DOI: http://dx.doi.org/10.21111/fij.v6i2.5536
Diterima: 22 Januari 2021 Revisi: 04 Juli 2021 Terbit: 21 Juli 2021

Analisis Sentimen Pada Tweet Tentang Penanganan Covid-19 Menggunakan Word Embedding Pada Algoritma Support Vector Machine Dan K-Nearest Neighbor

Trifebi Shina Sabrila [1], Veronica Retno Sari [2], Agus Eko Minarno [3] *

Universitas Muhammadiyah Malang[1)2)3)]
trifebiss@gmail.com [1], veronica.maralovi@gmail.com [2], aguseko@umm.ac.id [3] *

Abstrak

Analisis sentimen merupakan salah satu bidang dari pengolahan data berbentuk teks untuk mengidentifikasi isi yang terkandung dalam teks pada dataset dengan membagi dataset ke dalam dua kelas yaitu sentimen positif dan sentimen negatif. Pada penelitian ini, dilakukan analisis sentimen terhadap data yang diperoleh dari jejaring sosial Twitter mengenai penanganan Covid-19 oleh pemerintah di Indonesia yang menuai banyak pro dan kontra oleh masyarakat di Indonesia. Metode klasifikasi yang digunakan dalam penelitian ini adalah Support Vector Machine (SVM) dan K-Nearest Neighbor (KNN) dengan ekstraksi fitur Word Embedding untuk membandingkan performa dari kedua performa metode tersebut. Pengklasifikasian yang dilakukan dengan menggunakan algoritma Support Vector Machine (SVM) dengan menggunakan ekstraksi fitur Word Embedding yaitu Word2Vec menghasilkan akurasi sebesar 85%, presisi 86%, recall 85%, dan nilai AUC sebesar 0.92. Sementara pada algoritma K-Nearst Neighbor (KNN) dengan ekstraksi fitur yang sama, dihasilkan akurasi sebesar 76%, presisi 77%, recall 76% dan nilai AUC sebesar 0.87. Hasil perbandingan dari kedua metode menunjukkan bahwa algoritma Support Vector Machine (SVM) mendapatkan performa yang lebih baik dibandingkan algoritma K-Nearest Neighbor (KNN).

Kata kunci: *Analisis Sentimen, SVM, KNN, Word2Vec, Covid-19*

Abstract

[Sentiment Analysis of Tweets on the Handling of Covid-19 using Word Embedding on Support Vector Machine and K-Nearest Neighbor Algorithm] *Sentiment analysis is a field of text processing that identifies a set of data into two classes, positive and negative. In this study, sentiment analysis will be done with the use of data obtained from social media Twitter about the handling of Covid-19 by the government in Indonesia which has reaped many pros and cons by the people in Indonesia. The classification method used in this study are Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) using Word Embedding feature extraction to compare the performance of the two methods. Classification using the Support Vector Machine (SVM) algorithm using Word Embedding feature extraction, namely Word2Vec gets an accuracy of 85%, precision 86%, recall 85%, and an AUC value of 0.92. Meanwhile, the K-Nearest Neighbor (KNN) algorithm with the same feature extraction results in an accuracy of 76%, precision 77%, recall 76%, and an AUC value of 0.87. The result of the two methods show that the Support Vector Machine (SVM) algorithm gets better performance than the K-Nearest Neighbor (KNN) algorithm.*

Keywords: *Sentiment Analysis, SVM, KNN, Word2Vec, Covid-19*

1. PENDAHULUAN

Sejak awal tahun 2020, dunia dikejutkan dengan pandemi Covid-19 (*Coronavirus Disease 2019*) yang disebabkan oleh virus SARS-Cov-2 yang menginfeksi seluruh negara di dunia. WHO (*World Health Organization*) telah menyatakan dunia masuk ke dalam darurat global terkait virus ini. Virus ini pertama kali masuk ke Indonesia pada awal Maret 2020 lalu dan hingga saat ini diprediksi akan terus berlanjut hingga batas waktu yang belum diketahui [1]. Di Indonesia sendiri, hingga saat ini pasien yang terinfeksi Covid-19 telah menyebar ke Provinsi dan 432 Kabupaten/Kota, bahkan tercatat 598.933 kasus yang terkonfirmasi positif dan 18.336 kasus meninggal pertanggal 10 Desember 2020 [2]. Menurut penelitian terbaru dari tim AS, penyebaran Covid-19 membuat perilaku masyarakat berubah semenjak dilakukannya pembatasan jarak sosial, karantina, dan anjuran tinggal dirumah, dimana hal ini memiliki efek yang sangat signifikan pada penggunaan media sosial oleh masyarakat yang naik hingga 60% [3].

Penggunaan media sosial yang semakin meningkat menjadikan media sosial sebagai salah satu *platform* bagi masyarakat untuk menceritakan kehidupan sehari-hari, mengutarakan keluh kesah, hingga menyampaikan opini terhadap suatu isu. Salah satu yang hangat diperbincangkan di media sosial terutama Twitter hingga saat ini ialah mengenai opini masyarakat terkait penanganan pandemi Covid-19 oleh pemerintah di Indonesia. Peran pemerintah dalam penanganan Covid-19 ini menuai banyak pro dan kontra dari masyarakat di Indonesia. Sebagian masyarakat merasa pemerintah sudah melakukan yang terbaik dalam mengantisipasi dan menangani Covid-19, sebagian lainnya merasa pemerintah masih kurang tegas dan tidak serius dalam menangani Covid-19 di Indonesia. Permasalahan tersebut menarik perhatian penulis untuk dilakukannya analisis sentimen terhadap kumpulan *tweet* opini masyarakat mengenai topik tersebut untuk mengetahui respon masyarakat terkait topik tersebut.

Analisis sentimen adalah salah satu bidang ilmu dari *Natural Language Processing* (NLP) [4] dan merupakan proses yang digunakan untuk melakukan penggalian, ekstraksi, dan pengolahan data tekstual untuk membantu mengidentifikasi isi yang terkandung dalam teks opini tersebut termasuk ke dalam sentimen positif ataupun negatif [5][6][7].

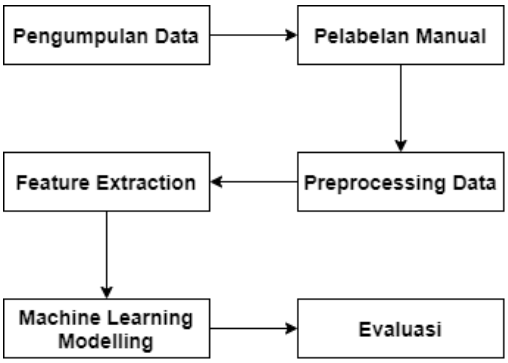
Pada beberapa penelitian terdahulu, Hennie Tuhuteru, dkk, telah melakukan analisis sentimen terhadap tweet yang mengandung topik tentang pembatasan sosial berskala besar menggunakan algoritma *Support Vector Machine* dan menghasilkan sentimen positif sebanyak 28%, sentimen negatif sebanyak 27% dan sentimen netral sebanyak 45% dengan hasil akurasi sebesar 82,07% [8]. Analisis sentimen terkait opini masyarakat mengenai *Corona Virus* pada sosial media Twitter juga dilakukan oleh Ricky, dkk, menggunakan 3 jenis algoritma *machine learning*, yaitu *Support Vector Machine* (SVM), *K-Nearest Neighbor* (KNN), dan *Naïve Bayes* dengan pembobotan menggunakan TF-IDF. Penelitian ini mengklasifikasikan sentimen ke dalam 3 polaritas, positif, netral, dan negatif. Hasil uji coba menunjukkan bahwa algoritma *Support Vector Machine* memberikan hasil yang terbaik jika dibandingan dengan kedua algoritma lainnya dengan akurasi sebesar 76.21% [9].

Farhan W.K., dkk, juga melakukan penelitian terkait analisis sentimen pada *tweet* berbahasa Indonesia dengan menggunakan model *Word2Vec* sebagai ekstraksi fiturnya dan algoritma *Support Vector Machine* (SVM) untuk pengklasifikasiannya. Model yang diusulkan menghasilkan nilai presisi sebesar 64.4%, *recall* sebesar 58%, dan *f-score* sebesar 61.1%. Rendahnya hasil pengujian yang didapatkan disebabkan oleh kurangnya jumlah dataset yang digunakan dalam penelitian [10]. Selain itu, Sharma, dkk, melakukan penelitian mengenai analisis sentimen *tweet* dengan digunakan metode *Word2Vec* yang merupakan salah satu model *Word Embedding* sebagai ekstraksi fiturnya dan algoritma *Random Forest* untuk proses klasifikasinya. Pada penelitian ini digunakan metode *Word2Vec* sebagai ekstraksi fiturnya dan algoritma *Random Forest* untuk proses klasifikasinya. Sebagai perbandingan, dilakukan juga ekstraksi fitur menggunakan model *Bag of Word* (BOW) dan TF-IDF. Hasil uji coba mendapatkan nilai akurasi sebesar 83.49% dengan menggunakan BOW, 84.49% dengan menggunakan TF-IDF, dan akurasi tertinggi sebesar 86.87% dengan menggunakan ekstraksi fitur *Word2Vec*. Namun, variasi algoritma klasifikasi yang digunakan hanya satu jenis, sehingga diperlukan pengembangan dengan menggunakan algoritma klasifikasi yang lainnya. Penelitian inilah yang dijadikan referensi utama oleh penulis dalam penelitian yang akan dilakukan [11].

Berdasarkan permasalahan yang telah dijabarkan sebelumnya, maka akan dilakukan penelitian mengenai analisis sentimen pada data *tweet* berbahasa Indonesia yang mengandung topik penanganan Covid-19 oleh pemerintah Indonesia menggunakan ekstraksi fitur *Word Embedding* (*Word2Vec*) dengan membandingkan dua jenis algoritma klasifikasi yaitu, algoritma *Support Vector Machine* dan *K-Nearest Neighbor*.

2. BAHAN DAN METODE

Pada bagian ini akan dijelaskan beberapa tahapan yang akan dilakukan dalam penelitian. Secara garis besar alur penelitian dapat dijelaskan pada Gambar 1.



Gambar 1. Alur Diagram Penelitian

2.1. Pengumpulan Data

Penelitian ini menggunakan data opini berupa *tweet* berbahasa Indonesia yang bersumber dari Twitter dengan topik penanganan Covid-19 oleh pemerintah di Indonesia. Data yang dikumpulkam bersumber dari Kaggle dan setelah dilakukan seleksi, terkumpul data dengan jumlah sebanyak 801 *tweet*.

2.2. Pelabelan Manual

Pelabelan pada dataset yang telah dikumpulkan dilakukan dilakukan dengan cara manual. Polaritas yang digunakan dalam penelitian ini hanya terdiri atas polaritas positif untuk data yang mengandung sentimen positif dan polaritas negatif untuk data yang mengandung sentimen negatif. Pelabelan manual

dalam penelitian ini dilakukan oleh tiga orang anotator, dua orang sebagai anotator primer dan satu orang sebagai anotator sekunder.

2.3. Preprocessing Data

Data yang telah diberikan label selanjutkan akan diproses pada tahap *preprocessing*. *Preprocessing* merupakan salah satu tahap yang paling penting karena digunakan untuk mengekstraksi pengetahuan yang penting dari data teks yang ada, serta mengubah data yang tidak terstruktur menjadi lebih terstruktur [12]. Adapun tahapan *preprocessing* yang dilakukan dalam penelitian ini sebagai berikut:

2.3.1 Cleaning

Pada tahap ini dilakukan penghapusan tanda baca, URL atau *link, hashtag* (#), dan *username* (@) yang tidak akan digunakan dalam proses analisis sentimen.

2.3.2 Tokenization

Tokenization merupakan tahap yang dilakukan untuk memecah sebuah kalimat menjadi potongan-potongan kata.

2.3.3 Case Folding

Case folding adalah suatu proses yang digunakan untuk mengubah seluruh huruf yang terdapat di dalam dokumen menjadi huruf kecil.

2.3.4 Stopword Removal

Stopword removal adalah proses penghapusan kata-kata yang tidak memiliki makna penting pada tweet, seperti kata ‘yang’, ‘di’, ‘ke’, ‘dengan’, dan lain-lain agar kata-kata yang tersisa dalam kumpulan *tweet* hanyalah kata yang memiliki makna penting saja.

2.3.5 Stemming

Setelah dilakukan penghapusan *stopword*, *stemming* yang merupakan proses mengubah kata-kata dalam dokumen menjadi kata dasar akan dilakukan dalam penelitian ini. Proses *stemming* dalam dokumen bahasa Indonesia cukup kompleks, karena harus dilakukan penghilangan seluruh imbuhan pada kata-kata yang terdapat pada *tweet* [13]. Peneliti menggunakan *library* “*Sastrawi”* untuk melakukan proses *stemming* dalam penelitian ini.

2.4. Feature Extraction

Setelah dataset selesai melalui tahap *preprocessing*, selanjutnya akan dilakukan proses ekstraksi fitur untuk mencari nilai pada tiap fitur yang terkandung di dalam dataset. Proses ini merupakan salah satu langkah paling penting dalam analisis sentimen karena mempengaruhi hasil yang akan didapatkan [14]. Dalam penelitian kali ini metode ekstraksi fitur yang digunakan adalah *Word embedding*. *Word embedding* adalah cara yang digunakan untuk menggambarkan kata-kata ke dalam bentuk vektor. *Word embedding* memiliki beberapa keunggulan jika dibandingkan dengan metode *Bag of Word* (BOW) maupun TF-IDF, diantaranya adalah mampu mengurangi dimensi fitur dan menangkap hubungan semantik dari kata-kata yang terdapat dalam kumpulan dokumen [11]. Algoritma *word embedding* yang akan digunakan dalam penelitian ini adalah *Word2Vec*.

2.4.1 Word2Vec

Algoritma *Word2Vec* merupakan jaringan saraf tiruan yang digunakan untuk memetakan kata ke variabel target yang dapat berupa kata maupun sekumpulan kata. Selain itu, teknik ini memberikan bobot kata yang direpresentasikan dalam bentuk vektor kata [11]. Ada dua jenis model *Word2Vec* yang dapat digunakan, yaitu *Continous Bag-of-Word* (CBOW) dan *Skip Gram*. Model *Word2Vec* yang akan digunakan dalam penelitian adalah model *Skip Gram*. Ukuran vektor pada model *Word2vec* dapat ditentukan jumlah fiturnya [15]. Pada penelitian ini dibuat model *Word2Vec* dengan jumlah fitur sebanyak 200 menggunakan bantuan *library* “*Gensim*” pada *python*.

2.5. Machine Learning Modelling

Setelah dilakukan proses ekstraksi fitur, selanjutnya akan dilakukan proses pemodelan dengan menggunakan salah satu metode *machine learning*, yaitu klasifikasi. Algoritma klasifikasi *Support Vector Machine* (SVM) dan *K-Nearest Neighbor* (KNN) akan digunakan sebagai perbandingan dalam penelitian kali ini..

2.5.1 Support Vector Machine

Support Vector Machine (SVM) merupakan salah satu algoritma klasifikasi yang sering digunakan dalam klasifikasi teks dan telah menunjukkan performa yang sangat baik bahkan mengalahkan performa algoritma pembelajaran mesin lainnya [16].

SVM bekerja dengan cara membuat garis yang disebut sebagai *hyperplane* yang akan memisahkan setiap kelas yang ada pada data [17]. Dalam memprediksi suatu data, SVM akan melabeli data tersebut sesuai dengan daerah kelas dimana data tersebut berada. SVM bekerja atas dasar prinsip *Structural Risk Management* (SRM) sehingga usaha dalam mendapatkan *hyperplane* yang paling baik sebagai pemisah antar kelas adalah inti dari metode SVM. Dalam proses pencarian *hyperplane* yang optimal, SVM sangat bergantung pada parameter yang digunakan, seperti nilai C, nilai *epsilon*, dan fungsi *kernel* yang digunakan. Ada beberapa fungsi kernel yang dikenal dalam SVM, diantaranya *adalah linear, sigmoid, radial,* dan *polynomial* [17][18]. Fungsi *kernel* yang akan digunakan dalam penelitian kali ini adalah *kernel linear*.

2.5.2 K-Nearest Neighbor

K-Nearest Neighbor (KNN) adalah algoritma yang digunakan untuk proses klasifikasi. Klasifikasi data dilakukan dengan berdasarkan mayoritas jumlah tetangga (*neighbor*) terdekat pada data yang akan diprediksi berdasarkan jumlah k yang telah ditentukan [19]. Penentuan jumlah nilai k akan sangat berpengaruh terhadap hasil klasifikasi yang dihasilkan. Dalam menentukan jumlah nilai k yang akan digunakan, sebaiknya gunakan nilai ganjil agar menghindari kemungkinan tidak ditemukannya jawaban. Dalam menghitung jarak antar titik pada kelas k, dapat digunakan metode perhitungan jarak *Euclidean Distance*. Persamaan yang digunakan adalah sebagai berikut [20]:

$$D(x_i, y_i) = \sqrt{\sum_{i=1}^{n}(x_i, y_i)^2} \quad (1)$$

Keterangan:
x_i : data train
y_i : data test
$D(x_i, y_i)$: jarak
i : variable data
n : dimensi data

2.6. Evalusasi

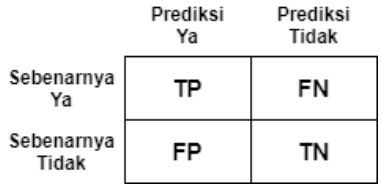
Hasil dari pemodelan yang telah dilakukan kemudian akan diuji untuk mengetahui performa yang dihasilkan. Pada penelitian ini, evaluasi akan dilakukan dengan menggunakan *k-fold cross validation* dengan nilai k = 10 dan confusion *matrix* serta nilai AUC pada kurva ROC.

2.6.1 K-fold Cross Validation

K-fold cross validation merupakan sebuah metode guna mengevaluasi performa suatu model dimana data akan dipecah ke dalam dua bagian yakni data *train* serta data *test*. *K-fold cross validation* dengan nilai k = 10, akan memecah data ke dalam 10 bagian yang terdiri atas 9 bagian sebagai data *train* dan satu bagian sebagai data *test.* Proses akan dilakukan secara bergantian sebanyak 10 kali hingga seluruh bagian data memiliki kesempatan untuk berperan sebagai data *test* menggunakan model yang sudah dibangun.

2.6.2 Confusion Matrix

Confusion matrix adalah sebuah tabel yang menyediakan informasi terkait perbandingan hasil prediksi klasifikasi yang telah dilakukan oleh sistem dengan nilai yang sebenarnya [21]. Contoh *confusion matrix* dapat dilihat pada Gambar 2.



Gambar 2. Contoh *confusion matrix*

Pada *confusion matrix*, terkandung 4 istilah nilai hasil klasifikasi yang kemudian dapat digunakan untuk menghitung nilang akurasi, presisi, dan *recall* dari model yang diuji, yaitu *True Positive* (TP) yang merupakan data yang diklasifikasikan secara benar pada kelas positif, *False Negative* (FN) yang merupakan data yang diprediksi negatif namun sebenarnya positif, *False Positive* (FP) yang merupakan data yang diprediksi positif namun sebenarnya negatif, dan *True Negative* (TN) yang merupakan data yang diklasifikasikan secara benar pada kelas negatif [22].

Berdasarkan nilai yang dihasilkan pada *confusion matrix* tersebut, berikut persamaan yang digunakan untuk menghitung nilai akurasi, presisi dan *recall* dari model yang akan diuji [23]:

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

$$Presisi = \frac{TP}{TP+FP} \quad (3)$$

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

2.6.3 Area Under Curve

Area Under Curve (AUC) merupakan nilai yang dapat digunakan untuk mengukur performa metode yang digunakan berdasar pada kurva ROC yang dihasilkan. Rumus AUC untuk menghitung performa dapat dilihat pada persamaan berikut [24]:

$$AUC = \frac{1}{2}\sum_{i=1}^{n}(X_{i+1}, X_i)(Y_{i+1} - Y_i) \quad (5)$$

Nilai AUC yang mendekati satu maka akan semakin baik. Interpretasi pada nilai AUC dapat dilihat pada Tabel 1 dibawah ini.

Tabel 1. Interpretasi nilai AUC [24]

Nilai AUC	Interpretasi
1.0 (100%)	*Perfect model*
0.9 – 0.99 (90 – 99%)	*Excellent model*
0.8 – 0.89 (80 – 89%)	*Very good model*
0.7 – 0.79 (70 – 79%)	*Fair model*
0.51 – 0.69 (51 – 69%)	*Poor model*
< 0.5 (50%)	*Worthless model*

3. HASIL DAN PEMBAHASAN

Pada bagian ini berisi mengenai hasil dari penelitian yang telah dilakukan. Penelitian dilakukan sesuai dengan tahap yang sudah diuraikan pada bagian bahan dan metode. Pengujian dalam penelitian dilakukan dengan menggunakan bahasa pemrograman *python* dengan bantuan *library* "*scikit learn*" untuk implementasinya. Hasil dari pengujian selanjutnya akan dievaluasi dengan pengukuran nilai akurasi, presisi, *recall,* dan nilai AUC yang dihasilkan.

Dataset yang digunakan dalam penelitian ini adalah data berupa *tweet* yang mengandung topik tentang penanganan Covid-19 oleh pemerintah di Indonesia, data yang dikumpulkan bersumber dari kaggle yang diunduh melalui link

https://www.kaggle.com/dionisiusdh/covid19-indonesian-twitter-sentiment. Setelah dilakukan pembersihan, data yang dikumpulkan berjumlah 801 *record*. Dari proses pelabelan manual yang dilakukan, didapatkan hasil *tweet* yang termasuk ke dalam kelas positif sebanyak 400 data dan *tweet* yang termasuk ke dalam kelas negatif sebanyak 401 data.

Confusion matrix dari pengujian sistem dengan menggunakan Algoritma *Support Vector Machine* (SVM) dan *K-Nearest Neighbor* (KNN) dengan fitur ekstraksi *Word Embedding* (*Word2Vec*) ditunjukkan pada Gambar 3 dan Gambar 4.

	Prediksi Negatif (0)	Prediksi Positif (1)
Sebenarnya Negatif (0)	326	75
Sebenarnya Positif (1)	70	330

Gambar 3. *Confusion matrix* dengan algoritma SVM

	Prediksi Negatif (0)	Prediksi Positif (1)
Sebenarnya Negatif (0)	286	115
Sebenarnya Positif (1)	161	239

Gambar 4. *Confusion matrix* dengan algoritma KNN

Pada klasifikasi menggunakan algoritma SVM, terdapat 396 data yang diprediksi negatif dan 405 data yang diprediksi positif. Namun, dari 396 data yang diprediksi negatif, terdapat kesalahan prediksi sebanyak 70 data yang seharusnya berada pada kelas positif, sedangkan dari 405 data yang diprediksi positif, terdapat kesalahan prediksi sebanyak 75 data yang seharusnya berada pada kelas negatif.

Selanjutnya, pada klasifikasi menggunakan algoritma KNN, terdapat 447 data yang diprediksi negatif, dan 354 data yang diprediksi positif. Namun, dari 447 data yang diprediksi negatif, terdapat kesalahan prediksi sebanyak 161 data yang seharusnya berada pada kelas positif, sedangkan dari 354 data yang diprediksi positif, terdapat kesalahan prediksi sebanyak 115 data yang seharusnya berada pada kelas negatif.

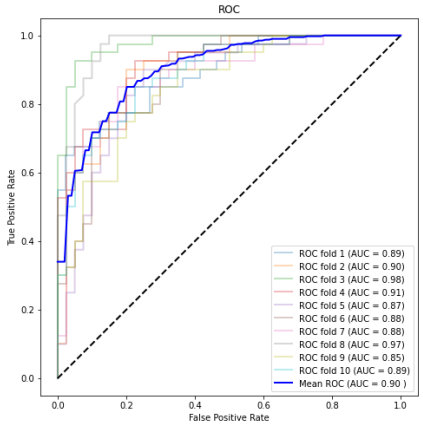
Berdasarkan nilai *confusion matrix* tersebut, diperoleh perbandingan hasil perhitungan akurasi, presisi, dan *recall* sistem yang ditampilkan pada Tabel 2.

Tabel 2. Perbandingan nilai akurasi, presisi dan recall

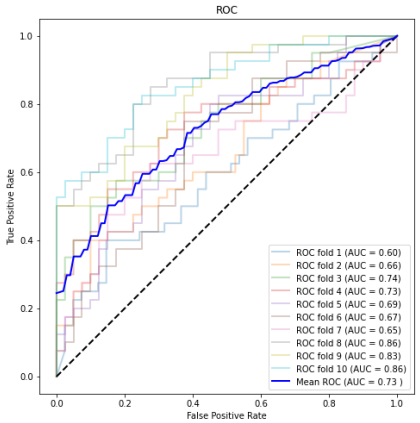
Algoritma	Akurasi	Presisi	Recall
SVM	81,90%	81,91%	81,90%
KNN	65,60%	65,75%	65,54%

Hasil pengujian tersebut menunjukkan bahwa algoritma *Support Vector Machine* (SVM) memiliki nilai akurasi, presisi, dan *recall* yang berturut-turut lebih unggul sebesar 16,3%, 16,16% dan 16,36% dibandingkan dengan algoritma *K-Nearest Neighbor* (KNN).

Kemudian hasil evaluasi berupa nilai AUC yang dihitung berdasarkan kurva ROC pada Gambar 5 dan Gambar 6 akan ditampilkan pada Tabel 3.



Gambar 5. Kurva ROC menggunakan algoritma SVM



Gambar 6. Kurva ROC menggunakan algoritma KNN

Tabel 3. Perbandingan Nilai AUC

Algoritma	Nilai AUC
SVM	0,90
KNN	0,73

Berdasarkan nilai pada Tabel 3, nilai AUC pada algoritma *Support Vector Machine* (SVM) lebih unggul dibandingkan dengan algoritma *K-Nearest Neighbor* (KNN) dan termasuk ke dalam kategori *excellent model*. Sedangkan nilai AUC pada algoritma *K-Nearest Neighbor* (KNN) termasuk ke dalam kategori *fair model.*

4. KESIMPULAN

Berdasarkan hasil klasifikasi analisis sentimen pada *tweet* tentang penanganan Covid-19 oleh pemerintah di Indonesia, dapat disimpulkan bahwa

kedua metode dapat menghasilkan performa yang baik dan algoritma *Support Vector Machine* (SVM) mendapatkan hasil akurasi, presisi, *recall,* dan nilai AUC yang lebih unggul dibandingkan dengan algoritma *K-Nearest Neighbor* (KNN).

Dalam pengembangan kedepannya, dapat dilakukan penambahan terhadap jumlah data yang digunakan serta percobaan terhadap algoritma pembelajaran mesin lainnya seperti *Naïve Bayes*, *Decision Tree*. dan lain-lain.

5. DAFTAR PUSTAKA

[1] M. Syarifuddin, "Analisis Sentimen Opini Publik Mengenai Covid-19 Pada Twitter Menggunakan Metode Naïve Bayes Dan Knn," *Inti Nusa Mandiri*, vol. 15, no. 1, pp. 23–28, 2020.

[2] S. T. P. Covid-19, "Peta Sebaran Covid-19," 2020. [Online]. Available: https://covid19.go.id/peta-sebaran.

[3] A. K. Fauziyyah, "Analisis Sentimen Pandemi Covid19 Pada Streaming Twitter Dengan Text Mining Python," *J. Ilm. SINUS*, vol. 18, no. 2, p. 31, 2020, doi: 10.30646/sinus.v18i2.491.

[4] S. Fanissa, M. A. Fauzi, and S. Adinugroho, "Analisis Sentimen Pariwisata di Kota Malang Menggunakan Metode Naive Bayes dan Seleksi Fitur Query Expansion Ranking | Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 8, pp. 2766–2770, 2018.

[5] F. Iqbal *et al.*, "A Hybrid Framework for Sentiment Analysis Using Genetic Algorithm Based Feature Reduction," *IEEE Access*, vol. 7, no. c, pp. 14637–14652, 2019, doi: 10.1109/ACCESS.2019.2892852.

[6] N. Hayatin, G. I. Marthasari, and L. Nuarini, "Optimization of Sentiment Analysis for Indonesian Presidential Election using Naive Bayes and Particle Swarm Optimization," *J. Online Inform.*, vol. 5, no. 1, pp. 81–88, 2020, doi: 10.15575/join.v5i1.558.

[7] Q. Xu, V. Chang, and C. H. Hsu, "Event Study and Principal Component Analysis Based on Sentiment Analysis – A Combined Methodology to Study the Stock Market with an Empirical Study," *Inf. Syst. Front.*, vol. 22, no. 5, pp. 1021–1037, 2020, doi: 10.1007/s10796-020-10024-5.

[8] H. Tuhuteru, "Analisis Sentimen Masyarakat Terhadap Pembatasan Sosial Berksala Besar Menggunakan Algoritma Support Vector Machine," *J. Inf. Syst. Dev.*, vol. 4, no. 1, 2020.

[9] R. Risnantoyo, A. Nugroho, and K. Mandara, "Sentiment Analysis on Corona Virus Pandemic Using Machine Learning Algorithm," *J. Informatics Telecommun. Eng.*, vol. 4, no. 1, pp. 86–96, 2020, doi: 10.31289/jite.v4i1.3798.

[10] F. W. Kurniawan and W. Maharani, "Indonesian Twitter Sentiment Analysis Using Word2Vec," *2020 Int. Conf. Data Sci. Its Appl. ICoDSA 2020*, pp. 31–36, 2020, doi: 10.1109/ICoDSA50139.2020.9212906.

[11] A. Sharma and A. Daniels, "Tweets Sentiment Analysis via Word Embeddings and Machine Learning Techniques."

[12] L. Hermawan and M. Bellaniar Ismiati, "Pembelajaran Text Preprocessing berbasis Simulator Untuk Mata Kuliah Information Retrieval," *J. Transform.*, vol. 17, no. 2, p. 188, 2020, doi: 10.26623/transformatika.v17i2.1705.

[13] R. Rismanto, "Rekomendasi Artikel Terkait Pada Berita Online Menggunakan Teknik Text Mining," pp. 268–271.

[14] X. Chen, Y. Xue, H. Zhao, X. Lu, X. Hu, and Z. Ma, "A novel feature extraction methodology for sentiment analysis of product reviews," *Neural Comput. Appl.*, vol. 31, no. 10, pp. 6625–6642, 2019, doi: 10.1007/s00521-018-3477-2.

[15] M. Rusli, "Ekstraksi Fitur Menggunakan Model Word2Vec Pada Sentiment Analysis Kolom Komentar Kuisioner Evaluasi Dosen Oleh Mahasiswa," *Klik - Kumpul. J. Ilmu Komput.*, vol. 7, no. 1, p. 35, 2020, doi: 10.20527/klik.v7i1.296.

[16] A. S. H. Basari, B. Hussin, I. G. P. Ananta, and J. Zeniarja, "Opinion mining of movie review using hybrid method of support vector machine and particle swarm optimization," *Procedia Eng.*, vol. 53, pp. 453–462, 2013, doi: 10.1016/j.proeng.2013.02.059.

[17] D. Maulina and R. Sagara, "Klasifikasi Artikel Hoax Menggunakan Support Vector Machine Linear Dengan Pembobotan Term Frequency-Inverse Document Frequency," *J. Mantik Penusa*, vol. 2, no. 1, pp. 35–40, 2018.

[18] W. R. U. Fadilah, D. Agfiannisa, and ..., "Analisis Prediksi Harga Saham PT. Telekomunikasi Indonesia Menggunakan Metode Support Vector Machine," ... *Informatics J.*, vol. 5, no. 2, 2020.

[19] N. Godara and S. Kumar, "Opinion Mining using Machine Learning Techniques," *Int. J. Eng. Adv. Technol.*, vol. 9, no. 2, pp. 4287–4292, 2019, doi: 10.35940/ijeat.b4108.129219.

[20] F. Tempola, M. Muhammad, and A. Khairan, "Perbandingan Klasifikasi Antara KNN dan Naive Bayes pada Penentuan Status Gunung Berapi dengan K-Fold Cross Validation," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 5, p. 577, 2018, doi: 10.25126/jtiik.201855983.

[21] M. I. Fikri, T. S. Sabrila, and Y. Azhar, "Perbandingan Metode Naïve Bayes dan Support Vector Machine pada Analisis

Sentimen Twitter," vol. 10, pp. 71–76, 2020.

[22] X. Deng, Q. Liu, Y. Deng, and S. Mahadevan, "An improved method to construct basic probability assignment based on the confusion matrix for classification problem," *Inf. Sci. (Ny).*, vol. 340–341, pp. 250–261, 2016, doi: 10.1016/j.ins.2016.01.033.

[23] M. E. Al Rivan, N. Rachmat, and M. R. Ayustin, "Klasifikasi Jenis Kacang-Kacangan Berdasarkan Tekstur Menggunakan Jaringan Syaraf Tiruan," vol. Vol 6 No 1, no. 1, pp. 89–98, 2020, doi: 10.35143/jkt.v6i1.3546.

[24] A. Yani, "Analisa Kelayakan Kredit Menggunakan Artifcial Neural Network dan Backpropogation (Studi Kasus German Credit Data)," *J. Ilm. Komputasi*, vol. 18, no. 4, pp. 385–390, 2019, doi: 10.32409/jikstik.18.4.2672.