
Analisis Sentimen Pada Maskapai Penerbangan di Platform Twitter Menggunakan Algoritma Support Vector Machine (SVM)

Hendry Cipta Husada^{1*}, Adi Suryaputra Paramita²

^{1,2}Program Studi Teknik Informatika, Universitas Ciputra, Surabaya, Jawa Timur
Email: ^{1*}hendrycipta7@gmail.com, ²adi.suryaputra@ciputra.ac.id

(Naskah masuk: 15 Des 2020, direvisi: 25 Jan 2021, diterima: 27 Jan 2021)

Abstrak

Perkembangan teknologi saat ini telah memberikan kemudahan bagi banyak orang dalam mendapatkan dan menyebarkan informasi di berbagai *social media platform*. Twitter merupakan salah satu media yang kerap digunakan untuk menyampaikan opini sebagai bentuk reaksi seseorang atas suatu hal. Opini yang terdapat di Twitter dapat digunakan perusahaan maskapai penerbangan sebagai parameter kunci untuk mengetahui tingkat kepuasan publik sekaligus bahan evaluasi bagi perusahaan. Berdasarkan hal tersebut, diperlukan sebuah metode yang dapat secara otomatis melakukan klasifikasi opini ke dalam kategori positif, negatif, atau netral melalui proses analisis sentimen. Proses analisis sentimen dilakukan dengan proses *data preprocessing*, pembobotan kata menggunakan metode TF-IDF, penerapan algoritma, dan pembahasan atas hasil klasifikasi. Klasifikasi opini dilakukan dengan *machine learning approach* memanfaatkan algoritma *multi-class Support Vector Machine (SVM)*. Data yang digunakan dalam penelitian ini adalah opini dalam bahasa Inggris dari para pengguna Twitter terhadap maskapai penerbangan. Berdasarkan pengujian yang telah dilakukan, hasil klasifikasi terbaik diperoleh menggunakan SVM kernel RBF pada nilai parameter $C(\text{complexity}) = 10$ dan $\gamma(\text{gamma}) = 1$, dengan nilai *accuracy* sebesar 84,37% dan 80,41% ketika menggunakan *10-fold cross validation*.

Kata Kunci: Twitter, Opini, Analisis Sentimen, Maskapai Penerbangan, *Machine Learning*, Klasifikasi, *Support Vector Machine*.

Sentiment Analysis of Airline on Twitter Platform Using Support Vector Machine (SVM) Algorithm

Abstract

Current technological developments have been given an easiness to a lot of people in terms of access to and distribute information on the various social media platform. Twitter is one of the media that often used for conveying opinion as a form of a reaction over a thing. The opinions contained on Twitter can be used by airline companies as a key parameter to measure the level of public satisfaction as well as evaluation materials for the company. Therefore, required a method that can automatically do opinion classifications into positive, negative, or neutral category through the sentiment analysis process. The sentiment analysis process is carried out with the data pre-processing process, word weighting using the TF-IDF method, the implementation of the algorithm, and analysis of the classification result. The opinion classification process is carried out with a machine learning approach using the multi-class Support Vector Machine (SVM) algorithm. The data used in this research are opinions in English vocabulary from Twitter users towards airlines. Based on experiments, the best classification results are obtained using SVM's RBF kernel on the parameter value $C(\text{complexity}) = 10$ and $\gamma(\text{gamma}) = 1$, with accuracy values 84.37% and 80.41% when using 10-fold cross validation.

Keywords: *Twitter, Opinion, Sentiment Analysis, Airline, Machine Learning, Classification, Support Vector Machine.*

I. PENDAHULUAN

Industri penerbangan adalah salah satu industri terbesar dan terkemuka di dunia karena memiliki peran besar terhadap kehidupan masyarakat. Salah satu negara yang memperoleh manfaat paling besar adalah Amerika yang setidaknya terdapat 23.911 penerbangan komersil dan membawa 2.246.004 penumpang pada setiap harinya [1]. Tingginya jumlah pengguna transportasi udara dikarenakan mobilitas masyarakat yang semakin meningkat dan tuntutan untuk melakukan perjalanan di dalam dan luar negeri dalam waktu yang singkat. Hal tersebut tentunya mendorong setiap perusahaan maskapai penerbangan untuk berusaha memperoleh keunggulan kompetitif dengan terus meningkatkan pelayanan mereka.

Perkembangan teknologi saat ini semakin mempermudah banyak orang dalam melontarkan opini di *social media online* sebagai bentuk respon dan tanggapan akan suatu hal. Dari sekian banyak *social media platform* yang tersedia, Twitter menjadi salah satu yang favorit untuk digunakan. Sepanjang tahun 2018, jumlah aktif pengguna Twitter pada tiap harinya berhasil mencapai angka 126 juta atau naik 9% dari tahun sebelumnya [2]. Twitter sendiri merupakan *social media* yang berjalan secara *real time*, memungkinkan para penggunanya membicarakan topik yang bervariasi dan mengemukakan pemikiran secara spontan [3]. Hal ini sering dimanfaatkan oleh industri-industri untuk menyebarkan dan mengumpulkan informasi di Twitter, salah satunya adalah industri penerbangan.

Opini yang diutarakan di Twitter dapat dijadikan sebagai parameter kunci untuk mengukur tingkat kepuasan publik terhadap suatu maskapai penerbangan. Informasi yang diperoleh dapat berupa pertanyaan, masukan, kritik, hingga apresiasi. Semua opini yang telah beredar tentunya dapat mempengaruhi citra dari perusahaan dan besar kemungkinan kecenderungan yang ada dalam opini tersebut dijadikan sebagai bahan evaluasi maupun alat penentu kebijakan. Maka dari itu penting bagi perusahaan untuk memberikan perhatian khusus pada hal tersebut. Namun, bukan perkara yang mudah bagi perusahaan dalam melakukan analisis terhadap kumpulan *tweet* dalam jumlah yang besar secara manual. Oleh sebab itu, dibutuhkan metode khusus yang dapat mengkategorisasi *tweet* ke dalam kelas positif, negatif, dan netral secara otomatis dengan menerapkan analisis sentimen.

Analisis sentimen merupakan bidang ilmu yang mempelajari cara untuk melakukan analisis terhadap opini, pendapat, dan penilaian dari suatu pihak terhadap objek tertentu yang dapat berupa layanan, produk, pihak, dan isu tertentu. Analisis sentimen dapat diimplementasikan pada beberapa level yang berbeda, baik itu teks yang berupa dokumen ataupun kalimat. Dalam melakukan analisis sentimen, terdapat beberapa algoritma yang dapat dimanfaatkan, salah satu diantaranya *Support Vector Machine* (SVM). Algoritma *Support Vector Machine* (SVM) merupakan algoritma yang telah digunakan dalam banyak bidang penelitian seperti *face detection*, bioinformatika, dan *text mining* [4]. Selain itu, algoritma ini juga memiliki tingkat *accuracy* yang lebih baik dalam proses klasifikasi teks

daripada algoritma lainnya seperti *K-Nearest Neighbors* (KNN), *C4.5* dan *Naïve Bayes* [5].

Berdasarkan hal tersebut, penelitian ini akan menerapkan analisis sentimen pada opini publik terhadap maskapai penerbangan di Twitter menggunakan algoritma *Support Vector Machine* (SVM) untuk melakukan klasifikasi *tweet* ke dalam kelas positif, negatif, dan juga netral.

II. LANDASAN TEORI

A. Twitter

Twitter merupakan layanan *social media micro blogging* yang memberi keleluasaan bagi penggunanya dalam menyampaikan pesan berupa opini dan penilaian atas suatu isu atau persoalan [2]. Pesan yang disampaikan umumnya disebut dengan istilah *tweet* atau kicauan yang memiliki kapasitas maksimal 280 karakter teks.

B. Analisis Sentimen

Analisis sentimen atau *opinion mining* adalah pengaplikasian dari ilmu komputasi linguistik, *text analysis*, dan *natural language processing* yang digunakan untuk mengekstrak subjektivitas dari sebuah sumber teks [6]. Analisis sentimen sendiri kerap digunakan untuk menyatakan suka atau tidak suka terhadap suatu hal melalui sentimen positif dan negatif. Hal tersebut dapat dijadikan sebagai parameter untuk proses pengambilan keputusan.

C. Text Mining

Text Mining adalah proses penemuan informasi dan pengetahuan dari sebuah sumber berbentuk teks yang belum terlihat sebelumnya, dengan tujuan memperoleh pola untuk tujuan yang spesifik [7]. Dalam penerapannya, *text mining* kerap digunakan dalam melakukan analisis informasi dan merupakan alat dalam proses pengambilan keputusan dengan memanfaatkan data teks dengan jumlah banyak. Sebagai tahap awal, perlu dilakukan proses *data preprocessing* yang terdiri atas beberapa metode untuk membersihkan dan menstandarisasi data teks yang akan diolah. Beberapa metode tersebut diantaranya *cleansing*, *case folding*, *tokenizing*, *filtering*, dan *stemming* yang dijelaskan lebih lanjut sebagai berikut:

1. Cleansing

Cleansing adalah proses pembersihan data teks dari komponen yang tidak diperlukan atau dianggap sebagai *noise*. Komponen yang akan dihilangkan antara lain karakter HTML, simbol *emoticon*, *hashtag* (#), *username account* (@username), *retweet* (RT), *link URL*, dan alamat *website*.

2. Case Folding

Case folding adalah proses standarisasi yaitu dengan mengubah seluruh elemen pada teks menjadi *lowercase*. Proses ini juga akan menghapus karakter selain huruf yang dianggap tidak berpengaruh terhadap proses pengolahan teks seperti angka dan tanda baca [8].

3. Tokenizing

Tokenizing adalah proses dimana kumpulan kata yang terangkai dalam sebuah kalimat akan dipisah menjadi pecahan kata tunggal atau berbentuk *token* [8].

4. *Filtering*

Filtering adalah proses penghapusan kata-kata yang termasuk dalam kategori *stopword* atau kata-kata yang dianggap tidak memiliki peran untuk proses analisis sentimen. Untuk menunjang proses ini digunakan *stopword list* yang akan dicocokkan pada data teks yang ada [8].

5. *Stemming*

Stemming adalah proses transformasi kata-kata yang sebelumnya terdapat imbuhan menjadi bentuk kata dasarnya sesuai dengan struktur morfologinya. Maka dapat diasumsikan bahwa kata yang mengalami transformasi memiliki arti dan makna yang sama dengan kata dasarnya [9].

D. Pembobotan TF-IDF

TF-IDF (*Term Frequency-Inverse Document Frequency*) merupakan salah satu metode untuk melakukan pembobotan kata dari proses ekstraksi kata dengan menerapkan perhitungan kata umum di *information retrieval*. Metode pembobotan ini merupakan penggabungan antara *term frequency* dan *inverse document frequency*. *Term frequency* merupakan jumlah kemunculan sebuah *term* pada sebuah dokumen. Besarnya jumlah *term* yang muncul berbanding lurus dengan pembobotan yang diberikan. Sedangkan *inverse document frequency* adalah proses untuk mengukur seberapa penting kata dalam suatu dokumen [10]. Nilai TF-IDF dapat diperoleh menggunakan persamaan (1), (2), dan (3).

$$W_{tf_{t,d}} = \begin{cases} 1 + \log_{10} tf_{t,d}, & \text{if } tf_{t,d} > 0 \\ 0, & \text{if } tf_{t,d} = 0 \end{cases} \quad (1)$$

$$idf_t = \log_{10} \frac{N}{df_t} \quad (2)$$

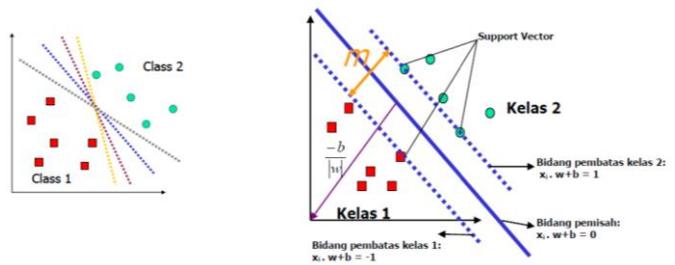
$$W_{t,d} = W_{tf_{t,d}} \cdot idf_t \quad (3)$$

Keterangan:

- $W_{tf_{t,d}}$ = Bobot kata di setiap dokumen
- $tf_{t,d}$ = Jumlah kemunculan *term* dalam dokumen
- N = Jumlah keseluruhan dokumen
- df = Jumlah dokumen yang mengandung *term*
- idf = Bobot *inverse* dalam nilai df
- $W_{t,d}$ = Pembobotan TF-IDF

E. *Support Vector Machine* (SVM)

Support Vector Machine (SVM) merupakan satu di antara banyak algoritma yang digunakan untuk klasifikasi dan termasuk dalam kategori *supervised learning*. Konsep kerja *Support Vector Machine* yaitu dengan mencari *hyperplane* atau garis pembatas paling optimal yang berfungsi untuk memisahkan dua kelas [11]. Konsep kerja dari algoritma *Support Vector Machine* (SVM) dapat dilihat pada Gambar 1.



Gambar 1. *Hyperplane Support Vector Machine* [12]

Untuk memperoleh garis *hyperplane* yang paling optimal dalam memisahkan data ke dua buah kelas tersebut, maka digunakan perhitungan *margin hyperplane* dan menemukan titik maksimal. Dalam memperoleh *hyperplane* pada SVM, dapat menggunakan persamaan (4).

$$(w \cdot x_i) + b = 0 \quad (4)$$

Di dalam data x_i , yang termasuk pada kelas -1 dapat dirumuskan seperti pada persamaan (5).

$$(w \cdot x_i + b) \leq 1, y_i = -1 \quad (5)$$

Sedangkan data x_i yang termasuk pada kelas +1 dapat dirumuskan seperti pada persamaan (6).

$$(w \cdot x_i + b) \geq 1, y_i = 1 \quad (6)$$

Dalam proses klasifikasi dengan *Support Vector Machine* (SVM) biasanya ditemui kondisi dimana *kernel linear* bekerja tidak optimal yang mengakibatkan hasil klasifikasi terhadap data menjadi buruk. Hal tersebut dapat diatasi dengan menggunakan *kernel non-linear* dengan memanfaatkan *kernel trick*. Dengan memanfaatkan *kernel trick*, akan dilakukan *mapping* data input ke *feature space* yang dimensinya lebih tinggi sehingga membuat data input yang dihasilkan akan terpisah secara *linear* dan membentuk *hyperplane* yang optimal [13]. Persamaan dari setiap kernel SVM dapat dilihat pada Tabel 1.

Tabel 1. Persamaan Setiap Kernel SVM

Kernel	Persamaan
<i>Linear</i>	$K(x_i, x) = x_i^T x$
<i>Polynomial</i>	$K(x_i, x) = (\gamma \cdot x_i^T x + r)^p, \gamma > 0$
<i>RBF</i>	$K(x_i, x) = \exp(-\gamma x_i - x ^2), \gamma > 0$
<i>Sigmoid</i>	$K(x_i, x) = \tanh(\gamma x_i^T x + r)$

F. *Confusion Matrix*

Confusion Matrix merupakan suatu instrumen yang digunakan untuk mengevaluasi performa dari model klasifikasi yang telah dihasilkan. Pada *confusion matrix*, hasil kelas prediksi akan dibandingkan dengan kelas data yang sebenarnya [14]. Hasil tersebut kemudian akan digunakan untuk menghitung nilai *accuracy*, *precision*, *recall*, dan *f-score*. Pengukuran evaluasi dengan *confusion matrix* dapat dilihat pada Tabel 2.

Tabel 2. *Confusion Matrix*

Data Aktual	Data Prediksi		
	TRUE	FALSE	TOTAL
TRUE	TP	FN	P
FALSE	FP	TN	N
TOTAL	P'	N'	P+N

Keterangan:

TP (*True Positive*) = Data positif yang terklasifikasi secara benar,

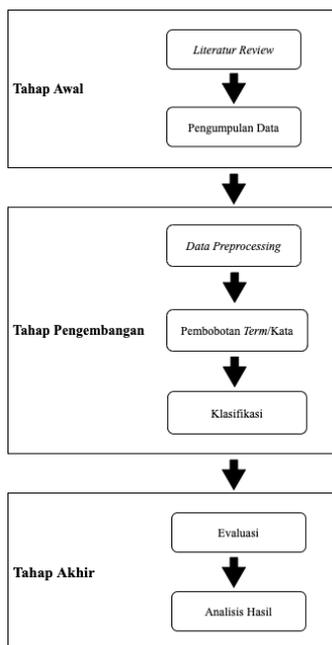
TN (*True Negative*) = Data negatif yang terklasifikasi secara benar.

FP (*False Positive*) = Data negatif yang terklasifikasi menjadi positif.

FN (*False Negative*) = Data positif yang terklasifikasi menjadi negatif.

III. METODOLOGI PENELITIAN

Dalam melakukan penelitian ini, setidaknya terbagi atas 3 tahapan besar, meliputi tahap awal, tahap pengembangan, dan tahap akhir. Setiap tahapan terdiri dari beberapa proses di dalamnya. Urutan tahapan secara lengkap dapat dilihat pada Gambar 2.



Gambar 2. Tahapan Penelitian

A. *Literatur Review*

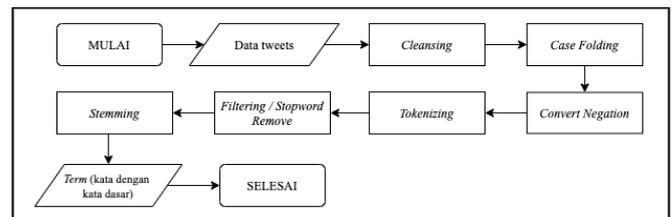
Dalam proses ini dilakukan pencarian sumber literatur mengenai analisis sentimen yang mampu mendukung penelitian ini secara teoritis. Literatur yang diperoleh bersumber dari buku, skripsi terdahulu, jurnal, dan sumber pustaka dari internet.

B. Pengumpulan Data

Dataset yang digunakan pada penelitian ini adalah kumpulan opini berupa *tweet* terhadap beberapa perusahaan maskapai penerbangan di negara Amerika pada periode 17 Februari - 24 Februari 2015 yang diambil dari tautan *dataset* yang digunakan dalam penelitian ini bersumber dari <https://www.figure-eight.com/data-for-everyone/>. Seluruh data *tweet* dimuat dalam bentuk format *CSV file*.

C. *Data Preprocessing*

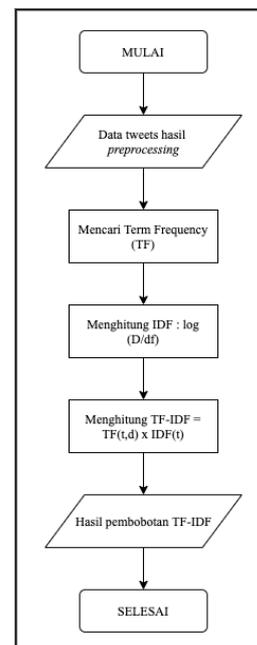
Sebelum masuk dalam proses klasifikasi, kumpulan data *tweet* harus melalui tahapan *preprocessing* untuk melakukan transformasi data yang awalnya berbentuk tidak terstruktur atau terdapat banyak *noise*, menjadi data yang terstruktur sehingga proses analisis menjadi lebih mudah. Beberapa metode yang akan diterapkan dalam proses *data preprocessing* dapat dilihat pada Gambar 3.



Gambar 3. Alur Proses *Data Preprocessing*

D. Pembobotan *Term/Kata*

Dalam proses ini, kumpulan *term/kata* akan divektorisasi atau diubah menjadi bentuk numerik. Hal tersebut perlu untuk dilakukan karena dalam klasifikasi hanya dapat memproses data dalam bentuk numerik saja. Proses pembobotan dalam penelitian ini menggunakan metode *Term Frequency-Inverse Document Frequency (TF-IDF)*. Alur proses pembobotan dengan TF-IDF dapat dilihat pada Gambar 4.



Gambar 4. Alur Proses Pembobotan TF-IDF

E. Klasifikasi *Support Vector Machine* (SVM)

Dalam proses klasifikasi, digunakan algoritma *Support Vector Machine* (SVM) untuk mengenali pola pada *data training* yang telah diberi label dan kemudian menghasilkan *machine learning model*. Model yang dihasilkan selanjutnya digunakan untuk mengenali data *tweet* baru yang belum diberi label sehingga dihasilkan sebuah prediksi kelas sentimennya apakah itu positif, negatif, atau netral.

F. Evaluasi

Proses selanjutnya setelah melakukan klasifikasi yaitu dengan melakukan evaluasi untuk menguji performa *model* dari algoritma *Support Vector Machine* (SVM). Evaluasi dilakukan menggunakan instrumen bernama *confusion matrix* untuk mengukur nilai *accuracy*, *precision*, *recall*, dan *f-score*.

G. Analisis Hasil

Pada proses terakhir dalam penelitian ini, dilakukan analisis secara deskriptif terhadap hasil pengujian algoritma *Support Vector Machine* (SVM) untuk melakukan klasifikasi sentimen.

IV. IMPLEMENTASI DAN HASIL

A. Dataset

Dataset yang digunakan dalam penelitian ini bersumber dari <https://www.figure-eight.com/data-for-everyone/> yang merupakan *platform* penyedia *dataset*. *Dataset* tersebut merupakan kumpulan opini dari pengguna Twitter yang ditunjukkan kepada beberapa maskapai penerbangan di Amerika dalam kurun periode 17-24 Februari 2015.

Dalam menunjang proses pengujian yang akan dilakukan, diperlukan data untuk proses *training* dan *testing*. Hal tersebut dilakukan agar dapat menghasilkan *machine learning model* yang digunakan untuk memprediksi kelas pada data baru. Untuk itu, dilakukan proses *sampling* data sebanyak 2.000 data *tweet* dari *dataset* semula yang memiliki lebih dari 14.000 *records* data *tweet*. Proses selanjutnya adalah labelisasi data *tweet* yang dilakukan dengan memanfaatkan kamus *lexicon* yang bersumber dari tautan <https://github.com/fnielsen/afinn/tree/master/afinn/data>. Di dalam kamus *lexicon* yang digunakan, terdapat setidaknya lebih dari 3.000 kata umum dalam bahasa Inggris yang memiliki sentimen positif dan negatif beserta nilai bobotnya masing-masing.

Penentuan label suatu data *tweet* adalah berdasarkan nilai akumulasi secara keseluruhan kata. Jika suatu *tweet* memiliki nilai > 0 maka termasuk dalam kelas positif, jika nilai < 0 maka termasuk dalam kelas negatif, dan jika nilai $= 0$ maka termasuk dalam kelas netral. Suatu *tweet* dapat termasuk dalam kelas netral karena 2 hal yaitu tidak terdapat kata yang memiliki sentimen pada *tweet* sesuai dengan kamus *lexicon*, dan jumlah nilai kata bersentimen positif sama dengan kata bersentimen negatif.

Setelah proses labelisasi data *tweet* berdasarkan kamus *lexicon*, dilakukan langkah validasi dengan melakukan pengecekan hasil secara manual. Hal ini dilakukan karena ada

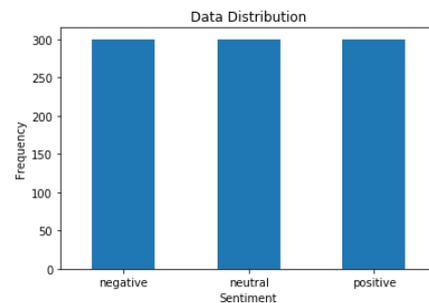
kemungkinan terdapat kata bersentimen pada data *tweet* yang tidak terdapat dalam kamus *lexicon* yang mengakibatkan hasil kelas atau label tidak sesuai.

Hasil dari proses labelisasi data *tweet* dapat dilihat pada Tabel 3.

Tabel 3. Hasil Labelisasi Data *Tweet*

Kelas Sentimen	Jumlah
Positif	320
Negatif	1257
Netral	423

Berdasarkan hasil labelisasi data *tweet*, terdapat ketidakseimbangan yang signifikan di antara 3 kelas. Kondisi tersebut jelas tidak ideal untuk dilakukan proses klasifikasi, sebab *unbalanced dataset* dapat membuat hasil prediksi dari *machine learning model* lebih cenderung pada data dengan kelas yang dominan. Untuk mengatasi kondisi tersebut, digunakan metode *under sampling* untuk menyeimbangkan jumlah data diantara tiga kelas sentimen dengan menyesuaikan pada kelas minoritas yaitu kelas sentimen positif. Hasil dari proses *under sampling* dapat dilihat pada Gambar 5.



Gambar 5. Distribusi Data Hasil *Under Sampling*

B. Data Preprocessing

Kumpulan data *tweet* yang diproses dalam penelitian ini termasuk dalam *unstructured text data*. Maka dari itu perlu dilakukan *data preprocessing* untuk membersihkan dan menstandarisasi data *tweet* tersebut yang meliputi proses *cleansing*, *case folding*, *convert negation*, *tokenizing*, *filtering*, dan *stemming*. Hal tersebut dilakukan agar menghasilkan data yang *proper* untuk diolah dan membuat hasil klasifikasi menjadi optimal. Hasil implementasi proses tersebut dapat dilihat pada Tabel 4, 5, 6, 7, 8, dan 9.

1. *Cleansing*

Tabel 4. Hasil Proses *Cleansing*

Sebelum	Hasil
@United is truly the drunk uncle of boarding. Don't believe gate agent that overhead is full; don't reserve aisle http://t.co/cdZhTyd0aK	is truly the drunk uncle of boarding. Do not believe gate agent that overhead is full; do not reserve aisle

2. Case Folding

Tabel 5. Hasil Proses Case Folding

Sebelum	Hasil
is truly the drunk uncle of boarding. Do not believe gate agent that overhead is full; do not reserve aisle	is truly the drunk uncle of boarding do not believe gate agent that overhead is full do not reserve aisle

3. Convert Negation

Tabel 6. Hasil Proses Convert Negation

Sebelum	Hasil
is truly the drunk uncle of boarding do not believe gate agent that overhead is full do not reserve aisle	is truly the drunk uncle of boarding do notbelieve gate agent that overhead is full do notreserve aisle

4. Tokenizing

Tabel 7. Hasil Proses Tokenizing

Sebelum	Hasil
is truly the drunk uncle of boarding do notbelieve gate agent that overhead is full do notreserve aisle	['is', 'truly', 'the', 'drunk', 'uncle', 'of', 'boarding', 'do', 'notbelieve', 'gate', 'agent', 'that', 'overhead', 'is', 'full', 'do', 'notreserve', 'aisle']

5. Filtering

Tabel 8. Hasil Proses Filtering

Sebelum	Hasil
['is', 'truly', 'the', 'drunk', 'uncle', 'of', 'boarding', 'do', 'notbelieve', 'gate', 'agent', 'that', 'overhead', 'is', 'full', 'do', 'notreserve', 'aisle']	['truly', 'drunk', 'uncle', 'boarding', 'notbelieve', 'gate', 'agent', 'overhead', 'full', 'notreserve', 'aisle']

6. Stemming

Tabel 9. Hasil Proses Stemming

Sebelum	Hasil
['truly', 'drunk', 'uncle', 'boarding', 'notbelieve', 'gate', 'agent', 'overhead', 'full', 'notreserve', 'aisle']	['truli', 'drunk', 'uncl', 'board', 'notbeliev', 'gate', 'agent', 'overhead', 'full', 'notreserv', 'aisl']

C. Pembobotan TF-IDF

Pada proses ini, dilakukan perhitungan jumlah kemunculan tiap term pada keseluruhan data tweet (TF), menghitung jumlah tweet yang mengandung setiap term (DF),

menghitung *inverse* dari nilai DF (IDF), dan mengkalikan nilai tf dan idf untuk memperoleh nilai bobot tiap tweet. Proses tersebut dilakukan menggunakan persamaan (1), (2), (3). Hasil pembobotan pada 3 sampel data tweet dapat dilihat pada Tabel 10.

Tabel 10. Hasil Pembobotan TF-IDF

Term	TF			DF	IDF	IDF		
	D1	D2	D3			D1	D2	D3
complet	1	0	0	1	0,477	0,477	0	0
awesom	2	0	0	1	0,477	0,954	0	0
experi	1	0	1	2	0,176	0,176	0	0,176
last	1	0	0	1	0,477	0,477	0	0
month	1	0	0	1	0,477	0,477	0	0
thank	1	0	0	1	0,477	0,477	0	0
flight	1	0	0	1	0,477	0,477	0	0
depart	1	0	0	1	0,477	0,477	0	0
time	1	0	0	1	0,477	0,477	0	0
...
travel	0	0	1	1	0,477	0	0	0,477
meet	0	0	1	1	0,477	0	0	0,477

D. Klasifikasi Support Vector Machine (SVM)

Proses klasifikasi ditentukan ke dalam 3 kelas yaitu positif, negatif, dan netral menggunakan *multi-class SVM one vs. one*. Dataset kemudian dibagi menjadi *training set* dan *testing set* menggunakan dengan *ratio* 0,1. Artinya 90 % data digunakan untuk proses *training*, dan 10% digunakan untuk proses *testing*.

Algoritma *Support Vector Machine* (SVM) sendiri memiliki beberapa fungsi kernel dan parameter yang dapat digunakan ketika proses klasifikasi. Parameter yang digunakan dapat ditentukan nilainya dan perbedaan nilai pada parameter dapat mempengaruhi performa dari *model*. Untuk itu, dilakukan proses pengujian berupa *parameter tuning* pada setiap fungsi kernel sehingga menghasilkan *machine learning model* yang memiliki performa optimal.

Pengujian dilakukan dengan memberikan *input* nilai pada masing-masing parameter yang digunakan seperti *degree*, *complexity*, *gamma*, dan *maximum iteration*. Proses tersebut dilakukan pada setiap fungsi kernel dari algoritma *Support Vector Machine* (SVM) sesuai dengan parameter yang dimiliki. Detail dari pengujian tersebut dapat dilihat pada Tabel 11.

Tabel 11. Skenario Pengujian Support Vector Machine

Kernel	Parameter	Nilai
Linear	Complexity (C)	[0,01; 0,1; 0,5; 1; 10; 20; 40; 50; 100; 1000]
	Maximum Iteration	[0,001; 0,01; 0,1; 1; 1,5; 2; 2,5; 10; 20; 100]
	Degree (d)	[0,01; 0,1; 0,5; 1; 10; 20; 40; 50; 100; 1000]
Polynomial	Complexity (C)	[0,001; 0,01; 0,1; 1; 1,5; 2; 2,5; 10; 20; 100]
	Gamma (γ)	[0,01; 0,1; 0,5; 1; 10; 20; 40; 50; 100; 1000]

RBF	Complexity (C)	[0,001; 0,01; 0,1; 1; 1,5; 2; 2,5; 10; 20; 100]
	Gamma (γ)	[0,01; 0,1; 0,5; 1; 10; 20; 40; 50; 100; 1000]
Sigmoid	Complexity (C)	[0,001; 0,01; 0,1; 1; 1,5; 2; 2,5; 10; 20; 100]
	Gamma (γ)	[0,01; 0,1; 0,5; 1; 10; 20; 40; 50; 100; 1000]

E. Hasil Pengujian Algoritma SVM

Proses pengujian dilakukan untuk menghasilkan *machine learning model* dengan performa yang paling optimal melalui evaluasi menggunakan *confusion matrix*.

1. Kernel Linear

Pada *kernel linear*, dilakukan pengujian terhadap parameter *complexity* dan *maximum iteration*. Dengan menggunakan nilai parameter *complexity* antara 0,01 hingga 1.000, nilai *testing accuracy* terus mengalami peningkatan hingga mencapai yang terbaik pada nilai *complexity* = 10, yaitu sebesar 83,33%. Nilai *complexity* yang semakin besar, menyebabkan nilai *accuracy* cenderung mengalami penurunan. Setelah itu dengan menggunakan nilai parameter *max iteration* antara 10 hingga 1.000 dan *complexity* = 10, nilai *testing accuracy* terus mengalami peningkatan hingga mencapai yang terbaik pada nilai *max iteration* = 250, yaitu sebesar 83,33%. Nilai *accuracy* juga tetap konstan pada nilai *max iteration* yang lebih besar.

Hasil *confusion matrix* dari penggunaan nilai *complexity* = 10 dan *maximun iteration* = 250 pada *kernel linear* dapat dilihat pada Gambar Tabel 12.

Tabel 12. Confusion Matrix Kernel Linear

Aktual	Prediksi		
	Negatif	Netral	Positif
Negatif	25	2	3
Netral	6	29	1
Positif	2	1	26

Dari *confusion matrix* tersebut, juga dilakukan perhitungan *precision*, *recall*, dan *f-score* sebagai berikut:

Precision

$$Positif = \frac{26}{26+2+2} = 0,87 \quad | \quad Netral = \frac{29}{29+6+1} = 0,81 \quad | \quad Negatif = \frac{25}{25+6+2} = 0,83$$

$$Average Recall = \frac{0,87+0,81+0,83}{3} \times 100\% = 83,67\%$$

Recall

$$Positif = \frac{26}{26+3+1} = 0,87 \quad | \quad Netral = \frac{29}{29+2+2} = 0,88 \quad | \quad Negatif = \frac{25}{25+6+2} = 0,76$$

$$Average Precision = \frac{0,87+0,88+0,76}{3} \times 100\% = 83,67\%$$

$$F-score = \frac{2 \times 83,67 \times 83,67}{83,67 + 83,67} = 83,67\%$$

2. Kernel Polynomial

Pada *kernel polynomial*, dilakukan pengujian terhadap parameter *complexity*, *gamma*, dan *degree*. Dengan menggunakan nilai parameter *complexity* antara 0,01 hingga

1.000, nilai *testing accuracy* terus mengalami peningkatan hingga mencapai yang terbaik pada nilai *complexity* = 10, yaitu sebesar 76,04%. Nilai *complexity* yang semakin besar, menyebabkan nilai *accuracy* cenderung mengalami penurunan. Setelah itu, dengan menggunakan nilai parameter *gamma* antara 0,001 hingga 100 dan *complexity* = 10, nilai *testing accuracy* mencapai yang terbaik pada nilai *gamma* = 1 yaitu sebesar 76,04%. Nilai *gamma* yang semakin besar juga menyebabkan nilai *accuracy* mengalami penurunan. Terakhir, dengan menggunakan nilai parameter *degree* antara 1 hingga 4, *complexity* = 10, dan *gamma* = 1, nilai *testing accuracy* mencapai yang tertinggi pada nilai *degree* = 1 yaitu sebesar 83,33%.

Hasil *confusion matrix* dari penggunaan nilai *complexity* = 10, *gamma* = 1, dan *degree* = 1 pada *kernel polynomial* dapat dilihat pada Tabel 13.

Tabel 13. Confusion Matrix Kernel Polynomial

Aktual	Prediksi		
	Negatif	Netral	Positif
Negatif	25	2	3
Netral	6	29	1
Positif	2	1	26

Dari *confusion matrix* tersebut, juga dilakukan perhitungan *precision*, *recall*, dan *f-score* sebagai berikut:

Precision

$$Positif = \frac{26}{26+2+2} = 0,87 \quad | \quad Netral = \frac{29}{29+6+1} = 0,81 \quad | \quad Negatif = \frac{25}{25+6+2} = 0,83$$

$$Average Recall = \frac{0,87+0,81+0,83}{3} \times 100\% = 83,67\%$$

Recall

$$Positif = \frac{26}{26+3+1} = 0,87 \quad | \quad Netral = \frac{29}{29+2+2} = 0,88 \quad | \quad Negatif = \frac{25}{25+6+2} = 0,76$$

$$Average Precision = \frac{0,87+0,88+0,76}{3} \times 100\% = 83,67\%$$

$$F-score = \frac{2 \times 83,67 \times 83,67}{83,67 + 83,67} = 83,67\%$$

3. Kernel RBF

Pada *kernel RBF*, dilakukan pengujian terhadap parameter *complexity* dan *gamma*. Dengan menggunakan nilai parameter *complexity* antara 0,01 hingga 1.000, nilai *testing accuracy* terus mengalami peningkatan hingga mencapai yang terbaik pada nilai *complexity* = 10 yaitu sebesar 84,37%. Dengan nilai *complexity* yang lebih besar, nilai *accuracy* yang dihasilkan tetap konstan. Setelah itu, dengan menggunakan nilai parameter *gamma* antara 0,001 hingga 100 dan *complexity* = 10, nilai *testing accuracy* terus mengalami peningkatan hingga mencapai yang terbaik pada nilai *gamma* = 1, yaitu sebesar 84,37%. Nilai *gamma* yang semakin besar, menyebabkan nilai *accuracy* yang dihasilkan cenderung mengalami penurunan.

Hasil *confusion matrix* dari penggunaan nilai *complexity* = 10, *gamma* = 1 pada *kernel RBF* dapat dilihat pada Tabel 14.

Tabel 14. Confusion Matrix Kernel RBF

Aktual	Prediksi		
	Negatif	Netral	Positif
Negatif	26	3	2
Netral	5	28	1
Positif	2	2	27

Dari confusion matrix tersebut, juga dilakukan perhitungan precision, recall, dan f-score sebagai berikut:

Precision

$$Positif = \frac{27}{27+2+2} = 0,87 \quad | \quad Netral = \frac{28}{28+5+1} = 0,82 \quad |$$

$$Negatif = \frac{26}{26+3+2} = 0,84$$

$$Average Recall = \frac{0,87+0,82+0,84}{3} \times 100\% = 84,33\%$$

Recall

$$Positif = \frac{27}{27+1+2} = 0,90 \quad | \quad Netral = \frac{28}{28+3+2} = 0,85 \quad |$$

$$Negatif = \frac{26}{26+5+2} = 0,79$$

$$Average Precision = \frac{0,90+0,85+0,79}{3} \times 100\% = 84,67\%$$

$$F-score = \frac{2 \times 84,67 \times 84,33}{84,67 + 84,33} = 84,50\%$$

4. Kernel Sigmoid

Pada kernel sigmoid, juga dilakukan pengujian terhadap parameter complexity dan gamma. Dengan menggunakan nilai parameter complexity antara 0,01 hingga 1.000, nilai testing accuracy terus mengalami peningkatan hingga mencapai yang terbaik pada nilai complexity = 10 yaitu sebesar 80,20%. Nilai complexity yang semakin besar menyebabkan nilai accuracy yang dihasilkan cenderung mengalami penurunan. Setelah itu, dengan menggunakan nilai parameter gamma antara 0,001 hingga 100 dan complexity = 10, nilai testing accuracy terus mengalami peningkatan hingga mencapai yang terbaik pada nilai gamma = 0,1 yaitu sebesar 82,29%. Nilai gamma yang semakin besar juga menyebabkan nilai accuracy yang dihasilkan mengalami penurunan.

Hasil confusion matrix dari penggunaan nilai complexity = 10, gamma = 0,1 pada kernel sigmoid dapat dilihat pada Tabel 15.

Tabel 15. Confusion Matrix Kernel Sigmoid

Aktual	Prediksi		
	Negatif	Netral	Positif
Negatif	26	2	2
Netral	5	27	2
Positif	2	4	26

Dari confusion matrix tersebut, juga dilakukan perhitungan precision, recall, dan f-score sebagai berikut:

Precision

$$Positif = \frac{26}{26+2+4} = 0,81 \quad | \quad Netral = \frac{27}{27+5+2} = 0,79 \quad |$$

$$Negatif = \frac{26}{26+2+2} = 0,87$$

$$Average Recall = \frac{0,81+0,79+0,87}{3} \times 100\% = 82,33\%$$

Recall

$$Positif = \frac{26}{26+2+2} = 0,87 \quad | \quad Netral = \frac{27}{27+2+4} = 0,82 \quad |$$

$$Negatif = \frac{26}{26+5+2} = 0,79$$

$$Average Precision = \frac{0,87+0,82+0,79}{3} \times 100\% = 82,67\%$$

$$F-score = \frac{2 \times 82,67 \times 82,33}{82,67 + 82,33} = 82,50\%$$

5. Rangkuman Pengujian Algoritma SVM

Berdasarkan pada pengujian yang telah dilakukan, dihasilkan model dengan nilai performa terbaik pada setiap fungsi kernel. Rangkuman dari hasil pengujian dapat dilihat pada Tabel 16.

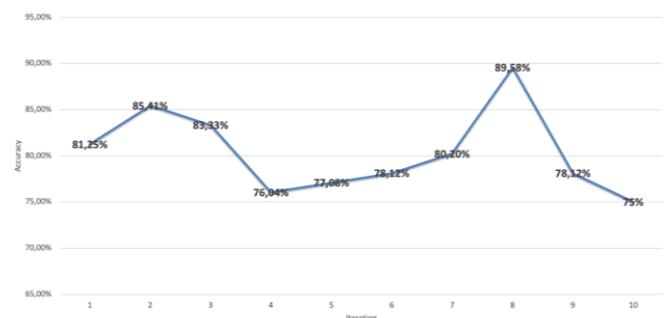
Tabel 16. Rangkuman Hasil Pengujian

Kernel	Parameter	Acc.	Prec.	Rec.	F-score
Linear	complexity = 10, max iteration = 250	83,33%	83,67%	83,67%	83,67%
Polynomial	complexity = 10, gamma = 1, degree = 1	83,33%	83,67%	83,67%	83,67%
RBF	complexity = 10, gamma = 1	84,37%	84,33%	84,67%	84,50%
Sigmoid	complexity = 10, gamma = 0,1	82,29%	82,33%	82,67%	82,50%

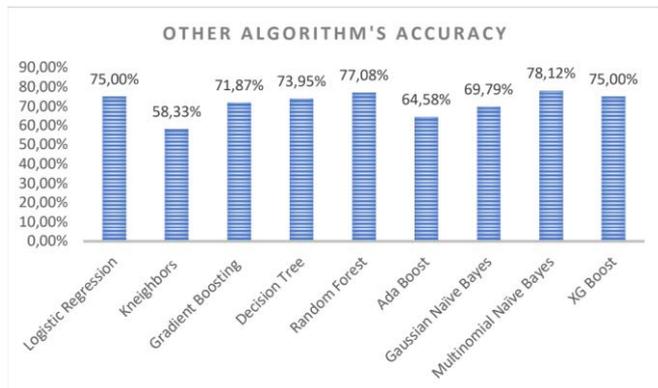
Hasil secara keseluruhan menunjukkan kernel RBF menghasilkan nilai accuracy yang terbaik. Penggunaan kernel RBF dengan parameter complexity = 10 dan gamma = 1 mampu menghasilkan nilai accuracy sebesar 84,37 %. Selain accuracy, kernel RBF juga menghasilkan nilai precision, recall, dan f-score yang terbaik dengan penggunaan nilai parameter yang sama.

F. Hasil Pengujian Cross Validation

Untuk melakukan validasi tingkat accuracy dari suatu model, maka dilakukan pengujian dengan metode 10-fold cross validation. Pengujian ini menggunakan fungsi kernel RBF dengan nilai parameter complexity = 10 dan gamma = 1 yang pada proses parameter tuning menghasilkan nilai accuracy, precision, recall, dan f-score terbaik. Hasil dari 10-fold cross validation dapat dilihat pada Gambar 6.



Gambar 6. Hasil 10-fold Cross Validation



Gambar 7. Grafik Perbandingan Accuracy Algoritma Lain.

Dari 10 kali eksperimen pengujian yang dilakukan, diperoleh nilai *accuracy* yang beragam dan diperoleh rata-rata *accuracy* sebesar 80,41% dan bisa terlihat hasil tersebut lebih baik dari *accuracy* algoritma yang lain, dimana pada algoritma yang *accuracy* terbaik diraih oleh *Multinomial Naïve Bayes* dengan 78,21%.

V. KESIMPULAN

Berdasarkan dari penelitian yang telah dilakukan, dapat disimpulkan bahwa algoritma *Support Vector Machine* (SVM) terbukti mampu diterapkan dengan baik untuk melakukan analisis 26sentiment dengan didukung oleh beberapa metode *preprocessing*, pembobotan *term* menggunakan TF-IDF, dan *parameter tuning* algoritma. Selain itu, pada proses pengujian algoritma *Support Vector Machine* (*parameter tuning*), nilai *accuracy* terbaik dihasilkan oleh kernel RBF dengan nilai parameter *complexity* = 10 dan *gamma* = 1, yaitu sebesar 84,37%. Sedangkan pada pengujian menggunakan *10-fold cross validation*, diperoleh rata-rata nilai *accuracy* sebesar 80,41%. Nilai *precision*, *recall*, dan *f-score* terbaik juga dihasilkan oleh kernel RBF dengan nilai parameter *complexity* = 10 dan *gamma* = 1, yang masing-masing sebesar 84,33%, 84,67%, dan 84,50%.

REFERENSI

- [1] FAA. (2017). *Federal Aviation Administration's Air Traffic by the Numbers*. Diakses dari: <https://www.transtats.bts.gov/airports.asp>
- [2] Zephoria. (2019). *Top 10 Twitter Statistics – Updated October 2019*. Diakses dari: <https://zephoria.com/twitter-statistics-top-ten/>
- [3] Hamdan, H., Bellot, P. & Bechet, F. (2015). Lsislif: Feature Extraction and Label Weighting for Sentiment Analysis in Twitter. *Proceedings of International Workshop on Semantic Evaluation (SemEval-2015)*, pp. 568.
- [4] Rainarli, E. & Romadhan, A. (2017). Perbandingan Simple Logistic Classifier Dengan Support Vector Machine Dalam Memprediksi Kemenangan Atlet. *Journal of Information Systems Engineering and Business Intelligence*, pp. 87-91.
- [5] Malheiro, R., Panda, R., Gomes, P. & Paiva, R. P. (2013). Music Emotion Recognition from Lyrics: A Comparative Study. *International Workshop on Machine Learning and Music*.
- [6] Barawi, M.H. & Seng, Y.Y. (2013). Evaluation of Resources Creations by Using Sentiment Analysis. *International Conference on Cognitive Science*. Elsevier Publisher Inc.
- [7] Mustafa, A., Akbar, A. & Sultan, A. (2009). Knowledge Discovery Using Text Mining: A Programmable Implementation on Information Extraction and Categorization. *International Journal of Multimedia and Ubiquitous Engineering*, Vol. 4(2), April.
- [8] Vijayarani, S., Ilamathi, J. & Nithya. (2015). Preprocessing Techniques for Text Mining - An Overview. *International Journal of Computer Science & Communication Networks*, Vol. 5(1), pp. 7-16.
- [9] Indriati & Ridok A. (2016). Sentiment Analysis For Review Mobile Application Using Neighbor Method Weighted K-Nearest Neighbor (NWKNN). *Journal of Environmental Engineering & Sustainable Technology*, Vol. 03(01), July 2016, pp. 23-32.
- [10] Akbari, M.I.H.A.D., Novianty, A. & Setianingsih, C. (2012). Analisis Sentimen Menggunakan Metode Learning Vector Quantization. *e-Proceeding of Engineering*, Vol. 4(2), Agustus 2017, pp. 2283-2292.
- [11] Lidya, S.K., Sitompul, O.S. & Efendi, S. (2015). Sentiment Analysis Pada Teks Bahasa Indonesia Menggunakan Support Vector Machine (SVM) dan K-Nearest Neighbor (K- NN). *Seminar Nasional Teknologi Informasi dan Komunikasi*.
- [12] Sembiring, K. (2007). *Penerapan Teknik Support Vector Machine Untuk Pendeteksian Intrusi Pada Jaringan* [Skripsi]. Program Studi Teknik Informatika, Institut Teknologi Bandung.
- [13] Shandra, E.N., Setiawan B.D., Sari Y.A. (2019). Klasifikasi Pola Sidik Bibir untuk Menentukan Jenis Kelamin Manusia dengan Metode Gray Level Co-Occurrence Matrix dan Support Vector Machine. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Vol. 3(3), pp. 2753-2760.
- [14] Salekhah, C.I. (2016). *Implementasi Metode Multi Class Support Vector Machine Untuk Klasifikasi Emosi pada Lirik Lagu Bahasa Indonesia*. Universitas Komputer Indonesia (Unikom).