
Analisis Performa K-Nearest Neighbor Dalam Mengklasifikasi Sentimen Pengguna Aplikasi PeduliLindungi

Sulfayanti¹, Ika Nurul Fadhillah^{*2}, Nahya Nur³

^{1,2,3}Program Studi Informatika, Fakultas Teknik, Universitas Sulawesi Barat

E-mail: ¹sulfayanti@unsulbar.ac.id, ²ika@gmail.com, ³nahyanur@unsulbar.ac.id

Abstrak

Virus Corona 19 (Covid-19) merupakan invensi vius menular yang kini telah menyebar ke berbagai negara, termasuk Indonesia. Pemantauan penyebaran Covid-19 di Indonesia ditangani langsung oleh Pemerintah Indonesia khususnya oleh Kementerian Komunikasi dan Informatika (KOMINFO) dengan pembuatan aplikasi PeduliLindungi yang dapat diunduh di Google Play Store. Pengguna memberikan ulasan atau komentar mereka tentang aplikasi dengan tujuan untuk mengevaluasi dan meningkatkan kualitas produk kedepannya. Salah satu cara dalam mengevaluasi ulasan adalah dengan menganalisis sentiment ulasan tersebut. Analisis terhadap sentiment ulasan dilakukan untuk menentukan apakah opini atau komentar terhadap suatu permasalahan atau isu terkait topik tertentu memiliki kecenderungan positif, negatif, atau netral. Sehingga peneliti ingin mengetahui sejauh mana gambaran umum masyarakat mengenai Aplikasi PeduliLindungi dengan memanfaatkan hasil analisis ini dengan melakukan teknik klasifikasi terhadap ulasan PeduliLindungi menggunakan Algoritma K-Nearest Neighbor. Hasil pengklasifikasian sentimen yang diperoleh yaitu opini kategori positif sebesar 58,66%, sentimen dengan kategori negatif sebesar 26,12%, dan sentimen dengan kategori netral sebesar 15,22% ulasan dari total 7.500 ulasan. Hasil ini menunjukkan bahwa sentiment positif masyarakat lebih tinggi terhadap aplikasi PeduliLindungi. Performa algoritma K-Nearest Neighbor dalam mengklasifikasi sentimen pengguna menghasilkan akurasi sebesar 60,66%, dengan nilai presisi 64%, recall 62,79%, dan F1-score 54,44%.

Kata kunci—PeduliLindungi, Analisis Sentimen, Text Mining, K-Nearest Neighbor

Abstract

Corona Virus 19 (Covid-19) is a contagious viral infection that has now spread to various countries, including Indonesia. Monitoring the spread of Covid-19 in Indonesia is handled directly by the Government of Indonesia, especially by the Ministry of Communication and Information (KOMINFO) by making the PeduliLindungi application which can be downloaded on the Google Play Store. Users provide their reviews or comments about the application with the aim of evaluating and improving the quality of the product in the future. One way to evaluate reviews is to use sentiment analysis. Sentiment analysis is carried out to determine whether opinions or comments on a problem or issue related to a particular topic have a positive, negative, or neutral tendency. So that researchers want to know the extent to which the general public regarding the Care for Protection Application by utilizing sentiment analysis and carrying out the PeduliLindungi review classification technique using the K-Nearest Neighbor Algorithm. Sentiment classification results obtained opinion in the positive category of 58.66%, sentiment in the negative category of 26.12%, and sentiment in the neutral

category of 15.22% of reviews from a total of 7,500 reviews. So it can be concluded that the public believes in the PeduliLindungi application and judges that the application is worthy of being installed. While the results of the classification of the K-Nearest Neighbor algorithm produce an accuracy of 60.66%, with a precision value of 64%, 62.79% recall, and 54.44% F1-score.

Keywords—PeduliLindungi, Sentiment Analysis, Text Mining, K-Nearest Neighbor

1. PENDAHULUAN

Pada akhir bulan Desember 2019 jenis baru virus *Corona* ditemukan di kota Wuhan, Cina. Virus ini merupakan keluarga besar virus yang menyerang manusia dan hewan, virus baru ini diberi nama *Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-COV2)*, dan menyebabkan penyakit *Coronavirus Disease-2019 (COVID-19)*. Virus *COVID-19* menyebar dengan sangat cepat di berbagai negara dengan total 431 juta kasus per tanggal 26 Februari 2022 [1]. Pada tanggal 11 Maret 2020 WHO (*World Health Organization*) telah menetapkan *COVID-19* sebagai pandemi global [2].

Di Indonesia kasus pertama virus *Corona* pertama kali ditemukan pada tanggal 2 Maret 2020 [2]. Pemerintah Republik Indonesia telah melakukan berbagai upaya untuk menangani penyebaran virus ini. Salah satunya ialah memantau penyebaran *COVID-19* dengan menerbitkan aplikasi *mobile* PeduliLindungi. Selain itu, pemerintah juga gencar dalam melakukan program vaksinasi massal terhitung sejak bulan Juli 2021 [3].

Aplikasi PeduliLindungi dibuat dengan tujuan untuk melindungi pengguna atau masyarakat dengan melacak riwayat kontak serta mengirimkan notifikasi kepada pengguna jika berada di zona merah atau area yang terinfeksi *COVID-19* [4]. Aplikasi PeduliLindungi juga memiliki fasilitas untuk mencetak kartu vaksin dan riwayat vaksin untuk pengguna yang telah melakukan vaksinasi.

Setiap aplikasi selalu memberi kesempatan bagi pengguna untuk dapat memberi rating dan ulasan terkait aplikasi. Hal ini digunakan oleh pengembang untuk dapat melakukan evaluasi guna meningkatkan kualitas aplikasi ke depannya [5]. Aplikasi PeduliLindungi hingga Maret 2022 mendapatkan rating sebesar 4,3 bintang dari total sebesar 5,0 dan ulasan dengan total 850.609 ulasan.

Kadangkala sebuah sistem membutuhkan perbaikan-perbaikan karena adanya kelemahan. Kelemahan pada aplikasi peduliLindungi dibuktikan dengan ulasan yang berisi keluhan-keluhan pengguna yang disampaikan melalui *user review* pada *Google Play Store*.

Banyaknya jumlah ulasan dapat menyulitkan pengguna dalam mengambil kesimpulan saat membaca ulasan. Pembacaan ulasan guna menarik kesimpulan dari hasil *review* dapat dipercepat dengan adanya ringkasan opini mengenai hasil *review* produk [6]. Salah satu cara untuk mendapatkan ringkasan opini guna mengevaluasi ulasan adalah dengan menganalisis sentimen yang terkandung di dalamnya.

Analisis yang dilakukan terhadap sentimen pada ulasan dapat digunakan untuk mengetahui kecenderungan opini publik terhadap suatu isu, topik permasalahan, atau bahkan produk/aplikasi tertentu dengan mengkategorikan opini tersebut ke dalam kategori positif, negatif, atau netral. Hal ini dapat dimanfaatkan untuk memperbaiki permasalahan juga untuk meningkatkan kualitas produk atau pelayanan yang ada [7].

Penelitian sebelumnya yang menganalisis sentimen pada ulasan yang terdapat pada *Google Play Store* telah banyak dikembangkan menggunakan algoritma klasifikasi, seperti penelitian yang dilakukan oleh Mustopa *et al.* [8] yang memanfaatkan *Support Vector Machine* dan *Naïve Bayes* berdasarkan *Particle Swarm Optimization* dalam melakukan klasifikasi sentimen aplikasi PeduliLindungi. Penelitian ini menunjukkan bahwa Algoritma *Support*

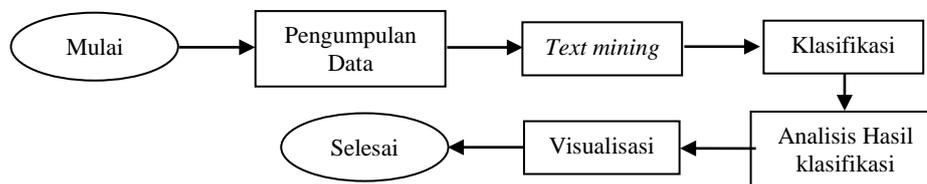
Vector Machine berbasis *PSO* menjadi algoritma unggulan yang memiliki nilai tertinggi dibandingkan *Naive Bayes* berbasis *PSO* yang memperoleh nilai akurasi 93,0% serta nilai AUC 0,977. Penelitian ini memiliki kesamaan dengan penelitian yang dilakukan yaitu untuk menganalisis sentiment pada aplikasi PeduliLindungi menggunakan komentar atau review yang terdapat pada *Google Play Store*. Hanya saja penelitian yang dilakukan ini memanfaatkan algoritma *K-Nearest Neighbor*.

Selanjutnya Indrayuni *et al.* [9] melakukan penelitian terkait ulasan aplikasi Halodoc dengan menggunakan algoritma *Naive Bayes*, *Support Vector Machine*, dan *K-Nearest Neighbors*. Hasil penelitian menunjukkan bahwa performa metode *K-Nearest Neighbor* lebih baik dan akurat dalam mengklasifikasikan sentimen yang berbahasa Inggris karena memiliki nilai akurasi yang paling tinggi yaitu 95,00% dan nilai AUC yang paling besar yaitu 0,985. Selain itu pada penelitian yang dilakukan oleh Fitrianti *et al.* [10] diketahui bahwa metode *K-Nearest Neighbor* dapat membantu dalam mengklasifikasikan analisis sentimen akun google restoran Taman Santap Rumah Kayu dengan sentimen berbahasa Indonesia yang menghasilkan nilai akurasi sebesar 96,61%. Penelitian ini menjadi dasar pemilihan metode yang digunakan untuk menyelesaikan proses klasifikasi pada penelitian yang dilakukan.

Berdasarkan uraian latar belakang diatas penelitian ini bertujuan untuk menganalisis performa Algoritma *K-Nearest Neighbor* dalam mengklasifikasikan sentimen terhadap ulasan aplikasi PeduliLindungi yang terdapat pada *Google Play Store*.

2. METODE

Secara garis besar alur pada penelitian ini dilakukan melalui beberapa tahapan yaitu pengumpulan data, *text-mining*, klasifikasi, analisis hasil klasifikasi, dan visualisasi sebagaimana Gambar 1.



Gambar 1 Alur penelitian

2.1 Pengumpulan Data

Data yang diolah berupa ulasan pada aplikasi PeduliLindungi pada situs *Google Play Store*. Pengumpulan data dilakukan dengan menggunakan teknik *scraping* data untuk mengesktraksi ulasan secara online dan memanfaatkan bahasa *Python*.

2. 2 Text Mining

Text Mining merupakan variasi teknik data mining yang dapat dimanfaatkan untuk mendapatkan intisari informasi dari sekumpulan dokumen *text* [11]. *Text mining* bekerja dengan mencari pola yang menarik dari sejumlah besar data tekstual [12]. Berikut adalah tahapan *text mining* yang dilakukan dalam penelitian ini, yaitu:

2. 2.1 Preprocessing

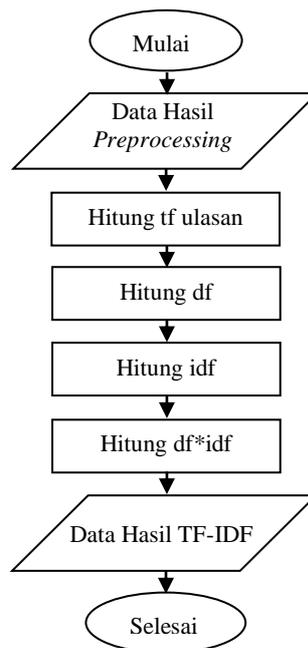
Preprocessing data adalah tahap awal yang mesti dilakukan dalam mengolah data guna menyeleksi data kemudian mentransformasinya menjadi data terstruktur. Terdapat beberapa tahapan proses yang dilakukan dalam *preprocessing*, seperti *Case folding*, *Tokenizing*, *Normalization*, dan *Filtering*. *Casefolding*, mengubah semua kata menjadi lowercase. *Tokenizing* adalah proses untuk mengubah kalimat atau teks pada ulasan sehingga menjadi kata terpisah yang tidak saling mempengaruhi. *Spelling normalization* adalah tahapan untuk

memperbaiki kata salah ketik (*typo*) dan kata tidak baku diperbaiki atau dikembalikan menjadi kata bakunya. *Filtering* merupakan proses membuang kata yang tidak memiliki arti dan pengaruh terhadap analisis ulasan namun seringkali muncul dalam dokumen. Proses *filtering* disebut dengan *Stopword Removal*. Proses untuk menghilangkan kata yang tidak dibutuhkan dilakukan dengan memanfaatkan *library* NLTK, yaitu sebuah *library* yang mempunyai kumpulan kata *stop word* berbahasa Indonesia.

2. 2.2 TF-IDF

Setelah proses *filtering* ulasan dilakukan pelabelan ulasan menggunakan kamus *InsetLexicon*. *InSet Lexicon* (Indonesia *Sentiment lexicon*) mengandung 3.609 kata positif dan 6.609 kata negatif dalam bahasa Indonesia. Setiap kata tersebut sudah memiliki bobot nilai (*polarity score*) yang berkisar antara -5 sampai dengan +5. *Polarity score* inilah yang kemudian digunakan dalam pengklasifikasian jenis sentiment [13]. Pelabelan dilakukan dengan mengategorikan ulasan ke dalam 3 kategori atau label yakni positif, netral, dan negatif.

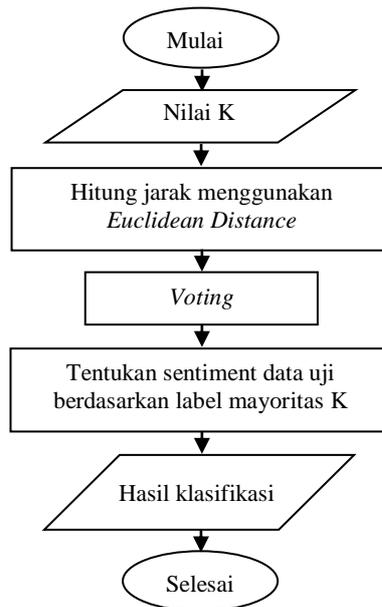
Tahap selanjutnya yaitu pembobotan TF-IDF. TF-IDF merupakan metode yang digunakan untuk menghitung nilai statistik kata berdasarkan jumlah kemunculannya dalam dokumen. TF (*Term Frequency*) menunjukkan banyaknya kemunculan kata pada suatu kalimat maupun dokumen. Sedangkan, DF (*Document Frequency*) menunjukkan jumlah kata yang muncul keseluruhan dokumen yang diselidiki [12]. TF-IDF akan menghitung bobot ulasan berdasarkan tiap kata yang sudah melalui tahap *preprocessing* sebelumnya kemudian mengalikan dengan idf.



Gambar 2 Alur proses TF-IDF

2. 3 Klasifikasi

Tahap ini diselesaikan melalui proses klasifikasi dengan memanfaatkan metode *K-Nearest Neighbor*. Tahapan klasifikasi KNN ditunjukkan oleh pada Gambar 3 berikut:



Gambar 3 Flowchart klasifikasi KNN

2. 4 Analisis Hasil Klasifikasi

Untuk mengukur hasil klasifikasi sentimen diperlukan *confusion matrix* yang akan menghasilkan nilai seperti akurasi, presisi, *recall*, dan juga *f1-score*.

2. 5 Visualisasi

Tahap visualisasi diselesaikan dengan melakukan ekstraksi terhadap seluruh informasi yang ada untuk mengetahui kata yang paling sering digunakan pada ulasan pengguna aplikasi PeduliLindungi dengan menggunakan *Wordcloud*.

3. HASIL DAN PEMBAHASAN

Pengumpulan data sebagai langkah awal penelitian dilakukan dengan teknik *scraping* dengan menggunakan *library python* pada ulasan pengguna PeduliLindungi di situs *Google Play Store*. Teknik ini dilakukan untuk mengambil sebanyak 7.500 ulasan. Gambar 4 menunjukkan hasil *scraping* yang telah diekstraksi:

```

1 content
2 Sangat membantu
3 Good
4 "Aplikasi sampah, mendaftar saja sulit masuk . Katanya sudah 3 kali daftar tapi tak bi
5 "Buat android 5.1.1 dong,malas bolak balik buka chrome terus di saat mau check in"
6 Setelah update malah tidak bisa cek lokasi dan ketersediaan vaksin terdekat. Mohon di
7 Serba guna
8 Parah ne aplikasi sertifikat vaksin w hilang g guna emnk gara2 pembaruan mulu bukanya
9 Sangat membantu
10 Sertifikat vaksin tidak muncul
11 "Ga bisa ngecek sertifikat vaksin, lokasi vaksin, Loading terus padahal jaringan stab:
12 aneh. Update malah makin jelek dari pada versi kemarin mending gaush update udah bias:
13 gak bisa didonload
14 Jos
15 Ok
16 keren
17 🍌🍌🍌🍌🍌
18 The bestie
19 Knapa gk bisa masuk padahal sudah punya akun? Ada apa ini?
20 Ok
21 Tni nentine
  
```

Gambar 4 Hasil *scraping* data

2. 1 preprocessing

Data yang diperoleh dari proses *scraping* tidak dapat digunakan secara langsung karena terdapat bentuk kalimat yang masih belum terstruktur sehingga masih perlu dilakukan *preprocessing* data. Beberapa hasil *preprocessing* data pada penelitian ini adalah sebagai berikut:

2. 1.1 Case Folding

Case folding pada tahap *preprocessing* data dilakukan dengan tujuan untuk mengubah huruf atau kata menjadi huruf kecil. Selain itu, pada proses ini juga akan menghilangkan tanda baca serta angka yang terdapat dalam kalimat pada ulasan. Hal ini bertujuan agar kata yang terdiri dari huruf kecil dan besar tidak dideteksi dan diartikan berbeda makna.

Tabel 1 Hasil *Case Folding*

No.	Input	Output
1	GAK BAKAL GUA UPGRADE !!! BOSEN, PERBAHARUI TERUS. GAK GENAH JUGA. YANG BALES JUGA BOOT KAYAKNYA. BOSEN SIH LU JAWABNY GITU MULU. UDAH MANA APLIKASI NGABISIN MEMORY. !!!	gak bakal gua upgrade bosen perbaharui terus gak genah juga yang bales juga boot kayaknya bosen sih lu jawabny gitu mulu udah mana aplikasi ngabisin memory
2	Dengan pedulilindungi semua perjalananku lancar berkat data Vaksinku dari 1-3 tersusun rapi d sini	dengan pedulilindungi semua perjalananku lancar berkat data vaksinku dari tersusun rapi d sini

2. 1.2 Spelling Normalization

Pada bagian *spelling normalization* dilakukan agar kata-kata singkatan atau kata yang tidak baku yang terdapat pada data kalimat ulasan dapat diperbaiki, misalnya kata “kyk” akan diubah menjadi “seperti,” lalu kata “d” diubah menjadi “di”. Kamus yang digunakan bersumber dari *repository* GitHub dengan menambahkan beberapa kata yang sebelumnya tidak ada pada *library* tersebut. Adapun contoh kalimat ulasan setelah *spelling normalization* dapat dilihat pada Tabel 2 berikut ini:

Tabel 2 Hasil *Spelling Normalization*

No.	Input	Output
1	aplikasi kyk sampah	aplikasi seperti sampah
2	dengan pedulilindungi semua perjalananku lancar berkat data vaksinku daritersusun rapi d sini	dengan pedulilindungi semua perjalananku lancar berkat data vaksinku dari tersusun rapi di sini

2. 1.3 Tokenizing

Proses *Tokenizing* adalah tahapan untuk memecah kalimat dalam dokumen *text*. Kalimat dipecah dengan cara memisahkan kata berdasarkan kata per-spasi. Token merujuk pada potongan kata yang telah berhasil dipisahkan. Pemisahan kata ini dilakukan dengan memberi tanda koma(,) pada setiap katanya. Adapun contoh hasil dari proses *tokenizing* ditunjukkan pada Tabel 3.

Tabel 3 Hasil *Tokenizing*

No.	Input	Output
1	aplikasi seperti sampah	['aplikasi', 'seperti', 'sampah']
2	dengan pedulilindungi semua	['dengan', 'pedulilindungi',

perjalananku lancar berkat data vaksinku dari tersusun rapi di sini	'semua', 'perjalananku', 'lancar', 'berkat', 'data', 'vaksinku', 'dari', 'tersusun', 'rapi', 'di', 'sini']
---	--

2. 1.3 Filtering

Proses *Filtering* adalah proses yang dilakukan untuk membuang kata yang tidak memiliki arti dan pengaruh terhadap analisis sentimen namun seringkali muncul dalam dokumen. Proses *filtering* disebut dengan *Stopword Removal*. Adapun proses penghilangan kata tersebut dilakukan dengan memanfaatkan *library* NLTK yang merupakan sebuah *library* yang terdiri dari kumpulan kata *stop word* berbahasa Indonesia. Contoh hasil proses stopword removal ditunjukkan pada Tabel 4.

Tabel 4 Tahap *Filtering*

No.	Input	Output
1	['aplikasi', ' seperti ', 'sampah']	['aplikasi', 'sampah']
2	[' dengan ', 'pedulilindungi', 'semua', 'perjalananku', 'lancar', 'berkat', 'data', 'vaksinku', ' dari ', 'tersusun', 'rapi', ' di ', 'sini']	['pedulilindungi', 'semua', 'perjalananku', 'lancar', 'berkat', 'data', 'vaksinku', 'tersusun', 'rapi', 'sini']

2. 2 TF-IDF

Setelah proses *filtering* ulasan dilakukan pelabelan ulasan menggunakan kamus *InsetLexicon*. Pelabelan dilakukan dengan mengategorikan ulasan ke dalam 3 kategori atau label yakni positif, netral, dan negatif. Ada 4 aturan yang berlaku pada proses *labeling* ini, yaitu:

1. Cari kalimat yang mengandung negasi.

Sebelum menjalankan proses ini, akan dibuat kamus negasi yaitu; “tidak” dan “bukan”. Jika ada kata negasi di depan suatu kata maka sentimennya akan menjadi penyangkalan dari sentimennya.

2. Cari kata yang termasuk dalam kamus *Insetlexicon*.

Memeriksa setiap kata, jika kata tersebut terdapat pada kamus *lexicon* maka sentimennya akan dihitung.

3. Jika kata tidak ditemukan, periksa *root word*.

Proses ini dilakukan dengan bantuan *library* Sastrawi untuk mengubah bentuk kata dari kata majemuk (kata yang memiliki imbuhan) menjadi kata dasar (*root word*). Jika *root word* terdapat pada kamus *Inset Lexion* maka sentimennya akan dihitung.

4. Menghitung *score*.

Proses perhitungan *score* sentimen yaitu dengan persamaan berikut:

$$Score = (\text{jumlah bobot kata positif}) + (\text{jumlah bobot kata negatif}) \quad (1)$$

Tabel 5 Contoh Menghitung Bobot Sentimen

Ulasan	Kata Positif (bobot)	Kata Negatif (bobot)	Score
sangat membantu	membantu (4)	-	4
sangat susah daftar	daftar (1)	susah (-3)	-4
tidak membantu sama sekali	sama (3)	tidak+membantu (-4) sekali (-1)	

Proses perhitungan bobot sentimen dapat dilihat pada Tabel 5. Apabila diakumulatikan kata-kata tersebut dapat dikategorikan menjadi sentimen positif, sentimen netral, dan sentimen negatif. Suatu kata dikategorikan sebagai sentimen negatif jika nilai skor < 0. Sedangkan kata akan dikategorikan sebagai sentimen positif jika nilai skor > 0. Jika nilai skor = 0 maka

dinyatakan kedalam sentimen netral. Contoh hasil pemberian kategori sentimen ditunjukkan oleh Tabel 6 untuk hasil *labeling lexicon*(positif) dan Tabel 7 untuk hasil *labeling lexicon*(netral).

Tabel 6 Hasil *Labeling Lexicon*(Positif)

No	Ulasan	Score	Label
1.	sangat membantu	4	positif
2.	Ok	4	positif
3.	bagus perlu ditingkatkan	5	positif
4.	Baik	3	positif
5.	bagus sekali aplikasinya masukan mohon ditambah merubah biodata pengguna mau buat qr scan tempat kerja bagaimana caranya	5	positif

Tabel 7 Hasil *Labeling Lexicon*(Netral)

No	Ulasan	Score	Label
1.	tambahin mode offline dong	0	netral
2.	bagaimana yah	0	netral
3.	update nya hampir hari	0	netral
4.	ntah lah	0	netral
5.	iya mayanlah	0	netral

Hasil akhir dari *labeling* yang menunjukkan jumlah ulasan untuk setiap label menggunakan *Inset lexicon* dapat dilihat pada Tabel 6. Dari 7.500 ulasan diperoleh sentimen kategori positif sebanyak 4.399 ulasan, sentimen dengan kategori negatif sebesar 1.959 ulasan, dan sentimen dengan kategori netral sebesar 1.142 ulasan.

Tabel 8 Hasil Pelabelan Ulasan Menggunakan *Inset Lexicon*

Positif	Netral	Negatif	Jumlah
4.399	1.142	1.959	7.500
58,66%	15,22%	26,12%	100%

Tahap selanjutnya adalah pembobotan TF-IDF, pada tahap ini akan dilakukan perhitungan bobot ulasan berdasarkan tiap kata yang sudah melalui tahap *preprocessing* sebelumnya kemudian mengalikan dengan idf. Contoh hasil pembobotan tf-idf ditunjukkan pada Gambar 5 berikut.

(0, 4290)	0.6527710572260347
(0, 2901)	0.7575552434298141
(1, 1668)	1.0
(2, 5024)	0.13333511705265003
(2, 4733)	0.326715451312348
(2, 4727)	0.17493609646271513
(2, 4275)	0.3420267841350138
(2, 2973)	0.417447436091977
(2, 2820)	0.4323850706755717
(2, 2141)	0.3507715024994921
(2, 2098)	0.2707198349874322
(2, 966)	0.25935275846404854
(2, 689)	0.15134560292409868
(2, 230)	0.24105617013199088
(2, 221)	0.14855063868136492
(3, 4987)	0.17864238706488125
(3, 2836)	0.17240738545955114
(3, 2733)	0.3906000895108179
(3, 1881)	0.2424095548734462
(3, 1338)	0.2577157102057354
(3, 888)	0.3906000895108179
(3, 880)	0.29840521426561767
(3, 790)	0.2008436029594875
(3, 779)	0.22124152968480815
(3, 735)	0.35349477508406857
:	:

Gambar 5 Hasil TF-IDF

2.3 Klasifikasi

Sebelum tahap klasifikasi algoritma, perlu dilakukan pembagian data. Dalam hal ini, data perlu dibagi ke dalam dua kelompok yaitu kelompok data latih (*training*) dan kelompok data uji (*testing*). Pengujian pada penelitian ini dilakukan menggunakan tiga perbandingan jumlah data latih dan data uji. Tabel 9 berikut menunjukkan perbandingan tersebut:

Tabel 9 *Splitting* Data

Jenis Perbandingan %	Data Latih	Data Uji
Perbandingan 70:30	5.250	2.250
Perbandingan 80:20	6.000	1.500
Perbandingan 90:10	6.750	750

Tahap pengklasifikasian dapat dilanjutkan setelah proses pembagian data. Proses klasifikasi ini akan memanfaatkan algoritma *K-Nearest Neighbors* dengan metode perhitungan jarak menggunakan *Euclidean Distance*, dengan menguji beberapa nilai K dari K=2, K=4, K=5, dan K=6 untuk melihat akurasi tertinggi yang dihasilkan dari pengujian beberapa nilai K tersebut. Berikut merupakan hasil perbandingan dari beberapa nilai K:

Tabel 10 Hasil Akurasi Algoritma

Nilai K	Akurasi Model		
	Perbandingan%		
	70:30	80:20	90:10
K2	60,31	60,26	60,66
K3	57,68	58,2	57,73
K4	57,42	58,2	57,86
K5	56,07	56,53	57
K6	55,55	56,66	56,13

Pada Tabel 10 dapat diketahui bahwa nilai K2 untuk klasifikasi menggunakan perbandingan 90:10 mendapatkan nilai akurasi tertinggi yakni 60,66% dengan 6.750 ulasan yang menjadi data latih dan 750 ulasan yang menjadi data uji.

Hasil pengujian menunjukkan bahwa nilai K memiliki pengaruh yang besar terhadap nilai akurasi. Namun secara umum, rata-rata nilai akurasi akan menurun seiring dengan penambahan nilai K. Dimana, semakin besar nilai K maka semakin banyak jumlah tetangga yang digunakan dalam mengklasifikasi data yang baru.

Metode *Euclidean distance* diimplementasikan untuk mengetahui nilai kedekatan diantara dua data dimana nilai yang semakin kecil menunjukkan jarak data yang semakin dekat. Nilai K yang kecil berarti bahwa tetangga tersebut memiliki kedekatan yang terbaik sehingga dapat pula diartikan bahwa kedua data tersebut memiliki karakteristik yang paling mirip. Misalkan nilai $K=2$, hal ini berarti terdapat dua data yang menjadi tetangga terdekat dan memiliki karakteristik data yang mirip dengan data yang diujikan sehingga dapat digolongkan dalam kelas yang sama dalam proses klasifikasi. Hal ini dapat membantu proses klasifikasi dapat dijalankan dengan baik. Sebaliknya, jika nilai K semakin besar maka semakin banyak pula tetangga yang diambil dalam proses klasifikasi meskipun data tersebut memiliki jarak yang cukup jauh terhadap data uji atau bahkan hampir tidak memiliki karakteristik yang sama. Penambahan *noise* dapat disebabkan oleh hal ini. Selain itu, dominasi kelas tertentu dikarenakan frekuensi data yang tidak seimbang untuk setiap kelas pada data latih menyebabkan data uji cenderung diklasifikasikan pada kelas yang mendominasi. Selain itu, proses pelabelan data dengan menggunakan kamus *lexicon* juga berdampak pada hasil yang diperoleh dimana masih terdapat banyak kata-kata yang tidak termuat dalam kamus tersebut sehingga hasil pelabelan data belum cukup optimal. Hasil presisi, *recall*, dan *f-1 score* dapat dilihat pada Tabel 11-13.

Tabel 11 Hasil Presisi Algoritma

Presisi Model			
Nilai K	Perbandingan%		
	70:30	80:20	90:10
K2	63,92	64,64	64,35
K3	67,26	67,98	69,94
K4	68,51	69,50	67,53
K5	68,83	68,07	66,79
K6	69,57	70,03	67,33

Tabel 12 Hasil *Recall* Algoritma

<i>Recall</i> Model			
Nilai K	Perbandingan%		
	70:30	80:20	90:10
K2	61,64	61,69	62,79
K3	60,49	60,57	60,45
K4	60,51	60,90	61,06
K5	59,60	60,18	60,01
K6	59,32	60,08	59,97

Tabel 13 Hasil *F1-Score* Algoritma

<i>F1-Score</i> Model			
Nilai K	Perbandingan %		
	70:30	80:20	90:10
K2	54,59	54,95	54,44
K3	52,05	52,73	51,21
K4	51,84	52,89	51,50
K5	50,50	51,64	49,81
K6	49,98	51,30	49,50

2. 4 Analisis Hasil

Perhitungan akurasi dengan *Confusion matrix* dapat dilakukan dengan mengetahui jumlah data uji yang diklasifikasi dengan benar dan salah dalam pengklasifikasiannya.

Tabel 14 Hasil *Confusion Matrix*

Predicted Class	Actual Class		
	Negatif	Netral	Positif
Negatif	65	112	17
Netral	4	86	7
Positif	18	137	304

Confusion matrix pada Tabel 14 berupa matrik dengan ukuran 3x3 yang menunjukkan setiap kelas klasifikasi yaitu positif, netral, dan negatif pada pengujian yang dilakukan melalui perbandingan data latih dan data uji sebesar 90:10 dan menggunakan $K=2$. Berdasarkan *confusion matrix* tersebut dapat disimpulkan bahwa model mengklasifikasikan secara benar sebanyak 65 data sebagai negatif, 86 data sebagai netral, dan 304 data sebagai data positif. Terdapat kesalahan prediksi pada model, dimana sebesar 17 data diklasifikasikan ke dalam kelas positif padahal seharusnya negatif, serta memprediksi 18 data ke dalam kelas negatif yang seharusnya positif.

2. 5 Analisis Hasil

Hasil sentimen dapat diperoleh setelah proses pembersihan data dan klasifikasi. Selanjutnya, visualisasi data dilakukan dengan menggunakan *Wordcloud*. Tampilan *wordcloud* sebagaimana Gambar 5 terdiri dari kata yang paling sering muncul dalam suatu data penelitian yang digunakan.



Gambar 5 *Wordcloud*

4. KESIMPULAN

Analisis sentimen masyarakat pada ulasan aplikasi PeduliLindungi di *Google Play Store* berdasarkan dari 7.500 data ulasan, menghasilkan opini masyarakat dengan sentimen kategori positif sebesar 58,66%, sentimen dengan kategori negatif sebesar 26,12%, dan sentimen dengan kategori netral sebesar 15,22%. Hal ini menunjukkan bahwa masyarakat percaya dengan aplikasi PeduliLindungi dan menilai bahwa aplikasi tersebut layak untuk di-*install*. Selain itu, melalui pembagian data latih dan data uji sebesar 90:10, penggunaan rumus *Euclidean Distance* serta nilai $K=2$ dalam proses klasifikasi *K-Nearest Neighbors*, diperoleh hasil akurasi sebesar 60,66%, dengan nilai *presisi* 64%, *recall* 62,79%, *F1-score* 54,44%.

REFERENSI

- [1] N. Google, "Peta Kasus Covid-19," *Google News*, 2022.

- [2] G. S. Putri, "WHO Resmi Sebut Virus Corona Covid-19 sebagai Pandemi Global, Kompas.com," 2020. <https://www.kompas.com/sains/read/2020/03/12/083129823/who-resmi-sebut-virus-corona-covid-19-sebagai-pandemi-global>
- [3] I. Syambudi, "Kemenkes: Vaksinasi untuk Umum Mulai Juli 2021, tirto.id," 2021. <https://tirto.id/kemenkes-vaksinasi-untuk-masyarakat-umum-mulai-juli-2021-ggMc> (accessed Feb. 27, 2022).
- [4] PeduliLindungi, "Apa Itu PeduliLindungi, PeduliLindungi.id," 2022. <https://www.pedulilindungi.id/> (accessed Feb. 02, 2022).
- [5] F. Gunawan, M. A. Fauzi, and P. P. Adikara, "Analisis Sentimen Pada Ulasan Aplikasi Mobile Menggunakan Naive Bayes dan Normalisasi Kata Berbasis Levenshtein Distance (Studi Kasus Aplikasi BCA Mobile)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 10, pp. 1082–1088, 2017, doi: 10.29080/systemic.v3i2.234.
- [6] F. A. Hirzani, W. Maharani, and M. A. Bijaksana, "Analisis Sentimen Review Produk Kosmetik menggunakan Pendekatan Berbasis Kamus," in *e-Proceeding of Engineering*, 2015, pp. 5891–5898.
- [7] R. Prasetya and A. Ridwan, "Data Mining Application on Weather Prediction Using Classification Tree, Naïve Bayes and K-Nearest Neighbor Algorithm With Model Testing of Supervised Learning Probabilistic Brier Score, Confusion Matrix and ROC," *Jaict J. Appl. Commun. Inf. Technol.*, vol. 4, no. 2, pp. 25–33, 2019, doi: 10.32497/jaict.v4i2.1690.
- [8] A. Mustopa, Hermanto, A. Thahir, E. B. Pratama, A. Hendini, and D. Risdiansyah, "Analysis of User Reviews for the PeduliLindungi Application on Google Play Using the Support Vector Machine and Naive Bayes Algorithm Based on Particle Swarm Optimization," in *2020 Fifth International Conference on Informatics and Computing (ICIC)*, 2020.
- [9] E. Indrayuni, A. Nurhadi, and D. A. Kristiyanti, "Implementasi Algoritma Naive Bayes, Support Vector Machine, dan K-Nearest Neighbors untuk Analisa Sentimen Aplikasi Halodoc," *Fakt. Exacta*, vol. 14, no. 2, pp. 64–71, 2021, doi: 10.30998/faktorexacta.v14i2.9697.
- [10] R. P. Fitrianti, A. Kurniawati, and D. Agusten, "Implementasi Algoritma K-Nearest Neighbor Terhadap Analisis Sentimen Review Restoran Dengan Teks Bahasa Indonesia," in *Seminar Nasional Aplikasi Teknologi Informasi (SNATi) 2019*, 2019, pp. A27–A32.
- [11] S. Budi, "Text Mining Untuk Analisis Sentimen Review Film Menggunakan Algoritma K-Means," *Techno.Com*, vol. 16, no. 1, pp. 1–8, 2017, doi: 10.33633/tc.v16i1.1263.
- [12] R. Feldman and J. Sanger, *The Text Mining Handbook*. New York: Cambridge University Press, 2006. doi: 10.1017/cbo9780511546914.
- [13] F. Koto and G. Y. Rahmaningtyas, "Inset lexicon: Evaluation of a Word List for Indonesian Sentiment Analysis in Microblogs," *Proc. 2017 Int. Conf. Asian Lang. Process. IALP 2017*, vol. 2017, pp. 391–394, 2017, doi: 10.1109/IALP.2017.8300625.