

Analisis Sentimen pada Ulasan Aplikasi Jakarta Terkini (JAKI) di *Google Play Store* Menggunakan Metode *Support Vector Machine*

Daniel Dwi Eryanto Manurung¹, Nur Hafifah Matondang², Desta Sandya Prasvita³
Program Studi Informatika / Fakultas Ilmu Komputer
Universitas Pembangunan Nasional Veteran Jakarta
Jl. RS. Fatmawati, Pondok Labu, Jakarta Selatan, DKI Jakarta, 12450, Indonesia
danielDEM@upnvj.ac.id¹, nurhafifahmatondang@upnvj.ac.id², desta.sandya@upnvj.ac.id³

Abstrak. Aplikasi Jakarta Terkini (JAKI) terus berkembang dengan baik. Namun beberapa pengguna mengeluh ketidakpuasan terhadap layanan aplikasi. Beberapa jenis keluhan pengguna JAKI diantaranya sertifikat vaksin yang tidak tersedia, kuota vaksin selalu penuh dan sebagainya. Berdasarkan masalah tersebut, peneliti melakukan analisis sentimen pada Aplikasi JAKI melalui ulasan pengguna untuk memberikan informasi kepada masyarakat terkait kinerja aplikasi menggunakan metode *Support Vector Machine* (SVM) dan seleksi fitur *chi-square*. Data yang diperoleh dilakukan pelabelan dan pembersihan data terlebih dahulu sebelum tahapan *text processing*, kemudian data diberikan bobot dengan *Term Frequency-Inverse Document Frequency* (TF-IDF), kemudian dilakukan seleksi fitur dengan *chi-square*. Tahapan selanjutnya dilakukan pembagian data sebesar 80% 20% dan diklasifikasikan dengan metode SVM. Hasil penelitian ini adalah fitur terseleksi sebanyak 66 fitur dan pemodelan menggunakan kernel RBF, C=40 dan gamma=0.1 dari SVM, diperoleh 120 data uji mendapatkan *accuracy* 97%, *precision* 100%, *recall* 93%, dan *specificity* 100%.

Kata Kunci: Analisis Sentimen, Klasifikasi, JAKI, SVM, *Chi-Square*.

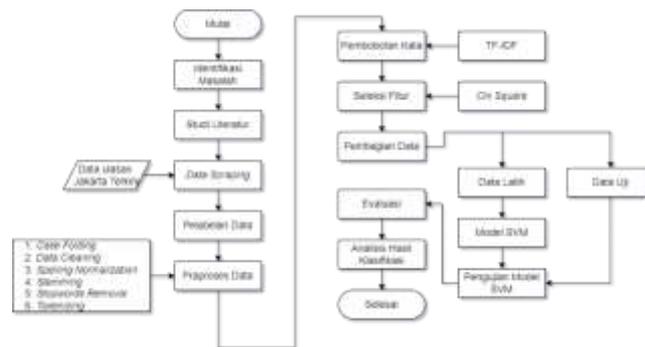
1 Pendahuluan

Indonesia secara signifikan dan pesat mengalami pertumbuhan teknologi pada era revolusi 4.0. Setiap pemerintah provinsi yang ada di Indonesia berlomba-lomba untuk melakukan pelayanan digitalisasi terbaik kepada masyarakat dengan berbagai cara. Pengembangan aplikasi merupakan bagian dari peningkatan kualitas pelayanan yang provinsi Jakarta lakukan bagi masyarakatnya. Unit Pengelola Jakarta *Smart City* (JSC) merupakan pengembang aplikasi ini, Pemerintah Provinsi Jakarta menamainya dengan aplikasi Jakarta Terkini (JAKI). Setelah dirilis pada tanggal 23 September 2019, lebih dari satu juta pengguna telah mengunduh aplikasi ini. Melalui aplikasi ini pengguna dapat mengetahui segala informasi seputar Jakarta dan menikmati beberapa layanan yang tersedia.

Seiring dengan berjalannya waktu aplikasi JAKI semakin berkembang, baik dari segi tampilan aplikasi serta layanan yang semakin banyak. Namun beberapa pengguna mengeluh ketidakpuasan mereka terhadap layanan yang disediakan oleh aplikasi JAKI. Hal ini dapat dilihat dari ulasan pengguna aplikasi JAKI di *Play Store* yang dimana masyarakat mengungkapkan kekecewaan mereka pada pelayanan JAKI yang mereka peroleh. Analisis sentimen dapat digunakan pada ulasan pengguna untuk menggali keterangan terkait evaluasi dan penilaian atas pelayanan JAKI.

Analisis sentimen menggunakan proses *text mining* menjadi solusi untuk mengatasi permasalahan tersebut. Pencarian pendapat pengguna terhadap suatu topik merupakan proses dari analisis sentimen. Pada analisis sentimen, ulasan-ulasan pengguna dikelompokkan menjadi dua, yaitu sentimen positif dan sentimen negatif [1]. Tujuan dasar pada analisis sentimen adalah untuk mengidentifikasi aspek positif atau negatif pada teks yang terdapat pada dokumen, kalimat atau pendapat, proses tersebut disebut pula polaritas teks [2]. Dengan penjabaran tersebut, maka metode klasifikasi dalam analisis sentimen dapat digunakan pada ulasan pengguna terkait aplikasi JAKI.

2 Metodologi Penelitian



Gambar 1. Penelitian ini terbagi menjadi beberapa tahap yakni identifikasi masalah, studi literatur, data *scraping*, pelabelan data, praproses data, pembobotan kata, seleksi fitur, pembagian data, pemodelan, evaluasi dan analisis hasil klasifikasi

2.1 Identifikasi Masalah

Proses identifikasi dilakukan dengan mengamati kondisi pada situs *Google Play Store* dan lainnya yang berhubungan dengan Jakarta Terkini (JAKI). Dari pengamatan yang dilakukan Jakarta Terkini (JAKI) merupakan aplikasi yang berbasis *Smart City* untuk peningkatan pelayanan pemerintah provinsi Jakarta terhadap masyarakat. Dengan banyaknya ungkapan kekecewaan pengguna pada aplikasi Jakarta Terkini, peneliti melakukan analisis sentimen terhadap aplikasi Jakarta Terkini dengan menggunakan metode klasifikasi *support vector machine*.

2.2 Studi Literatur

Tahapan selanjutnya yang dilakukan penulis yaitu studi literatur, tahap ini dilakukan dengan tujuan sebagai bahan pendukung dalam memecahkan pada penelitian yang digunakan sebagai sumber pustaka dengan cara mencari dan mengumpulkan referensi seperti *e-book*, jurnal, *website* dan penelitian lainnya yang terkait dengan permasalahan pada analisis sentimen, *text mining*, *preprocessing*, dan metode klasifikasi algoritma *support vector machine*.

2.3 Data Scraping

Data Scraping dibutuhkan dalam hal penarikan dan pengumpulan *review* pengguna yang didapatkan dari situs *google play* menggunakan bantuan *library* yang disediakan oleh *python* yaitu *google-play-scrapers*. Data *review* dikumpulkan adalah data berbahasa Indonesia dan diurutkan berdasarkan sentimen yang relevan. Data yang berhasil dikumpulkan selanjutnya dilakukan tahap pelabelan.

2.4 Pelabelan Data

Setelah data yang dibutuhkan telah terkumpul, proses selanjutnya yaitu melakukan pelabelan pada sentimen yang sudah terkumpul sebelumnya. Hasil pelabelan otomatis akan dilakukan oleh annotator yang berjumlah tiga orang. Kemudian hasil pelabelan tersebut dievaluasi menggunakan Persamaan *fleiss kappa* (1) untuk mengetahui pelabelan tersebut bagus atau tidak. *Fleiss kappa* adalah salah satu metode yang digunakan untuk mengukur kesepakatan label yang sebelumnya telah dilabelkan oleh annotator [3]. *Fleiss kappa* berfungsi mengetahui valid atau tidaknya pelabelan yang telah dilakukan oleh annotator [4]. Persamaan untuk menghitung nilai dari persamaan *fleiss kappa value* terdapat pada Persamaan (1):

$$kappa\ value = \frac{p_0 - p_e}{1 - p_e} \quad (1)$$

$$p_0 = \frac{1}{N} (\sum_{i=1}^N p_i) \quad (2)$$

$$p_e = \sum_{j=1}^k p_j^2 \quad (3)$$

$$p_i = \frac{1}{n(n-1)} (\sum_{j=1}^k n_{ij}^2 - n_{ij}) \quad (4)$$

$$p_j = \frac{n_{.j}}{N.n} = \frac{1}{Nn} \sum_{i=1}^N n_{ij} \quad (5)$$

Berikut interpretasi Kappa pada Tabel 1:

Tabel 1. Interpretasi Kappa

Indeks Kappa	Agreement
1.00 – 0.81	<i>Almost perfect agreement</i>
0.80 – 0.61	<i>Substantial agreement</i>
0.60 – 0.41	<i>Moderate agreement</i>
0.40 – 0.21	<i>Fair agreement</i>
0.20 – 0.00	<i>Slight agreement</i>
<0	<i>Poor agreement or no agreement</i>

2.5 Praproses Data

Tahap *preprocessing* dilakukan sebelum tahap *modelling* atau pemodelan data, dimana terjadi proses pembentukan dan pembersihan informasi sehingga dapat diolah pada tahap berikutnya [5]. Data yang sudah diberi label pada proses sebelumnya akan diolah pada tahap praproses data. Tahapan ini dilakukan proses membersihkan data untuk mengklasifikasi pada analisis sentimen. Dengan tujuan untuk mengurangi dan mengatasi *noise* data. Pada tahap ini juga bertujuan untuk mengubah data yang tidak terstruktur menjadi terstruktur. Adapun tahap praproses yang akan dilakukan yaitu.

2.5.1 Case Folding

Case Folding ialah proses menyeragamkan atau menyamakan seluruh teks yang akan digunakan menjadi huruf kecil (*lowercase*) [6].

2.5.2 Data Cleaning

Data Cleaning adalah tahapan untuk menghilangkan *noise* yang kurang penting dalam teks [6]. Pada tahapan ini dilakukan untuk menghilangkan value yang tidak memiliki nilai yang berarti seperti *noise* seperti URL, angka, simbol-simbol, tanda baca dan lain-lain.

2.5.3 Spelling Normalization

Spelling Normalization adalah tahap seleksi kata yang tidak baku, singkat atau salah eja untuk diperbaiki menjadi kata baku sesuai KBBI. *Spelling Normalization* dilakukan untuk memperkecil dimensi kata yang memiliki ejaan berbeda namun mengandung makna yang sama [7].

2.5.4 Stemming

Proses ini adalah tahapan mengubah suatu data yang memiliki imbuhan menjadi kata dasar. *Stemming* merupakan proses pencarian kata dasar dengan menghilangkan imbuhan yang terdapat pada kata tersebut sehingga tersisa kata dasar [6].

2.5.5 Stopword Removal

Pada tahap *stopword removal* dilakukan proses untuk menghilangkan kata atau *term* yang tidak memiliki makna. *Stopword* adalah kata-kata yang acap kali terlihat namun tidak mempengaruhi sentimen, maka kata tersebut diklasifikasikan sebagai *noise* untuk dihilangkan [8].

2.5.6 Tokenizing

Tokenizing adalah proses dimana teks dipecah menjadi beberapa kata sesuai dengan kebutuhan [9]. Hasil dari proses *tokenizing* adalah deretan token dari proses pemecahan teks.

2.6 Pembobotan Kata

Setelah dilakukan praproses data, langkah selanjutnya adalah masuk ke tahap pembobotan kata. Pada tahap ini setiap kata atau *term* yang terdapat pada dokumen akan dihitung frekuensinya dengan menggunakan metode pembobotan TF-IDF. Penghitungan IDF dapat dilakukan dengan menggunakan Persamaan (6).

$$w_{ij} = tf_{ij} \cdot \log \frac{D}{df} \quad (6)$$

Keterangan:

tf_{ij} : Jumlah kemunculan kata atau *term* pada suatu dokumen.

D : Jumlah keseluruhan dokumen.

df : Jumlah kemunculan dokumen yang mengandung kata.

2.7 Seleksi Fitur *Chi-square*

Setelah dilakukan pembobotan angka menggunakan TF-IDF, data selanjutnya dilakukan seleksi fitur. Seleksi fitur sendiri berfungsi untuk pengurangan atau mereduksi fitur suatu data. Seleksi fitur dipercaya dapat meningkatkan akurasi pada suatu klasifikasi. Pada tahapan ini akan dilakukan seleksi fitur menggunakan algoritma *chi-square*. *Chi-square* termasuk dalam metode seleksi fitur tipe *supervised*, yang dimana mampu mengurangi atau mereduksi fitur tanpa mengurangi akurasi [10]. Berikut Persamaan (7) *chi-square*:

$$x^2(t, c) = \sum \frac{(O_i - E_i)^2}{E_i} \quad (7)$$

Keterangan:

O_i : *Observed value* (nilai yang sebenarnya)

E_i : *Expected value* (nilai yang diharapkan)

2.8 Pembagian Data

Data yang sudah melewati tahap seleksi fitur, selanjutnya akan dibagi menjadi data latih dan data uji. Persentase data latih akan lebih besar dari data uji, dimana pembagiannya yaitu 80% untuk data latih dan 20% untuk data uji.

2.9 Pemodelan *Support Vector Machine*

Data latih yang sudah dibagi akan dilakukan klasifikasi dengan menggunakan metode algoritma *support vector machine*. *Support Vector Machine* merupakan metode pembelajaran linier yang mencari *Hyperplane* optimal untuk pemisahan dua buah kelas [6]. *Hyperplane* berfungsi sebagai pemisah dua kelompok label +1 atau label -1, dan setiap label memiliki polanya sendiri [11]. Penentuan parameter *support vector machine* akan menggunakan *library GridSearchCV* yang sudah disediakan oleh *python*. *GridSearchCV* berfungsi untuk mendapatkan nilai parameter terbaik untuk klasifikasi. Digunakan *hyper-parameter* C, kernel dan gamma untuk klasifikasinya dengan nilai interval C yaitu 0,5, 0,01, 0,1, 1, 10, 20, 40, 50, 100 kemudian interval kernel yaitu *rbf*, *poly*, *sigmoid*, *linear* dan untuk interval gamma yaitu 1, 0,1, 0,01, 0,001. Kemudian data uji akan dilakukan *testing* dengan model yang sudah dibentuk.

2.10 Evaluasi

Setelah melalui proses uji pada proses sebelumnya. Hasil dari *testing* akan dievaluasi untuk menguji performa kinerja dari model yang terbentuk menggunakan algoritma *support vector machine*. Tabel 2 dibawah ini mendeskripsikan pengujian hasil klasifikasi data untuk menghitung *accuracy*, *precision*, *recall*, dan *specificity* menggunakan *confusion matrix*.

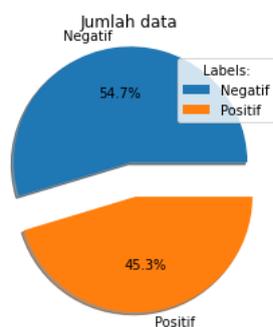
Tabel 2. *Confusion Matrix*

	Aktual	
	Positif	Negatif

Tabel 3. Ulasan dengan label hasil

Ulasan	Annotator 1	Annotator 2	Annotator 3	Hasil
Tolong diperbaiki min, masa untuk pilihan nama kelurahan dan kecamatan terbalik. Harusnya pilihan kecamatan pasar rebo yg tampil jd pilihan nama kelurahan kalisari, cijantung, baru dll	Negatif	Negatif	Negatif	Negatif
Tolong dong jaki, beri pelayanan yang baik. Kami sudah divaksin sertifikat nya susah minta ampun. Harusnya disertakan call center jaki. Biar komunikasi gak susah 😊	Negatif	Negatif	Negatif	Negatif

Pada Tabel 3 menunjukkan hasil penilaian oleh tiga annotator terhadap suatu data ulasan. Pada Tabel 3 di ulasan kedua, terlihat annotator 1, 2, dan 3 memberi label “Negatif”, maka dapat diambil keputusan untuk hasil label pada ulasan pertama adalah label “Negatif”. Setelah tiap orang memberikan label untuk setiap ulasan, penting untuk mengetahui apakah kesepakatan tersebut konsisten atau tidak. Kesepakatan tersebut diukur dengan *fleiss kappa*. Dimana *fleiss kappa* bertujuan untuk mengetahui valid atau tidaknya pelabelan yang telah dilakukan oleh annotator. Jika nilai *kappa* hanya mencapai $<0,60$ maka label data yang memiliki kategori berbeda dan akan diskusikan kembali oleh ketiga annotator, bila nilai *kappa* mencapai $>0,60$ kesepakatan sudah dapat dikatakan baik. Berdasarkan evaluasi hasil pelabelan data menggunakan *fleiss kappa*, didapatkan hasil sebesar 0,990592 yang merupakan kategori *Almost perfect agreement*, maka kekuatan kesepakatan dari pelabelan data memiliki nilai yang sangat baik dan dapat dilakukan ke proses selanjutnya. Hasil dari proses pelabelan didapatkan data berlabel positif sebanyak 453 data dan negatif 547 data. Rincian jumlah data diilustrasikan pada Gambar 3.



Gambar 3. Diagram pie jumlah label

3.3 Praproses

Langkah-langkah yang dilakukan dalam praproses yaitu *case folding*, *data cleaning*, *normalization*, *stemming*, *stopwords removal*, dan *tokenizing* dapat dilihat hasil dari praproses pada Tabel 4.

Tabel 4. Hasil Praproses

Sebelum Praproses	Sesudah Praproses
Tolong diperbaiki min, masa untuk pilihan nama kelurahan dan kecamatan terbalik. Harusnya pilihan kecamatan pasar rebo yg tampil jd pilihan nama kelurahan kalisari, cijantung, baru dll	['admin', 'pilih', 'nama', 'lurah', 'camat', 'pilih', 'camat', 'pasar', 'rebo', 'tampil', 'pilih', 'nama', 'lurah', 'kalisari', 'cijantung']

Tolong dong jaki, beri pelayanan yang baik. Kami sudah divaksin sertifikat nya susah minta ampun. Harusnya disertakan call center jaki. Biar komunikasi gak susah 😊

['jaki', 'layan', 'vaksin', 'sertifikat', 'susah', 'ampun', 'call', 'center', 'jaki', 'komunikasi', 'susah']

3.4 Pembobotan TF-IDF

Pada pembobotan kata menggunakan metode TF-IDF ini ada hal yang perlu dilakukan yaitu menghitung TF (*Term Frequency*) pada setiap dokumen dan IDF (*Inverse Document Frequency*) di tiap dokumen menggunakan Persamaan (7), lalu menghitung TFIDF dengan mengalikan hasil dari TF dengan IDF dan jumlah fitur yang didapatkan sebanyak 1857 kata. Hasil perhitungan dari TF-IDF dapat dilihat pada Tabel 5.

Tabel 5. Hasil Penghitungan Bobot TF-IDF

Token	TF					DF	IDF	TD-IDF				
	D1	D2	D3	D4	D5			D1	D2	D3	D4	D5
admin	1	0	0	0	0	1	0,69897	0,69897	0	0	0	0
ampun	0	0	1	0	0	1	0,69897	0	0	0,69897	0	0
aplikasi	0	1	0	1	0	2	0,39794	0	0,39794	0	0,39794	0
susah	0	1	2	0	0	2	0,39794	0	0,39794	0,79588	0	0
tuju	0	1	0	0	0	1	0,69897	0	0,69897	0	0	0
vaksin	0	0	1	0	2	2	0,39794	0	0	0,39794	0	0,79588

3.5 Seleksi Fitur

Setelah dilakukannya proses pembobotan pada setiap fitur, tahap selanjutnya adalah proses seleksi fitur dimana fitur kata terbaik dihasilkan berdasarkan nilai *chi-square* tertinggi dengan menggunakan Persamaan (7). Proses penyeleksian fitur kata dilakukan dengan cara memilih yang akan digunakan dalam klasifikasi sedangkan yang tidak terpilih akan dibuang atau dihapus. Proses seleksi akan dilakukan dengan pengujian terhadap nilai *chi-square*. Jika nilai *chi-square* suatu fitur memiliki nilai diatas nilai kritis, maka fitur tersebut akan terpilih sebagai fitur yang akan digunakan dalam klasifikasi. Pengujian akan menggunakan beberapa nilai kritis pada taraf nyata α 0.950, 0.900, 0.100, 0.050, 0.250, 0.010, dan 0.005 yaitu 0.004, 0.016, 2.706, 3.841, 5.024, 6.635, dan 7.879. Hasil dari penyeleksian fitur dengan menggunakan beberapa *critical value* terdapat pada Tabel 6.

Tabel 6. Jumlah fitur setelah diseleksi

<i>Critical Value</i>	Jumlah Fitur	Jumlah Fitur yang Tereliminasi
0.004	1832	25
0.016	1811	46
2.706	111	1746
3.841	66	1791
5.024	50	1807
6.635	36	1821
7.879	26	1831

3.6 Pembagian Data

Setiap pengujian data akan dibagi menjadi 80% untuk data *training* dan 20% untuk data *testing* untuk diteruskan ke dalam proses pemodelan. Jumlah pembagian data dapat dilihat pada Tabel 7.

Tabel 7. Pembagian Data

	Positif	Negatif	Total
Data Training	359	441	800
Data Testing	94	106	200

3.7 Pemodelan Data

Setiap data yang sudah dibagi menjadi 80:20 akan dilakukan pelatihan terhadap data latih untuk mendapatkan sebuah bentuk model klasifikasi. Setelah dilakukan pelatihan, data model akan diuji untuk mengetahui kemampuan model dengan mengklasifikasikan data uji. Sebelum dilakukannya pemodelan data dilakukan pencarian parameter terbaik menggunakan *GridSearchCV*. Uji parameter menggunakan parameter kernel yaitu rbf, poly dan linear kemudian parameter C yaitu 0.5, 0.01, 0.1, 1, 10, 20, 40, 50, 100 dan parameter gamma yaitu 1, 0.1, 0.01, 0.001.

Hasil dari proses pencarian parameter terbaik menggunakan *GridSearchCV* Tabel 4.15, didapatkan parameter terbaik yaitu kernel = rbf, C = 40 dan gamma = 0.1 dengan *score* sebesar 0.94 dan akurasi yang didapat 0.97. Sehingga data yang digunakan adalah data yang menggunakan *critical value* = 3.841 yang menyeleksi fitur sebanyak 66 kata. Hasil pencarian parameter terbaik serta akurasi yang didapatkan untuk setiap pengujian data terdapat pada Tabel 8.

Tabel 8. Hasil Pencarian Parameter Terbaik

<i>Critical Value</i>	Jumlah Fitur	<i>Best Parameter</i>				Akurasi
		<i>Kernel</i>	<i>Cost</i>	<i>Gamma</i>	<i>Score</i>	
0.004	1832	rbf	10	1	0.9525	0.945
0.016	1811	linear	0.5	1	0.9487	0.95
2.706	111	linear	40	1	0.9462	0.96
3.841	66	rbf	40	0.1	0.9400	0.97
5.024	50	rbf	50	1	0.9375	0.965
6.635	36	rbf	20	1	0.9362	0.965
7.879	26	rbf	1	1	0.9137	0.93

3.8 Evaluasi

Pemodelan menggunakan kernel RBF, C = 40 dan gamma = 0.1 dari *Support Vector Machine* (SVM), diperoleh bahwa 200 data uji (*testing*) mendapatkan *accuracy* sebesar 0.97 yang berarti bahwa 97% dari data uji diprediksi dengan benar berdasarkan kelas yang sebenarnya. Kelas positif diprediksi benar sebanyak 88 data dan kelas negatif sebanyak 106 data. Nilai *recall* sebesar 0.936 yang berarti model dapat memprediksi dengan benar kelas positif sebesar 93.6% dari total keseluruhan data yang benar-benar kelas positif. Sedangkan untuk tingkat *precision* dari model untuk memprediksi benar kelas positif dari keseluruhan data yang diprediksi kelas positif sebesar 1 atau 100%. Pada model ini nilai *recall* memiliki nilai yang rendah artinya masih terdapat kesalahan pada kelas positif yang diprediksikan sebagai kelas negatif. Hasil dari *Confusion Matrix* dapat dilihat pada Tabel 9.

Tabel 9. Hasil *Confusion Matrix*

		Kelas Sebenarnya	
		Positif	Negatif
Kelas Prediksi	Positif	88 (TP)	0 (FP)
	Negatif	6 (FN)	106 (TN)

3.9 Analisis Hasil Klasifikasi

Data *input* yang digunakan untuk melakukan proses visualisasi kata-kata apa saja yang terkandung dalam sentimen positif yaitu data berlabel positif yang sudah melalui tahapan pra-proses



Gambar 3. Wordcloud Sentimen Positif

Berdasarkan *wordcloud* pada Gambar 3 menunjukkan bahwa beberapa kata-kata yang sering muncul seperti “terima”, “kasih”, “cepat”, “bagus”, “bantu”, “keren” dan lain-lain. Disimpulkan bahwa sentimen positif pada ulasan pengguna aplikasi JAKI memberikan respon yang baik dikarenakan pengguna merasa terbantu dengan adanya aplikasi JAKI.



Gambar 4. Wordcloud Sentimen Negatif

Berdasarkan *wordcloud* pada Gambar 4 menunjukkan bahwa beberapa kata-kata yang sering muncul seperti “susah”, “daftar”, “vaksin”, “lokasi”, “penuh”, “kuota” dan lain-lain. Disimpulkan bahwa sentimen negatif pada ulasan pengguna aplikasi JAKI memberikan respon yang kurang baik dikarenakan pengguna merasa kesulitan atau kesusahan dalam menggunakan aplikasi Jakarta Terkini terutama pada layanan vaksinasi.

4 Kesimpulan dan Saran

4.1 Kesimpulan

Data yang digunakan merupakan ulasan pengguna dari *Google Play Store* pada aplikasi Jakarta Terkini (JAKI) sebanyak 1000 ulasan. Data dilakukan pelabelan dengan cara manual dan didapatkan jumlah kelas masing-masing sebanyak 453 positif dan 547 negatif. Kemudian data dilakukan pembersihan terlebih dahulu di praproses sebelum dilakukan pembobotan, kemudian data yang sudah bersih diberikan bobot setiap kata dengan *Term Frequency-Invers Document Frequency* (TF-IDF) yang nantinya akan dijadikan sebagai fitur setelah itu fitur yang diperoleh akan diseleksi kembali dengan menggunakan metode *chi-square*. Fitur yang sudah terseleksi akan dilakukan pembagian data menjadi data latih dan data uji untuk membentuk suatu model dengan menggunakan metode *Support Vector Machine*. Performa pengujian model yang terbentuk dengan metode *Support Vector Machine* dilakukan dengan membandingkan hasil dari model tersebut dengan data uji sebanyak 200 ulasan pengguna. Selanjutnya dilakukan evaluasi tersebut didapatkan hasil *accuracy* sebesar 97%, dengan nilai *precision* 100%, *recall* 93.6%, dan *specificity* 100%.

4.2 Saran

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

- Diharapkan untuk penelitian berikutnya untuk meningkatkan pada proses *pre-processing* dan pemodelan data yang lebih baik sehingga dapat menganalisis dengan baik dan evaluasi model yang sangat baik.
- Label yang digunakan tidak hanya positif dan negatif saja, tetapi menambahkan label netral.
- Penelitian selanjutnya dapat menggunakan algoritma klasifikasi lainnya seperti *Naïve Bayes*, *K-Nearest Neighbor* dan lain-lain sebagai perbandingan untuk performa model.

5 Referensi

- [1] F. Bei and S. Saepudin, "ANALISIS SENTIMEN APLIKASI TIKET ONLINE DI PLAY STORE MENGGUNAKAN METODE SUPPORT VECTOR MACHINE (SVM)," *SISMATIK (Seminar Nasional Sistem Informasi dan Manajemen Informatika)*, pp. 91–97, Aug. 2021.
- [2] R. Wahyudi and G. Kusumawardhana, "Analisis Sentimen pada review Aplikasi Grab di Google Play Store Menggunakan Support Vector Machine," *JURNAL INFORMATIKA*, vol. 8, pp. 200–207, Sep. 2021.
- [3] T. R. Nichols, P. M. Wisner, G. Cripe, and L. Gulabchand, "Putting the Kappa Statistic to Use," *Qual Assur J*, pp. 57–61, 2010.
- [4] J. A. Zulqornain, Indriati, and P. P. Adikara, "Analisis Sentimen Tanggapan Masyarakat Aplikasi Tiktok Menggunakan Metode Naïve Bayes dan Categorical Proportional Difference (CPD)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, pp. 2886–2890, Jul. 2021.
- [5] N. Herlinawati, Y. Yuliani, S. Faizah, W. Gata, and Samudi, "ANALISIS SENTIMEN ZOOM CLOUD MEETINGS DI PLAY STORE MENGGUNAKAN NAÏVE BAYES DAN SUPPORT VECTOR MACHINE," *CESS (Journal of Computer Engineering System and Science)*, vol. 5, pp. 293–298, Jul. 2020.
- [6] F. F. Irfani, M. Triyanto, A. D. Hartanto, and Kusnawi, "Analisis Sentimen Review Aplikasi Ruangguru Menggunakan Algoritma Support Vector Machine," *Jurnal Bisnis, Manajemen dan Informatika*, vol. 16, pp. 258–266, Feb. 2020.
- [7] D. Y. Praptiwi, "ANALISIS SENTIMEN ONLINE REVIEW PENGGUNA E-COMMERCE MENGGUNAKAN METODE SUPPORT VECTOR MACHINE DAN MAXIMUM ENTROPY," UNIVERSITAS ISLAM INDONESIA, Yogyakarta, 2018.
- [8] L. B. Ilmawana and M. A. Mude, "Perbandingan Metode Klasifikasi Support Vector Machine dan Naïve Bayes untuk Analisis Sentimen pada Ulasan Tekstual di Google Play Store," *ILKOM Jurnal Ilmiah*, vol. 12, pp. 154–161, Aug. 2020.
- [9] M. I. Ahmadi, F. Apriani, M. Kurniasari, S. Handayani, and G. Gustian, "SENTIMENT ANALYSIS ONLINE SHOP ON THE PLAY STORE USING METHOD SUPPORT VECTOR MACHINE (SVM)," *Seminar Nasional Informatika 2020 (SEMNASIF 2020)*, pp. 196–203, 2020.
- [10] O. Somantri and D. Apriliani, "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)*, vol. 5, pp. 537–548, Oct. 2018.
- [11] W. A. Luqyana, I. Cholissodin, and R. S. Perdana, "Analisis Sentimen Cyberbullying pada Komentar Instagram dengan Metode Klasifikasi Support Vector Machine," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, pp. 4704–4713, Nov. 2018.