

## Klasifikasi Sentimen Ulasan Pengguna Aplikasi PeduliLindungi di *Google Play* Menggunakan Algoritma *Support Vector Machine* dengan Seleksi Fitur *Chi-Square*

Cindy Chairunnisa<sup>1</sup>, Iin Ernawati<sup>2</sup> dan Mayanda Mega Santoni<sup>3</sup>

Program Studi Informatika / Fakultas Ilmu Komputer

Universitas Pembangunan Nasional Veteran Jakarta

Jl. RS. Fatmawati, Pondok Labu, Jakarta Selatan, DKI Jakarta, 12450, Indonesia

cindyc@upnvj.ac.id, iinernawati@upnvj.ac.id, megasantoni@upnvj.ac.id<sup>3</sup>

**Abstrak.** Upaya pemerintah untuk mengurangi penyebaran wabah virus corona yang semakin meluas hampir di setiap negara di dunia termasuk di Indonesia telah banyak dilakukan. Salah satu upaya yang dilakukan oleh pemerintah dengan memanfaatkan teknologi yang ada pada saat ini adalah membuat sebuah aplikasi bernama PeduliLindungi. Aplikasi ini bertujuan untuk melakukan *tracing* dan monitoring lokasi penyebaran virus corona sehingga dapat menurunkan kasus corona di Indonesia. Banyak ulasan yang diberikan oleh masyarakat terhadap aplikasi ini baik yang berupa kritik maupun kepuasan. Namun, untuk mengetahui seluruh ulasan yang diberikan tidak mudah. Oleh sebab itu, penelitian dilakukan dengan tujuan untuk mengetahui hasil sentimen masyarakat terhadap aplikasi PeduliLindungi. Analisis sentimen yang dilakukan dengan mengklasifikasikan ulasan menjadi ulasan positif dan ulasan negatif menggunakan algoritma *Support Vector Machine* dengan seleksi fitur *chi-square*. Pengumpulan data ulasan dilakukan dengan melakukan *scrapping* di *google play* dengan menggunakan bahasa pemrograman *Python*. Hasil dari klasifikasi sentimen terhadap aplikasi PeduliLindungi menghasilkan performa yang baik dan menghasilkan nilai akurasi sebesar 93%, *recall* sebesar 86%, *precision* sebesar 98%, *specificity* sebesar 98% dan *f1-score* sebesar 92%.

**Kata Kunci:** Analisis Sentimen, Seleksi Fitur, Chi-Square, Support Vector Machine, PeduliLindungi

### 1 Pendahuluan

Pada awal tahun 2020 wabah virus corona melanda hampir semua negara di dunia termasuk Indonesia. Penyebaran dan penularan virus corona yang sangat cepat menyebabkan kepanikan dan ketakutan yang luar biasa di kalangan penduduk di dunia [13]. Beberapa upaya telah dilakukan oleh Pemerintah diantaranya adalah menetapkan kebijakan protokol kesehatan, menutup tempat umum serta pembatasan sosial. Namun kebijakan tersebut dirasa kurang efektif sehingga pemerintah menerapkan *contact tracing*. *Contact tracing* yang diterapkan pada awalnya bersifat manual *tracing* dimana ketika orang yang terinfeksi virus memberikan informasi dengan siapa dan dimana ia melakukan kontak fisik dengan orang lain. Namun, penerapan manual *tracing* ini membuang banyak waktu dan sumber daya [2].

Dalam perkembangan teknologi pada saat ini, teknologi seperti smartphone dapat dimanfaatkan untuk membantu pemerintah melakukan *tracing*. *Tracing* dilakukan dengan menginformasikan lokasi seseorang yang terinfeksi virus corona secara otomatis agar lebih cepat dan akurat. Oleh sebab itu, pemerintah Indonesia menciptakan sebuah aplikasi bernama PeduliLindungi. Aplikasi ini diharapkan mampu melindungi masyarakat Indonesia dengan saling berbagi lokasi saat bepergian agar *contact tracing* dengan orang yang terinfeksi virus dapat terus dilakukan sehingga dapat memperlambat dan menurunkan penyebaran virus corona [11]. Namun masih terdapat beberapa ulasan negatif dalam aplikasi tersebut diantaranya kesalahan informasi lokasi, aplikasi yang terlalu banyak menghabiskan daya, *log-out* otomatis sehingga pengguna harus memasukkan kode OTP berulang kali dan membutuhkan kecepatan internet tinggi. Selain itu masih terdapat kekhawatiran para pengguna aplikasi ini terkait efektivitas dan privasi dikarenakan aplikasi tersebut memerlukan pelacakan lokasi setiap individu. Berdasarkan permasalahan tersebut data ulasan pengguna aplikasi PeduliLindungi pada Google Play dapat dimanfaatkan dan diolah dengan melakukan pengklasifikasian terhadap ulasan berupa analisis sentimen.

Penelitian terkait analisis sentimen telah dilakukan sebelumnya menggunakan metode *Naïve Bayes* dan *Support Vector Machine* (SVM) untuk memprediksi label sentimen ulasan pengguna aplikasi *zoom meetings* di *google play*. Hasil akurasi dan AUC yang didapatkan dengan menggunakan algoritma *Support Vector Machine* (SVM)

adalah sebesar 81.22% dan 0.886, sedangkan menggunakan *Naïve Bayes* adalah sebesar 74.37% dan 0.659. Sehingga dapat diketahui bahwa tingkat akurasi yang dihasilkan menggunakan algoritma *Support Vector Machine* (SVM) lebih tinggi 6,85% dibandingkan algoritma *Naïve Bayes* (NB) pada 1.007 record dataset [4]. Penelitian lain terhadap *user feedback* pada aplikasi dengan melakukan perbandingan metode *Support Vector Machine* dengan seleksi fitur *chi-square* dan metode SVM tanpa seleksi fitur *chi-square*. Hasil penelitian ini menghasilkan nilai akurasi sebesar 77% dengan menggunakan seleksi fitur dan 69% tanpa menggunakan seleksi fitur pada model SVM [8].

Oleh sebab itu, penelitian ini akan menerapkan metode *Support Vector Machine* (SVM) yang dinilai mampu melakukan klasifikasi analisis sentimen terhadap data ulasan dan penggunaan seleksi fitur *chi-square* untuk menghilangkan fitur yang tidak relevan. Tujuan dilakukan penelitian ini adalah mengklasifikasikan sentimen masyarakat terhadap aplikasi PeduliLindungi menjadi dua kategori kelas yaitu positif dan negatif. Dengan luaran akhir penelitian yang berupa model klasifikasi dan hasil analisis sentimen yang dapat dijadikan bahan pertimbangan untuk melakukan evaluasi terhadap aplikasi PeduliLindungi.

## 2 Landasan Teori

### 2.1 Analisis Sentimen

Analisis sentimen atau dapat disebut juga dengan *opinion mining* merupakan proses mengumpulkan, mengolah dan menganalisis ulasan pengguna terhadap suatu produk atau topik tertentu yang dibuat dalam suatu blog post, *comments*, *reviews* atau *tweets* [15]. Ulasan yang diberikan ini tidak hanya bernilai positif namun juga dapat bernilai negatif. Bahkan suatu keadaan tertentu ulasan yang bernilai positif dapat dikatakan bernilai negatif hal ini disebabkan karena setiap orang tidak dapat memberikan penilaian dengan cara yang sama.

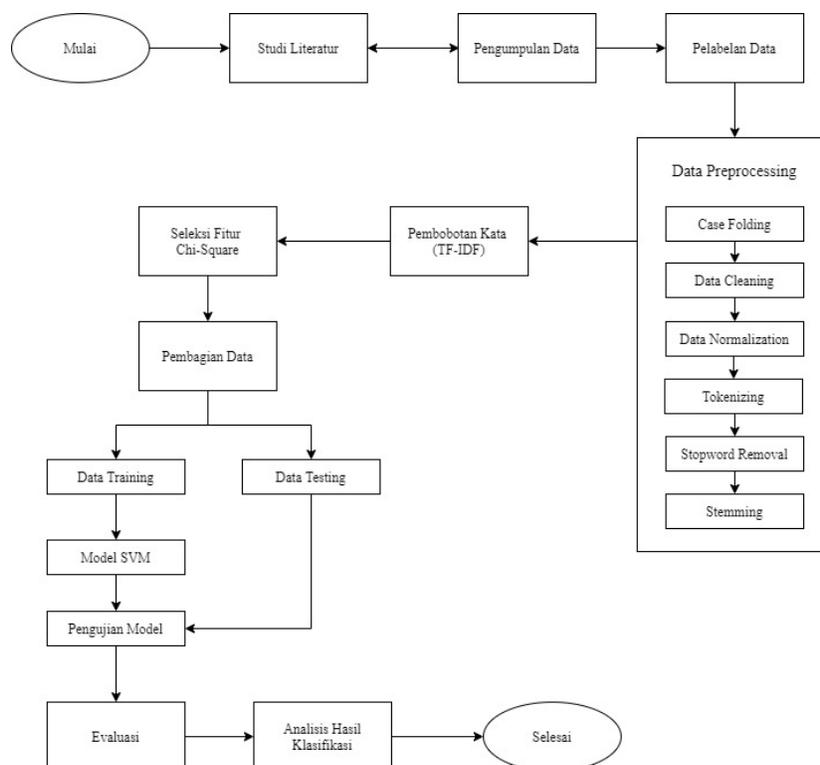
### 2.2 Support Vector Machine (SVM)

*Support Vector Machine* (SVM) pertama kali diperkenalkan oleh Vapnik pada awal tahun 1990an dimana merupakan sebuah teknik algoritma yang efektif dan menjanjikan untuk diterapkan dalam data mining. Algoritma SVM hingga saat ini sudah banyak diterapkan dalam berbagai bidang seperti pengkategorian teks, pengenalan suara, analisa gambar, peramalan waktu, keamanan informasi dan sebagainya. SVM merupakan algoritma yang dapat mengatasi beberapa kesulitan seperti *overfitting* dan *curse of dimensionality*. Sehingga algoritma ini menjadi lebih populer dan telah banyak diterapkan [3]. *Support Vector Machine* (SVM) merupakan salah satu contoh penerapan algoritma yang dapat digunakan untuk permasalahan klasifikasi, analisis regresi dan prediksi dengan menggunakan model secara *linear* dimana data terpisah menjadi dua buah atau lebih kelas yang terpisah berdasarkan nilai *hyperplane* optimal diantara masing-masing kelas. *Hyperplane* yang optimal merupakan nilai maksimum jarak antar kelas dalam suatu model [1].

### 2.3 Chi Square

*Chi-square* merupakan salah satu metode seleksi fitur *supervised* yang mampu menghilangkan fitur tanpa mengurangi tingkat akurasi yang dihasilkan [12]. *Chi-square* bertujuan untuk melihat ketergantungan suatu fitur terhadap kategori atau label kelasnya [9]. Algoritma dari seleksi fitur *chi square* adalah mengurangi data yang besar menjadi lebih kecil sehingga proses yang dilakukan menjadi lebih cepat [14].

### 3 Metode Penelitian



Gambar. 1. Bagan alur

#### 3.1 Pengumpulan Data

Pengumpulan data dilakukan dengan melakukan scraping data ulasan pengguna aplikasi PeduliLindungi di *Google Play*. Proses *scraping* data dilakukan dengan memanfaatkan *library python* yang telah tersedia yaitu *google play scraper* sebagai akses untuk mengambil data. Data ulasan kemudian disimpan dalam bentuk format file CSV.

#### 3.2 Pelabelan Data

Pelabelan data tidak dilakukan berdasarkan rating pengguna melainkan berdasarkan ulasan yang diberikan. Rating pengguna memberikan kemudahan untuk menentukan sebuah ulasan bernilai positif atau negatif, namun masih terdapat beberapa rating yang tidak memberikan ulasan yang sesuai dengan rating yang diberikan sehingga dilakukan berdasarkan data ulasan pengguna. Pelabelan data dilakukan secara manual dengan yang akan dilakukan oleh tiga anator dimana setiap anator akan memberikan penilaiannya dan mengkategorikan ulasan ke dalam 2 label kelas yaitu label positif ataupun negatif. Penentuan akhir akan dilakukan proses *vote* untuk mendapatkan suatu kesepakatan atau label hasil setiap data ulasan. Hasil pelabelan pada seluruh data yang telah disepakati diukur dengan menggunakan perhitungan fleiss kappa. Berikut merupakan rumus dasar untuk mencari nilai *kappa* [16].

$$Kappa(k) = \frac{P_{(A)} - P_e}{1 - P_e} \quad (1)$$

Keterangan :

K = Koefisien Kappa

P(A) = Proporsi banyaknya penilaian yang sama

P(E) = Nilai persetujuan

### 3.3 Data Preprocessing

Data *preprocessing* merupakan tahapan dilakukannya pembersihan dan pemilihan data untuk mendapatkan data yang memiliki kualitas dan bersih sehingga siap untuk digunakan. Data *preprocessing* dilakukan karena data yang akan digunakan masih mengandung *noise* dan masih kotor sehingga perlu dilakukan praproses sebelum dilakukan pengolahan data [7].

#### 3.3.1 Case Folding

*Case folding* merupakan tahapan proses mengubah semua huruf dalam data menjadi bentuk yang sama dan biasanya huruf tersebut diubah menjadi huruf kecil [7].

#### 3.3.2 Data Cleaning

Data *Cleaning* merupakan proses membersihkan karakter atau kata yang kurang memiliki pengaruh dalam proses *text mining*. Karakter yang dihapus atau dihilangkan berupa *punctuation* atau tanda baca, simbol-simbol, *emoticon*, *link*, karakter, angka dan lain-lain [11].

#### 3.3.3 Data Normalization

Normalisasi teks merupakan proses melakukan perbaikan terhadap kata-kata yang masih memiliki ejaan yang salah atau merupakan singkatan dan bentuk kata yang lain. Proses ini dilakukan agar perhitungan dimensi kata tidak meluas meskipun memiliki arti yang sama dikarenakan masih terdapat kata yang salah eja atau singkatan [5].

#### 3.3.4 Data Stemming

*Stemming* merupakan proses mengubah setiap kata yang memiliki imbuhan menjadi kata dasar. Proses *stemming* bertujuan agar kata yang memiliki makna yang sama atau mirip akan dikelompokkan menjadi satu makna sesuai makna kata dasarnya [6].

#### 3.3.5 Stopwords Removal

*Stopword Removal* atau *filtering* merupakan proses menghapus *stopwords* dari daftar kata (*token*). *Stopword* merupakan kata – kata yang biasanya tidak relevan dengan teks sehingga perlu dihapus agar proses *text mining* menjadi lebih efisien. Kata – kata seperti ‘dan’, ‘atau’, ‘yang’, ‘dari’ dan ‘ke’ merupakan contoh kata yang termasuk *stopwords* yang dimana sering muncul dalam sebuah teks [6].

#### 3.3.6 Tokenizing

*Tokenizing* merupakan proses membagi sekumpulan kata atau teks dalam sebuah dokumen menjadi sebuah kata (*token*) yang dipisahkan oleh spasi dan tanda baca. Pada proses *tokenizing* sebuah teks diberikan sebagai *input* kemudian *output* yang dihasilkan berupa daftar kata (*token*) [6].

### 3.4 Pembobotan Kata (TF-IDF)

Pembobotan kata (*term weighting*) merupakan proses perhitungan dan penetapan bobot setiap kata sebagai penentuan seberapa pentingnya suatu kata terhadap data teks. TF-IDF merupakan salah satu metode populer yang digunakan untuk menghitung bobot sebuah kata. Berikut merupakan persamaan rumus TF-IDF [6].

$$w_{ij} = tf_{ij} \times idf_{ij} \quad (2)$$

Keterangan :

$tf_{ij}$  : jumlah kemunculan kata dalam dokumen

$idf_{ij}$  : jumlah dokumen yang mengandung kata

$w_{ij}$  : bobot kata dalam dokumen

### 3.5 Seleksi Fitur

Seleksi fitur merupakan proses untuk memilih fitur yang relevan yang akan digunakan untuk membangun model agar mendapatkan fitur yang lebih optimal. *Chi Square* merupakan salah satu metode seleksi fitur *supervised* yang mampu menghilangkan fitur tanpa mengurangi tingkat akurasi yang dihasilkan [11]. *Chi Square* akan melihat ketergantungan suatu fitur terhadap kategori atau label kelasnya [9]. Berikut merupakan persamaan rumus untuk mendapatkan uji *chi square*.

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} \quad (3)$$

Keterangan :

$O_i$  = Hasil yang diamati

$E_i$  = Hasil yang diharapkan

### 3.6 Klasifikasi *Support Vector Machine* (SVM)

*Support Vector Machine* (SVM) merupakan salah satu contoh penerapan algoritma yang dapat digunakan untuk permasalahan klasifikasi, analisis regresi dan prediksi dengan menggunakan model secara *linear* dimana data terpisah menjadi dua buah atau lebih kelas yang terpisah berdasarkan nilai *hyperplane* optimal diantara masing-masing kelas. *Hyperplane* yang optimal merupakan nilai maksimum jarak antar kelas dalam suatu model. Pada dasarnya SVM merupakan algoritma yang digunakan untuk mengklasifikasi data yang *linear* namun pada kasus *non-linear separable* dataset pada algoritma SVM dapat dilakukan dengan menggunakan fungsi kernel. Pada algoritma SVM terdapat empat jenis kernel yang dapat digunakan untuk melakukan klasifikasi diantaranya yaitu *linear*, *polynomial*, *sigmoid* dan *rbf* [1].

### 3.7 Evaluasi

Proses evaluasi dilakukan untuk mengevaluasi performa model klasifikasi yang telah dibentuk. Metode evaluasi pada penelitian ini menggunakan metode *Confusion Matrix* dengan menghitung nilai akurasi, *recall*, *precision*, *f-1 score* dan *specificity*.

### 3.8 Analisis Hasil Klasifikasi

Analisis hasil klasifikasi dilakukan berdasarkan hasil klasifikasi sentimen yang telah dilakukan dengan melakukan visualisasi dalam bentuk wordcloud agar lebih mudah dipahami dan mendapatkan informasi.

## 4 Hasil dan Pembahasan

### 4.1 Pengumpulan Data

Data yang digunakan adalah data pada situs *Google Play* terkait ulasan pengguna aplikasi *PeduliLindungi*. Data ulasan yang diambil sebanyak 1000 data yang paling relevan dalam rentang waktu dari bulan November 2021 hingga April 2022. Hasil dari pengambilan data ulasan dapat dilihat pada Gambar 2.

	userName	content	score	at
0	Christian S	Mungkin manfaat bagi regulator cukup memadai, ...	3	2022-02-01 05:04:13
1	kurniadi frans	Untuk pengisian tanggalnya tidak efisien, teru...	3	2022-01-12 03:14:49
2	Hassan Mahmud	Selain untuk memudahkan pemaksaan vaksin, enta...	1	2022-02-03 21:50:45
3	Muhammad Arif Bagus Pratama	Apk ini kalau menurut saya sih bagus ya,cuma y...	4	2022-01-13 13:03:55
4	mila permatasari	Aplikasi parah setelah di update gak bisa di g...	1	2022-02-28 02:36:19

Gambar. 2. Hasil *Scrapping* Data Ulasan

### 4.2 Pelabelan Data

Proses pelabelan pada 1000 data ulasan aplikasi *PeduliLindungi* dihasilkan sebanyak 446 data ulasan positif dan 554 ulasan negatif. Hasil pelabelan data dihasilkan berdasarkan voting mayoritas dari ketiga anator. Setelah didapatkan hasil pelabelan data kemudian dilakukan evaluasi kesepakatan dengan menggunakan nilai *kappa*. Pada penelitian ini dihasilkan nilai *kappa* sebesar 0.991 sehingga dapat dikatakan bahwa kekuatan kesepakatan dari hasil pelabelan data sangat baik dan data dapat digunakan.

### 4.3 Data Preprocessing

Tahapan pra proses data pada penelitian dilakukan dengan melalui beberapa tahapan yaitu *case folding*, *data cleaning*, *data normalization*, *stemming*, *stopword removal* dan *tokenizing*.

#### 4.3.1 Case Folding

Contoh tahap *case folding* dilakukan dengan mengubah kata “Aplikasinya” menjadi “aplikasinya”, “Kalender” menjadi “kalender” menjadi huruf kecil ditunjukkan pada Tabel 1.

Tabel 1. *Case Folding*

Sebelum	Sesudah
Aplikasinya sangat membantu dan memudahkan jika berpergian ke area tertentu..cuma saran untuk memasukan tgl lahir dan tahun kalau bisa jangan menggunakan sistim Kalender	aplikasinya sangat membantu dan memudahkan jika berpergian ke area tertentu..cuma saran untuk memasukan tgl lahir dan tahun kalau bisa jangan menggunakan sistim kalender

#### 4.3.2 Data Cleaning

Contoh tahap data *cleaning* dilakukan pada data ulasan dengan menghapus tanda baca yaitu “.” ditunjukkan pada Tabel 2.

**Tabel 2.** Data *Cleaning*

Sebelum	Sesudah
aplikasinya sangat membantu dan memudahkan jika berpergian ke area tertentu..cuma saran untuk memasukan tgl lahir dan tahun kalau bisa jangan menggunakan sistim kalender	aplikasinya sangat membantu dan memudahkan jika berpergian ke area tertentu cuma saran untuk memasukan tgl lahir dan tahun kalau bisa jangan menggunakan sistim kalender

### 4.3.3 Data *Normalization*

Contoh tahap data *normalization* dilakukan pada data ulasan dengan mengubah kata “berpergian” menjadi “bepergian”, “sistim” menjadi “sistem”, “memasukan” menjadi “memasukkan”, “tgl” menjadi “tanggal” yang ditunjukkan pada Tabel 3.

**Tabel 3.** Data *Normalization*

Sebelum	Sesudah
aplikasinya sangat membantu dan memudahkan jika berpergian ke area tertentu cuma saran untuk memasukan tgl lahir dan tahun kalau bisa jangan menggunakan sistim kalender	aplikasinya sangat membantu dan memudahkan jika bepergian ke area tertentu cuma saran untuk memasukkan tanggal lahir dan tahun kalau bisa jangan menggunakan sistem kalender

### 4.3.4 Data *Stemming*

Contoh tahap data *normalization* dilakukan pada data ulasan dengan mengubah kata “aplikasinya” menjadi “aplikasi”, “membantu” menjadi “bantu”, “memudahkan” menjadi “mudah”, “bepergian” menjadi “pergi”, “tertentu” menjadi “tentu”, “memasukkan” menjadi “masuk”, “menggunakan” menjadi “guna” yang ditunjukkan pada Tabel 4.

**Tabel 4.** Data *Stemming*

Sebelum	Sesudah
aplikasinya sangat membantu dan memudahkan jika bepergian ke area tertentu cuma saran untuk memasukkan tanggal lahir dan tahun kalau bisa jangan menggunakan sistem kalender	aplikasi sangat bantu dan mudah jika pergi ke area tentu cuma saran untuk masuk tanggal lahir dan tahun kalau bisa jangan guna sistem kalender

### 4.3.5 *Stopwords Removal*

Contoh tahap *stopwords removal* dilakukan pada data ulasan dengan melakukan penghapusan kata yang terdapat dalam kamus *stopwords*. Hasil kata yang telah dihapus dari kedua data tersebut yaitu “sangat”, “dan”, “jika”, “ke”, “tentu”, “cuma”, “untuk”, “tahun”, “kalau”, “bisa”, “jangan”, “guna”, “yang”, “dua”, “tidak”, “sampai” “sekarang”, “padahal”, “sejak”, “bulan”, “lalu” “tetapi”, “belum” yang ditunjukkan pada Tabel 5.

**Tabel 5.** *Stopwords Removal*

Sebelum	Sesudah
aplikasi sangat bantu dan mudah jika pergi ke area tentu cuma saran untuk masuk tanggal lahir dan tahun kalau bisa jangan guna sistem kalender	aplikasi bantu mudah pergi area saran masuk tanggal lahir sistem kalender

#### 4.3.6 Tokenizing

Contoh tahap *tokenizing* menjadikan setiap kata menjadi terpisah menjadi satuan kata atau *token* dan menjadi sebuah *list* kata yang ditunjukkan pada Tabel 6.

Tabel 6. *Tokenizing*

Sebelum	Sesudah
aplikasi bantu mudah pergi area saran masuk tanggal lahir sistem kalender	[aplikasi, bantu, mudah, pergi, area, saran, masuk, tanggal, lahir, sistem, kalender]

#### 4.4 Pembobotan Kata (TF-IDF)

Pembobotan kata dilakukan terhadap 934 kata dari jumlah seluruh dokumen sebanyak 1000 dokumen data ulasan. Perhitungan TF-IDF dilakukan berdasarkan persamaan (1) dengan memanfaatkan *library python* sehingga hasil yang dikeluarkan dari proses pembobotan kata adalah seluruh kata telah memiliki nilai atau bobot agar dapat digunakan pada proses selanjutnya.

#### 4.5 Seleksi *Fitur Chi Square*

Proses seleksi fitur *chi square* dilakukan dengan menggunakan rumus pada persamaan (2). Perhitungan *chi square* dilakukan dengan menghitung nilai *Observed* dan *Expected* terlebih dahulu. Nilai *Observed* didapatkan dengan cara menjumlahkan nilai TF-IDF pada kelas yang sama, sedangkan nilai *Expected* didapatkan dengan cara melakukan perkalian probabilitas kemunculan suatu kelas dengan jumlah total TF-IDFnya.

Proses seleksi fitur pada penelitian ini dilakukan dengan melakukan pengujian nilai *chi square* yang telah didapatkan terhadap nilai kritis. Pengujian seleksi fitur ini akan menggunakan beberapa nilai taraf nyata  $\alpha$  0.950, 0.900, 0.100, 0.050, 0.025, 0.010 dan 0.005 dengan nilai kritis yaitu 0.004, 0.016, 2.706, 3.841, 5.024, 6.635, 7.879. Berikut merupakan jumlah hasil seleksi fitur ditunjukkan pada Tabel 7.

Tabel 7. Hasil Seleksi Fitur

Nilai Taraf Nyata $\alpha$	0.950	0.900	0.100	0.050	0.025	0.010	0.005
Nilai Kritis	0.004	0.016	2.706	3.841	5.024	6.635	7.879
Jumlah Fitur Awal	934	934	934	934	934	934	934
Jumlah Fitur Terseleksi	17	31	853	887	896	904	911
Jumlah Fitur Akhir	917	903	81	47	38	30	23

Pada Tabel 7. menunjukkan bahwa nilai kritis yang digunakan mempengaruhi banyak sedikitnya jumlah fitur yang akan digunakan. Nilai kritis semakin besar maka semakin banyak jumlah fitur yang terseleksi. Pemilihan fitur yang akan digunakan berdasarkan nilai kritis. Apabila hasil perhitungan nilai *chi square* lebih besar daripada nilai kritis maka fitur tersebut akan digunakan.

#### 4.6 Klasifikasi *Support Vector Machine*

Proses klasifikasi yang dilakukan dengan membagi data menjadi data latih dan data uji dengan menggunakan perbandingan yang umum dilakukan dalam melakukan klasifikasi yaitu sebesar 80% : 20%. Pada penelitian ini akan membandingkan tiga kernel yang ada yaitu *linear*, *polynomial* dan *rbf*. Untuk uji parameter yang akan

digunakan adalah *parameter cost* ( $C$ ) = [0.1, 1, 10, 20, 30, 40, 50 100] serta *parameter gamma* ( $\gamma$ ) = [0.01, 0.1, 1] dengan menggunakan metode *Grid Search CV*. Pada *Grid Search CV* akan menghasilkan nilai skor terbaik atau *grid score* sebagai penentu parameter terbaik yang akan digunakan untuk melakukan klasifikasi. Berikut merupakan hasil *Grid Search CV* dengan parameter terbaik ditunjukkan pada Tabel 8.

**Tabel 8.** Hasil *Grid Search CV* + Seleksi Fitur

Nilai Taraf Nyata	Nilai Kritis	Jumlah Fitur	Paramater			Grid Score	Akurasi
			Kernel	Cost	Gamma		
0.950	0.004	917	rbf	10	1	0.9125	0.93
0.900	0.016	903	rbf	1	1	0.9075	0.93
0.100	2.706	81	rbf	30	0.1	0.9012	0.92
0.050	3.841	47	rbf	100	0,01	0.8912	0.90
0.025	5.024	38	rbf	1	1	0.8937	0.90
0.010	6.635	30	linear	40	0.01	0.8962	0.90
0.005	7.879	23	rbf	30	0.1	0.8825	0.90

### 4.3 Evaluasi

Evaluasi dilakukan dengan menggunakan *confusion matrix* terhadap 200 data uji. Dihasilkan tabel *confusion matrix* sebagai berikut ditunjukkan pada Tabel 9.

**Tabel 9.** *Confusion Matrix*

True Label	Predicted Label	
	0	1
0	104	2
1	13	81

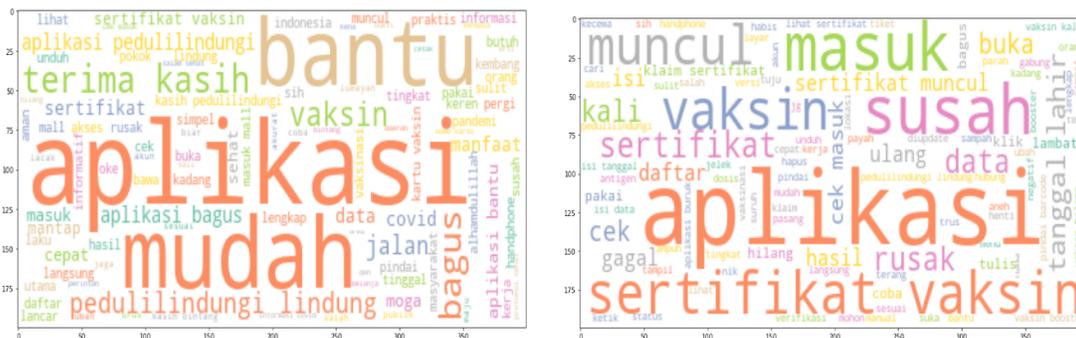
Dari hasil *confusion matrix* tersebut kemudian akan dilakukan evaluasi dengan melakukan perhitungan *accuracy*, *recall*, *precision*, *f-1 score* dan *specificity*. Berikut merupakan hasil evaluasi klasifikasi pada penelitian ini ditunjukkan pada Tabel 10.

**Tabel 10.** Hasil Evaluasi Klasifikasi

Klasifikasi	Akurasi	Recall	Precision	Specificity	F-1 Score
SVM + Chi Square	93%	86%	98%	98%	92%

### 4.8 Analisis Hasil Klasifikasi

Analisis hasil klasifikasi dilakukan untuk menampilkan visualisasi gambaran terhadap hasil penelitian yang telah dilakukan selain itu untuk memudahkan pembaca dalam memahami hasil penelitian dengan menampilkan *wordcloud*.



Gambar. 3. Wordcloud positif dan negatif

Kata yang muncul pada wordcloud positif diantaranya ‘aplikasi’, ‘bantu’, ‘mudah’, ‘bagus’ dan lain-lain. Berdasarkan visualisasi tersebut sentimen positif terhadap aplikasi menyatakan bahwa aplikasi PeduliLindungi cukup bagus dan membantu untuk melakukan vaksinasi. Sedangkan pada wordcloud negatif kata yang muncul adalah ‘aplikasi’, ‘susah’, ‘cek’, ‘vaksin’, ‘gagal’, ‘masuk’ dan lain-lain. Berdasarkan visualisasi tersebut sentimen negatif terhadap aplikasi menyatakan bahwa aplikasi pada PeduliLindungi untuk melakukan cek sertifikat vaksin dan masuk ke dalam ruang publik susah dan memiliki kendala

## 5 Kesimpulan

Berdasarkan hasil pembahasan dan analisis yang telah dilakukan pada penelitian ini maka dapat disimpulkan sebagai berikut :

1. Klasifikasi sentimen pada data ulasan pengguna aplikasi PeduliLindungi di Google Play dilakukan pada data dengan rentang waktu bulan November 2021 hingga April 2022 sebanyak 1000 data dengan jumlah data positif 446 dan data negatif 554. Pelabelan data dilakukan secara manual dengan menggunakan tiga anotator berbeda dan dievaluasi dengan nilai *kappa* kemudian dilakukan praproses hingga data menjadi bersih dan siap digunakan. Setelah data menjadi bersih kemudian data tersebut diberi nilai bobot di setiap kata dengan TF-IDF selanjutnya dilakukan pemilihan fitur terbaik dengan metode seleksi fitur *chi square*. Setelah mendapatkan fitur terbaik untuk tahap klasifikasi dilakukan pembagian data dengan perbandingan data 80% : 20%. Selanjutnya dilakukan klasifikasi data dengan menggunakan metode *Support Vector Machine* untuk mencari nilai parameter yang terbaik.
2. Performa hasil kinerja dan akurasi pengujian model hasil klasifikasi dengan metode *Support Vector Machine* dan seleksi fitur *chi square* terhadap data ulasan aplikasi PeduliLindungi mendapatkan hasil performa yang baik dengan kernel rbf dengan nilai parameter  $cost (C) = 1$  dan  $gamma (\gamma) = 10$  menghasilkan nilai akurasi sebesar 93%, *recall* sebesar 86%, *precision* sebesar 98%, *specificity* sebesar 98% dan *f1-score* sebesar 92%.

## 6 Saran

Berdasarkan hasil pembahasan dan analisis terhadap penelitian yang telah dilakukan, adapun saran yang dapat diberikan sebagai berikut :

1. Melakukan penambahan data ulasan yang akan digunakan pada penelitian dan melibatkan seseorang yang lebih ahli terkait frasa kata dalam suatu sentimen pada tahap pelabelan data.
2. Meningkatkan performa pada tahap preprocessing data misalnya pada tahap normalisasi data sehingga analisis data dapat menjadi lebih baik serta dapat meningkatkan nilai akurasi.

## Referensi

- [1] Abubakar, A. (2016). A Support Vector Machine Classification of Computational Capabilities of 3D Map on Mobile Device for Navigation Aid. *International Journal of Interactive Mobile Technologies (IJIM)*, 10(3). Retrieved from <http://dx.doi.org/10.3991/ijim.v10i3.5056>
- [2] Ahmad, K., Qadir, J., Qolomany, B., Khan, I., Khan, T., Sulaeman, M., . . . Al-Fuqaha, A. (2021). Sentiment Analysis of Users' Reviews on COVID-19 Contact Tracing Apps with a Benchmark Dataset. *arXiv*. Retrieved from <https://arxiv.org/abs/2103.01196>
- [3] Deng, N., Tian, Y., & Zhang, C. (2013). *Support Vector Machines Optimization Based Theory, Algorithms, and Extensions*. Florida: Taylor & Francis Group.
- [4] Herlinawati, N., Yuliani, Y., Faizah, S., Gata, W., & Samudi. (2020). Analisis Sentimen Zoom Clouds Meetings di Google Play Store Menggunakan Naive Bayes dan Support Vector Machine. *Journal of Computer Engineering System and Science*, 5(2), 293-298.
- [5] Indraini, A. N. (2021). Analisis Sentimen Terhadap Pembelajaran Daring di Indonesia Dengan Menggunakan Support Vector Machine (SVM). *Skripsi Thesis*.
- [6] Jo, T. (2019). *Text Mining Concepts, Implementation, and Big Data Challenge* (Vol. 45). Switzerland: Springer International Publishing AG.
- [7] Kumar, L., & Bhatia, P. K. (2013). Text Mining : Concepts, Process and Applications. *Journal of Global Research in Computer Science*, 4(3).
- [8] Luthfiana, L., Young, J. C., & Rusli, A. (2020). Implementasi Algoritma Support Vector Machine dan Chi Square untuk Analisis Sentimen User Feedback Aplikasi. *ULTIMATICS*, 12(2).
- [9] Nisa, A., Darwiyanto, E., & Asror, I. (2019). Analisis Sentimen Menggunakan Naive Bayes Classifier dengan Chi-Square Feature Selection Terhadap Penyedia Layanan Telekomunikasi. *e-Proceeding of Engineering*, 6(2), 8650.
- [10] Pratama, A. Y., Umidah, Y., & Voutama, A. (2021). Analisis Sentimen Media Sosial Twitter Dengan Algoritma K-Nearest Neighbor dan Seleksi Fitur Chi-Square (Kasus Omnibus Law Cipta Kerja). *Jurnal Sains Komputer & Teknologi (J-SAKTI)*, 5(2), 897-910.
- [11] Putri, C. E., & Hamzah, R. E. (2021). Aplikasi Pedulilindungi Mitigasi Bencana Covid-19 Di Indonesia. *Jurnal Pustaka Komunikasi*, 4(1), 66-78.
- [12] Somantri, O., & Apriliani, D. (2018). Support Vector Machine Berbasis Feature Selection untuk Sentiment Analysis Kepuasan Pelanggan Terhadap Pelayanan Warung dan Restoran Kuliner Kota Tegal. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 5(5), 537-548.
- [13] Sudiarsa, I. W., & Wiraditya, I. G. (2020). Analisis Usability Pada Aplikasi Peduli Lindungi Sebagai Aplikasi Informasi Dan Tracking Covid-19 Dengan Heuristic Evaluation. *Journal of Information Technology and Computer Sains*, 3(2).
- [14] Tsani, M. R., Rupaka, A. P., Asmoro, L., & S B A, B. P. (2020). Analisis Sentimen Review Transportasi Menggunakan Algoritma Support Vector Machine Berbasis Chi-Square. *Smart Comp*, 9(1).
- [15] Vinodhini, G., & Chandrasekaran, R. (2012). Sentiment Analysis and Opinion Mining : A Survey. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2(6).
- [16] Zulqornain, J. A., Indriati, & Adikara, P. P. (2021). Analisis Sentimen Tanggapan Masyarakat Aplikasi Tiktok Menggunakan Metode Naive Bayes dan Categorical Proportional Difference (CPD). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 5(7), 2886-2890.