

Analisis Sentimen Ulasan Aplikasi Pospay dengan Algoritma Support Vector Machine

Dea Safryda Putri ^a, Taufik Ridwan ^b

^a Universitas Singaperbangsa Karawang, Jl. HS. Ronggo Waluyo, Puseurjaya, Kec. Telukjambe Timur, Karawang 41361, Indonesia

^b Universitas Singaperbangsa Karawang, Jl. HS. Ronggo Waluyo, Puseurjaya, Kec. Telukjambe Timur, Karawang 41361, Indonesia

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima Redaksi: 31 Desember 2022

Revisi Akhir: 12 Januari 2023

Diterbitkan Online: 10 Maret 2023

KATA KUNCI

Analisis Sentimen

Pospay

Support Vector Machine

Text Mining

KORESPONDENSI

E-mail: dea.safryda19039@student.unsika.ac.id

A B S T R A C T

Pospay application is a form of financial technology belonging to Pos Indonesia. Pospay application on the Google Play Store has more than 25 thousand user reviews. The more user reviews, the more difficult and longer it will take for prospective users and application managers to conclude information on user sentiment trends that are useful in making decisions about application use and evaluation. Sentiment analysis is the solution to this problem because sentiment analysis is able to classify unstructured data to generate sentiment information efficiently by applying data mining algorithms. This research uses the Knowledge Discovery in Database (KDD) method, where at the data mining stage, the Support Vector Machine algorithm is applied in making the model. Grid search was applied to test 3 scenarios so that the proportion of data distribution with the best accuracy was obtained, namely, 90:10, using RBF kernel with parameters: $C = 1$, $\gamma = 1$. The results of this research were a model with 95% accuracy, 91% precision, 100% recall, and 95% f1-score. Information was also obtained that the sentiment of Pospay application users on the Google Play Store tends to be positive (54.1%) but not much far from the percentage of negative sentiments (45.9%).

1. PENDAHULUAN

Teknologi finansial berkembang semakin pesat sehingga masyarakat Indonesia semakin tidak asing dengan beragam teknologi finansial, seperti pembayaran digital, dompet digital, bank digital, pinjaman uang digital dan lain-lain [1]. Salah satu produk teknologi finansial yaitu aplikasi Pospay. Pospay merupakan aplikasi berbasis giropos milik Pos Indonesia yang dapat memfasilitasi layanan transaksi keuangan secara *mobile*. Aplikasi Pospay tersedia dalam Google Play Store dan sudah diunduh oleh lebih dari 1 juta pengguna. Pada bulan November 2022, aplikasi ini sempat menduduki peringkat pertama dalam aplikasi terpopuler kategori keuangan di Google Play Store [2].

Pada Google Play Store, terdapat fitur ulasan bagi masing-masing aplikasi. Ulasan merupakan penilaian positif (kepuasan dan pujian) atau negatif (keluhan, kritik dan saran) dari pengguna aplikasi atau layanan. Ulasan pengguna dapat menjadi acuan calon pengguna dalam membuat keputusan memakai atau tidak suatu aplikasi atau layanan [3]. Bagi pihak pengelola aplikasi, ulasan pengguna bermanfaat untuk evaluasi aplikasi,

meningkatkan kualitas layanan dan menyusun strategi dalam mempertahankan loyalitas pengguna [4].

Aplikasi Pospay di Google Play Store memiliki lebih dari 25 ribu ulasan pengguna [2]. Semakin banyak ulasan pengguna maka akan semakin sulit dan lama waktu yang diperlukan oleh calon pengguna dan pihak pengelola aplikasi untuk menyimpulkan informasi kecenderungan sentimen pengguna [3]. Analisis sentimen adalah proses *text mining* yang mengklasifikasikan data tidak terstruktur untuk menghasilkan informasi sentimen secara efisien dengan algoritma *data mining* [5]. Analisis sentimen perlu diterapkan pada ulasan aplikasi Pospay di Google Play Store.

Algoritma Support Vector Machine (SVM) merupakan salah satu algoritma klasifikasi *data mining* dalam kategori *supervised learning*. SVM memiliki keunggulan yaitu mampu menerapkan pemisahan linear pada input data nonlinear berdimensi besar dengan pemanfaatan fungsi kernel [6]. Menurut Hartmann (2018) dalam [7], banyak peneliti yang melaporkan bahwa klasifikasi teks dengan SVM memberikan hasil yang lebih akurat.

Berdasarkan informasi yang telah dipaparkan, maka penelitian ini menerapkan algoritma SVM dalam analisis sentimen pada ulasan

aplikasi Pospay. Penelitian ini dilakukan dengan tujuan untuk mengetahui informasi menyeluruh terkait sentimen pengguna aplikasi Pospay di Google Play Store cenderung positif atau negatif sekaligus mengetahui performa algoritma SVM.

2. TINJAUAN PUSTAKA

2.1 Text Mining

Text mining adalah cabang ilmu dari data mining. Text mining merupakan proses penambangan data teks (tidak terstruktur) berjumlah banyak untuk menemukan informasi tertentu [8]. Salah satu bentuk implementasi text mining yang populer yaitu analisis sentimen [9]. Analisis sentimen adalah proses text mining yang mengklasifikasikan data tidak terstruktur untuk menghasilkan informasi sentimen secara efisien dengan algoritma data mining [5]. Penerapan text mining memiliki konsep serupa dengan data mining. Sebelum data teks diimplementasikan menggunakan metode dan algoritma text mining, data teks harus terlebih dahulu melalui pra-pemrosesan teks (text preprocessing) [8].

Tahapan pra-pemrosesan teks terdiri dari:

1. *Case folding*: mengubah huruf besar menjadi kecil
2. *Cleaning*: menghapus angka, tanda baca, simbol, *whitespace*, dan sebagainya.
3. *Tokenize*: memotong string berdasarkan tiap kata.
4. *Normalize*: memperbaiki kata yang tidak sesuai EYD.
5. *Filtering*: menyaring kata-kata yang penting.
6. *Stemming*: mengubah kata berimbuhan menjadi kata dasarnya.

2.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) adalah salah satu algoritma klasifikasi data mining kategori supervised learning. Cara kerja SVM yaitu dengan memisahkan antar kelas menggunakan hyperplane dengan jarak margin maksimum. SVM mampu menerapkan pemisahan linear pada input data nonlinear berdimensi besar dengan fungsi kernel [6]. Kernel yang umum digunakan dalam SVM yaitu kernel linear, RBF dan polynomial [10]. Efektivitas kernel bergantung pada parameter yang diterapkan pada suatu kernel.

Dalam [11], beberapa parameter pengaturan kernel diantaranya sebagai berikut.

1. Parameter C merupakan nilai parameter regulasi yang jika tidak diatur nilainya, maka akan diterapkan nilai default yaitu C = 1.
2. Parameter *coef()* merupakan independen dalam fungsi kernel yang hanya diimplementasikan pada kernel polynomial dan sigmoid. Apabila parameter *coef()* tidak diatur nilainya, maka akan diterapkan nilai default yaitu 0.
3. Parameter gamma (γ) merupakan kernel koefisien dengan nilai awal adalah *scale*.
4. Parameter *degree* merupakan fungsi kernel kusus polynomial, dimana suku pangkat tertinggi pengaturan sebesar 3 dan pengaturan default sebesar 0,001.
5. Parameter epsilon merupakan nilai yang berfungsi untuk menaksir nilai limit dan kebenaran nilainya. Apabila parameter epsilon tidak diatur nilainya, maka akan diterapkan nilai default yaitu 0,1.

2.3 Grid Search

Grid search merupakan cara untuk memperoleh kombinasi hyperparameter yang paling sesuai. Cara kerja grid search adalah dengan menerapkan setiap kombinasi parameter yang diinginkan dan mengevaluasi model sehingga menghasilkan parameter yang mampu memberikan hasil terbaik bagi model [12].

2.4 Proportionate Stratified Random Sampling

Proportionate stratified random sampling merupakan teknik pengambilan sampel berdasarkan strata atau tingkatan data dalam populasi secara proporsional. Teknik ini termasuk dalam kelompok probability sampling dimana setiap data dalam populasi memiliki peluang yang sama untuk dipilih dan sampel terpilih dapat mewakili keseluruhan populasi yang ada [13]. Dalam pengambilan sampel minimal sebesar 10% dari jumlah populasi yang ada [14].

2.5 Term Frequency - Inverse Document Frequency

Term frequency (TF) adalah jumlah kata/term dalam suatu dokumen, sedangkan inverse document frequency (IDF) adalah frekuensi kemunculan kata/term diseluruh dokumen [15]. TF-IDF bermanfaat dalam meningkatkan performa model.

Berikut merupakan persamaan dari TF-IDF.

$$TF - IDF = \frac{n_{yx}}{\sum t_y} \cdot \log \frac{\sum d}{n_{dx}} \tag{1}$$

keterangan:

x = kata ke-x y = dokumen ke-y n = jumlah
t = term/kata d = dokumen

2.6 Confusion matrix

Confusion matrix merupakan suatu alat untuk mengukur performa model berdasarkan hasil data aktual dan data prediksi yang terdiri dari true positif (TP), true negative (TN), false positif (FP) dan false negative (FN). True positif dan true negative merupakan data kelas yang selaras secara aktual dan prediksi, sedangkan false positif dan false negative merupakan data kelas tidak selaras secara aktual dan prediksi sehingga dianggap kesalahan [10].

Tabel 1. Confusion Matrix

	Prediction	True (positif)	False (negative)
Actual			
True (positif)		TP	FP
False (negative)		FN	TN

Sumber: [10]

2.7 Visualisasi data

Visualisasi data merupakan bentuk penyajian informasi secara lebih ringkas, jelas dan menarik. *Bar chart* merupakan bentuk visualisasi yang cocok untuk mengetahui perbedaan jumlah/persentase data antar kategori [16].

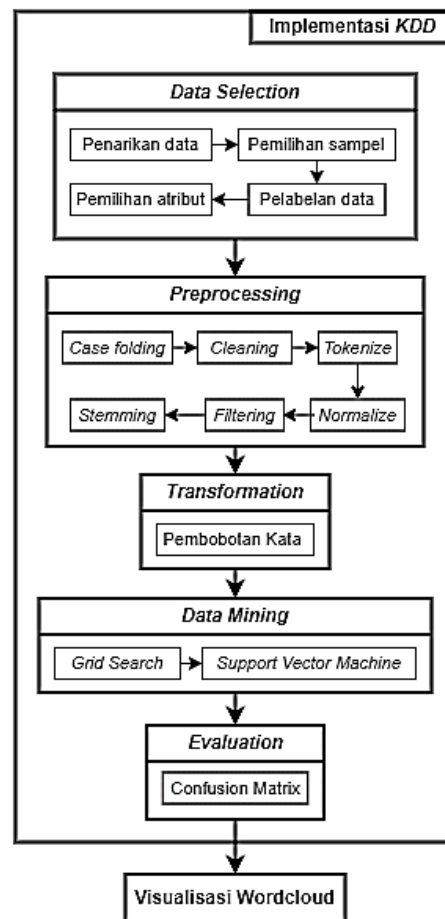
Pada *text mining*, visualisasi data yang umum diimplementasikan yaitu wordcloud. Wordcloud menyajikan kata yang sering muncul ke dalam sebuah *frame*, dimana semakin sering suatu kata muncul maka semakin besar ukuran kata tersebut dan berlaku pula sebaliknya [3].

2.8 Penelitian Terkait

Analisis sentimen menggunakan algoritma SVM ditemukan dalam penelitian yang dilakukan oleh Nana Mulyana Maghfur dkk. mengenai topik dinamika Covid-19 khususnya varian dari Covid-19 menghasilkan akurasi sebesar 88,5% [17]. Penelitian analisis sentimen yang dilakukan oleh Auliya Rahman Isnain dkk., dimana topik yang dibahas yaitu kebijakan *lockdown* menghasilkan akurasi sebesar 74% menggunakan algoritma SVM [18]. Penelitian oleh Katherine Ivana Ruslim dkk. juga mengimplementasikan SVM dalam kasus analisis sentimen ulasan aplikasi *mobile banking* yaitu *BCA Mobile* menghasilkan akurasi sebesar 84,6% [19]. Penelitian yang dilakukan oleh Nidaul Hasanati dkk. mengimplementasikan algoritma SVM dalam melakukan analisis sentimen pada data *tweet* pengguna Twitter mengenai topik vaksin Covid-19 dan menerapkan *grid search* yang menghasilkan parameter dengan akurasi terbaik yaitu sebesar 85% dari penerapan parameter kernel = RBF, C = 100 dan *degree* = 0,01 [20].

3. METODOLOGI

Penelitian dilaksanakan pada tanggal 17 Desember 2022 terhadap ulasan pengguna aplikasi Pospay di Google Play Store dengan menggunakan bahasa pemrograman Python via Google Colaboratory. Penelitian ini mengimplementasikan metode *Knowledge Discovery in Database (KDD)* yang merupakan salah satu metode yang banyak digunakan dalam penelitian terkait *data mining*. *Knowledge Discovery in Database (KDD)* memiliki lima tahapan utama yaitu *data selection*, *preprocessing*, *transformation*, *data mining* dan *evaluation*. Berikut merupakan alur penelitian yang dilakukan.



Gambar 1. Alur Penelitian

3.1 Implementasi KDD

Semua tahap dalam implementasi KDD, kecuali pelabelan data, dilakukan menggunakan bahasa pemrograman Python via Google Colaboratory.

1. Data selection

Pada tahap ini, dilakukan penarikan data ulasan pengguna berbahasa Indonesia aplikasi Pospay di Google Play Store. Hasil data populasi diolah dengan teknik *proportionate stratified random sampling* sehingga menghasilkan data sampel. Strata yang dijadikan acuan *sampling* yaitu atribut *score (rating)* dan jumlah sampel yang diambil sebanyak 10% dari jumlah populasi. Hasil data sampel diberi label secara manual dan dipilih atribut yang relevan.

2. Preprocessing

Pada tahap ini, dilakukan *text preprocessing* terhadap atribut *content* (ulasan) yang terdiri dari:

- a. *Case folding*: mengubah huruf besar menjadi kecil. Contoh: 'Pospay sangat MANTAP' menjadi 'pospay sangat mantap'.
- b. *Cleaning*: tahap ini diterapkan menggunakan *library regex*. Proses yang dilakukan yaitu menghilangkan simbol dan tanda petik (#', '<.*@:!?; dan lain-lain), angka (0-9), memperbaiki duplikasi karakter beruntun lebih dari tiga (contoh: bagussss menjadi bagus) dan spasi berlebih.
- c. *Tokenize*: tahap ini diterapkan menggunakan *library NLTK*. Setiap kata dipisahkan pada string data. Contoh: 'sangat bagus' menjadi ['sangat', 'bagus'].

- d. *Normalize*: menormalisasi kata yang tidak sesuai ejaan bahasa Indonesia yang disempurnakan. Contoh: ‘dkt’ menjadi ‘dekat’, ‘jlk’ menjadi ‘jelek’ dan lain-lain.
 - e. *Filtering*: tahap ini diterapkan menggunakan *library* NLTK. Proses yang dilakukan pada tahap ini yaitu menyeleksi kata-kata bermakna dengan membuang kata-kata tidak bermakna. Contoh kata tidak bermakna: wkwk, di, pada, setelah, dan lain-lain.
 - f. *Stemming*: tahap ini diterapkan menggunakan *library* Sastrawi. Kata berimbuhan diubah kedalam bentuk dasar, contoh: ‘pendaftaran’ menjadi ‘daftar’.
3. *Transformation*
 Proses pembobotan kata diterapkan menggunakan TF-IDF dengan *library* Sklearn. Sebelum penerapan TF-IDF, data dibersihkan dari *missing value* untuk meningkatkan performa model.
4. *Data Mining*
 Pada tahap ini, data dibagi menjadi data latih dan data uji dengan skenario sebagai berikut.

Tabel 2. Skenario Pembagian Data

Skenario	Data Latih : Data Uji
Skenario 1	90 : 10
Skenario 2	80 : 20
Skenario 3	70 : 30

Kemudian *grid search* diterapkan pada masing-masing skenario pembagian data untuk memperoleh kernel dan parameter dengan performa terbaik. Berikut merupakan kernel dan parameter yang diuji coba menggunakan *grid search*.

Tabel 3. Kernel dan Parameter Uji Coba

Kernel	Linear, RBF, Polynomial.
Parameter C	0,1; 0,25; 0,5; 0,75; 1; 1,1; 1,25; 1,5; 1,75.
Parameter	1; 0,1; 0,01; 0,001;
Gamma (γ)	0,0001.

Parameter selain yang diujikan pada tabel 3 akan dibiarkan menggunakan nilai *default*. Kernel dan parameter dengan performa terbaik menurut hasil *grid search* akan diterapkan menggunakan algoritma Support Vector Machine (SVM) dengan *library* Sklearn.

5. *Evaluation*
 Pada tahap ini, model dari masing-masing skenario dievaluasi menggunakan *confusion matrix* dengan *library* Sklearn, Mathplotlib dan Seaborn agar diketahui skenario mana yang menghasilkan model dengan performa terbaik. Model dengan performa terbaik diekspor dalam format *.sav* menggunakan *library* Pickle.

3.2 Visualisasi Wordcloud

Wordcloud positif dan negatif dibuat berdasarkan teks hasil *preprocessing*. Library Python yang digunakan dalam membuat visualisasi Wordcloud yaitu *library* Wordcloud dan PIL.

4. HASIL DAN PEMBAHASAN

4.1. Implementasi KDD

Hasil implementasi KDD pada setiap tahapan yaitu sebagai berikut.

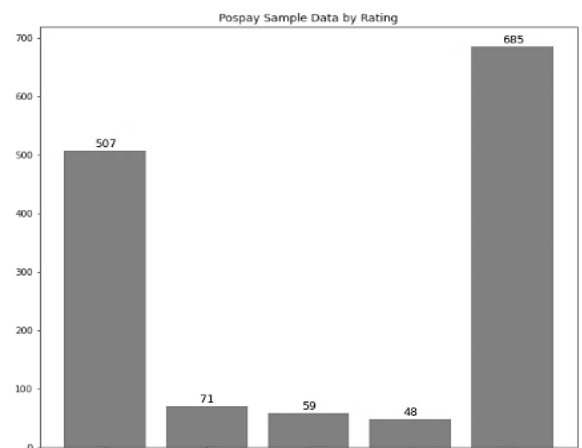
1. Data Selection

Hasil penarikan data ulasan berbahasa Indonesia aplikasi Pospay di Google Play Store yang dilaksanakan pada tanggal 17 Desember 2022 sebanyak 13.698 data relevan.



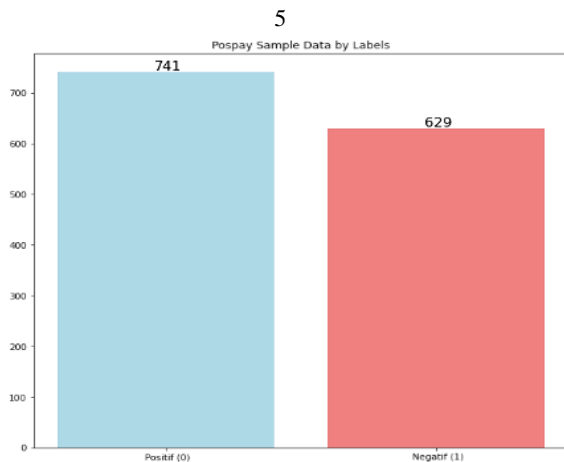
Gambar 2. Data Populasi

Proportionate stratified random sampling diterapkan berdasarkan strata *rating* 1-5 dengan masing-masing proporsi sebesar 10% dari populasi. Total data sampel yang diperoleh sebanyak 1.370 data dengan jumlah data setiap *rating* sebagai berikut.



Gambar 3. Jumlah Data Sampel Berdasarkan Rating

Sebanyak 1.370 data sampel diberi label 0 (positif) dan 1 (negatif) secara manual. Berikut merupakan perbandingan jumlah data berdasarkan label.



Gambar 4. Jumlah Data Sampel Berdasarkan Label

Data sampel memiliki 11 atribut yang terdiri dari *reviewId*, *username*, *userImage*, *content*, *score*, *thumbsUpCount*, *reviewCreatedVersion*, *at*, *replyContent*, *repliedAt* dan label. Atribut yang kurang relevan bagi penelitian ini dieliminasi sehingga tersisa 2 atribut yaitu *content* dan label. Berikut hasil akhir tahap *data selection*.

	content	label
0	Transaksi pending, dikomplen sudah 1 bulan bel...	1.0
1	Semakin di upgrade semakin turun sih kualitasn...	1.0
2	Sudah d instal kok g bisa kebuka	1.0
3	Kok gak bisah di bukak pos giro nya iya ? pada...	1.0
4	Sekelas BUMN kantor pos, bikin app ko gini ama...	1.0
...
1365	Terimakasih ,Cs fast respon jika ada kendala	0.0
1366	Lancar	0.0
1367	Bagus	0.0
1368	Layanan terbaik buat kantor pos terima kasih 🙏🙏🙏	0.0
1369	Mantap, pos giro emang membantu dan berkualitas	0.0

1370 rows × 2 columns

Gambar 5. Data Sampel

2. *Preprocessing*

Data sampel kemudian diolah melalui tahapan *text preprocessing* sehingga diperoleh hasil sebagai berikut.

a. *Case Folding*

Penerapan *case folding* terhadap data pada atribut *content* telah mengubah semua huruf besar (*uppercase*) menjadi huruf kecil (*lowercase*) dengan menggunakan fungsi `lower()`. Berikut merupakan hasil yang diperoleh dari proses *preprocessing* pertama yaitu *case folding*.

	content	label
0	transaksi pending, dikomplen sudah 1 bulan bel...	1.0
1	semakin di upgrade semakin turun sih kualitasn...	1.0
2	sudah d instal kok g bisa kebuka	1.0
3	kok gak bisah di bukak pos giro nya iya ? pada...	1.0
4	sekelas bumn kantor pos, bikin app ko gini ama...	1.0
...
1365	terimakasih ,cs fast respon jika ada kendala	0.0
1366	lancar	0.0
1367	bagus	0.0
1368	layanan terbaik buat kantor pos terima kasih 🙏🙏🙏	0.0
1369	mantap, pos giro emang membantu dan berkualitas	0.0

1370 rows × 2 columns

Gambar 6. Hasil *Case Folding*

b. *Cleaning*

Penerapan *cleaning* terhadap data pada atribut *content* telah menghilangkan simbol, puntuasi, angka, duplikasi karakter dan spasi berlebih diawal kalimat, akhir kalimat maupun diantara kata. Data yang telah melalui proses *cleaning* yaitu sebagai berikut.

	content	label
0	transaksi pending dikomplen sudah bulan belum ...	1.0
1	semakin di upgrade semakin turun sih kualitasn...	1.0
2	sudah d instal kok g bisa dibuka	1.0
3	kok gak bisa di buka pos giro nya iya padahal ...	1.0
4	Sekelas bumn kantor pos bikin app ko gini amat...	1.0
...
1365	terimakasih cs fast respon jika ada kendala	0.0
1366	lancar	0.0
1367	bagus	0.0
1368	layanan terbaik buat kantor pos terima kasih t...	0.0
1369	mantap pos giro emang membantu dan berkualitas	0.0

1370 rows × 2 columns

Gambar 7. Hasil *Cleaning*

c. *Tokenize*

Penerapan *tokenize* terhadap atribut *content* telah memisahkan setiap kata pada string data. Memisahkan setiap kata pada data bertujuan untuk mempermudah dan membuat proses *preprocessing* selanjutnya menjadi lebih mudah dan optimal. Berikut merupakan hasil dari proses *tokenize*.

	content	label
0	[transaksi, pending, dikomplen, sudah, bulan, ...	1.0
1	[semakin, di, upgrade, semakin, turun, sih, ku...	1.0
2	[sudah, d, instal, kok, g, bisa, dibuka]	1.0
3	[kok, gak, bisa, di, buka, pos, giro, nya, iya...	1.0
4	[Sekelas, bum, kantor, pos, bikin, app, ko, g...	1.0
...
1365	[terimakasih, cs, fast, respon, jika, ada, ken...	0.0
1366	[lancar]	0.0
1367	[bagus]	0.0
1368	[layanan, terbaik, buat, kantor, pos, terima, ...	0.0
1369	[mantap, pos, giro, emang, membantu, dan, berk...	0.0

1370 rows x 2 columns

Gambar 8. Hasil *Tokenize*

d. *Normalize*

Penerapan *normalize* terhadap data pada atribut *content* telah mengubah kata yang tidak sesuai ejaan bahasa Indonesia yang disempurnakan, kata bahasa asing dan kata *slang*. Beberapa kata yang telah diubah yaitu *pending* menjadi *tertunda*, *dikomplen* menjadi *dikomplain*, *emang* menjadi *memang*, dan sebagainya. Berikut merupakan hasil yang diperoleh dari tahap *normalize*.

	content	label
0	[transaksi, tertunda, dikomplain, sudah, bulan...	1.0
1	[semakin, di, meningkatkan, semakin, turun, si...	1.0
2	[sudah, di, install, kok, tidak, bisa, dibuka]	1.0
3	[kok, tidak, bisa, di, buka, pos, giro, nya, i...	1.0
4	[Sekelas, bum, kantor, pos, buat, aplikasi, k...	1.0
...
1365	[terima kasih, cs, cepat, tanggapan, jika, ada...	0.0
1366	[lancar]	0.0
1367	[bagus]	0.0
1368	[layanan, terbaik, buat, kantor, pos, terima, ...	0.0
1369	[mantap, pos, giro, memang, membantu, dan, ber...	0.0

1370 rows x 2 columns

Gambar 9. Hasil *Normalize*

e. *Filtering*

Penerapan *filtering* terhadap data pada atribut *content* telah menghilangkan kata yang kurang bermakna seperti kata 'sudah', 'di', 'kok', 'tidak', 'bisa', 'cs', 'buat', 'memang', 'dan', 'wkwk', 'jika', 'ada', dan sebagainya. Berikut hasil yang diperoleh dari proses *filtering*.

	content	label
0	[transaksi, tertunda, dikomplain, selesai, sel...	1.0
1	[meningkatkan, turun, kualitasnya, kencang, me...	1.0
2	[install, dibuka]	1.0
3	[buka, pos, giro, pos, giro, membantu]	1.0
4	[Sekelas, bum, kantor, pos, aplikasi, daftar,...	1.0
...
1365	[terima kasih, cepat, tanggapan, kendala]	0.0
1366	[lancar]	0.0
1367	[bagus]	0.0
1368	[layanan, terbaik, kantor, pos, terima, kasih,...	0.0
1369	[mantap, pos, giro, membantu, berkualitas]	0.0

1370 rows x 2 columns

Gambar 10. Hasil *Filtering*

f. *Stemming*

Penerapan *stemming* terhadap data pada atribut *content* telah mengubah setiap kata berimbuhan dari string data kedalam bentuk dasarnya. Beberapa kata yang telah diubah yaitu *dikomplain* menjadi *komplain*, *tanggapan* menjadi *tanggap*, dan sebagainya. Berikut hasil proses *stemming*.

	content	label
0	[transaksi, tunda, komplain, selesai, selesai]	1.0
1	[tingkat, turun, kualitas, kencang, menang, wa...	1.0
2	[install, buka]	1.0
3	[buka, pos, giro, pos, giro, bantu]	1.0
4	[kelas, bum, kantor, pos, aplikasi, daftar, m...	1.0
...
1365	[terima kasih, cepat, tanggap, kendala]	0.0
1366	[lancar]	0.0
1367	[bagus]	0.0
1368	[layan, baik, kantor, pos, terima, kasih, tang...	0.0
1369	[mantap, pos, giro, bantu, kualitas]	0.0

1370 rows x 2 columns

Gambar 11. Hasil *Stemming*

3. *Transformation*

Data yang telah melalui tahap *text preprocessing* kemudian dibersihkan dari *missing value* dan diterapkan pembobotan kata menggunakan TF-IDF, sehingga diperoleh matriks berukuran 1.295 x 971.

	abal	acara	acc	acung	ada	adil	administrator	adu	agen	ahli	...	wajah	wajib
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.000000	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.232287	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.000000	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.000000	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.000000	0.0
...
1290	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.000000	0.0
1291	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.000000	0.0
1292	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.000000	0.0
1293	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.000000	0.0
1294	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.000000	0.0

1295 rows x 971 columns

Gambar 12. Hasil Pembobotan Kata

4. *Data Mining*

Hasil pembobotan kata dibagi menjadi data latih dan data uji sesuai perbandingan pada masing-masing skenario, sehingga diperoleh hasil berikut.

Tabel 4. Hasil Pembagian Data

Skenario	Latih : Uji	Data Latih	Data uji
Skenario 1	90 : 10	1165	130
Skenario 2	80 : 20	1036	259
Skenario 3	70 : 30	906	389

Selanjutnya, kernel dan parameter uji coba diterapkan pada masing-masing skenario menggunakan *grid search*, sehingga diperoleh hasil berikut.

Tabel 5. Hasil *Grid Search*

Skenario	Kernel dan Parameter Terbaik
Skenario 1	Kernel: RBF, parameter C: 1, (90 : 10) parameter gamma (γ): 1.
Skenario 2	Kernel: Linear; parameter C: 1,25; (80 : 20) parameter gamma (γ): 1.
Skenario 3	Kernel: Linear; parameter C: 1, (70 : 30) parameter gamma (γ): 1.

Kernel dan parameter terbaik dari masing-masing skenario diimplementasikan menggunakan algoritma SVM, sehingga diperoleh hasil akurasi model sebagai berikut.

Tabel 6. Hasil Akurasi Model

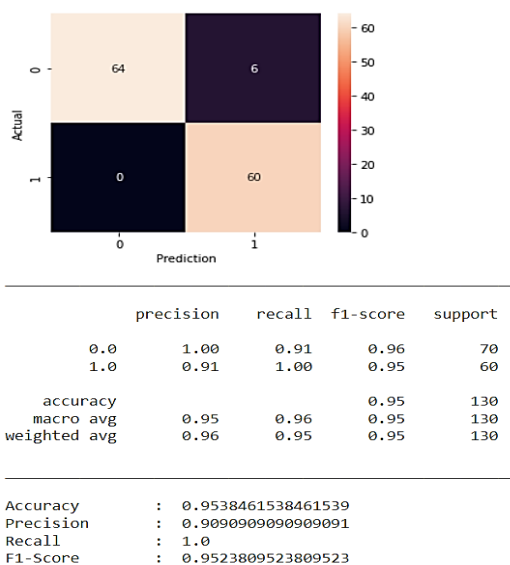
Skenario	Akurasi Model
Skenario 1	95%
Skenario 2	92%
Skenario 3	89%

5. *Evaluation*

Evaluasi model menggunakan *confusion matrix* menghasilkan *output* sebagai berikut.

a. Skenario 1

Berikut merupakan hasil performa model skenario 1.

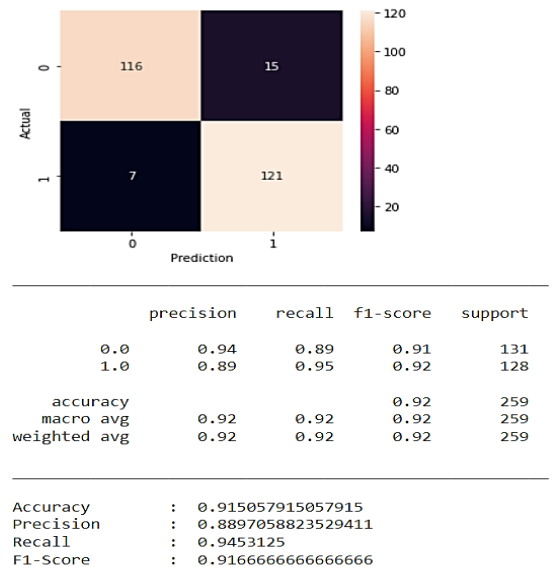


Gambar 13. Hasil Performa Model Skenario 1

Confusion matrix diatas memberi informasi data *true positif* sebanyak 64, *true negative* sebanyak 60, *false positif* sebanyak 6 dan *false negative* tidak ada. Berdasarkan evaluasi yang telah dilakukan, dapat diketahui bahwa total data yang tidak diprediksi dengan benar sejumlah 6 data dari total 130 data uji dan diperoleh nilai akurasi 95%, presisi 91%, recall 100% dan f1-score 95%.

b. Skenario 2

Berikut merupakan hasil performa model skenario 2.

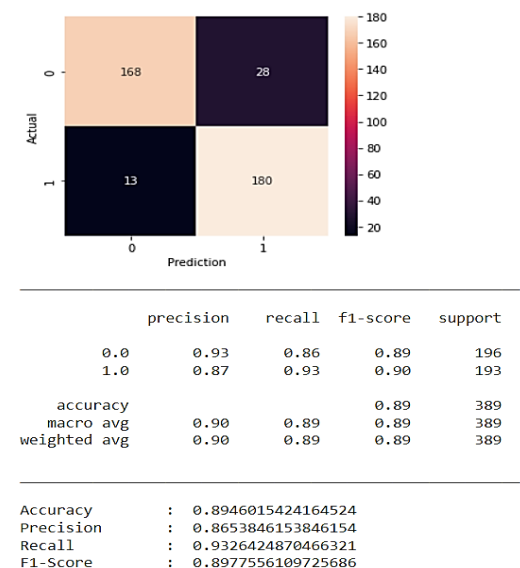


Gambar 14. Hasil Performa Model Skenario 2

Confusion matrix diatas memberi informasi data *true positif* sebanyak 116, *true negative* sebanyak 121, *false positif* sebanyak 15 dan *false negative* sebanyak 7. Berdasarkan evaluasi yang telah dilakukan, dapat diketahui bahwa total data yang tidak diprediksi dengan benar sejumlah 22 data dari total 259 data uji dan diperoleh nilai akurasi 92%, presisi 89%, recall 95% dan f1-score 92%.

c. Skenario 3

Berikut merupakan hasil performa model skenario 3.



Gambar 15. Hasil Performa Model Skenario 1

Confusion matrix diatas memberi informasi data *true positif* sebanyak 168, *true negative* sebanyak 180, *false positif* sebanyak 28 dan *false negative* sebanyak 13. Berdasarkan evaluasi yang telah dilakukan, dapat diketahui bahwa total data yang tidak diprediksi dengan benar sejumlah 41 data dari total 389 data uji dan diperoleh nilai akurasi 89%, presisi 87%, recall 93% dan f1-score 90%.

Berdasarkan hasil evaluasi model dari tiga skenario, maka dapat diperoleh model dengan performa terbaik yaitu model skenario 1. Model skenario 1 selanjutnya diekspor menggunakan *library* Pickle sehingga menghasilkan modelPospay.sav.

4.2. Visualisasi Wordcloud

Wordcloud yang dihasilkan terdiri dari wordcloud sentimen positif dan wordcloud sentimen negatif. Berikut merupakan tampilan wordcloud sentimen positif.



Gambar 16. Wordcloud Sentimen Positif

Lima kata dengan frekuensi kemunculan terbanyak pada wordcloud sentimen positif adalah kata ‘mantap’, ‘aplikasi’, ‘bayar’, ‘mudah’ dan ‘bagus’. Berikut merupakan tampilan wordcloud sentimen negatif.



Gambar 17. Wordcloud Sentimen Negatif

Lima kata dengan frekuensi kemunculan terbanyak pada wordcloud sentimen negatif adalah kata ‘aplikasi’, ‘masuk’, ‘pakai’, ‘salah’ dan ‘buka’.

5. KESIMPULAN DAN SARAN

Penelitian ini, menghasilkan kesimpulan bahwa sentimen pengguna aplikasi Pospay di Google Play Store cenderung positif namun tidak berpaut jauh dengan jumlah sentimen negatif. Secara spesifik, dari total data sampel sebanyak 1.370 data, ditemukan sentimen positif sebanyak 741 data (54,1%) dan sentimen negatif sebanyak 629 data (45,9%). Berdasarkan informasi tersebut, kinerja aplikasi dan layanan Pospay sudah cukup baik namun perlu dievaluasi dan ditingkatkan kembali oleh pihak Pos Indonesia selaku pengelola aplikasi Pospay agar kualitas aplikasi, layanan dan citra dalam masyarakat menjadi lebih baik lagi. Selain itu, dalam penelitian ini mengimplementasikan algoritma Support Vector Machine (SVM) dalam membuat model analisis sentimen ulasan pengguna aplikasi Pospay di Google Play Store. Dalam proses pembuatan model, untuk mendapatkan performa terbaik, *grid search* diterapkan dengan 3 skenario pembagian data serta kernel dan parameter uji coba sehingga diperoleh skenario terbaik yaitu skenario 1. Skenario 1 memiliki proporsi data latih : data uji yaitu 90 : 10, kernel = RBF, parameter C = 1 dan parameter gamma (γ) = 1. Performa model skenario 1 menggunakan SVM menghasilkan akurasi sebesar 95%, presisi sebesar 91%, recall sebesar 100% dan f1-score sebesar 95%. Algoritma SVM telah memberikan performa yang sangat baik dalam analisis sentimen ulasan aplikasi Pospay.

Saran pengembangan dari penelitian ini dimasa mendatang yaitu dengan mengimplementasikan algoritma maupun metode yang berbeda dengan yang telah diterapkan dalam penelitian ini. Selain itu, model analisis sentimen ulasan aplikasi Pospay diharapkan dapat diimplementasikan ke dalam sebuah sistem.

DAFTAR PUSTAKA

- [1] I. Martinelli, “Menilik Financial Technology dalam Bidang Perbankan,” *J. Sos. Hum. Komun.*, vol. 2, no. 1, pp. 32–43, 2021.
- [2] Google Play Store, “Google Play Store: Pospay.” <https://play.google.com/store/apps/details?id=com.posi.Indonesia.giropos&hl=id&gl=US> (accessed Dec. 20, 2022).
- [3] A. Nabillah, S. Alam, and M. G. Resmi, “Twitter User Sentiment Analysis Of TIX ID Applications Using Support Vector Machine Algorithm,” vol. 3, no. 1, pp. 14–27, 2022.
- [4] O. Irnawati and K. Solecha, “Analisis Sentimen Ulasan Aplikasi Flip Menggunakan Naïve Bayes dengan Seleksi Fitur PSO,” vol. 4, no. 02, pp. 189–199, 2022.
- [5] G. Sanjaya and K. M. Lhaksana, “Analisis Sentimen Komentar YouTube tentang Terpilihnya Menteri Kabinet Indonesia Maju Menggunakan Lexicon Based,” vol. 7, no. 3, pp. 9698–9710, 2020.
- [6] R. A. Saputra, D. Puspitasari, and T. Baidawi, “Deteksi Kematangan Buah Melon dengan Algoritma Support Vector Machine Berbasis Ekstraksi Fitur GLCM,” vol.

- 4, no. 2, 2022.
- [7] R. Wahyudi and G. Kusumawardhana, "Analisis Sentimen pada Aplikasi Grab di Google Play Store Menggunakan Support Vector Machine," *J. Inform.*, vol. 8, no. 2, pp. 200–207, 2021, doi: 10.31294/ji.v8i2.9681.
- [8] D. A. C. Rachman, R. Goejantoro, and F. D. T. Amijaya, "Implementasi Text Mining Pengelompokan Dokumen Skripsi Menggunakan Metode K-Means Clustering," *J. Eksponensial*, vol. 11, no. 2, pp. 167–174, 2020.
- [9] W. Widayat, "Analisis Sentimen Movie Review menggunakan Word2Vec dan metode LSTM Deep Learning," *J. Media Inform. Budidarma*, vol. 5, no. 3, p. 1018, 2021, doi: 10.30865/mib.v5i3.3111.
- [10] F. Romadoni, Y. Umidah, and B. N. Sari, "Text Mining Untuk Analisis Sentimen Pelanggan Terhadap Layanan Uang Elektronik Menggunakan Algoritma Support Vector Machine," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 9, no. 2, pp. 247–253, 2020, doi: 10.32736/sisfokom.v9i2.903.
- [11] D. Suprayogi and H. F. Pardede, "Support Vector Regression Dalam Prediksi Penurunan Jumlah Kasus Penderita Covid-19," *Jointecs (Journal Inf. Technol. Comput. Sci.)*, vol. 7, no. 2, pp. 63–70, 2022, doi: 10.31328/jointecs.v7i2.3687.
- [12] M. I. Gunawan, D. Sugiarto, and I. Mardianto, "Peningkatan Kinerja Akurasi Prediksi Penyakit Diabetes Mellitus Menggunakan Metode Grid Search pada Algoritma Logistic Regression," *J. Edukasi dan Penelit. Inform.*, vol. 6, no. 3, p. 280, 2020, doi: 10.26418/jp.v6i3.40718.
- [13] Yurisyah Maisyiroh, "Pengaruh Pelaksanaan WFH terhadap Burnout Karyawan dengan Work Family Conflict sebagai Variabel Intervening," *J. Ris. Manaj. dan Bisnis*, vol. 2, no. 1, pp. 47–54, 2022, doi: 10.29313/jrmb.v2i1.934.
- [14] I. Islamy, "Penelitian Survei dalam Pembelajaran dan Pengajaran Bahasa Inggris," *Japanese Soc. Biofeedback Res.*, vol. 19, no. 5, pp. 463–466, 2019.
- [15] L. Maharani Siniwi, A. Prahutama, and A. Rachman Hakim, "Query Expansion Ranking Pada Analisis Sentimen Menggunakan Klasifikasi Multinomial Naïve Bayes (Studi Kasus : Ulasan Aplikasi Shopee pada Hari Belanja Online Nasional 2020)," *J. Gaussian*, vol. 10, no. 3, pp. 377–387, 2021.
- [16] Irhamah, N. A. Rakhmawati, and H. Nurhadi, "Pengembangan Sistem Informasi Sederhana untuk Pengelolaan dan Pengolahan Data Tol Laut PT. PELNI (Persero) Cabang Surabaya," *Sewagati*, vol. 4, no. 2, p. 95, 2020, doi: 10.12962/j26139960.v4i2.6163.
- [17] N. M. Maghfur, F. Muhammad, and A. Voutama, "Analysis of the Relationship between Public Sentiment on Social Media and Indonesian Covid-19 Dynamics," *Systematics*, vol. 3, no. 3, pp. 336–345, 2021.
- [18] A. Rahman Isnain, A. Indra Sakti, D. Alita, and N. Satya Marga, "Sentimen Analisis Publik Terhadap Kebijakan Lockdown Pemerintah Jakarta Menggunakan Algoritma Svm," *Jdmsi*, vol. 2, no. 1, pp. 31–37, 2021, [Online]. Available: <https://t.co/NfhmfMjtXw>.
- [19] K. I. Ruslim, P. P. Adikara, and Indriati, "Analisis Sentimen Pada Ulasan Aplikasi Mobile Banking Menggunakan Metode Support Vector Machine dan Lexicon Based Features," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 7, pp. 6694–6702, 2019.
- [20] N. Hasanati, Q. Aini, and A. Nuri, "Implementation of Support Vector Machine with Lexicon Based for Sentiment Analysis on Twitter," in *2022 10th International Conference on Cyber and IT Service Management (CITSM)*, Sep. 2022, pp. 1–4, doi: 10.1109/CITSM56380.2022.9935887.

BIODATA PENULIS



Dea Safryda Putri

Mahasiswi Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang.



Taufik Ridwan, S.T., M.T.

Dosen Program Studi Sistem Informasi Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang dan Dosen Pengampu Mata Kuliah *Text Mining*.