

Klasifikasi Respon Masyarakat Indonesia Terhadap COVID-19 di Twitter

The Classification of Indonesian Perspective to Word COVID-19 on Twitter

Panji Bintoro¹, Ratnasari²

¹Universitas Aisyah Pringsewu, Indonesia

²Universitas Aisyah Pringsewu, Indonesia

Article Info	ABSTRAK
<p>Kata Kunci:</p> <p>COVID-19, Analisis Sentimen, Support Vector Machine, Multinomial Naïve Bayes, Stochastic Gradient Descent, Random Forest</p>	<p>Penanganan COVID-19 (<i>Coronavirus Disease-2019</i>) di Indonesia pernah menjadi topik hangat di media sosial yaitu Twitter. Penanganan pemerintah Indonesia tersebut menimbulkan pro dan kontra di masyarakat. Opini publik di Twitter dapat digunakan sebagai sistem pendukung keputusan dalam pengambilan kebijakan yang tepat untuk mengevaluasi kinerja pemerintah. Metode analisis sentimen dapat digunakan untuk menganalisis opini publik di Twitter. Penelitian ini bertujuan untuk memahami fenomena opini publik tentang COVID-19 di Indonesia. Kami menggunakan dataset tweet mengenai COVID-19 di Indonesia untuk diklasifikasikan menjadi sentimen positif, dan negatif. Kami melakukan <i>preprocessing</i> untuk data duplikat dan tidak relevan dan kami juga membandingkan <i>text feature extraction</i> untuk analisis sentimen pada data baru. Algoritma pembelajaran mesin divalidasi menggunakan beberapa metode serta menggunakan 10-fold validasi. Performa terbaik terjadi pada skenario kedua dengan <i>classifier</i> terbaik adalah <i>Support Vector Machine</i> dengan akurasi yang didapatkan sebesar 84.03%.</p>
<p>Keywords:</p> <p>COVID-19 Sentiment Analysis Support Vector Machine Multinomial Naïve Bayes Stochastic Gradient Descent Random Forest</p>	<p>ABSTRACT</p> <p><i>Handling COVID-19 (Coronavirus Disease-2019) in Indonesia has been a hot topic on social media, is Twitter. The handling of the Indonesian government raises pros and cons in the community. Public opinion on Twitter can be used as a decision support system to develop appropriate guidelines for evaluating government performance. On Twitter, the sentiment analysis method can be used to analyze public opinion. This research aims to understand the phenomenon of public opinion about COVID-19 in Indonesia. We use a dataset of tweets about COVID-19 in Indonesia to classify them into positive and negative sentiments. We perform duplicate and irrelevant data preprocessing and compare text feature extraction for sentiment analysis on new data. Machine learning algorithms were evaluated using several methods and using 10-fold validation. The best performance occurs in the second scenario with the best classifier being the Support Vector Machine with an accuracy of 84.03%.</i></p>
	<p style="text-align: right;">This is an open access article under the CC BY-SA license.</p> 

Penulis Korespondensi:

Panji Bintoro,
Program Studi Rekayasa Perangkat Lunak,
Universitas Aisyah, Indonesia
Email: panjibintoro09@aisyahuniversity.ac.id

1 PENDAHULUAN

Penggunaan internet khususnya media sosial cukup tinggi di Indonesia. Berdasarkan hasil riset Hootsuite Social Wear yang dirilis pada Januari 2019, pengguna media sosial di Indonesia mencapai 150 juta atau 56% dari total populasi, meningkat 20% dari survei sebelumnya [1]. Salah satu media sosial yang banyak digunakan dan berkembang adalah Twitter [2]. Konten yang dibagikan oleh pengguna Twitter sebagian besar adalah opini tentang masalah saat kebijakan pemerintah dan masalah sosial – politik, serta ulasan produk baru (yang dapat digunakan untuk business intelligence). Bagaimanapun, konten dapat digunakan sebagai sumber informasi yang efektif untuk membantu proses pengambilan keputusan.

Pandemi COVID-19 telah berdampak pada semua sektor dan masyarakat Indonesia. Berdasarkan data dari covid.go.id pada 14 Mei 2020 terdapat 16.006 kasus yang dikonfirmasi, 3.518 pasien sembuh, dan 1.043 pasien meninggal [3]. Pemerintah Indonesia dinilai lamban dalam menangani COVID-19 [4][5] sehingga kebijakan pemerintah Indonesia dalam menangani COVID-19 menjadi trending topik di Twitter, menimbulkan pro dan kontra, terlebih karena COVID-19 telah berdampak di banyak sektor yang ada.

Kebijakan pemerintah harus diinformasikan dengan baik agar tepat sasaran. Opini publik seperti yang dipublikasikan di Twitter diperlukan untuk menginformasikan pembuatan kebijakan. Mengekstrak opini publik di Twitter tidaklah mudah, karena berantakan dan sering sekali ditulis dalam bahasa yang tidak standar atau tidak baku. Jika proses ini memadai, pengetahuan dapat diperoleh untuk membantu dalam pembuatan kebijakan. Oleh karena itu, diperlukan metode atau teknik tertentu yang dapat mengklasifikasikan pendapat dengan cepat.

Analisis sentimen dalam text mining mempelajari sentimen, emosi, dan sikap dalam teks opini. Prinsip dasar dari analisis sentimen adalah untuk mengklasifikasikan polaritas teks tertentu dalam menentukan apakah isinya lebih positif, negatif, atau netral [6]. Oleh karena itu, penelitian ini menggunakan analisis sentimen untuk mengetahui tanggapan masyarakat tentang kinerja pemerintah terhadap penanganan COVID-19 [7].

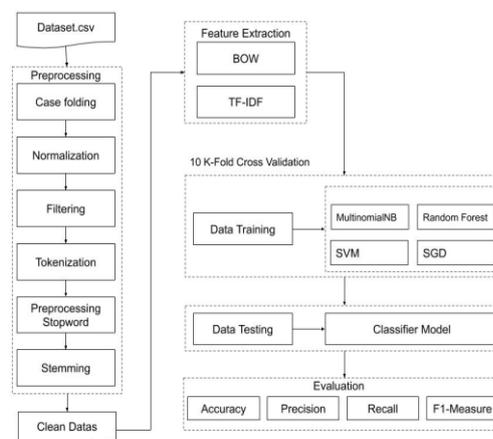
Penelitian tentang analisis sentimen berkembang pesat dan ada pendekatan baru yang diluncurkan. Analisis sentimen berbasis machine learning telah menjadi pendekatan populer yang dianggap efektif dalam menganalisis data berbasis teks [8]. Dalam penelitian sebelumnya Prastyo et.al. [9] melakukan analisis sentimen terhadap opini masyarakat terhadap kinerja pemerintah dalam menangani kasus COVID-19 ini. Dalam penelitiannya membandingkan dua algoritma yaitu Support Vector Machine dan Multinomial Naive Bayes. Hasil menunjukkan bahwa Support Vector Machine (SVM) memiliki akurasi lebih tinggi dari Multinomial Naive Bayes (MNB).

G.Singh et. al. [10] membandingkan Multinomial Naive Bayes (MNB) dan Bernoulli Naive Bayes (BNB) untuk memprediksi apakah sentimen artikel berita tertentu positif atau negatif. Hasil percobaan menunjukkan bahwa kerja MNB lebih baik dari BNB. A. Lutfi et. al. [11] melakukan analisis sentimen dalam tinjauan penjualan dari tempat pasar Indonesia dengan membandingkan Support Vector Machine (SVM) dan Naive Bayes Classifier (NBC). Hasil menunjukkan SVM mengungguli NBC karena memperoleh akurasi 93,65%.

Terakhir, dalam penelitian ini, kami membandingkan algoritma machine learning dan membandingkan beberapa algoritma diantaranya SVM, SGD, MNB, dan Random Forest untuk menentukan algoritma mana yang bekerja paling baik dalam melakukan analisis sentimen Bahasa Indonesia tentang pemerintah dalam penanganan COVID-19.

2 METODE PENELITIAN

Dalam penelitian ini tweet berbahasa Indonesia dikelompokkan menjadi dua kelas (positif dan negatif). Beberapa tahapan yang diperlukan untuk mendapatkan hasil klasifikasi yang optimal yaitu *preprocessing*, *feature extraction*, pengembangan model klasifikasi dan evaluasi. Kerangka kerja ditunjukkan pada Gambar 1.



Gambar 1. Kerangka kerja.

2.1 Data Collection

Dataset yang digunakan adalah *dataset* dari penelitian Prastyo et.al. [9], sebanyak 2269 data berlabel yang terdiri dari 874 positif, 1044 negatif, dan 351 netral. Namun untuk penelitian ini hanya menggunakan dua kelas yaitu positif dan negatif.

2.2 Preprocessing

Data teks dari Twitter pertama kali masih berantakan, jadi pada *final project* ini ada tahapan *preprocessing* untuk menghasilkan tweet dengan data yang relevan atau rapih agar memudahkan kinerja algoritma nantinya.

a. Normalization

Normalization yang dimaksud disini yaitu mengganti kata - kata yang tidak baku menjadi baku. Pada proses *normalization* ini menggunakan dataset kamus-alay [14].

b. Filtering

Filtering merupakan tahap untuk menghapus kata - kata yang mengandung seperti *username*, *hashtag*, *urls*, dan *special characters*.

c. Tokenization

Tokenization merupakan proses membagi tweet menjadi kata - kata, sebagai contoh: “Virus corona di Indonesia terus meningkat, pemerintah Indonesia tarik rem darurat” menjadi “[virus] [corona] [di] [indonesia] [terus] [meningkat] [pemerintah] [indonesia] [tarik] [rem] [darurat].

d. Stopword

Stopword merupakan pengambilan kata-kata penting dari hasil proses *tokenization* dengan *library* Sastrawi dan menggunakan *module* StopWordRemoverFactory.

e. Stemming

Stemming merupakan proses perubahan bentuk kata menjadi kata dasar, dengan dilakukannya proses ini setiap kata berimbuhan akan berubah menjadi kata dasar sehingga dapat lebih mengoptimalkan proses teks *mining*.

f. Clean Data

Clean data yang dimaksud yaitu data yang sudah melewati tahap *preprocessing* dan data tersebut sudah relevan atau bersih.

2.3 Feature Extraction

Pada penelitian ini menggunakan dua skenario yang dimana dua skenario tersebut berbeda prosesnya pada tahap *feature extraction*. Pada skenario pertama hanya menggunakan BOW (*Bag of Word*) sedangkan pada skenario kedua menggunakan TF-IDF (*Term Frequency Inverse Document Frequency*). BOW merupakan membagi setiap kata dalam dokumen sebagai kata kunci yang berpotensi penting dari dokumen tersebut. Pendekatannya adalah memperlakukan setiap dokumen sebagai kumpulan kata - kata individual. TF merupakan frekuensi dari setiap kata - kata untuk mewakili relevansi kata ke dokumen. Sebuah dokumen yang panjang biasanya memiliki lebih banyak kata, dengan demikian lebih banyak kemunculan kata. IDF bertujuan mengukur seberapa lazim suatu frekuensi dalam suatu dokumen. Beberapa metode *feature extraction* diatas dilakukan dengan tujuan meningkatkan kinerja khususnya pada nilai *recall* dan *precision*. Persamaan TF dapat dilihat pada persamaan 1.

$$TF_{(d,t)} = \frac{f_{t,d}}{\sum_{t,d}} \quad (1)$$

$f_{t,d}$ merupakan munculnya frekuensi setiap kata (t) dalam dokumen (d) dan $\sum_{t,d}$ merupakan jumlah semua akta yang terkandung di dalam dokumen (d). Persamaan IDF dapat dilihat pada persamaan 2.

$$IDF_{(t)} = \log \frac{|D|}{f_{t,d}} \quad (2)$$

$|D|$ merupakan jumlah dokumen dalam koleksi dan $f_{t,d}$ merupakan munculnya frekuensi dari setiap kata (t) dalam dokumen (d). Kemudian nilai TF-IDF dapat dihitung menggunakan persamaan 3.

$$W_{(d,t)} = TF_{(d,t)} \cdot IDF_{(t)} \quad (3)$$

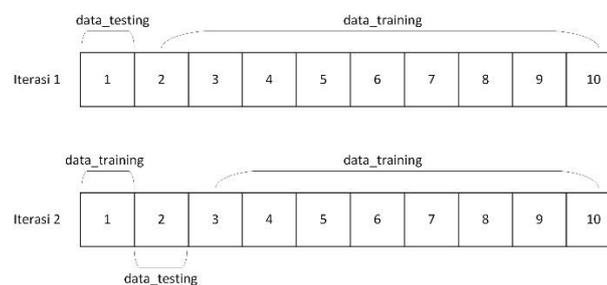
Pada persamaan TF-IDF ini hasil dari TF dikalikan dengan IDF. Hasil akhir perhitungan tersebut disebut *feature extraction*.

2.4 Development Classification Model

Setelah *preprocessing* dan ekstraksi ciri, proses selanjutnya adalah membuat model untuk klasifikasi. Pada proses ini data dibagi menjadi dua jenis data yaitu data *training* dan data *testing* dengan perbandingan 70% untuk data *training* dan 30% data *testing*.

a. Data Training

Data *training* yang telah disiapkan digunakan untuk melatih model menggunakan algoritma pembelajaran mesin antara lain Multinomial Naïve Bayes (MNB), Support Vector Machine (SVM), Stochastic Gradient Descent (SGD), dan Random Forest. Pada percobaan ini kami menggunakan dua kelas data yaitu positif dan negatif dengan menggunakan empat algoritma klasifikasi. Kami juga membandingkan hasil kinerja dari algoritma tersebut untuk mendapatkan nilai pengujian terbaik. Pada proses pengklasifikasian menggunakan data *training* yang diuji menggunakan 10 *k-fold cross validation* untuk mengevaluasi kinerja dari model. Hal ini diperlukan karena dapat mengurangi waktu komputasi dengan tetap menjaga keakuratan estimasi dan menghindari *overfitting*. Seluruh jumlah data *training* dibagi menjadi dua yaitu data *training* dan *testing*, kemudian diambil nilai $k=10$ dimana data dibuat menjadi 10 lipatan/*fold* dengan ukuran yang sama atau mendekati sama, sehingga memiliki 10 subset data untuk mengevaluasi kinerja model atau algoritma. Selanjutnya untuk masing-masing dari 10 *subset* data tersebut *cross-validation* akan menggunakan 9-*fold* sebagai pelatihan dan 1-*fold* sebagai pengujian. Ilustrasi *k-fold cross validation* ditunjukkan pada Gambar 2.



Gambar 2. Ilustrasi *k-fold cross validation*.

b. Data Testing

Proses kedua adalah pengujian dengan data *testing* dengan perbandingan 30% dari total semua dataset. Proses ini dilakukan untuk menguji model yang sudah dibangun apakah dapat bekerja secara optimal terhadap data baru. Jumlah data *testing* sebagai data pengujian berjumlah lebih sedikit dibandingkan data latih, hal ini disebabkan karena jika suatu model dengan kasus yang banyak dan bervariasi maka model akan menjadi lebih cerdas sehingga prediksi kesalahan pada pengujian akan lebih sedikit.

2.5 Evaluation dan Validation

Proses evaluasi menggunakan *confusion matrix*, untuk mengevaluasi performa algoritma pembelajaran mesin yang berisi informasi mengenai prediksi dan klasifikasi dari model. Ada empat indikator yang digunakan dalam proses evaluasi dalam *confusion matrix* yaitu *accuracy*, *precision*, *recall*, dan *F1-measure*.

	1 (Positive)	0 (Negative)
1 (Positive)	True Positive	False Positive
0 (Negative)	False Negative	True Negative

Gambar 3. *Confusion matrix*.

Berdasarkan gambar di atas terdapat empat istilah sebagai representasi hasil proses klasifikasi pada *confusion matrix*. Istilah tersebut yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) yang dapat diperoleh nilai *accuracy*, *precision*, *recall*, dan *f1-score*. Nilai *accuracy* menggambarkan seberapa akurat sistem dapat mengklasifikasikan data

secara benar. Dengan kata lain, nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data. Nilai *precision* menggambarkan jumlah data kategori positif yang diklasifikasikan secara benar dibagi dengan total data yang diklasifikasi positif. Nilai *recall* menunjukkan berapa persen data kategori positif yang terklasifikasikan dengan benar oleh sistem. Nilai *f1-score* menggambarkan perbandingan rata-rata *precision* dan *recall* yang dibobotkan.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100\% \quad (4)$$

$$Precision = \frac{TP}{FP + TP} * 100\% \quad (5)$$

$$Recall = \frac{TP}{FN + TP} * 100\% \quad (6)$$

$$F1 - score = 2 * \frac{Recall * Precision}{Recall + Precision} \quad (7)$$

dimana:

- a. TP merupakan *True Positive*, yaitu jumlah data *positive* yang terklasifikasi dengan benar oleh sistem.
- b. TN merupakan *True Negative*, yaitu jumlah data *negative* yang terklasifikasi dengan benar oleh sistem.
- c. FP merupakan *False Positive*, yaitu jumlah data *positive* namun terklasifikasi salah oleh sistem.
- d. FN merupakan *False Negative*, yaitu jumlah data *negative* namun terklasifikasi salah oleh sistem.

3 HASIL DAN ANALISIS

Pada penelitian ini lebih berfokus pada tugas atau *task Binary Classification* dimana kelas data dibagi menjadi dua yaitu positif dan negatif. Beberapa *classifier* digunakan untuk melakukan klasifikasi sebuah *tweet* dan memprediksi data baru yang belum terlihat sebelumnya. Adapun *classifier* yang digunakan dalam penelitian ini adalah MultinomialNB, Support Vector Machine, Stochastic Gradient Descent, dan Random Forest. Percobaan yang dilakukan terhadap empat *classifier* tersebut bertujuan untuk menentukan *classifier* mana yang memiliki performa yang terbaik.

Dataset yang diperoleh dari penelitian Prastyo et.al [9] terdiri dari 3 kelas yaitu positif, negatif, dan netral. Pada penelitian ini berfokus pada *binary classification task* maka hanya digunakan 2 kelas saja yaitu kelas positif dan negatif dengan membuang kelas netral maka detail dataset dapat ditunjukkan pada Tabel 1.

Tabel 1. Jumlah tiap kelas pada dataset.

No	Kelas	Jumlah
1	Positif	874
2	Negatif	1044

Data tweet tersebut dilakukan *preprocessing* terlebih dahulu dengan tahap *case folding*, *filtering*, *normalized alay world*, *tokenization*, *stopword removal* dan *stemming* sebelum dilakukan *training* pada setiap algoritma *classifier*. Pada penelitian ini digunakan dua eksperimen yaitu:

- a. Skenario 1

Pada skenario 1 dilakukan *preprocessing* dengan ekstraksi fitur berupa *Bag of Word* dengan memperhatikan kemunculan kata atau *term frequency* saja.

- b. Skenario 2

Pada skenario 2 dilakukan *preprocessing* dengan ekstraksi fitur berupa *Term Frequency Inverse Document Frequency (TF-IDF)*.

Pada setiap skenario dataset dibagi menjadi dua bagian yaitu data *training* dan data *testing* dengan persentase 70%:30%. Data *training* adalah data yang digunakan sebagai data latih untuk setiap *classifier* sedangkan data *testing* adalah data yang belum pernah terlihat sebelumnya oleh *classifier*, dimana nantinya data *testing* ini diuji performa setiap *classifier* dalam melakukan tugas klasifikasi.

3.1 Skenario 1

Pada skenario 1, dataset *training* yang telah diproses dan diambil ekstraksi fitur dengan menggunakan fungsi *CountVectorizer* dari *library Sklearn* dilakukan *training* pada setiap algoritma *classifier* dengan mempertimbangkan skema *k-fold cross validation* dengan nilai *k* sebesar 10 dan dihitung nilai akurasi. Tabel 2 menunjukkan detail dari proses *training* pada skenario 1 untuk setiap *classifier*.

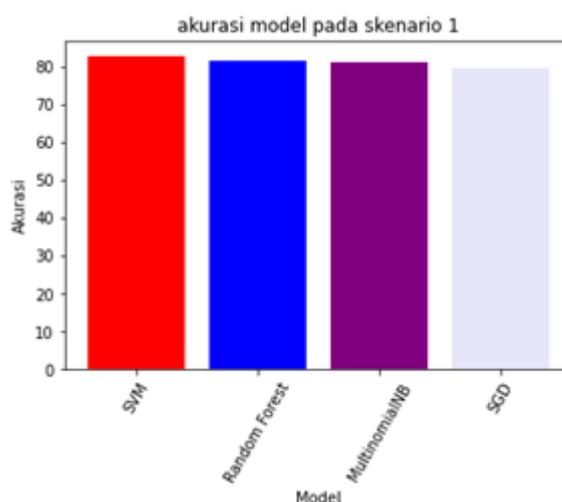
Tabel 2. Skenario 1.

10-Fold	Akurasi			
	MNB	SVM	SGD	RF
1	0.77037	0.822222	0.755556	0.82963
2	0.807407	0.777778	0.755556	0.77037
3	0.843284	0.835821	0.80597	0.828358
4	0.80597	0.820896	0.768657	0.80597
5	0.843284	0.850746	0.813433	0.850746
6	0.798507	0.791045	0.761194	0.791045
7	0.858209	0.820896	0.828358	0.798507
8	0.843284	0.791045	0.783582	0.80597
9	0.828358	0.783582	0.820896	0.783582
10	0.880597	0.835821	0.843284	0.813433
Avg	0.827927	0.812985	0.793649	0.807761
Min	0.77037	0.777778	0.755556	0.77037
Max	0.880597	0.850746	0.843284	0.850746

Dapat dilihat bahwa MultinomialNB mempunyai nilai rata-rata akurasi terbaik untuk 10-fold validation dari pada classifier yang lain, yaitu sebesar 82.79%. Setelah dilakukan training, maka akan dilakukan hyper parameter tuning dengan bantuan Grid Search CV dari Sklearn untuk mendapatkan parameter terbaik dari setiap parameter. Setelah didapatkan parameter terbaik maka dilakukan prediksi terhadap data testing. Tabel 3 dan Gambar 4 menunjukkan hasil eksperimen dengan menggunakan data testing.

Tabel 3. Hasil skenario 1.

No	Classifier	Akurasi
1	SVM	82.64%
2	Random Forest	81.25%
3	MultinomialNB	80.90%
4	SGD	79.34%



Gambar 4. Akurasi model skenario 1.

Hasil dari skenario 1 menunjukkan bahwa algoritma classifier SVM menunjukkan performa terbaik dalam melakukan prediksi pada data testing yakni sebesar 82.64%.

3.2 Skenario 2

Pada skenario 2, dataset training yang telah dipreprocessing dan diambil ekstraksi fitur dengan menggunakan fungsi TF-IDF Vectorizer dari library Sklearn dilakukan training pada setiap algoritma classifier dengan mempertimbangkan skema k-fold cross validation dengan nilai k sebesar 10 dan dihitung nilai akurasinya. Tabel 4 menunjukkan detail dari proses training pada skenario 2 untuk setiap classifier.

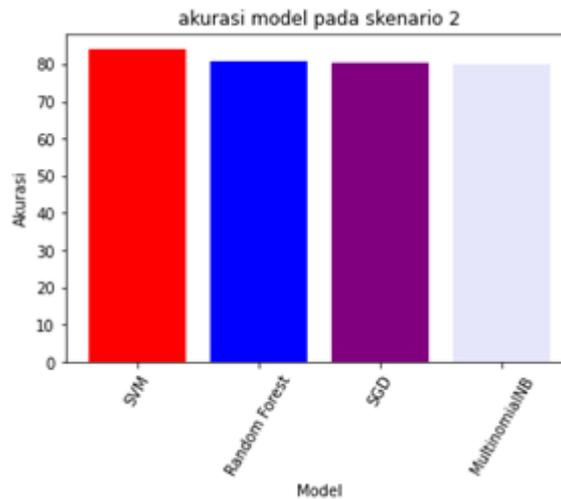
Tabel 4. Skenario 2.

10-Fold	Akurasi			
	MNB	SVM	SGD	RF
1	0.770337	0.822222	0.8	0.807407
2	0.777778	0.785185	0.77037	0.762963
3	0.798507	0.850746	0.798507	0.798507
4	0.843284	0.828358	0.820896	0.828358
5	0.828358	0.858209	0.835821	0.865672
6	0.783582	0.80597	0.761194	0.776119
7	0.813433	0.820896	0.843284	0.798507
8	0.828358	0.813433	0.798507	0.791045
9	0.850746	0.835821	0.80597	0.80597
10	0.798507	0.843284	0.88806	0.813433
Avg	0.809292	0.826412	0.812261	0.804798
Min	0.77037	0.785185	0.761194	0.762963
Max	0.850746	0.858209	0.88806	0.865672

Berdasarkan data hasil *experiment* skenario 2 menunjukkan bahwa SVM *classifier* menunjukkan rata-rata akurasi terbaik untuk 10-fold *validation* dari pada *classifier* yang lain, yaitu dengan nilai 82.64%. Setelah dilakukan *training*, maka dilakukan *hyper parameter tuning* dengan bantuan fungsi *Grid Search CV* dari *Sklearn* untuk mendapatkan parameter terbaik dari setiap *classifier*. Setelah didapatkan parameter terbaik maka dilakukan prediksi terhadap data *testing*. Tabel 5 dan Gambar 5 menunjukkan hasil eksperimen dengan menggunakan data *testing*.

Tabel 5. Hasil skenario 2.

No	Classifier	Akurasi
1	SVM	84.02%
2	Random Forest	80.9%
3	MultinomialNB	80.55%
4	SGD	80.03%



Gambar 5. Akurasi model skenario 2

Hasil *experiment* pada skenario 2 menunjukkan SVM *classifier* mempunyai akurasi nilai tinggi yaitu sebesar 84.02% jika dibandingkan dengan *classifier* lain pada skenario 2. Hal ini menunjukkan bahwa terjadi peningkatan performa jika dibandingkan dengan skenario 1 dimana SVM *classifier* pada skenario 1 hanya sebesar 82.64%. Tabel 6 menunjukkan hasil *metric* evaluasi pada performa SVM *classifier* dengan ekstraksi fitur BOW TF-IDF.

Tabel 6. Hasil metric evaluasi SVM.

No	Classifier	Akurasi
1	Accuracy	84.02%
2	Precision	79.15%
3	Recall	84.36%
4	F1-measure	81.67%

4 KESIMPULAN

Dalam penelitian ini kami telah membangun sebuah model klasifikasi yang dapat menganalisa *sentiment* dari sebuah data tweet mengenai COVID-19 di Indonesia ke dalam kelas positif maupun kelas negatif. Kami juga telah membandingkan ekstraksi fitur dengan menggunakan TF-IDF dan tanpa TF-IDF ke dalam BOW *representation*, serta melakukan *hyper parameter tuning* untuk setiap *classifier*. Hasil menunjukkan bahwa SVM dengan RBF dan nilai C sebesar 10 serta *gamma* 1 menunjukkan performa terbaik yaitu *accuracy* sebesar 84.02%, *precision* sebesar 79.15%, *recall* sebesar 84.36% dan *F1-measure* sebesar 81.67%. Hal ini menunjukkan peningkatan sebesar 2% akurasi dari model yang dibangun oleh Prastyo[9] dimana dalam penelitian ini menggunakan normalisasi kamus alay dan *stemming* yang tidak dilakukan pada model Prastyo. Untuk pengembangan kedepannya diharapkan dapat menggunakan data yang lebih banyak, serta menggunakan fitur lain sebagai perbandingan untuk dapat menemukan model yang lebih baik lagi.

REFERENSI (10 PT)

- [1] Katadata, “Berapa pengguna media sosial Indonesia?” katadata.id, 2019 [Online]. Tersedia: <https://databoks.katadata.co.id/datapublish/2019/02/08/berapa-pengguna-media-sosial-indonesia> (accessed May 05, 2020).
- [2] AntaraNews, “Pengguna Twitter indonesia tumbuh pesat di 2018,” AntaraNews.com, 2019 [Online]. Tersedia: <https://www.antaranews.com/berita/839825/pengguna-twitter-indonesia-tumbuh-pesat-pada-2018> (accessed May 05, 2020).
- [3] Gugus Percepatan Penanganan COVID-19, “Data Covid-19 di Indonesia,” covid19.go.id, 2020 [Online]. Tersedia: <https://covid19.go.id/> (accessed May 14, 2020).
- [4] Suara, “Pemerintah Indonesia dinilai lambat mengantisipasi covid-19 sejak dini,” suara.com, 2020 [Online]. Tersedia: <https://www.suara.com/news/2020/04/10/025500/pemerintah-indonesia-dinilai-lambat-mengantisipasi-covid-19-sejak-dini?page=all> (accessed May 14, 2020).
- [5] DetikNews, “Pemerintah dinilai lambat tangani corona, Jokowi: Kita tak ingin grusa-grusu,” news.detik.com, 2020 [Online]. Tersedia: <https://news.detik.com/berita/d-4987368/pemerintah-dinilai-lambat-tangani-corona-jokowi-kita-tak-ingin-grasa-grusu> (accessed May 14, 2020).
- [6] I. Zulfa and E. Winarko, “Sentimen analisis tweet berbahasa Indonesia dengan deep belief network,” IJCCS (Indonesian J. Comput. Cybern. Syst., vol. 11, no. 2, pp. 187, 2017, doi: 10.22146/ijccs.24716.
- [7] J. Michie, “The covid-19 crisis – and the future of the economy and economics,” Int. Rev. Appl. Econ., vol. 34, no. 3, pp. 301–303, 2020, doi: 10.1080/02692171.2020.175604
- [8] R. Shahid, S. T. Javed, and K. Zafar, “Feature selection based classification of sentiment analysis using biogeography optimization algorithm,” in 2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT), 2017, pp. 1–5, doi: 10.1109/ICIEECT.2017.7916549
- [9] Prastyo, Sumi, Dian, & Permanasari, “Tweets Responding to the Indonesian Government’s Handling of COVID-19: Sentiment Analysis Using SVM with Normalized Poly Kernel”, Journal of Information Systems Engineering and Business Intelligence, 2020, 6 (2), 112-122
- [10] G. Singh, B. Kumar, L. Gaur, and A. Tyagi, “Comparison between Multinomial and Bernoulli Naïve Bayes for Text Classification,” in 2019 International Conference on Automation, Computational and Technology Management, ICACTM, 2019, pp. 593–596, doi: 10.1109/ICACTM.2019.8776800
- [11] F. Ratnawati and E. Winarko, “Sentiment Analysis of Movie Opinion in Twitter Using Dynamic Convolutional Neural Network Algorithm,” IJCCS (Indonesian J. Comput. Cybern. Syst., vol. 12, no. 1, pp. 1, 2018, doi: 10.22146/ijccs.19237.
- [12] N. D. Putranti and E. Winarko, “Analisis Sentimen Twitter untuk Teks Berbahasa Indonesia dengan *Maximum Entropy* dan *Support Vector Machine*,” IJCCS (Indonesian J. Comput. Cybern. Syst., vol. 8, no. 1, pp. 91, 2014.ISSN: 1978-1520
- [13] L. B. Ilmawan and E. Winarko, “Aplikasi Mobile untuk Analisis Sentimen pada Google Play” IJCCS (Indonesian J. Comput. Cybern. Syst., vol. 9, no. 1, pp. 53, 2015.ISSN: 1978-1520