ISSN: 2339-2541
JURNAL GAUSSIAN, Volume 9, Nomor 3, Tahun 2020, Halaman 376 -390
Online di: https://ejournal3.undip.ac.id/index.php/gaussian/

ANALISIS SENTIMEN GOJEK PADA MEDIA SOSIAL TWITTER DENGAN KLASIFIKASI SUPPORT VECTOR MACHINE (SVM)

Nur Fitriyah [1], Budi Warsito[2], Di Asih I Maruddani [3]
[1,2,3]Departemen Statistika FSM Universitas Diponegoro
budiwrst2@gmail.com

ABSTRACT

Appearance of PT Aplikasi Karya Anak Bangsa or as known as Gojek since 2015 give a convenience facility to people in Indonesia especially in daily activities. Sentiment analysis on Twitter social media can be the option to see how Gojek users respond to the services that have been provided. The response was classified into positive sentiment and negative sentiment using Support Vector Machine method with model evaluation 10-fold cross validation. The kernel used is the linear kernel and the RBF kernel. Data labeling can be done with manually and sentiment scoring. The test results showed that the RBF kernel gets overall accuracy and the highest kappa accuracy on manual data labeling and sentiment scoring. On manual data labeling, the overall accuracy is 79.19% and kappa accuracy is 16.52%. While the labeling of data with sentiment scoring obtained overall accuracy of 79.19% and kappa accuracy of 21%. The greater overall accuracy value and kappa accuracy obtained, the better performance of the classification model.

Keywords: Gojek, Twitter, Support Vector Machine, overall accuracy, kappa accuracy

1. PENDAHULUAN

Twitter merupakan salah satu media sosial yang populer di Indonesia maupun di berbagai negara lainnya. Berdasarkan laporan *Wearesocial Hootsuite* pengguna media sosial Twitter di Indonesia mencapai 52% dari jumlah pengguna internet atau mencapai 78 juta pengguna [7].

Gojek merupakan ojek *online* yang berasal dari Indonesia. Jumlah unduh aplikasi Gojek mencapai 142 juta, layanan pesan-antar makanan mencapai 400 ribu mitra di 370 kota di Indonesia, dan bermitra dengan 28 institusi keuangan [9]. Setiap pelanggan memiliki respon yang berbeda terhadap pelayanan Gojek. Berdasarkan pernyataan sebelumnya, penulis ingin meneliti bagaimana respon pengguna Gojek terhadap pelayanan Gojek dengan menganalisis cuitan (*tweet*) pengguna Gojek di media sosial Twitter. Analisis sentimen pada media sosial Twitter menjadi pilihan penulis untuk melihat bagaimana respon pengguna Gojek terhadap pelayanan yang sudah diberikan.

Pada penelitian ini akan diklasifikasikan respon pengguna Gojek ke dalam dua sentimen, yaitu sentimen positif dan sentimen negatif. Pada umumnya masalah dalam dunia nyata (*real world problem*) jarang yang bersifat *linear separable*. Salah satu metode yang dapat mengatasi *problem non-linear* secara efisien yaitu *Support Vector Machine* (SVM). Untuk mengatasi permasalahan *nonlinear* digunakan fungsi *kernel*. *Kernel* yang akan digunakan adalah *kernel linear* dan *kernel Radial Basis Function* (RBF). Secara umum, *kernel* RBF adalah pilihan pertama yang tepat untuk pembentukan model. *Kernel* RBF secara *nonlinear*

memetakan sampel ke ruang dimensi yang lebih tinggi sehingga dapat menangani kasus *nonlinear* [8]. *Kernel linear* adalah kasus khusus RBF [10], karena *kernel linear* dengan parameter C memiliki kinerja yang sama dengan *kernel* RBF dengan beberapa parameter (C,γ). *Kernel* polinomial dan sigmoid memiliki lebih banyak parameter daripada kernel RBF, maka dari itu *kernel* tersebut tidak akan digunakan.

2. TINJAUAN PUSTAKA

2.1. *Text Mining*

Text mining adalah proses ekstraksi pola (informasi dan pengetahuan yang berguna) dari sejumlah sumber data melalui identifikasi pola yang menarik. Pada kasus *text mining*, sumber data adalah kumpulan data tekstual yang tidak terstruktur pada dokumen [3]. *Text mining* bertujuan untuk menemukan informasi yang tidak diketahui, sesuatu yang belum diketahui dan belum dapat ditulis [6].

2.2. Analisis Sentimen

Analisis sentimen atau *opinion mining* merupakan proses memahami, mengekstrak dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat opini. Analisis sentimen dilakukan untuk melihat pendapat atau kecenderungan opini terhadap sebuah masalah atau objek oleh seseorang, apakah cenderung berpandangan atau beropini negatif atau positif [14].

2.3. *Pre-Processing* Data

Karena bahasa Twitter memiliki banyak item teks yang unik, maka item teks tersebut dihilangkan untuk mengurangi ruang fitur. Item tersebut perlu dihilangkan karena tidak memiliki nilai informasi dalam konteks klasifikasi sentimen [2]. *Pre-processing* yang akan dilakukan adalah sebagai berikut: *Case folding, remove* URL, *unescape* HTML, *remove mention, remove number, remove punctuation, remove emoticon, dan* normalisasi kalimat.

2.4. *Sentiment Scoring*

Pada *sentiment scoring* terdapat input kamus sentimen, *boosterwords* dan input kata negasi. Kamus sentimen berisi kumpulan kata yang telah diberi bobot dengan kekuatan sentimen 1 s.d. 5 (memiliki sentimen positif), dan -1 s.d. -5 (memiliki sentimen negatif). *Boosterwords* adalah kata yang dapat meningkatkan atau mengurangi intensitas sentimen kata disebelahnya [19]. Kata ini diberi bobot 1-2 untuk menambah atau mengurangi skor kata disampingnya. Kata negasi merupakan kata yang terdapat dalam suatu kalimat yang dapat mengubah orientasi dari suatu opini [19].

2.5. *Feature Selection*

1. *Stopwords*

Stopwords adalah kata umum (*common words*) yang muncul dalam jumlah besar dan dianggap tidak memiliki makna [20]. Penggunaan *stopwords removal* terbukti dapat meningkatkan hasil akurasi sistem klasifikasi sentimen dibandingkan tanpa penggunaan *stopword* [4].

2. Tokenizing

Pada prinsipnya proses *tokenizing* adalah memisahkan setiap kata yang menyusun suatu dokumen. Pada umumnya setiap kata terindentifikasi atau

terpisahkan dengan kata yang lain oleh karakter spasi, sehingga proses *tokenizing* mengandalkan karakter spasi pada dokumen untuk melakukan pemisahan kata [12].

2.6. Pembobotan

[13] menyatakan bahwa pembobotan kata bertujuan untuk memberikan bobot pada fitur kata berdasarkan frekuensi kemunculan kata. *Term Frequency* (TF) merupakan jumlah kemunculan atau frekuensi kata pada suatu dokumen. Sedangkan *Inverse Document Frequency* (IDF) bertujuan untuk mengetahui apakah *term* yang dicari cocok dengan kata kunci yang diinginkan, *term* yang sering muncul akan memberikan pengaruh yang kecil dalam menentukan keterkaitan kata kunci dokumen.

TF-IDF dihitung dengan menggunakan persamaan seperti berikut [15]:

$$W_{j,i} = \frac{n_{j,i}}{\sum_k n_{k,i}} . \log_2 \frac{D}{d_j} \quad (1)$$

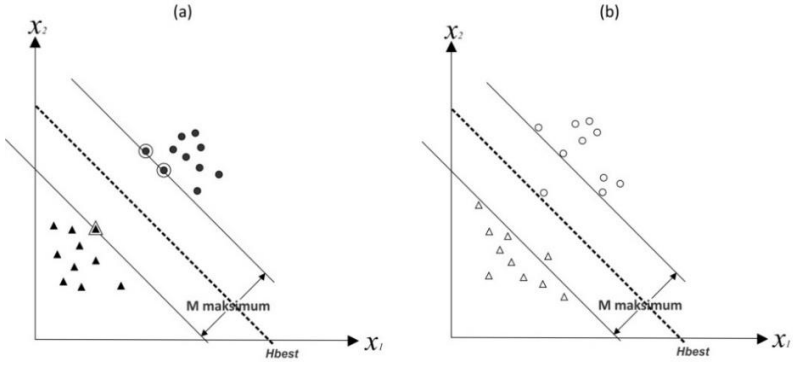
dimana:

$W_{j,i}$ = Pembobotan TF-IDF untuk *term* ke j pada dokumen ke i.
$n_{j,i}$ = Jumlah kemunculan *term* ke j pada dokumen ke i.
$\sum_k n_{k,i}$ = Jumlah kemunculan seluruh *term* pada dokumen ke i.
D = Banyaknya dokumen yang dibangkitkan.
d_j = Banyaknya dokumen yang mengandung *term* ke j.

2.7. *Support Vector Machine* (SVM)

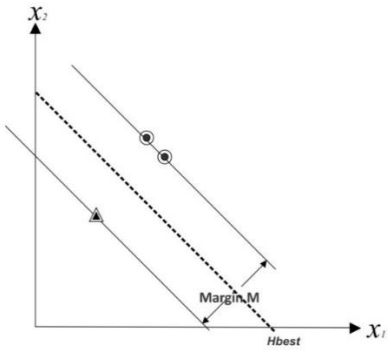
2.7.1. Bentuk Dasar *Support Vector Machine* (SVM)

SVM berusaha menemukan *hyperplane* dengan memaksimalkan jarak antar kelas [17], seperti diilustrasikan pada Gambar 1(a). Dengan cara ini, SVM menjamin kemampuan generalisasi yang tinggi untuk data-data yang akan datang, seperti diilustrasikan pada Gambar 1(b).



Gambar 1. *Hyperplane* Terbaik dan Margin Maksimum

Dalam SVM, objek-objek data terluar yang paling dekat dengan *hyperplane* ini disebut *support vector*. Hanya *support vector* inilah yang diperhitungkan oleh SVM untuk menemukan *hyperplane* paling optimal sedangkan objek-objek data yang lain tidak diperhitungkan sama sekali, seperti diilustrasikan pada Gambar 2. Dengan cara ini, SVM dapat bekerja secara lebih efisien.



Gambar 2. Memperoleh Hbest dengan Memperhitungkan Tiga *Support Vector*

Misalkan data yang terdapat pada himpunan data latih dinotasikan sebagai $x_i \in R^d$ sedangkan label kelas dinyatakan sebagai $y_i \in \{-1, +1\}$. Model linier secara umum yang dipakai dalam SVM untuk menghasilkan *hyperplane* adalah sebagai berikut [16]:

$$y = w^T x_i + b \quad , i = 1,2, \dots, l \tag{2}$$

dengan

x_i = $[x_1, x_2, \dots, x_k]$ adalah vektor baris berdimensi k (banyaknya fitur)
$y \in \{-1, +1\}$ = nilai target dari himpunan data x_i
l = jumlah data
w = $[w_1, w_2, \dots, w_k]$ adalah vektor baris yang menjadi parameter bobot
b = bias atau *error*

Dua pemisah sejajar dengan *hyperplane* yang diperoleh dari data terdekat pada masing-masing kelas yang diformulasikan sebagai berikut:

$$(w^T . x_i) + b = 1, \text{ untuk kelas positif} \tag{3}$$

$$(w^T . x_i) + b = -1, \text{ untuk kelas negatif} \tag{4}$$

Pada permasalahan SVM, margin dimaksimalkan, sehingga :

$$\max_{w,b} \frac{2}{\|w\|} \tag{5}$$

Untuk mendapatkan *hyperplane* terbaik dapat digunakan metode *Quadratic Programming* (QP) yaitu dengan cara meminimalkan

$$\min \ \tau(w) = \frac{1}{2}||w||^2 \tag{6}$$

dengan syarat

$$y_i(w^T . x_i + b) \geq 1 \ , \ i = 1,2, \dots, l \tag{7}$$

Persoalan ini akan lebih mudah diselesaikan jika diubah ke dalam formula *lagrangian* yang menggunakan *lagrange multiplier*. Dengan demikian solusi untuk optimalisasi ini adalah sebagai berikut:

$$L(w, b, \alpha) = \frac{1}{2} \| w \|^2 - \sum_{i=1}^{l} \alpha_i (y_i (w^T . x_i + b) - 1) \tag{8}$$

Nilai optimal dari persamaan tersebut dapat dihitung dengan meminimumkan L terhadap *w* dan *b* sekaligus memaksimalkan L terhadap α_i. Hal ini seperti kasus *dual problem* $\max_{\alpha}(\min_{w,b} L)$ dimana nilai minimum *lagrange* dengan syarat sebagai berikut [5]:

$$\max_{\alpha} Ld = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j (x_i^T . x_j) \quad (9)$$

dengan syarat

$$\alpha_i \geq 0 \text{ dan } \sum_{i=1}^{l} \alpha_i y_i = 0 \quad (10)$$

data latih dengan

$\alpha_i > 0$ = terletak pada *hyperplane* yang disebut *support vector*

$\alpha_i = 0$ = tidak terletak pada *hyperplane*

2.7.2. *Soft Margin*

Formulasi untuk mengklasifikasi dua kelas yang terpisah secara nonlinier dan lebih tahan terhadap derau maupun pencilan dapat menggunakan konsep *soft margin* [11], yaitu dengan mengubah masalah *Quadratic Programming* (QP) dan batasan di atas dengan menambahkan sebuah variabel kendur atau *slack variable* $\xi_i > 0$ sehingga menjadi

$$min\, \tau(w, \xi) = \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{l} \xi_i \quad (11)$$

dengan batasan

$$y_i (w^T x_i + b) \geq 1 - \xi_i \,, \forall_i \quad (12)$$

Parameter C berfungsi untuk mengontrol optimasi antara margin dan kesalahan klasifikasi ξ. Semakin besar parameter C, semakin besar pula penalti terhadap kesalahan klasifikasi. Optimasi untuk persamaan (49) dengan batas bawah persamaan (50) fungsi *lagrange* sebagai berikut [18]:

$$L = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \left({x_i}^T . x_j \right) \quad (13)$$

dengan batas

$$0 \leq \alpha_i \leq C, i = 1,2,..,l \; dan \sum_{i=1}^{l} a_i y_i = 0 \quad (14)$$

Pada dasarnya SVM adalah sebuah *linear classifier* [11]. Namun, SVM dapat dikembangkan menjadi *nonlinear classifier*. Konsep *kernel trick* pada ruang berdimensi lebih tinggi dapat menangani *nonlinear classifier*. Secara umum fungsi untuk mengkonversi himpunan data pada ruang masukan (*input space*) ke dalam ruang fitur (*feature space*) yang berdimensi lebih tinggi dapat diformulasikan sebagai

$\Phi : R^p \rightarrow R^q$, dimana $p < q$ (15)

Pada dasarnya proses pembelajaran SVM untuk menemukan *support vector* hanya bergantung pada *dot product* dari data pada ruang fitur, yaitu $\Phi_i \Phi_j$. Perhitungan *dot product* dapat digantikan dengan fungsi kernel $K(x_i,x_j)$ yang mendefinisikan secara implisit fungsi transformasi Φ tersebut. Inilah yang disebut *Kernel Trick*, yang diformulasikan sebagai

$$K(x_i, x_j) = \Phi_i(x_i) \cdot \Phi_j(x_j) \quad (16)$$

Pada umumnya terdapat empat jenis fungsi *kernel* yang dapat digunakan [8] yaitu:

1. *Kernel Linear*

$$K(x_i, x_j) = x_i^T x_j \quad (17)$$

2. *Kernel Polynomial*

$$K(x_i, x_j) = (\gamma\, x_i^T x_j + r)^d, \gamma > 0 \quad (18)$$

3. *Kernel Gaussian* (*Radial Basis Function*)

$$K(x_i, x_j) = \exp\{-\gamma \parallel x_i - x_j \parallel^2\}, \gamma > 0 \quad (19)$$

4. *Kernel Sigmoid*

$$K(x, x_k) = \tanh[\gamma\, x_i^T x_j + r] \quad (20)$$

2.8. Seleksi Model Klasifikasi dengan Metode *k-Fold Cross Validation*

Sesuai dengan namanya, metode *k-Fold Cross-Validation* mempartisi himpunan data D secara acak menjadi *k fold* (sub himpunan) yang saling bebas: f_1, $f_{2,\ldots,}$ f_k, sehingga masing-masing *fold* berisi *1/k* bagian data [17]. Selanjutnya dapat membangun *k* himpunan data: D_1, D_2,…, D_k yang masing-masing berisi (k-1) *fold* untuk data latih, 1 *fold* untuk data uji. Pada metode *k-fold cross-validation* ini jumlah kemunculan setiap sampel dalam data latih pasti sama dan setiap sampel pasti muncul satu dan hanya satu kali dalam data uji.

2.9. Evaluasi Model dengan *Confusion Matrix*

Evaluasi bertujuan untuk menilai kuliatas *classifier*. Sebuah sistem klasifikasi harus dinilai performanya agar dapat mengukur tingkat akurasi dari prediksi klasifikasi yang dihasilkan. Akurasi *Kappa* sangat dianjurkan karena dalam perhitungan akurasinnya menggunakan seluruh elemen dalam *confusion matrix* [1].

Tabel 1. Ukuran Evaluasi Model Klasifikasi

Ukuran	Rumus
Akurasi keseluruhan	$\frac{TP + TN}{P + N}$
Akurasi *Kappa*	$\frac{([P+N][TP+TN] - ([TP+FN][TP+FP] + [\mathrm{FP}+\mathrm{TN}][FN+TN]))}{(P+N)^2 - ([TP+FN][TP+FP] + [\mathrm{FP}+\mathrm{TN}][FN+TN])}$

Keempat istilah tersebut dapat digambarkan sebagai *confusion matrix* seperti diilustrasikan pada Tabel 3.

Tabel 2. *Confusion Matrix*

		Kelas Prediksi		Jumlah
		Positif	**Negatif**	
Kelas Aktual	**Positif**	TP	FN	TP+FN
	Negatif	FP	TN	FP+TN
Jumlah		TP+FP	FN+TN	P+N

3. METODOLOGI PENELITIAN

3.1. Jenis dan Sumber Data

Data yang digunakan merupakan data kualitatif berupa *tweets* dari pengguna media sosial Twitter. *Extrating tweets* dilakukan pada tanggal 18 Januari 2019 dengan kata kunci "gojek" dan kategori *tweets* berbahasa Indonesia sebanyak 10.000 *tweets*. Data yang diperoleh merupakan respon pengguna Gojek pada tanggal 12-18 Januari 2019. Data hasil *extrating tweets* tersebut dilakukan deteksi duplikat sehingga menjadi 6.917 *tweets*. Dari 6.917 *tweets* tersebut dipilih 1.500 *tweets* terbaru yang akan digunakan pada penelitian ini. Data 1.500 *tweets* tersebut merupakan respon pengguna Gojek pada tanggal 17-18 Januari 2019.

3.2. Langka-Langkah Analisis Data

Analisis data pada penelitian ini dilakukan dengan bantuan *software* RStudio 1.1.463 dan Microsoft Excel 2010. Adapun langkah-langkah analisis yang dilakukan adalah sebagai berikut:

1. *Extracting tweets*
2. *Pre-Processing* data.
3. Pelabelan data dengan *sentiment scoring* dan manual: melakukan pelabelan data menjadi label positif atau negatif dengan menggunakan *sentiment scoring* dan secara manual.
4. *Feature selection* : *filtering stopwords* dan *tokenization*
5. Pembobotan dokumen dengan TF-IDF
6. Membangun model klasifikasi *Suport Vector Machine* menggunakan fungsi *kernel linear* dan RBF, serta menggunakan evaluasi model *10-Fold Cross Validation.*
7. Menghitung akurasi keseluruhan dan akurasi *kappa* berdasarkan *confusion matrix.*
8. Menentukan fungsi *kernel* terbaik.

4. ANALISIS DAN PEMBAHASAN

4.1 *Extracting Tweets*

Extracting tweets dilakukan untuk mengumpulan data teks dari aplikasi Twitter dengan menggunakan API. Untuk melakukan *extracting tweets* dibutuhkan empat kode akses, yaitu API *Key,* API *Secret, Access Token, and Access Token Secret.* Kode-kode tersebut diperoleh setelah mendaftarkan akun Twitter pada https://apps.twitter.com/app/new.

4.2 *Pre-Processing* Data

Proses ini merupakan tahap yang dilakukan untuk transformasi data dari yang berbentuk tidak terstruktur menjadi data yang terstruktur agar mudah untuk tahap penelitian selanjutnya. Tahapan yang dilakukan diantaranya adalah sebagai berikut:

1. *Case Folding*

Case folding akan mengubah huruf menjadi satu bentuk dalam dokumen teks. Pada tahapan ini semua teks ditransformasi menjadi huruf kecil (*lowercase*).

2. *Remove* URL

Remove URL akan menghapus *link* URL (*Uniform Resource Locator*) yang terdapat pada dokumen teks. *Link* URL biasanya mengandung kata "http://".

3. *Unescape* HTML

Unescape HTML akan menghapus *file* HTML dan menghapus jejak karakter yang bisa dianggap sebagai *markup*.

4. *Remove Mention*

Remove Mention akan menghapus kata yang ditandai dengan simbol "@" pada bagian depan sebelum subjek. *Mention* berarti menyebutkan *username* pengguna Twitter lain pada saat mengunggah cuitan di Twitter.

5. *Remove Number*

Remove number akan menghapus semua angka yang terdapat pada dokumen teks karena angka tidak menunjukkan suatu perasaan.

6. *Remove punctuation*

Remove punctuation akan menghapus tanda baca yang ada pada dokumen. Karena penelitian ini hanya mengklasifikasikan data teks, maka selain karakter alphabet akan dihapus dari dokumen teks.

7. *Remove Emoticon*

Emoticon akan mempengaruhi hasil sentimen secara signifikan, maka peneliti menghapus *emoticon* pada dokumen teks. Karena *emoticon* yang terbaca pada dokumen tersusun dari tanda baca, maka *emoticon* sudah otomatis terhapus saat *remove punctuation.*

8. Normalisasi Kalimat

Kamus normalisasi kalimat dibuat secara manual di *notepad* sebanyak 548 kata. Jika ada kata singkatan yang cocok dengan kamus akan digantikan dengan kata baku.

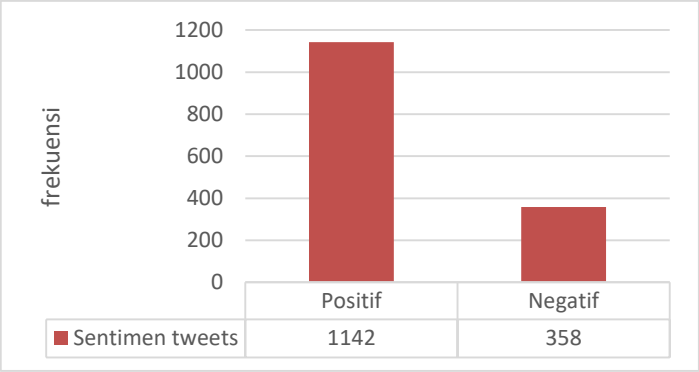
4.3 Pelabelan Data

Pelabelan data dilakukan dengan dua cara, yaitu secara manual dan *sentiment scoring*.

4.3.1. Pelabelan Data Secara Manual

Histogram di bawah ini diperoleh berdasarkan hasil pelabelan data *tweets* secara manual. Berdasarkan hasil *crawling* 1500 *tweets* teratas yang membicarakan Gojek diperoleh sentimen positif sebanyak 1142 *tweets* dan sentimen negatif sebanyak 358 *tweets*.

Sentimen negatif yang diberikan oleh pengguna terhadap Gojek adalah mengenai kekecewaan terhadap *driver* Gojek maupun terhadap sistem aplikasi Gojek yang merugikan pengguna. Dilihat dari banyaknya pengguna yang memberikan sentimen positif terhadap Gojek, menunjukan bahwa Gojek telah mampu memuaskan pengguna dalam fitur-fitur yang tersedia.



Gambar 3. Histogram Pelabelan Data Secara Manual

4.3.2. Pelabelan Data dengan Sentiment Scoring

setiap *tweets* yang ada pada dokumen teks dihitung skornya dengan cara menjumlahkannya dengan ketentuan sebagai berikut:

1. Setiap kata pada *tweets* yang terdapat pada kamus sentimen akan mendapat skor sesuai dengan yang ada pada kamus sentimen, jika kata tersebut tidak terdapat pada kamus sentimen akan mendapat skor 0.
2. Setiap kata yang terdapat kata negasi pada kata sebelumnya akan mendapatkan skor yang berlawanan pada kamus sentimen. Contoh: kata "adil" pada suatu *tweets* akan mendapat skor +5, jika sebelum kata "adil" terdapat kata negasi "tidak" maka skor yang diperoleh adalah -5.
3. Jika suatu kata yang terdapat pada kamus sentimen bernilai >0 yang diikuti kata *boosterwords* pada kata sebelumnya atau kata selanjutnya, maka skor kata sentimen ditambah skor kata *boosterwords.*
4. Jika suatu kata yang terdapat pada kamus sentimen bernilai <0 yang diikuti kata *boosterwords* pada kata sebelumnya atau kata selanjutnya, maka skor kata sentimen dikurang skor kata *boosterwords.*

Setiap *tweets* diberi label positif jika skor akhir yang diperoleh ≥ 0, dan diberi label negatif jika skor akhir yang diperoleh < 0 seperti perintah pada Lampiran 10. Hasil penjumlahan skor pada setiap *tweets* yang diperoleh dari ketiga kamus tersebut dapat dilihat pada perhitungan berikut:

1 gojek lebih enak saja

lebih (*boosterwords*)	= 1
enak (kamus sentimen)	= 4
Total Skor	**= 5**

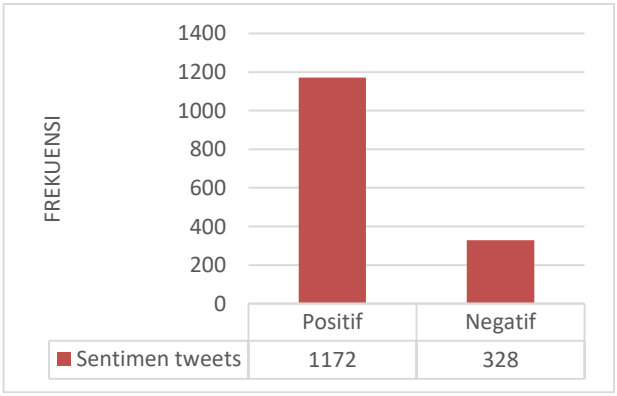
Label kelas : Positif

2 aku juga sudah tidak setia ke gojek

setia (kamus sentimen)	= 5
tidak (negasi)	
Maka :	
tidak setia (negasi dari setia)	= -5
Total Skor	**= -5**

Label kelas : Negatif

Berdasarkan pelabelan data yang telah dilakukan dengan *sentiment scoring*, diperoleh *tweets* yang berlabel positif sebanyak 1172, sedangkan *tweets* yang berlabel negatif sebanyak 328.



Gambar 4. Histogram Pelabelan Data dengan *Sentiment Scoring*

Sentiment scoring memberikan banyak kesalahan dalam melakukan pelabelan data. Setelah dilakukan pelabelan data secara manual pada 1500 *tweets* terdapat 172 *tweets* yang salah dalam pelabelan, artinya sebesar 11.46 % *tweets* mendapat label yang tidak tepat.

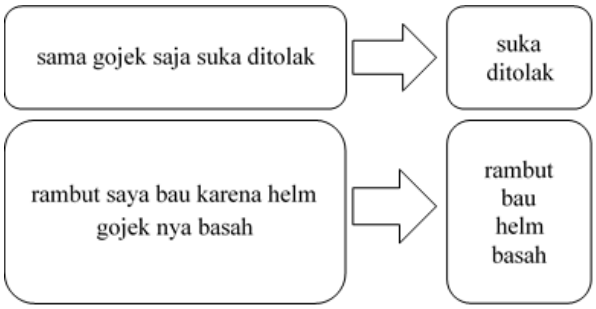
4.4 *Feature Selection*

4.4.1 *Stopwords*

Banyaknya *tweets* yang digunakan dalam penelitian menyebabkan banyaknya frasa atau fitur yang terbentuk. Pemilihan kata yang bermakna dengan menghilangkan kata yang kurang penting dalam membangun model dapat meningkatkan hasil akurasi sistem klasifikasi. *Stopwords* yang digunakan pada penelitian ini berjumlah 589 kata. *Stopwords* yang digunakan dapat dilihat pada Lampiran 7. *Term* yang terbentuk dari 1500 data sebanyak 4102 *term*. Setelah melalui proses *filtering stopwords*, *term* yang terbentuk sebanyak 3857 *term*.

4.4.2 *Tokenizing*

Pada proses *Tokenizing* dilakukan pemotongan dokumen menjadi bagian-bagian kata yang disebut token. Spasi digunakan untuk memisahkan antar kata tersebut. Gambar 20 merupakan token-token dari proses *tokenizing* yang terbentuk setelah dilakukan *filtering stopwords.*



Gambar 5. Proses *Filtering Stopwords* dan *Tokenizing*

4.5 Pembobotan

Nilai pembobotan kata dengan *Term Frequency-Inverse Document Frequency* (TF-IDF) dapat dilihat pada Tabel. Pembobotan kata akan digunakan untuk membangun model klasifikasi.

Tabel 3. Pembobotan dengan *Term Frequency-Inverse Document Frequency*

	Bagus	**cepat**	**dikirim**	**halal**	**percaya**	**senang**	**tidak**	...	**zonk**
Tweet ke-1	0,686	0	0	0,748	0	0	0,205	...	0
Tweet ke-2	0	0,638	0,638	0	0	0	0	...	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮
Tweet ke-1500	0	0	0	0	0,815	0,724	0	...	0

4.6 Klasifikasi Sentimen

4.6.1 Data Latih dan Data Uji

Algoritma klasifikasi SVM menggunakan data latih untuk membentuk model *classifier*, model yang terbentuk akan digunakan sebagai prediksi kelas data baru yang belum pernah ada sebelumnya. Data latih dan data uji yang digunakan adalah data yang telah memiliki label kelas, dengan perbandingan data latih dan data uji adalah 90% : 10%.

Tabel 4. Proporsi Kelas Sentimen Hasil Pelabelan Secara Manual Pada Data Latih dan Data Uji

Klasifikasi	Positif	Negatif	Jumlah
Data Latih	1025	326	1351
Data Uji	117	32	149
Jumlah	1142	358	1500

Tabel 5. Proporsi Kelas Sentimen Hasil Pelabelan dengan *Sentiment Scoring* Pada Data Latih dan Data Uji

Klasifikasi	Positif	Negatif	Jumlah
Data Latih	1058	293	1351
Data Uji	114	35	149
Jumlah	1172	328	1500

4.6.2 Klasifikasi *Support Vector Machine*

Pada penelitian ini digunakan metode *Support Vector Machine* (SVM) dengan fungsi *kernel* yang digunakan adalah *kernel linear* dan *kernel* RBF. Pengujian pengaruh parameter *Support Vector Machine* dilakukan untuk mengetahui nilai-nilai parameter SVM yang optimal untuk proses analisis sentimen. Pada *kernel linear* terdapat satu parameter yang diuji yaitu nilai *Cost* dengan nilai parameter *Cost* (C) : 0,01; 0,1; 1; 10; 100; 1000 untuk data latih. Hasil dari pengujian pengaruh nilai *Cost* pada model linier hasil pelabelan secara manual ditunjukkan pada Tabel 17, sedangkan pada model linier hasil pelabelan dengan *sentiment scoring* ditunjukkan pada Tabel 18. Penelitian ini menggunakan 10-*cross validation* untuk menguji performa *machine* dalam membentuk klasifikasi.

Tabel 6. Nilai Akurasi Keseluruhan dan Akurasi Kappa pada Model Linier Hasil Pelabelan Secara Manual

Evaluasi Model	*Cost* (C)					
	0,01	**0,1**	**1**	**10**	**100**	**1000**
Akurasi Keseluruhan	0,7919	0,7919	0,7852	0,7785	0,7785	0,7785
Akurasi *Kappa*	0,1105	0,1105	0,0966	0,0832	0,0832	0,0832

Tabel 7. Nilai Akurasi Keseluruhan dan Akurasi Kappa pada Model Linier Hasil Pelabelan dengan *Sentiment Scoring*

Evaluasi Model	*Cost* (C)					
	0,01	**0,1**	**1**	**10**	**100**	**1000**
Akurasi Keseluruhan	0,7919	0,7718	0,7584	0,7651	0,7651	0,7651
Akurasi *Kappa*	0,21	0,1673	0,0948	0,0833	0,0571	0,0571

Nilai C yang paling optimal pada Tabel 7 adalah 0,01 dan 0,1 karena memperoleh nilai akurasi keseluruhan dan akurasi *kappa* paling besar yaitu 79,19%

dan 11,05%. Sedangkan pada Tabel 8 menunjukkan bahwa nilai C paling optimal adalah 0,01, karena memperoleh nilai akurasi keseluruhan dan akurasi *kappa* paling besar yaitu 79,19% dan 21%.

Berdasarkan persamaan (69) pada *kernel* RBF terdapat dua parameter yang diuji yaitu nilai *Cost* (C) dan *Gamma* (γ). *Gamma* (γ) yang digunakan adalah 0,00026 dan nilai parameter *Cost* (C): 0,01; 0,1; 1; 10; 100; 1000 untuk data latih. Hasil dari pengujian pengaruh nilai *Cost* pada performa *kernel* RBF dengan nilai *gamma* tetap ditunjukkan pada Tabel 21 dan Tabel 22. Penelitian ini menggunakan 10-*cross validation* untuk menguji performa *machine* dalam membentuk klasifikasi.

Tabel 8. Nilai Akurasi Keseluruhan dan Akurasi Kappa pada Model RBF Hasil Pelabelan Secara Manual

Evaluasi Model	***Cost* (C)**					
	0,01	**0,1**	**1**	**10**	**100**	**1000**
Akurasi Keseluruhan	0,7852	0,7919	0,7919	0,7919	0,7919	0,7919
Akurasi *Kappa*	0	0,0804	0,1105	0,1105	0,1105	0,1652

Tabel 9. Nilai Akurasi Keseluruhan dan Akurasi Kappa pada Model RBF Hasil Pelabelan dengan *Sentiment Scoring*

Evaluasi Model	***Cost* (C)**					
	0,01	**0,1**	**1**	**10**	**100**	**1000**
Akurasi Keseluruhan	0,7584	0,7852	0,7919	0,7919	0,7785	0,7584
Akurasi *Kappa*	-0,0132	0,1501	0,21	0,1881	0,1812	0,0948

Berdasarkan pada Tabel 9 dari hasil pengujian nilai *Cost* dengan nilai *gamma* tetap, nilai C yang paling optimal adalah 1000 karena memperoleh nilai akurasi keseluruhan dan akurasi *kappa* paling besar yaitu 79,19% dan 16,52%. Sedangkan pada Tabel 10 menunjukkan bahwa nilai C paling optimal adalah 1, karena memperoleh nilai akurasi keseluruhan dan akurasi *kappa* paling besar yaitu 79,19% dan 21%.

Setelah melakukan analisis dengan menggunakan *kernel linear* dan *kernel* RBF diperoleh model terbaik dari masing-masing model sebagai berikut:

Tabel 10. Model Terbaik Kernel Linear dan Kernel Radial dari Hasil Pelabelan Data Secara Manual

Evaluasi Model	***Kernel Linear* (C=0.1)**	***Kernel* RBF (C=1000 dan γ=0.00026)**
Akurasi Keseluruhan	0,7919	0,7919
Akurasi *Kappa*	0,1105	0.1652

Tabel 11. Model Terbaik Kernel Linier dan Kernel RBF dari Hasil Pelabelan Data dengan *Sentiment Scoring*

Evaluasi Model	***Kernel Linear* (C=0.1)**	***Kernel* RBF (C=1 dan γ=0.00026)**
Akurasi Keseluruhan	0,7919	0,7919
Akurasi *Kappa*	0,21	0.21

Berdasarkan Tabel 11 dan 12, model *kernel linear* dan *kernel* RBF dari hasil pelabelan data secara manual dan *sentiment scoring* memiliki akurasi keseluruhan tertinggi yang sama yaitu 79,19%. Pada pelabelan data secara manual nilai akurasi *kappa* pada *kernel* RBF lebih besar daripada nilai akurasi *kappa* pada *kernel linear*, sedangkan pada pelabelan data dengan *sentiment scoring* mengasilkan akurasi *kappa* tertinggi yang sama yaitu 21%. Hal tersebut menunjukkan bahwa model dengan *kernel* RBF memiliki kecocokan hasil klasifikasi sentimen pada Gojek dengan benar dibandingkan menggunakan *kernel linear*.

5. KESIMPULAN

Berdasarkan analisis dan pembahasan terhadap sentimen pada Gojek di Twitter didapatkan beberapa kesimpulan sebagai berikut :

1. Pengguna Gojek pada tanggal 17 Januari 2019-18 Januari 2019 cenderung bersentimen positif di *twitter*, hal tersebut dibuktikan dari 1.500 *tweets* yang dianalisis dengan pelabelan secara manual sebanyak 1.142 *tweets* bersentimen positif dan sentimen negatif sebanyak 358 *tweets*.
2. Pelabelan data dengan menggunakan *sentiment scoring* belum bisa digunakan karena jumlah kesalahan yang dihasilkan terlalu banyak yaitu 172 *tweets* memperoleh label yang tidak tepat.
3. Hasil klasifikasi sentimen dari hasil pelabelan data secara manual menggunakan metode SVM pada Gojek menghasilkan tingkat akurasi keseluruhan terbaik sebesar 79,19% dan akurasi *kappa* terbaik sebesar 16,52%. Nilai akurasi keseluruhan dan *kappa* tersebut diperoleh dari pemodelan menggunakan *kernel* RBF dengan nilai *Cost*=1000 dan γ=0,00026.
4. Hasil klasifikasi sentimen dari hasil pelabelan data *sentiment scoring* menggunakan metode SVM pada Gojek menghasilkan tingkat akurasi keseluruhan terbaik sebesar 79,19% dan akurasi *kappa* terbaik sebesar 21%. Nilai akurasi keseluruhan dan *kappa* tersebut diperoleh dari pemodelan menggunakan *kernel* RBF dengan nilai *Cost*=1 dan γ=0,0002.

DAFTAR PUSTAKA

[1] Arisondang, V., Sudarsono, B. dan Prasetyo,Y. 2015. Klasifikasi Tutupan Lahan Menggunakan Metode Segmentasi Berbasis Algoritma Multiresolusi. *Jurnal Geodesi Undip* Volu. 4, No. 1, Hal: 9-19.

[2] Faret, J. dan Reitan, J. 2015. *Twitter Sentiment Analysis: Exploring the Effects of Linguistic Negation*. Norwegia: Norwegian University of Science and Technology,.

[3] Feldman, R dan Sanger, J. 2007. *The Text Mining Handbook*. New York: Cambridge University Press.

[4] Ghag, K. V. dan Shah, K. 2015. Comparative Analysis of Effect of Stopwords Removal on Sentiment Classification. *International Conference on Computer, Communication and Control* (IC4). India: Institute of Electrical and Electronics Engineers (IEEE).

[5] Gunn, S.R. 1998. *Support Vector Machine for Classification and Regression.* Southsmpton: Image Speech & Intelligent Systems Group University of Southampton.

[6] Gupta, V dan Lehal, G. S. 2009. A Survey of Text Mining Techniques and Applications. *Jurnal Emerging Technologies in Web Intelligence* Vol.1, No.1: Hal 60-76.
[7] Hootsuite. 2019. Local Insights. https://datareportal.com/reports/digital-2019-indonesia. Diakses: 7 April 2019.
[8] Hsu, C. W., Chang, C. C., dan Lin, C.J. 2010. *A Practical Guide to Support Vector Classification.* http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf. Diakses: 21 Juni 2019.
[9] Katadata. 2019. Persaingan Ketat Gojek dan Grab Menjadi SuperApp. https://katadata.co.id/telaah/2019/04/16/persaingan-ketat-gojek-dan-grab-menjadi-superapp. Diakses: 23 Juli 2019.
[10] Keerthi, S.S. dan Lin, C. J. 2003. Asymptotic Behaviors of Support Vector Machines with Gaussian Kernel. *Neural Computation* Vol. 15, No.7: 1667-1689.
[11] Kumar, V dan Wu, X. 2009. *The Top Ten Algorithms in Data Mining.* Boca Raton: Taylor & Francis Group.
[12] Nurhuda, F., Sihwi, S. W., dan Doewes, A. 2013. Analisis Sentimen Masyarakat terhadap Calon Presiden Indonesia 2014 berdasarkan Opini dari Twitter Menggunakan Metode Naive Bayes Classifier. *Jurnal* IT SMART Vol. 2, No. 2, Hal: 35-42.
[13] Rofiqoh, U., Perdana, R. S., dan Fauzi, M. A. 2017. Analisis Sentimen Tingkat Kepuasan Pengguna Penyedia Layanan Telekomunikasi Seluler Indonesia Pada Twitter Dengan Metode Support Vector Machine dan Lexicon Based Features. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* Vol. 1, No. 12, Hal: 1725-1732.
[14] Rozi, I. F., Pramono, S. H., dan Dahlan, E. A. 2012. Implementasi Opinion Mining (Analisis Sentimen) untuk Ekstraksi Data Opini Publik pada Perguruan Tinggi. *Jurnal Electrics, Electronics, Communications, Controls, Informatics, Systems* (EECCIS) Vol.6, No.1, Hal: 37-43.
[15] Salton, G. dan Buckley, C. 1988. Term-Weighting Approaches in Automatic Text Retrieval. *Jurnal Information Processing and Management* Vol.24, No. 5, Hal: 512-523.
[16] Santosa, B. 2007. *Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Yogyakarta: Graha Ilmu.
[17] Suyanto. 2019. *Data Mining: untuk Klasifikasi dan Klasterisasi Data Edisi Revisi*. Bandung: Informatika.
[18] Vapnik, V. dan Cortes, C. 1995. Support Vector Networks. *Jurnal Machine Learning*, 20, 273-297.
[19] Wahid, D. H. dan Azhari. 2016. Peringkasan Sentimen Esktraktif di Twitter Menggunakan Hybrid TF-IDF dan Cosine Similarity. *Indonesian Journal of Computing and Cybernetics Systems* (IJCCS), Vol. 10 , No. 2, Hal: 207-218.
[20] Yates, R. B., dan Neto, B. R. 1999. *Modern Information Retrieval*. New York: ACM Press.