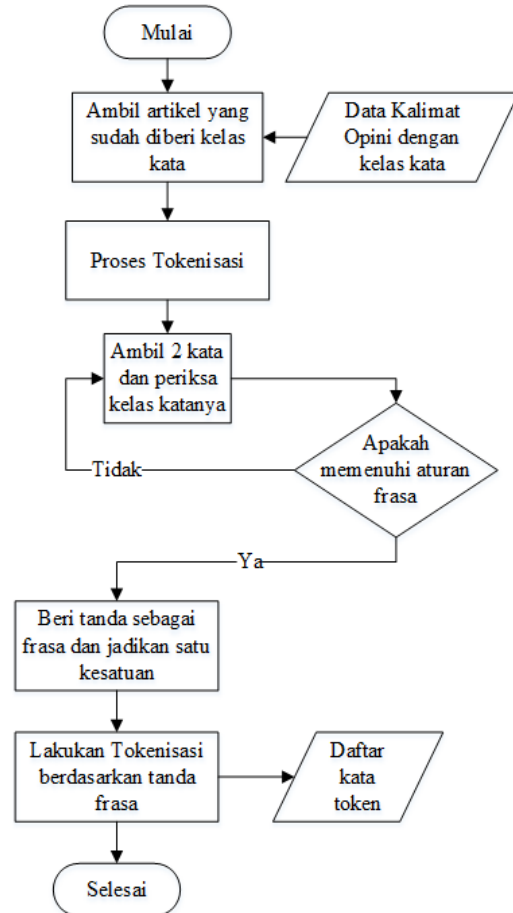
Pendekatan *k-fold cross validation* menggeneralisasi pendekatan *cross validation* dengan menyegmentasikan data ke dalam *k* partisi berukuran sama. Selama proses, salah satu dari partisi dipilih untuk *training*, sedangkan sisanya untuk *testing*. Prosedur ini diulang sebanyak *k* kali sehingga setiap partisi digunakan untuk *testing* tepat hanya satu kali. Total *error* ditentukan dengan menjumlahkan *error* untuk semua *k* proses tersebut.

III. HASIL DAN PEMBAHASAN

Dari dua perlakuan yang sudah dijelaskan sebelumnya, didapatkan data hasil klasifikasi yang dirangkum ke dalam Tabel II. Proses klasifikasi dilakukan menggunakan *library LibSVM* [16] dan *software Weka* [17] dengan parameter *kernel* linier dan nilai $\gamma = 1$. Pengujian menggunakan pendekatan *k-fold cross validation* dengan nilai $k = 2, 4, 6, 8, 10$ untuk *Dataset I* dan $k = 2, 6, 10, 12, 16, 20, 22, 26, 30, 32$ untuk *Dataset II*. Hasil dari klasifikasi ini disimpan pada sebuah *confusion matrix*. Matriks ini berisi jumlah komentar yang terklasifikasi sebagai TP, FP, TN, dan FN.

Pada nilai $k = 2$ berarti *dataset* (d) dibagi menjadi dua bagian yang sama besar, yaitu d_1 dan d_2 . Kemudian d_1 digunakan sebagai data latih (*training*), dan d_2 digunakan

sebagai data uji (*testing*). Setelah didapatkan hasilnya, proses klasifikasi diulang dengan d_2 sebagai data latih dan d_1 sebagai data uji. Hasil yang didapatkan dari seluruh proses disimpan dan dicari rata-ratanya. Hal ini dilakukan untuk mencari hasil klasifikasi terbaik.



Gbr. 3 Diagram alir proses deteksi frasa.

Dari *confusion matrix* ini didapatkan nilai *precision*, *recall*, *F-Measure*, *ROC*, dan akurasi dari tiap perlakuan. Nilai *precision* dan *recall* didapatkan dari (1) dan (2), Sedangkan untuk akurasi dan *F-measure* didapatkan dari (3) dan (4).

$$Precision : \frac{TP}{TP+FP} \quad (1)$$

$$Recall : \frac{TP}{TP+FN} \quad (2)$$

$$Accuracy : \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

$$F-Measure : 2 \frac{precision \cdot recall}{precision+recall} \quad (4)$$

dengan TP adalah *True Positive*, FP adalah *False Positive*, TN adalah *True Negative*, dan FN adalah *False Negative*.

Tabel II berisi *confusion matrix* dari hasil proses klasifikasi menggunakan *Dataset I*. Matriks ini berisi jumlah kalimat yang terklasifikasi ke dalam kelas positif dan negatif. Setelah mendapatkan nilai TP, FP, TN, dan FN, nilai *precision*, *recall*, *F-Measure* dan akurasi dapat dicari menggunakan (1), (2), (3),

dan (4), sedangkan nilai ROC diperoleh dari proses validasi model klasifikasi. Hasil dari perhitungan tersebut dimasukkan ke dalam Tabel III dan Tabel IV.

Pada Tabel III dapat dilihat bahwa nilai akurasi pada proses klasifikasi tanpa pendekatan berbasis deteksi frasa yang paling tinggi, yaitu 83,67 %, terjadi pada $k = 8$. Sedangkan dari Tabel IV, dapat dilihat bahwa nilai akurasi pada proses klasifikasi dengan pendekatan berbasis deteksi frasa yang paling tinggi, yaitu 90,14 %, didapat pada $k = 6$. Dengan hasil tersebut dapat disimpulkan bahwa terdapat peningkatan nilai akurasi sekitar 6% pada proses klasifikasi menggunakan pendekatan berbasis deteksi frasa pada Dataset I.

Tabel V berisi *confusion matrix* dari hasil proses klasifikasi menggunakan Dataset II. Menggunakan persamaan yang sama, didapatkan nilai *precision*, *recall*, *F-Measure*, *ROC* dan akurasi untuk Dataset II. Tabel VI menunjukkan hasil perhitungan untuk proses klasifikasi tanpa pendekatan berbasis deteksi frasa, sedangkan Tabel VII menunjukkan hasil perhitungan untuk proses klasifikasi dengan pendekatan

berbasis deteksi frasa. Dari Tabel VI dapat diketahui nilai akurasi tertinggi, yaitu 78,13% didapat pada $k = 30$, dan dari Tabel VII diketahui nilai akurasi tertinggi, yaitu 81,25% , didapat pada $k = 26$.

Bila dibandingkan dengan proses klasifikasi pada Dataset I, terjadi penurunan tingkat akurasi pada proses klasifikasi pada Dataset II. Penurunan ini terjadi pada kedua perlakuan. Hal ini kemungkinan disebabkan oleh tercampurnya kalimat fakta dan kalimat opini pada Dataset II, mengingat dataset ini diambil dari *review* film yang tidak hanya berisi opini penulis, tetapi juga sinopsis dari film yang bersangkutan. Sedangkan Dataset I hanya berisi kalimat opini saja.

Hasil ini juga menunjukkan bahwa dengan digunakannya pendekatan berbasis deteksi frasa, model klasifikasi yang didapat lebih optimal. Hal ini diperkuat dengan hasil ROC terbaik yang didapatkan yaitu 0,897, yang diperoleh pada proses klasifikasi dengan menggunakan pendekatan berbasis deteksi frasa pada Dataset I.

TABEL II
CONFUSION MATRIX DATASET I

Tanpa Deteksi Frasa					Dengan Deteksi Frasa				
k	Terklasifikasi sebagai				k	Terklasifikasi Sebagai			
	Positif		Negatif			Positif		Negatif	
2	TP	164	FN	32	2	TP	181	FN	16
	FP	55	TN	92		FP	49	TN	99
4	TP	174	FN	22	4	TP	175	FN	22
	FP	36	TN	111		FP	34	TN	114
6	TP	171	FN	25	6	TP	183	FN	14
	FP	32	TN	115		FP	20	TN	128
8	TP	175	FN	21	8	TP	179	FN	18
	FP	35	TN	112		FP	27	TN	121
10	TP	170	FN	26	10	TP	177	FN	20
	FP	35	TN	112		FP	26	TN	122

TABEL III
NILAI PRECISION, RECALL, F-MEASURE, ROC, DAN AKURASI DARI KLASIFIKASI TANPA PENDEKATAN BERBASIS DETEKSI FRASA UNTUK DATASET I

k	Negatif				Positif				Akurasi
	Precision	Recall	F-Measure	ROC	Precision	Recall	F-Measure	ROC	
2	0.749	0.837	0.79	0.731	0.742	0.626	0.679	0.731	74.64%
4	0.829	0.888	0.857	0.821	0.835	0.755	0.793	0.821	83.09%
6	0.842	0.872	0.857	0.827	0.821	0.782	0.801	0.827	83.38%
8	0.833	0.893	0.862	0.827	0.842	0.762	0.8	0.827	83.67%
10	0.829	0.867	0.848	0.815	0.812	0.762	0.786	0.815	82.22%

TABEL IV
NILAI PRECISION, RECALL, F-MEASURE, ROC, DAN AKURASI DARI KLASIFIKASI DENGAN PENDEKATAN BERBASIS DETEKSI FRASA UNTUK DATASET I

k	Negatif				Positif				Akurasi
	Precision	Recall	F-Measure	ROC	Precision	Recall	F-Measure	ROC	
2	0.787	0.919	0.848	0.794	0.861	0.669	0.753	0.794	81.16%
4	0.837	0.888	0.862	0.829	0.838	0.77	0.803	0.829	83.77%
6	0.901	0.929	0.915	0.897	0.901	0.865	0.883	0.897	90.14%
8	0.869	0.909	0.888	0.863	0.871	0.818	0.843	0.863	86.96%
10	0.872	0.898	0.885	0.861	0.859	0.824	0.841	0.861	86.67%

TABEL V
CONFUSION MATRIX DATASET II

Tanpa Deteksi Frasa					Dengan Deteksi Frasa				
<i>k</i>	Terklasifikasi sebagai				<i>k</i>	Terklasifikasi Sebagai			
	Positif		Negatif			Positif		Negatif	
2	TP	30	FN	2	2	TP	29	FN	3
	FP	15	TN	17		FP	14	TN	18
6	TP	28	FN	4	6	TP	26	FN	6
	FP	13	TN	19		FP	10	TN	22
10	TP	27	FN	5	10	TP	26	FN	6
	FP	11	TN	21		FP	10	TN	22
12	TP	27	FN	5	12	TP	27	FN	5
	FP	16	TN	16		FP	10	TN	22
16	TP	27	FN	5	16	TP	26	FN	6
	FP	12	TN	20		FP	9	TN	23
20	TP	27	FN	5	20	TP	26	FN	6
	FP	12	TN	20		FP	10	TN	22
22	TP	26	FN	6	22	TP	26	FN	6
	FP	12	TN	20		FP	8	TN	24
26	TP	27	FN	5	26	TP	27	FN	5
	FP	10	TN	22		FP	7	TN	25
30	TP	27	FN	5	30	TP	26	FN	6
	FP	9	TN	23		FP	9	TN	23
32	TP	27	FN	5	32	TP	26	FN	6
	FP	11	TN	21		FP	9	TN	23

TABEL VI
NILAI *PRECISION*, *RECALL*, *F-MEASURE*, *ROC*, DAN AKURASI PADA PROSES KLASIFIKASI TANPA DETEKSI FRASA PADA DATASET II

<i>k</i>	Negatif				Positif				Akurasi
	<i>Precision</i>	<i>Recall</i>	<i>F-Measure</i>	ROC	<i>Precision</i>	<i>Recall</i>	<i>F-Measure</i>	ROC	
2	0.882	0.469	0.612	0.703	0.638	0.938	0.759	0.703	70.31%
6	0.826	0.594	0.691	0.734	0.683	0.875	0.767	0.734	73.44%
10	0.808	0.656	0.724	0.75	0.711	0.844	0.771	0.75	75%
12	0.762	0.5	0.604	0.672	0.628	0.844	0.72	0.672	67.19%
16	0.8	0.625	0.702	0.734	0.692	0.844	0.761	0.734	73.44%
20	0.8	0.625	0.702	0.734	0.692	0.844	0.761	0.734	73.44%
22	0.769	0.625	0.69	0.719	0.684	0.813	0.716	0.719	71.88%
26	0.815	0.688	0.746	0.766	0.73	0.844	0.783	0.766	76.56%
30	0.821	0.719	0.767	0.781	0.75	0.844	0.794	0.781	78.13%
32	0.808	0.656	0.724	0.75	0.711	0.844	0.711	0.75	75%

TABEL VII
NILAI *PRECISION*, *RECALL*, *F-MEASURE*, *ROC* DAN *ACCURACY* PADA PROSES KLASIFIKASI DENGAN DETEKSI FRASA PADA DATASET II

<i>k</i>	Negatif				Positif				Akurasi
	<i>Precision</i>	<i>Recall</i>	<i>F-Measure</i>	ROC	<i>Precision</i>	<i>Recall</i>	<i>F-Measure</i>	ROC	
2	0.857	0.563	0.679	0.734	0.674	0.906	0.773	0.734	73.44%
6	0.786	0.688	0.733	0.75	0.722	0.813	0.765	0.75	75%
10	0.786	0.688	0.733	0.75	0.722	0.813	0.765	0.75	75%
12	0.815	0.688	0.746	0.766	0.73	0.844	0.783	0.766	76.56%
16	0.793	0.719	0.754	0.766	0.743	0.813	0.776	0.766	76.56%
20	0.786	0.688	0.733	0.75	0.722	0.813	0.765	0.75	75%
22	0.8	0.75	0.774	0.781	0.765	0.813	0.788	0.781	78.13%
26	0.833	0.781	0.806	0.813	0.794	0.844	0.818	0.813	81.25%
30	0.793	0.719	0.754	0.766	0.743	0.813	0.776	0.766	76.56%
32	0.793	0.719	0.754	0.766	0.743	0.813	0.776	0.766	76.56%

IV. KESIMPULAN

Proses tokenisasi dengan menggunakan pendekatan berbasis frasa mampu meningkatkan akurasi pada klasifikasi sentimen berbahasa Indonesia. Hal ini disebabkan oleh berkurangnya kejadian terhapusnya gabungan dua kata yang berupa frasa pada proses filterisasi, karena komputer sudah mampu membedakan antara gabungan dua kata yang berupa frasa dan yang bukan, kemudian menyimpannya sebagai satu entitas.

Dengan menggunakan HMM POS-Tagger didapatkan peningkatan nilai akurasi pada proses klasifikasi menggunakan SVM dengan pendekatan berbasis deteksi frasa, lebih kurang sebesar 6% pada *Dataset I* dan sekitar 3% pada *Dataset II*. Hal ini dibuktikan dengan banyaknya kalimat dan paragraf yang terklasifikasi dengan benar, sesuai dengan kelas sentimennya, baik pada *Dataset I* maupun pada *Dataset II*.

Hasil yang berbeda didapatkan dari proses klasifikasi yang menggunakan deteksi frasa pada *Dataset I* dan *Dataset II*. Hal ini mungkin disebabkan oleh tercampurnya kalimat fakta dan kalimat opini pada *Dataset II*, mengingat *dataset* ini diambil dari *review* film yang tidak hanya berisi opini penulis, namun juga sinopsis dari film yang bersangkutan, sedangkan *Dataset I* hanya berisi kalimat opini saja.

Untuk mendapatkan hasil yang lebih baik, diperlukan pengembangan pembelajaran dalam proses pemberian kelas kata dalam bahasa Indonesia, khususnya pada pembelajaran yang bisa membedakan kata benda ke dalam kelas kata turunannya, sehingga bisa dibuat aturan-aturan yang lebih baik dalam pengenalan ciri-ciri frasa, khususnya frasa benda.

REFERENSI

- [1] N. Saputra, T. B. Adji dan A. E. Permasari, "Analisis Sentimen Data Presiden Jokowi dengan Preprocessing Normalisasi dan Stemming menggunakan metode Naive Bayes dan SVM," *Jurnal Dinamika Informatika*, vol. 5, no. 1, 2015.
- [2] Harlili dan Y. Wibisono, "Sistem Analisis Opini *Microblogging* Berbahasa Indonesia", Bandung: UPI Bandung, 2013.
- [3] Helmy, "Pencarian Pola Akses Pengunjung Toko Online Menggunakan *Weighted Graph Web Usage Mining*," *JNTETI*, vol. 3, no. 1, 2014.
- [4] G. A. Buntoro, "*Sentiment Analysis Tweeter* dengan Kombinasi *Lexicon Based* dan *Double Propagation*", *CITEE*, 2014.
- [5] W. H. Ian, E. Frank dan H. A. Mark, *Data Mining: Practical Machine Learning Tools and Techniques*, Elsevier, 2011.
- [6] A. S. Nugroho, A. B. Witarto dan D. Handoko, "Application of Support Vector Machine in Bioinformatics", *Proceeding of Indonesian Scientific Meeting*, Gifu, Japan, 2003.
- [7] Tantiny, B. Susanto dan W. Hapsari, "Klasifikasi Email dengan Menggunakan Metode *Naive Bayesian* Studi Kasus: Mailing List www.tux.org," *Jurnal Informatika*, 2007.
- [8] J. Pustejovsky dan A. Stubbs, *Natural Language Annotation and Machine Learning*, O'Riley, 2012.
- [9] T. Chandrawati, Pengembangan *Part of Speech Tagger* untuk Bahasa Indonesia Berdasarkan Metode *Conditional Random Fields* dan *Transformation Based Learning*, Jakarta: Fakultas Ilmu Komputer Universitas Indonesia, 2008.
- [10] Y. Yusuf, S. Nurdiati dan B. P. Silalahi, "Analisis Pembentukan Pola Graf pada Kalimat Bahasa Indonesia menggunakan metode *Knowledge Graph*," *Lingua Jurnal Bahasa dan Sastra*, vol. 10, No. 1, 2014.
- [11] V. Chandani, R. S. Wahono dan Purnomo, "Komparasi Algoritma Klasifikasi *Machine Learning* Dan *Feature Selection* pada Analisis Sentimen *Review Film*", *Journal of Intelligent Systems*, vol. 1, 2015.
- [12] I. F. Rozi, S. H. Pramono dan E. A. Dahlan, "Implementasi *Opinion Mining* (Analisis Sentimen) untuk Ekstraksi Data Opini Publik pada Perguruan Tinggi", *EEICIS*, vol. 6, 2012.
- [13] B. Pang dan L. Lee, *A sentimental education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts*, Itacha, New York: Cornell University, 2002.
- [14] A. A. Armana, A. B. Putra, A. Purwarianti dan Kuspriyanto, "Syntactic Phrase Chunking for Indonesian Language", *Science Direct*, pp. 635-640, 2013.
- [15] A. Purwarianti dan A. F. Wicaksono, "HMM Based Part-Of-Speech Tagger for Bahasa Indonesia", *Proceedings of 4th International MALINDO (Malay and Indonesian Language) Workshop*, 2010.
- [16] W. H. Chih, C. C. Chih dan J. L. Chih, *A Practical Guide to Support Vector Classification*, Taipei: Department of Computer Science, National Taiwan University, 2003.
- [17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann dan I. H. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explorations*, vol. 11, no. 1, 2009.