

KOMPARASI ALGORITMA DATA MINING UNTUK ANALISIS SENTIMEN APLIKASI PEDULILINDUNGI

Hiras Parasian Doloksaribu ^{a,1,*}, Yusran Timur Samuel ^{b,2}

^a Fakultas Teknologi Informasi, Universitas Advent Indonesia, Kab. Bandung Barat

^b Fakultas Teknologi Informasi, Universitas Advent Indonesia, Kab. Bandung Barat

¹ 1881011@unai.edu*; ² y.tarihoran@unai.edu

* corresponding author

ARTICLE INFO

ABSTRACT

Keywords

PeduliLindungi
Analisis Sentimen
Komparasi
Naive Bayes, Support
Vector Machine
TF IDF Vectorizer
Count Vectorizer

The COVID-19 pandemic has caused many changes to occur in Indonesia. In the PeduliLindungi application to the community, the government makes the application to the community in the hope of being able to provide a warning if it enters the covid-19 zone and various other information from covid-19 [1]. The main purpose of this study is to analyze the sentiments of PeduliLindungi users who are currently used during the Covid-19 pandemic, where this application has begun to be used to travel anywhere and anytime to find out whether the user has vaccinated or not and various other things. such as the spread of the virus and the location of vaccination. The dataset for this study was taken from the Play Store. The algorithm used is Support Vector Machine and Naive Bayes to classify the data set. The data collection technique is Text Mining and compares the results of the two specified algorithms. The results of this research are Support Vector Machine with TF IDF Vectorizer with 89.05% accuracy followed by Support Vector Machine with Count Vectorizer, Naive Bayes with TF IDF Vectorizer and Naive Bayes with Count Vectorizer.

1. Pendahuluan

Pandemi Covid-19 menyebabkan banyak hal berubah di Indonesia saat ini, demikian pula aplikasi perangkat lunak dan komputer yang makin berkembang pesat di era pandemi ini menyebabkan Pemerintah membuat suatu aplikasi bernama PeduliLindungi dimana aplikasi tersebut berguna untuk mengetahui penyebaran virus Covid-19, lokasi vaksinasi, zona dari suatu daerah hingga para masyarakat saling berbagi data lokasi untuk bepergian agar riwayat kontak terhadap pasien Covid-19 dapat diketahui. Dengan adanya aplikasi ini, pengguna akan mendapatkan pemberitahuan jika sedang berada di zona merah Covid-19 pada suatu tempat yang telah terdaftar pasien pasien positif Covid-19 atau pasien yang masih dalam pengawasan [2].

Dalam penelitian ini analisis sentimen digunakan untuk menganalisis pendapat atau opini dari pengguna aplikasi PeduliLindungi berdasarkan ulasan (*review*) pengguna yang diambil dari Play Store. Analisis sentimen bisa dibidang sebagai penambahan opini dan dalam penelitian ini berpusat pada opini positif dan opini negative [3]. Analisis sentimen menggunakan algoritma Data Mining dalam hal analisa, proses, dan mengubah data berupa teks pada suatu objek atau entitas, misalnya pelayanan, suatu barang, tokoh publik, suatu kejadian, atau suatu subjek terpilih [3] dan dalam hal ini peneliti memilih aplikasi PeduliLindungi sebagai objek penelitian.

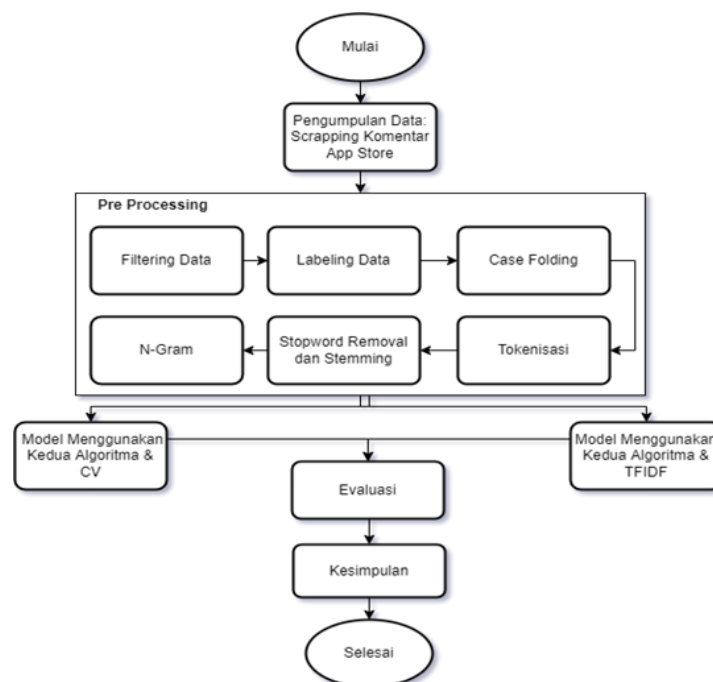
Mardiana, 2019 dalam penelitiannya menguji lima algoritma Data Mining yaitu; *Naive Bayes*, *Neural Network*, *Decision Tree*, *Support Vector Machine* dan *k-Nearest Neighbor* dengan hasil algoritma *Naive Bayes* dan *Support Vector Machine* menjadi algoritma dengan performa terbaik dalam mengolah data berupa teks [4]. Maka dari itu penelitian ini menggunakan kedua algoritma tersebut untuk melakukan komparasi dengan tambahan komparasi untuk pembobotan data menggunakan *Term Frequency-Inverse Document Frequency Vectorizer* dan *Count Vectorizer* di dalam kedua algoritma tersebut.

Ni Putu, 2019 dalam penelitiannya menganalisis sentimen Facebook menggunakan *Naive Bayes* melakukan pengujian akurasi sebanyak tujuh kali pengujian dengan jumlah sampel yang berbeda guna mengetahui data terbaik dalam menganalisis sentiment untuk dataset tersebut [5]. Fitriyani, 2021 dalam penelitiannya menganalisis sentiment aplikasi Sambara menggunakan *Support Vector Machine* juga menggunakan jumlah sampel yang berbeda sebanyak lima jumlah sampel berbeda [6]. Pada penelitian ini peneliti melakukan hal yang sama untuk mengetahui sampel terbaik dalam menganalisis sentiment dengan dataset yang berasal dari ulasan (*review*) pengguna PeduliLindungi di PlayStore.

Dalam penelitian ini untuk melakukan komparasi, digunakannya algoritma *Naive Bayes* dan *Support Vector Machine* sebagai algoritma yang bisa membantu pengimplementasian guna mendapatkan hasil terbaik dari komparasi kedua algoritma tersebut. Penelitian ini menggunakan metode *Text Mining* sebagai metode untuk mengumpulkan data dimana data yang diambil adalah data tulisan berupa opini publik yang akan diolah. Sumber data untuk penelitian ini adalah mengumpulkan data dari Play Store berupa ulasan (*review*) atau tinjauan dari para pengguna aplikasi PeduliLindungi tersebut. Dengan adanya penelitian ini diharapkan bisa mengetahui sentimen terhadap pengguna aplikasi PeduliLindungi dan mengetahui algoritma dan pembobotan kata terbaik dalam menganalisis sentimen pengguna aplikasi PeduliLindungi.

2. Metodologi Penelitian

Metode penelitian merupakan acuan, rencana dan langkah-langkah dalam pelaksanaan penelitian ini guna mencapai tujuan penelitian. Metode penelitian memiliki langkah-langkah rencana yang teratur sehingga hasil yang diperoleh sesuai dengan yang diharapkan. Oleh karena itu, diperlukan langkah yang terstruktur [8]. Gambar 1 berikut ini menerangkan langkah-langkah yang dilakukan dalam penyusunan Penelitian ini:



Gambar 1. Metodologi Penelitian

2.1 Text Mining

Text mining merupakan teknik mengolah data yang berbentuk teks, biasanya mengekstraksi sumber data dari dokumen dengan tujuan mendeteksi kata-kata yang bisa memperantarai isi artikel atau teks untuk melakukan analisis hubungan antar dokumen. Oleh karena itu, basis untuk bahan yang dipakai dalam *Text Mining* adalah berupa kelompok dokumen dengan format yang tidak teratur dengan mengidentifikasi dan mencari pola yang menarik. *Text Mining* dimanfaatkan untuk mengolah dokumen yang bermanfaat dari berbagai data yang sumber datanya adalah teks, yang memiliki format tidak terstruktur. Proses pengambilan informasi dalam teks mining dapat menghasilkan analisis emosional yang mengidentifikasi pernyataan secara emosional, baik positif maupun negatif [7].

2.2 Naive Bayes

Pengklasifikasi *Naive Bayes* adalah pengklasifikasi probabilitas sederhana yang menerapkan Teorema Bayesian dengan asumsi nilai Entitas independensi tinggi dimana label kelas diambil dari beberapa himpunan berhingga [8] [9]. Metode *Naive Bayes* merupakan metode yang memiliki beberapa keunggulan, seperti kesederhanaan dalam perhitungan, presisi tinggi dan kecepatan dalam memproses database besar [10].

$$P(H|X) = \frac{P(X|H) P(H)}{P(X)} \quad (1)$$

Dimana X merupakan Data pada label yang belum diketahui, H adalah dugaan dari data X yang merupakan suatu label yang lebih khas, P (H|X) merupakan peluang dugaan H beralaskan keadaan X (*posteriori probability*), P (H) merupakan peluang dugaan H (*prior probability*), P (X|H) merupakan Peluang dari X beralaskan keadaan pada dugaan H, P (X) merupakan Peluang dari X [11].

2.3 Support Vector Machine

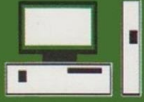
Support Vector Machine (SVM) adalah algoritma klasifikasi memproses data dengan menemukan *hyperplanes*, dengan tepi terbesar dari *hyperplanes* menjadi pembagi data antar kelas. *Margin* adalah jarak antara *hyperplanes* dengan data terdekat di setiap layer [12]. Data yang paling dekat dengan *hyperplanes* untuk masing-masing layer disebut *support vector*. Tujuan *Support Vector Machine* adalah untuk menguraikan data Tergantung pada margin *hyperplane* maksimum. Oleh karena itu, *Support Vector Machine* memastikan untuk memaksimalkan jarak antara data yang paling dekat dengan *hyperplane* [13].

$$y(x) = wT\phi(x) + b \quad (2)$$

Dimana x merupakan vektor input, w merupakan parameter bobot, $\phi(x)$ merupakan fungsi basis, dan b adalah suatu bias.

2.4 Pengumpulan Data

Pengumpulan data adalah langkah awal yang penting dalam metodologi penelitian. Dimana di tahap ini data dikumpulkan untuk keperluan penelitian. Data yang digunakan untuk penelitian harus benar-benar akurat dan jelas sumbernya. Metode pengambilan data dalam penelitian ini menggunakan metode scrapping yang dipergunakan untuk memetik data komentar, ulasan dari sebuah halaman website atau aplikasi Android dari Play Store kemudian mengolah data tersebut kedalam suatu format yang dapat disimpan [14] menggunakan python dengan link tujuan aplikasi PeduliLindungi dalam Play Store yaitu “com.telkom.tracencare” menggunakan library *pandas* dan *numpy* pada ulasan (*review*) aplikasi PeduliLindungi pada versi 4.0.2 dengan code sebagai berikut:



```
from google_play_scraper import Sort, reviews_all

result = reviews_all(
    'com.telkom.tracencare',
    sleep_milliseconds=0,
    lang='id',
    country='id',
    sort=Sort.MOST_RELEVANT,
)

df = pd.DataFrame(np.array(result), columns=['review'])

df.to_csv("pedulilindungiscrap.csv", index=False)
```

2.6 Filtering

Filtering data digunakan untuk menghilangkan atau menghapus data duplikat dan memilih attribute apa saja yang akan digunakan seperti username beserta data ulasan (*review*) lalu disimpan dalam format excel atau csv.

2.7 Labeling

Proses pelabelan data diperlukan untuk menentukan kelas daripada suatu ulasan (*review*) dalam dokumen apakah ulasan tersebut termasuk kedalam kelas berlabel positif atau negatif dimana kelas positif berisi sanjungan, usulan, dan perasaan positif seperti kepuasan dari pengguna aplikasi, kesenangan dan kebahagiaan dari para pengguna aplikasi tersebut. Sedangkan dalam kelas berlabel negatif berisi keluh kesah, keberatan, protes, kecaman, dan perasaan negatif seperti kemarahan, kemurkaan dan kekecewaan [15]. Penelitian ini menggunakan pelabelan secara *crowdsourcing* dengan menggunakan beberapa orang pelabel guna memaksimalkan hasil pelabelan [16] sesuai dengan sentimen dari ulasan (*review*) aplikasi PeduliLindungi tersebut dengan contoh hasil pada Tabel 1.

Tabel 1. Labeling

	<i>Sentiment</i>	<i>Content</i>
1	<i>Negative</i>	Gimana ini aplikasi kok ga bisa digunakan
2	<i>Negative</i>	Setelah di update versi terbaru kok gabisa dibuka
3	<i>Positive</i>	Sangat Membantu
4	<i>Negative</i>	Sekarang Update Terbaru malah ga bisa masuk

2.8 Case Folding

Case folding adalah proses penyatuan alfabet dengan mengubah semua huruf menjadi *lowercase* [17] dan menghapus punctuasi dan nominal (dalam hal ini hanya karakter dari a hingga z) guna merapihkan data dengan contoh hasil seperti pada Tabel 2.

Tabel 2. Case Folding

I	<i>text_real</i>	<i>setelah_caseFolding</i>
1	Gimana ini aplikasi kok ga bisa digunakan	gimana ini aplikasi kok ga bisa digunakan
2	Setelah di update versi terbaru kok gabisa dibuka	setelah di update versi terbaru kok gabisa dibuka
3	Sangat Membantu	sangat membantu
4	Sekarang Update Terbaru malah ga bisa masuk	sekarang update terbaru malah ga bisa masuk

2.9 Tokenisasi

Tokenisasi adalah teknik pemecahan beralaskan kata-kata yang membentuk string input, yang bisa diartikan sebagai teknik pemecahan kalimat menjadi kata-kata melalui cara membelah kata-kata tersebut untuk mengurai dan menentukan sekumpulan kata struktur sintaks pada setiap kata [17] dengan contoh hasil pada Tabel 3.

Tabel 3. Tokenisasi

	<i>setelah_caseFolding</i>	<i>setelah_Tokenisasi</i>
1	gimana ini aplikasi kok ga bisa digunakan	[gimana, ini, aplikasi, kok, ga, bisa, digunakan]
2	setelah di update versi terbaru kok gabisa dibuka	[setelah, di, update, versi, terbaru, kok, gabisa, dibuka]
3	sangat membantu	[sangat, membantu]
4	sekarang update terbaru malah ga bisa masuk	[sekarang, update, terbaru, malah, ga, bisa, masuk]

2.10 Stopword Removal and Stemming

Stopword removal adalah teknik untuk menghapus konjungsi atau kata berimbuhan seperti ‘dan’, ‘yang’, atau ‘ke’ hingga hanya menempatkan kata-kata bermakna yang memiliki arti penting [18]. Selanjutnya mengoreksi kata-kata yang memiliki kesalahan dalam eja dan kata-kata pendek tertentu dalam format template data yang sudah ada dengan contoh hasil seperti pada Tabel 4.

Tabel 4. Stopword Removal and Stemming

	<i>setelah_Tokenisasi</i>	<i>Stopword & Stemming</i>
1	[gimana, ini, aplikasi, kok, ga, bisa, digunakan]	[aplikasi, ga, bisa, guna]
2	[setelah, di, update, versi, terbaru, kok, ga, bisa, dibuka]	[update, versi, baru, ga, bisa, buka]
3	[sangat, membantu]	[bantu]
4	[sekarang, update, terbaru, malah, ga, bisa, masuk]	[update, baru, ga, bisa, masuk]

2.11 N-Gram

Setelah penghapusan *stopword* dan proses *stemming* proses selanjutnya adalah membuat *n-gram* dimana setiap kata yang ada digabungkan secara berurutan sesuai dengan n yang ditentukan [6]. pada penelitian ini n yang ditentukan adalah (1,2) dimana minimal 1 dan maksimal 2. Dengan Proses *n-gram* ini dicontohkan pada Tabel 5.

Tabel 5. N-Gram

<i>Stopword & Stemming</i>	<i>setelah_N-Gram (1,2)</i>
1 [aplikasi, ga, bisa, guna]	[aplikasi, aplikasi ga, ga, ga bisa, bisa, bisa guna, guna]
2 [update, versi, baru, ga, bisa, buka]	[update, update versi, versi, versi baru, baru, baru ga, ga, ga bisa, bisa, bisa buka, buka]
3 [bantu]	[bantu]
4 [update, baru, ga, bisa, masuk]	[update, update baru, baru, baru ga, ga, ga bisa, bisa, bisa masuk, masuk, masuk]

2.12 Pengujian

Pada tahap ini dilakukannya penerapan algoritma Data Mining untuk mengolah data berupa teks opini publik terhadap aplikasi PeduliLindungi yang sudah didapatkan. Pada tahap ini dilakukannya klasifikasi terhadap data ulasan (*review*) terhadap aplikasi PeduliLindungi di Play Store menggunakan algoritma *Support Vector Machine* dan *Naïve Bayes* dibantu dengan pembobotan *TF-IDF Vectorizer* atau *Count Vectorizer* dengan jumlah data set 1948 ulasan (*review*) aplikasi PeduliLindungi dengan data training lima kali pengujian sebanyak 200 data, 400 data, 600 data, 800 data 1948 data dengan data testing memiliki jumlah yang sama dengan data testing untuk masing masing pengujian.

2.13 Penarikan Kesimpulan

Kesimpulan adalah suatu langkah untuk menentukan hasil dari pengujian yang telah dilakukan. Hal ini bertujuan untuk memahami dan memberikan informasi tentang algoritma Data Mining mana yang lebih baik untuk mengolah data teks yang ada [3]. Ini juga menganalisis hasil dari algoritma terbaik yang juga dikombinasikan dengan *Count vectorizer* atau *TF IDF vectorizer*.

3. Hasil dan Pembahasan

3.1 Pengujian Menggunakan *Naive Bayes* dan *Support Vector Machine*

Pengujian dilakukan menggunakan bahasa pemrograman python dengan jupyter notebook menggunakan *X Cross Validation* dengan algoritma *Naive bayes* dan *Support vector machine* dengan rumus dan hasil sebagai berikut ini:

<pre>print('Naive Bayes & Count Vectorizer') X = cvzr_df.to_numpy() y = df_skripsi.label y_pred = cross_val_predict(nb, X, y, cv=10) conf_mat = confusion_matrix(y_pred, y) print(conf_mat) print(classification_report (y, y_pred, digits = 4))</pre>	<pre>Naive Bayes & Count Vectorizer [[1293 165] [174 313]]</pre> <table border="1"> <thead> <tr> <th></th> <th>precision</th> <th>recall</th> <th>f1-score</th> <th>support</th> </tr> </thead> <tbody> <tr> <td>Negative</td> <td>0.8868</td> <td>0.8814</td> <td>0.8841</td> <td>1467</td> </tr> <tr> <td>Positive</td> <td>0.6427</td> <td>0.6548</td> <td>0.6487</td> <td>478</td> </tr> <tr> <td>accuracy</td> <td></td> <td></td> <td>0.8257</td> <td>1945</td> </tr> <tr> <td>macro avg</td> <td>0.7648</td> <td>0.7681</td> <td>0.7664</td> <td>1945</td> </tr> <tr> <td>weighted avg</td> <td>0.8268</td> <td>0.8257</td> <td>0.8263</td> <td>1945</td> </tr> </tbody> </table>		precision	recall	f1-score	support	Negative	0.8868	0.8814	0.8841	1467	Positive	0.6427	0.6548	0.6487	478	accuracy			0.8257	1945	macro avg	0.7648	0.7681	0.7664	1945	weighted avg	0.8268	0.8257	0.8263	1945
	precision	recall	f1-score	support																											
Negative	0.8868	0.8814	0.8841	1467																											
Positive	0.6427	0.6548	0.6487	478																											
accuracy			0.8257	1945																											
macro avg	0.7648	0.7681	0.7664	1945																											
weighted avg	0.8268	0.8257	0.8263	1945																											

Gambar 2. *Naive Bayes* dengan *Count Vectorizer*

Pada Gambar 4 hasil pengujian Algoritma *Naive Bayes* dengan *Count Vectorizer* menghasilkan nilai akurasi 82,57%. Nilai *Class Precision* Negatif 88,68% , *Class Precision* Positif 64,27% , *Class Recall* Negatif 88,14% dan *Class Recall* Positif 65,48%.

<pre>print('Naive Bayes & TF-IDF Vectorizer') X = tfidf_df.to_numpy() y = df_skripsi.label y_pred = cross_val_predict(nb, X, y, cv=10) conf_mat = confusion_matrix(y_pred, y) print(conf_mat) print(classification_report (y, y_pred, digits = 4))</pre>	<table border="1"> <thead> <tr> <th colspan="5">Naive Bayes & TFIDF Vectorizer</th> </tr> <tr> <th></th> <th>precision</th> <th>recall</th> <th>f1-score</th> <th>support</th> </tr> </thead> <tbody> <tr> <td>Negative</td> <td>0.8863</td> <td>0.8821</td> <td>0.8842</td> <td>1467</td> </tr> <tr> <td>Positive</td> <td>0.6433</td> <td>0.6527</td> <td>0.6480</td> <td>478</td> </tr> <tr> <td>accuracy</td> <td></td> <td></td> <td>0.8257</td> <td>1945</td> </tr> <tr> <td>macro avg</td> <td>0.7648</td> <td>0.7674</td> <td>0.7661</td> <td>1945</td> </tr> <tr> <td>weighted avg</td> <td>0.8266</td> <td>0.8257</td> <td>0.8261</td> <td>1945</td> </tr> </tbody> </table>	Naive Bayes & TFIDF Vectorizer						precision	recall	f1-score	support	Negative	0.8863	0.8821	0.8842	1467	Positive	0.6433	0.6527	0.6480	478	accuracy			0.8257	1945	macro avg	0.7648	0.7674	0.7661	1945	weighted avg	0.8266	0.8257	0.8261	1945
Naive Bayes & TFIDF Vectorizer																																				
	precision	recall	f1-score	support																																
Negative	0.8863	0.8821	0.8842	1467																																
Positive	0.6433	0.6527	0.6480	478																																
accuracy			0.8257	1945																																
macro avg	0.7648	0.7674	0.7661	1945																																
weighted avg	0.8266	0.8257	0.8261	1945																																

Gambar 3. *Naive Bayes* dengan *TF-IDF Vectorizer*

Pada Gambar 5 hasil pengujian Algoritma *Naive Bayes* dengan *TF IDF Vectorizer* menghasilkan nilai akurasi 82,57%. Nilai *Class Precision* Negatif 88,63% , *Class Precision* Positif 64,33% , *Class Recall* Negatif 88,21% dan *Class Recall* Positif 65,27%.

<pre>print('SVM & Count Vectorizer') X = cvzr_df.to_numpy() y = df_skripsi.label y_pred = cross_val_predict(svm, X, y, cv=10) conf_mat = confusion_matrix(y_pred, y) print(conf_mat) print(classification_report (y, y_pred, digits = 4))</pre>	<table border="1"> <thead> <tr> <th colspan="5">SVM & Count Vectorizer</th> </tr> <tr> <th></th> <th>precision</th> <th>recall</th> <th>f1-score</th> <th>support</th> </tr> </thead> <tbody> <tr> <td>Negative</td> <td>0.8856</td> <td>0.9502</td> <td>0.9168</td> <td>1467</td> </tr> <tr> <td>Positive</td> <td>0.8032</td> <td>0.6234</td> <td>0.7020</td> <td>478</td> </tr> <tr> <td>accuracy</td> <td></td> <td></td> <td>0.8699</td> <td>1945</td> </tr> <tr> <td>macro avg</td> <td>0.8444</td> <td>0.7868</td> <td>0.8094</td> <td>1945</td> </tr> <tr> <td>weighted avg</td> <td>0.8654</td> <td>0.8699</td> <td>0.8640</td> <td>1945</td> </tr> </tbody> </table>	SVM & Count Vectorizer						precision	recall	f1-score	support	Negative	0.8856	0.9502	0.9168	1467	Positive	0.8032	0.6234	0.7020	478	accuracy			0.8699	1945	macro avg	0.8444	0.7868	0.8094	1945	weighted avg	0.8654	0.8699	0.8640	1945
SVM & Count Vectorizer																																				
	precision	recall	f1-score	support																																
Negative	0.8856	0.9502	0.9168	1467																																
Positive	0.8032	0.6234	0.7020	478																																
accuracy			0.8699	1945																																
macro avg	0.8444	0.7868	0.8094	1945																																
weighted avg	0.8654	0.8699	0.8640	1945																																

Gambar 4. *Support Vector Machine* dengan *Count Vectorizer*

Pada Gambar 6 hasil pengujian Algoritma *Support Vector Machine* dengan *Count Vectorizer* menghasilkan nilai akurasi 86,99%. Nilai *Class Precision* Negatif 88,56% , *Class Precision* Positif 80,32% , *Class Recall* Negatif 95,02% dan *Class Recall* Positif 62,34%.

<pre>print('SVM & TFIDF Vectorizer') X = tfidf_df.to_numpy() y = df_skripsi.label y_pred = cross_val_predict(svm, X, y, cv=10) conf_mat = confusion_matrix(y_pred, y) print(conf_mat) print(classification_report (y, y_pred, digits = 4))</pre>	<table border="1"> <thead> <tr> <th colspan="5">SVM & TFIDF Vectorizer</th> </tr> <tr> <th></th> <th>precision</th> <th>recall</th> <th>f1-score</th> <th>support</th> </tr> </thead> <tbody> <tr> <td>Negative</td> <td>0.8837</td> <td>0.9843</td> <td>0.9313</td> <td>1467</td> </tr> <tr> <td>Positive</td> <td>0.9260</td> <td>0.6025</td> <td>0.7300</td> <td>478</td> </tr> <tr> <td>accuracy</td> <td></td> <td></td> <td>0.8905</td> <td>1945</td> </tr> <tr> <td>macro avg</td> <td>0.9049</td> <td>0.7934</td> <td>0.8307</td> <td>1945</td> </tr> <tr> <td>weighted avg</td> <td>0.8941</td> <td>0.8905</td> <td>0.8818</td> <td>1945</td> </tr> </tbody> </table>	SVM & TFIDF Vectorizer						precision	recall	f1-score	support	Negative	0.8837	0.9843	0.9313	1467	Positive	0.9260	0.6025	0.7300	478	accuracy			0.8905	1945	macro avg	0.9049	0.7934	0.8307	1945	weighted avg	0.8941	0.8905	0.8818	1945
SVM & TFIDF Vectorizer																																				
	precision	recall	f1-score	support																																
Negative	0.8837	0.9843	0.9313	1467																																
Positive	0.9260	0.6025	0.7300	478																																
accuracy			0.8905	1945																																
macro avg	0.9049	0.7934	0.8307	1945																																
weighted avg	0.8941	0.8905	0.8818	1945																																

Gambar 5 *Support Vector Machine* dengan *TF-IDF Vectorizer*

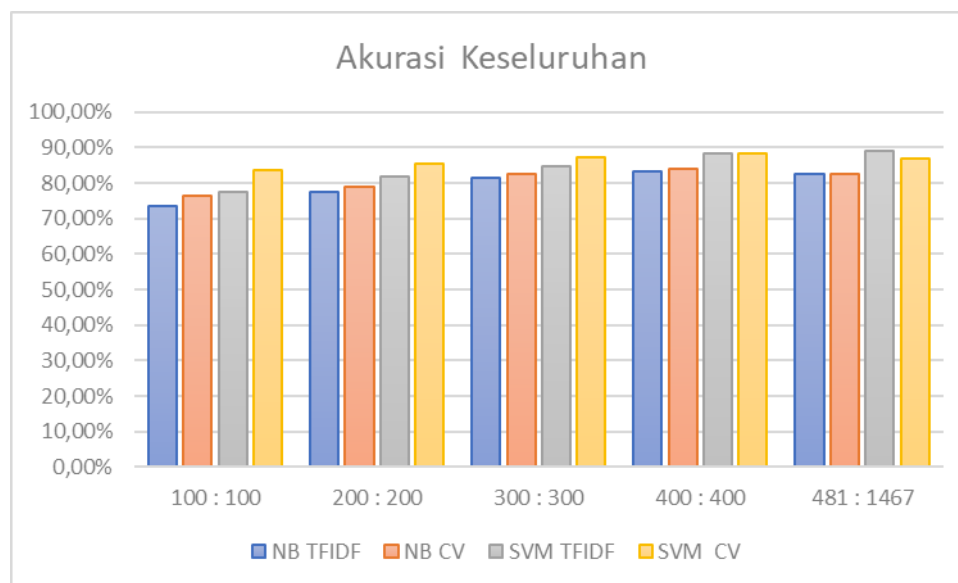
Pada Gambar 7 hasil pengujian Algoritma *Support Vector Machine* dengan *TF IDF Vectorizer* menghasilkan nilai akurasi 89,05%. Nilai *Class Precision* Negatif 88,37% , *Class Precision* Positif 92,60% , *Class Recall* Negatif 98,43% dan *Class Recall* Positif 60,25%.

Jumlah Data		Naïve Bayes & TFIDF					Naïve Bayes & CV				
		Precision		Recall		Accuracy	Precision		Recall		Accuracy
Positive	Negative	Positive	Negative	Positive	Negative		Positive	Negative	Positive	Negative	
100	100	84,06%	67,94%	58%	89%	73,50%	86,30%	70,87%	63%	90%	76,50%
200	200	86,67%	72%	65%	90%	77,50%	87,18%	73,77%	68%	90%	79%
300	300	91,63%	75,34%	69,33%	93,67%	81,50%	91,49%	76,71%	71,67%	93,33%	82,50%
400	400	91,56%	77,71%	73,25%	93,25%	83,25%	91,74%	78,86%	75%	93,25%	84,13%
481	1467	64,33%	88,63%	65,27%	88,21%	82,57%	64,27%	88,68%	65,48%	88,14%	82,57%

Gambar 6. Hasil Uji Naive Bayes Keseluruhan

Jumlah Data		SVM & TFIDF					SVM & CV				
		Precision		Recall		Accuracy	Precision		Recall		Accuracy
Positive	Negative	Positive	Negative	Positive	Negative		Positive	Negative	Positive	Negative	
100	100	79,57%	75,70%	74%	81%	77,50%	87,64%	80,18%	78%	89,00%	83,50%
200	200	81,91%	81,59%	81,50%	82%	81,75%	86,98%	84,13%	83,50%	87,50%	85,50%
300	300	85,19%	84,49%	84,33%	85,33%	84,83%	88,58%	85,85%	85,33%	89%	87,17%
400	400	90,48%	86,26%	85,50%	91%	88,25%	89,64%	86,96%	86,50%	90%	88,25%
481	1467	92,60%	88,37%	60,25%	98,43%	89,05%	80,32%	88,56%	62,34%	95,02%	86,99%

Gambar 7. Hasil Uji SVM Keseluruhan



Gambar 8. Grafik Akurasi Keseluruhan

Pada Gambar 6, Gambar 7 dan Gambar 8 dengan jumlah sampel data 1:1 antara positif dan negatif dari 100 : 100 hingga 400 : 400 data mengalami peningkatan akurasi di masing masing algoritma. Namun disaat data tidak lagi 1:1 pada pengujian kelima yaitu dengan 481 data positif dan 1467 data negatif hanya Algoritma *Support Vector Machine* dengan *TF IDF Vectorizer* yang mengalami kenaikan Akurasi hingga 89,05% sedangkan pengujian lain mengalami penurunan akurasi bila dibandingkan saat pengujian 400:400 data yaitu *Support Vector Machine* dengan *Count Vectorizer* dari 88,25% turun ke 86,99% diikuti *Naive Bayes* dengan *TF IDF Vectorizer* dari 83,25% turun ke 82,57% dan *Naive Bayes* dengan *Count Vectorizer* dari 84,13% turun ke 82,57%.

3.2 Word cloud

Berikut ini adalah kata kata yang sering muncul di masing masing label positif dan negatif yang sudah dibentuk ke dalam *word cloud* guna memperlihatkan ke dalam bentuk visual.



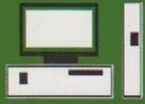
Gambar 8. Word Cloud Positif

Gambar 9. Word Cloud Negatif

4. Kesimpulan

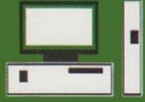
Pada penelitian ini peneliti menggunakan dua algoritma data mining yaitu *Naive Bayes* dan *Support Vector Machine* dalam melakukan klasifikasi sentimen dan membandingkan kedua algoritma tersebut guna melihat algoritma mana yang lebih baik dalam hal kasus tersebut dengan komentar pengguna aplikasi PeduliLindungi sebagai data penelitian dengan lima kali pengujian dengan jumlah sampel yang berbeda. Dalam hasil dan pembahasan pengujian bisa disimpulkan bahwa Algoritma *Support Vector Machine* yang dibantu *TF IDF Vectorizer* dalam kasus ini menjadi algoritma yang terbaik dalam tingkat akurasi dan perbandingan data yang bukan 1:1 pada masing masing label tidak menjadi halangan bagi algoritma ini untuk meningkatkan performanya dengan data yang lebih banyak. Dimana hal ini berarti untuk kasus ini algoritma *Support Vector Machine* dengan *TF IDF Vectorizer* memiliki juga tingkat ketelitian yang tinggi dibandingkan menggunakan *Count Vectorizer* ataupun dengan Algoritma *Naive Bayes*.

Pada hasil *word cloud* positif di Gambar 8 dan *word cloud* negatif di Gambar 9 bisa dilihat perspektif positif dan perspektif negatif dari para pengguna aplikasi PeduliLindungi. Dimana dalam *word cloud* negatif bisa dilihat bahwa para pengguna memiliki beberapa ketidakpuasan dan kekecewaan terhadap aplikasi tersebut dimana kata-kata yang sering muncul dalam *word cloud* negatif adalah 'Login', 'Aplikasi', 'Muncul', 'Sertifikat', 'Vaksin', 'Tolong', 'Susah' dengan frekuensi yang tinggi dimana para pengguna memiliki permasalahan login dalam aplikasi dan kesulitan dalam penggunaan juga banyak dari para pengguna yang mengeluh tentang sertifikat vaksin yang tidak kunjung muncul. Berdasarkan hasil ini aplikasi PeduliLindungi masih memiliki beberapa kekurangan terlihat dari banyaknya pengguna aplikasi tersebut yang merasa kecewa pada Oktober - November 2021. Dalam penelitian selanjutnya diharapkan menggunakan algoritma lain dengan tambahan *Word 2 Vectorizer*, *BERT* atau *Bag of Words* dalam pengimplementasian analisis opini publik.



Daftar Pustaka

- [1] KEMENTERIAN KOMUNIKASI DAN INFORMATIKA, “PeduliLindungi,” *Apa itu PeduliLindungi*. <https://www.pedulilindungi.id/> (diakses Nov 07, 2021).
- [2] I. W. Sudiarsa dan I. G. B. Wiraditya, “Analisis Usability Pada Aplikasi Peduli Lindungi Sebagai Aplikasi Informasi Dan Tracking Covid-19 Dengan Heuristic Evaluation,” *INTECOMS*, vol. 3, no. 2, hlm. 354–364, Desember 2020, doi: 10.31539/intecom.v3i2.1901.
- [3] P. Herino, “Klasifikasi Sentimen Layanan Ojek Online Menggunakan Metode Naive Bayes Classifier,” Skripsi, Universitas Islam Negeri Sultan Syarif Kasim, Riau, 2018. Diakses: Nov 08, 2021. [Daring]. Tersedia pada: <http://repository.uin-suska.ac.id/id/eprint/15975>
- [4] T. Mardiana, H. Syahreva, dan T. Tuslaela, “Komparasi Metode Klasifikasi Pada Analisis Sentimen Usaha Waralaba Berdasarkan Data Twitter,” *J. PILAR Nusa Mandiri*, vol. 15, no. 2, hlm. 267–274, Sep 2019, doi: 10.33480/pilar.v15i2.752<https://doi.org/10.33480/pilar.v15i2.752>.
- [5] P. S. M. Suryani, L. Linawati, dan K. O. Saputra, “Penggunaan Metode Naive Bayes Classifier pada Analisis Sentimen Facebook Berbahasa Indonesia,” *Maj. Ilm. Teknol. Elektro*, vol. 18, no. 1, hlm. 145–148, Apr 2019, doi: 10.24843/MITE.2019.v18i01.P22.
- [6] F. Fitriyani dan T. Arifin, “Penerapan Word N-Gram Untuk Sentiment Analysis Review Menggunakan Metode Support Vector Machine (Studi Kasus: Aplikasi Sambara),” *SISTEMASI*, vol. 9, no. 3, hlm. 610, Sep 2020, doi: 10.32520/stmsi.v9i3.954.
- [7] E. K. Putri dan T. Setiadi, “Penerapan Text Mining Pada Sistem Klasifikasi,” *JSTIF*, vol. 2, no. 3, hlm. 73–83, Oktober 2014, doi: 10.12928/jstie.v2i3.2877.
- [8] N. R. Ain, “Text Mining Dengan Metode Naive Bayes Classifier Untuk Mengklasifikasikan Berita Berdasarkan Konten,” Institut Teknologi Sepuluh November, Surabaya, 2018. Diakses: Nov 07, 2021. [Daring]. Tersedia pada: https://repository.its.ac.id/51007/1/1213100009-Undergraduate_Theses.pdf
- [9] A. Hermanto, “Implementasi Text Mining Menggunakan Naive Bayes Untuk Penentuan Kategori Tugas Akhir Mahasiswa Berdasarkan Abstraksinya,” *KONVERGENSI*, vol. 12, no. 2, hlm. 1–10, Jul 2016, doi: 10.30996/konv.v12i2.1310.g1107.
- [10] I. Ernawati, “Naive Bayes Classifier Dan Support Vector Machine Sebagai Alternatif Solusi Untuk Text Mining,” *J. Teknol. Inf. Dan Pendidik.*, vol. 12, no. 2, hlm. 33–39, Desember 2019, doi: 10.24036/tip.v12i2.219.
- [11] F. Ratnawati, “Implementasi Algoritma Naive Bayes Terhadap Analisis Sentimen Opini Film Pada Twitter,” *INOVTEK Polbeng - Seri Inform.*, vol. 3, no. 1, hlm. 50–59, Jun 2018, doi: 10.35314/isi.v3i1.335.
- [12] O. Somantri, S. Wiyono, dan D. Dairoh, “Optimalisasi Support Vector Machine (SVM) Untuk Klasifikasi Tema Tugas Akhir Berbasis K-Means,” *TELEMATIKA*, vol. 13, no. 2, hlm. 59–68, Jul 2016, doi: 10.31315/telematika.v13i2.1722.
- [13] D. Gunawan, D. Riana, D. Ardiansyah, F. Akbar, dan S. Alfarizi, “Komparasi Algoritma Support Vector Machine Dan Naive Bayes Dengan Algoritma Genetika Pada Analisis Sentimen Calon Gubernur Jabar 2018-2023,” *J. PILAR Nusa Mandiri*, vol. 6, no. 1, hlm. 121–129, Jan 2020.
- [14] A. Priyanto dan M. R. Ma’arif, “Implementasi Web Scraping dan Text Mining untuk Akuisisi dan Kategorisasi Informasi Laman Web Tentang Hidroponik,” *IJIS*, vol. 1, no. 1, hlm. 25–33, Agustus 2018.
- [15] S. Samsir, A. Ambiyar, U. Verawardina, F. Edi, dan R. Watrianthos, “Analisis Sentimen Pembelajaran Daring Pada Twitter di Masa Pandemi COVID-19 Menggunakan Metode Naive



- Bayes,” *J. MEDIA Inform. BUDIDARMA*, vol. 5, no. 1, hlm. 157–163, Jan 2021, doi: 10.30865/mib.v5i1.2580.
- [16] I. Ihsan, “Sentiment Analysis RKUHP Pada Twitter Menggunakan Metode Support Vector Machine,” *E-Proceeding Eng.*, vol. 8, no. 2, hlm. 3521–3536, Apr 2021.
- [17] N. Haqqizar dan T. N. Larasyanti, “Analisis Sentimen Terhadap Layanan Provider Telekomunikasi Telkomsel Di Twitter Dengan Metode Naïve Bayes,” *TAU SNAR-TEK*, vol. 1, no. 1, hlm. 30–33, Nov 2019.
- [18] S. Khairunnisa, A. Adiwijaya, dan S. A. Faraby, “Pengaruh Text Preprocessing terhadap Analisis Sentimen Komentar Masyarakat pada Media Sosial Twitter (Studi Kasus Pandemi COVID-19),” *J. MEDIA Inform. BUDIDARMA*, vol. 5, no. 2, hlm. 406–414, Apr 2021, doi: 10.30865/mib.v5i2.2835.