

# Optimalisasi Klasifikasi *Support Vector Machine* dengan SMOTE: Studi Kasus Ulasan Pengguna Aplikasi Alfagift

## *Optimizing Support Vector Machine Classification with SMOTE: Case Study of Alfagift Application User Reviews*

Adhani Mulianti<sup>1)</sup>, Yulison Herry Chrisnanto<sup>2)</sup>, Herdi Ashaury<sup>3)</sup>

<sup>1,2,3</sup>Fakultas Sains dan Informatika, Program Studi Informatika, Universitas Jenderal Achmad Yani

<sup>1,2,3</sup>Jl. Terusan Jend. Sudirman, Cibeber, Kec. Cimahi Selatan, Kota Cimahi, Jawa Barat, 40531, Telp. 022-6656190, Indonesia

adhanimulianti20@if.unjani.ac.id<sup>1)</sup>, yhc@if.unjani.ac.id<sup>2)</sup>, hay@if.unjani.ac.id<sup>3)</sup>

Diterima: 26 April 2024 || Direvisi: 11 Mei 2024 || Disetujui: 21 Mei 2024

**Abstrak** – *Support Vector Machine (SVM)* adalah algoritma supervised learning yang bekerja dengan mengklasifikasi berdasarkan kelas yang mengacu pada pola hasil dari proses pelatihan. SVM memiliki beberapa kernel yang umum dan populer digunakan salah satunya adalah kernel linear. Kelemahan SVM adalah dalam “pemilihan parameter” dan performanya cenderung buruk pada kasus dataset yang tidak seimbang. Tujuan penelitian ini adalah untuk mengatasi kelemahan dari algoritma SVM dengan metode yang diusulkan. Penelitian ini menggunakan kernel linear dengan ekstraksi fiturnya yaitu Word2Vec dengan model Skip-gram, dan dalam menangani masalah ketidakseimbangan data menggunakan teknik SMOTE (oversampling). Hasil penelitian menunjukkan bahwa dataset yang tidak seimbang menghasilkan akurasi sebesar 90% dan dataset yang seimbang (SMOTE) menghasilkan akurasi sebesar 92%, sehingga teknik oversampling SMOTE terbukti meningkatkan hasil akurasinya sebesar 2%.

**Kata Kunci:** Kernel Linear, Skip-gram, SMOTE, Support Vector Machine (SVM), Word2Vec

**Abstract** – *Support Vector Machine (SVM)* is a supervised learning algorithm that works by classifying based on classes that refer to patterns resulting from the training process. SVM has several commonly and popularly used kernels, one of which is the linear kernel. The weakness of SVM is in the "parameter selection" and its performance tends to be poor in the case of unbalanced datasets. The purpose of this study is to overcome the weaknesses of the SVM algorithm with the proposed method. This research uses a linear kernel with feature extraction that is Word2Vec with Skip-gram model, and in handling the data imbalance problem using SMOTE (oversampling) technique. The results showed that the unbalanced dataset produced an accuracy of 90% and the balanced dataset (SMOTE) produced an accuracy of 92%, so the SMOTE oversampling technique was proven to increase the accuracy results by 2%.

**Keywords:** Kernel Linear, Skip-gram, SMOTE, Support Vector Machine (SVM), Word2Vec

## PENDAHULUAN

Alfamart adalah jaringan ritel modern dengan banyak gerai yang tersebar di seluruh Indonesia. Makanan, minuman, dan kebutuhan pokok merupakan barang yang sering untuk dibeli di Alfamart. Alfamart memiliki aplikasi bernama Alfagift, yang menyediakan berbagai pilihan produk melalui perangkat lunak digital (Octaviani et al., 2022). Alfagift adalah aplikasi belanja *online* yang dikembangkan oleh Alfamart. Alfamart merupakan toko yang menjual produk ritel atau eceran di Indonesia dengan melayani berbagai kelompok masyarakat. Per tanggal 27 Oktober 2023, aplikasi Alfagift memiliki jumlah unduhan melewati angka 10 juta dan mendapatkan *rating* 4,2 di Google Play. Di samping itu, terdapat 197.000 ribu ulasan pengguna di bagian komentar aplikasi Alfagift di situs web Google

Play. Komentar tekstual dan nilai *rating* merupakan dua unsur dari ulasan pengguna. Komentar tekstual dapat memberikan gambaran yang lebih komprehensif daripada nilai *rating*, yang menggunakan skala numerik untuk memberikan penilaian umum pengalaman pengguna. Ulasan pengguna di Google Play mencakup pujian, kritik, dan rekomendasi, serta komentar yang menguntungkan (Wahyudi & Kusumawardana, 2021).

Penelitian terdahulu tentang analisis sentimen terhadap ulasan film menggunakan metode SVM dengan kernel *linear* memiliki keunggulan dalam mekanisme klasifikasi teks seperti data teks dipisahkan secara linear atau garis lurus, kernel banyak mendukung fitur (kata) dalam teks jika dibandingkan dengan kernel yang lain, dan menggunakan library

LibLinear, kernel *linear* dapat melatih model SVM lebih cepat dibandingkan kernel lainnya. Perbandingan hasil akurasi dari algoritma Support Vector Machine (SVM), Decision Tree (DT), Neural Network (NN), Random Forest (RF), dan Naive Bayes (NB) menunjukkan hasil 87,56% untuk metode SVM dengan kernel *linear* dan merupakan hasil yang paling tinggi dibandingkan algoritma lainnya (Sheik Abdullah et al., 2021). Salah satu kelemahan dari algoritma SVM adalah dalam pemilihan parameter. Pemilihan parameter yang tepat dengan menggunakan Grid Search atau Random Search belum dilakukan pada penelitian tersebut, sehingga menjadi celah penelitian untuk algoritma SVM (Cervantes et al., 2020). Penelitian terdahulu tentang analisis sentimen yang membandingkan dua metode NB dan SVM dengan mengatur parameter menggunakan Grid Search. Setelah disesuaikan parameternya algoritma SVM menghasilkan akurasi yang lebih baik yaitu sebesar 85,65% dibandingkan dengan algoritma NB yaitu sebesar 68,70%. Hal ini menjadi landasan penggunaan Grid Search untuk pemilihan parameternya (Chong & Shah, 2022). Penelitian terdahulu terkait analisis sentimen yang membandingkan beberapa metode, yaitu NB, SVM, DT, dan RF menunjukkan bahwa algoritma SVM dengan kernel *linear* menghasilkan tingkat akurasi tertinggi sebesar 89%, dibandingkan dengan kernel RBF yang hanya mencapai 73% (Guia et al., 2019). Penelitian terdahulu terkait analisis sentimen telah melakukan perbandingan antara ekstraksi fitur TF-IDF dan Word2Vec. Hasilnya menunjukkan bahwa Word2Vec menghasilkan akurasi klasifikasi sentimen yang lebih tinggi daripada TF-IDF. Hal tersebut menjadi celah penelitian ini untuk menggunakan Word2Vec sebagai metode ekstraksi fitur (Sohrabi & Hemmatian, 2019).

Masalah ketidakseimbangan kelas dapat diatasi dengan menggunakan pendekatan *Synthetic Minority Oversampling Technique (SMOTE)*. SMOTE meningkatkan kinerja algoritma klasifikasi dengan menghasilkan data sintesis baru dari kelas minoritas (Anam et al., 2023). Pada penelitian ini data yang diperoleh dari *web scraping* dari pustaka *google-play-scraper* berjumlah 3.394 ribu data mentah. *Dataset* ini akan diberi 2 *class* data yaitu *positive* dan *negative* dan memiliki 4 atribut (fitur) yaitu *username*, *score*, *date*, dan *content*. *Dataset* penelitian ini memiliki atribut "*score*" yang dapat mewakili sentimen ulasan pengguna. Ulasan dengan skor 5 dan 4 dikategorikan sebagai positif, skor 3 dapat mengandung campuran

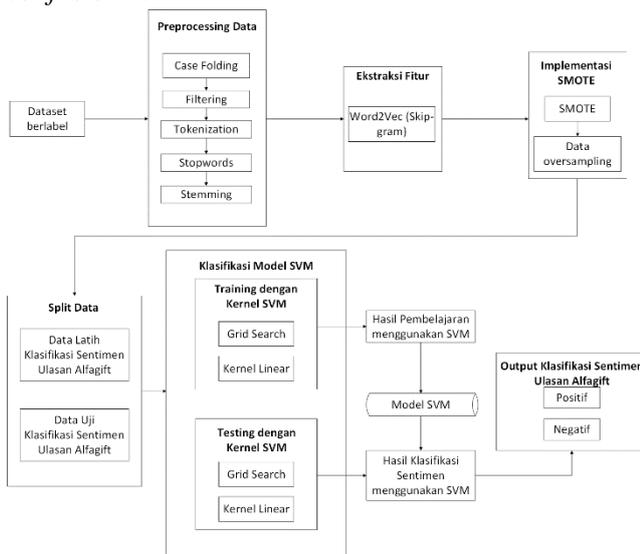
sentimen positif dan negatif, dan skor 2 dan 1 dikategorikan sebagai negatif. Hipotesis awal adalah dataset tersebut tidak seimbang, dengan proporsi tanggapan positif sebesar 63,8% dan tanggapan negatif 36,2% berdasarkan skor ulasan pengguna. Penelitian terdahulu yang melakukan survei mendalam pada Algoritma SVM memiliki kelemahan utama pada performa klasifikasinya ketika dihadapkan dengan dataset yang tidak seimbang. Algoritma ini cenderung bias terhadap kelas mayoritas dan mengabaikan kelas minoritas, sehingga menghasilkan *margin* yang bias dan menurunkan kinerja pada kelas minoritas. Teknik *oversampling*, seperti SMOTE, dapat diterapkan untuk mengatasi kelemahan ini (Cervantes et al., 2020).

Penelitian ini bertujuan untuk menguji kinerja algoritma SVM dengan kernel *linear* dan menggunakan ekstraksi fitur Word2Vec dengan model Skip-gram untuk klasifikasi sentimen. Dalam upaya meningkatkan akurasi SVM, penelitian ini menerapkan strategi pemilihan parameter yang optimal untuk kernel *linear*, serta menggunakan teknik *oversampling* SMOTE untuk menangani ketidakseimbangan *dataset*. Tujuannya adalah untuk menghasilkan model SVM yang memiliki performa klasifikasi tinggi, mengurangi risiko *overfitting*, dan meningkatkan kemampuan generalisasi SVM. Dengan demikian, penelitian ini mengkaji potensi penggunaan SVM dalam konteks klasifikasi sentimen dengan memperhatikan teknik pengoptimalan parameter dan penanganan ketidakseimbangan *dataset*.

Dalam konteks mengatasi kelemahan dari algoritma SVM, pengujian kinerja SVM menjadi penting untuk memaksimalkan hasil kinerja SVM. Berdasarkan temuan penelitian terdahulu terdapat dua kelemahan utama SVM yaitu: pemilihan parameter dan kinerja SVM dalam menghadapi *dataset* yang tidak seimbang. Solusi untuk mengatasi kelemahan pertama, pemilihan parameter pada kernel *linear* yaitu pemilihan parameter C dengan metode Grid Search. Peran dari Grid Search digunakan untuk mencari parameter C terbaik pada kernel *linear* SVM. Selain itu, untuk mengatasi kelemahan kedua, penanganan ketidakseimbangan dataset dengan menerapkan teknik *oversampling* SMOTE. SMOTE digunakan untuk menyeimbangkan *dataset* sehingga diharapkan dapat meningkatkan kinerja model SVM.

## METODE PENELITIAN

Penelitian ini dilakukan dengan tahapan sebagai berikut: Pertama, pengumpulan *dataset* dilakukan dengan proses *scraping data* menggunakan *library google-play-scrapers*, dan dilakukan proses pelabelan *dataset* berbasis leksikon dengan menggunakan TextBlob. Kedua, proses selanjutnya adalah *preprocessing* data. Ketiga melakukan ekstraksi fitur menggunakan Word2Vec dengan model Skip-gram. Keempat, implementasi SMOTE untuk menanganikan *dataset* yang tidak seimbang, serta *splitting* data menjadi *training* data dan *testing* data. Kelima, proses pelatihan dan pengujian algoritma SVM menggunakan kernel *linear* dan mengevaluasi model menggunakan *confusion matrix*.



Gambar 1 Metode Penelitian

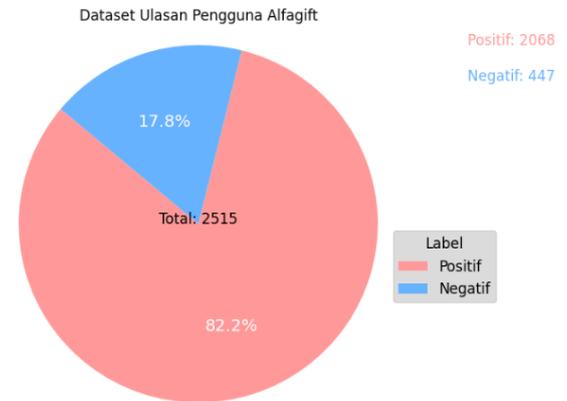
### Pengumpulan dataset

Data ulasan pengguna aplikasi Alfagift dikumpulkan dari pustaka *google-play-scrapers*. Data dikumpulkan untuk menyortir ulasan pengguna yang paling relevan dari situs Google Play. Jumlah dataset mentah yang dikumpulkan sebanyak 3.394 ulasan pengguna, memiliki rentang waktu dari bulan April 2020 sampai Oktober 2023. Setelah memperoleh *dataset*, kemudian terdapat proses pelabelan data yang dilakukan dengan pendekatan berbasis leksikon, yaitu dengan menggunakan TextBlob.

TextBlob adalah teknik berbasis leksikon yang dapat digunakan untuk berbagai tugas NLP termasuk penandaan bagian dari ucapan, analisis sentimen, ekstraksi frasa kata benda, parafrase, dan penyortiran, dll. Penelitian yang dilakukan menunjukkan bahwa Textblob lebih unggul dalam pelabelan dibandingkan dengan VADER dan SentiWordNet, sehingga

penelitian ini menggunakan TextBlob untuk proses pelabelan *datasetnya* (Mujahid et al., 2021).

Setelah proses pelabelan menggunakan Textblob, total jumlah data ulasan Alfagift menjadi 2.515 data, untuk data berlabel positif berjumlah 2068 atau sekitar 82,2% dan untuk data berlabel negatif berjumlah 447 atau sekitar 17,8%.



Gambar 2 Rasio jumlah dataset ulasan pengguna Alfagift

### Preprocessing data

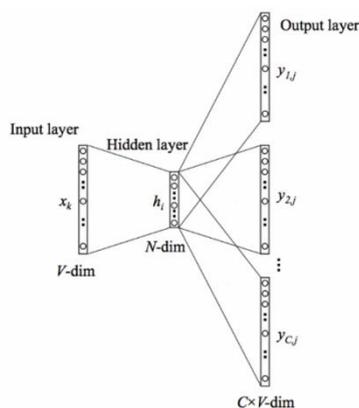
Hasil dari *web scraping* yang dilakukan tentu menghasilkan data mentah yang belum terstruktur dengan baik. Maka dari itu, perlu dilakukan tahapan *preprocessing* data, agar *dataset* siap untuk digunakan dalam klasifikasi model SVM. Pada tahapan *preprocessing* adalah sebagai berikut:

1. *Case folding*: Huruf semua kata dalam teks diubah menjadi huruf kecil untuk membentuk format standar.
2. *Filtering*: Proses pembersihan data melibatkan penghilangan istilah-istilah seperti sebutan, tautan, *emoticon*, tagar, dan tanda baca.
3. *Tokenization*: Tokenisasi adalah proses memecah kalimat menjadi kata-kata individual dengan menggunakan pemisah seperti koma dan titik.
4. *Stopwords*: Kata-kata seperti "yang", "untuk", "mereka", "untuk", dan kata kunci berlebihan lainnya yang sering muncul dapat dihilangkan dengan menggunakan proses penghapusan kata.
5. *Stemming*: Proses pemecahan kalimat menjadi elemen-elemen yang paling mendasar disebut frasa.

### Ekstraksi fitur

Model Word2Vec adalah pendekatan representasi vektor kata yang dapat menghasilkan hasil terbaik dalam NLP. Sebuah *neural network* digunakan untuk menghitung arsitektur model Word2Vec, dengan ruang vektor berfungsi sebagai output dan badan teks berfungsi sebagai input. Vektor kata yang muncul

adalah vektor ruang berdimensi rendah yang merangkum makna semantik dari kata tersebut. Model arsitektur Word2Vec terdiri dari dua jenis: Model Skip-Gram dan model Continuous Bag of Words (CBOW) (Nawangari et al., 2019).



**Gambar 3** Model Arsitektur Skip-gram

Model Word2Vec memiliki dua teknik pengukuran yaitu: Negative Sampling dan Hierarchical Softmax. Negative Sampling tidak sekompleks Hierarchical Softmax karena hanya melakukan pembaruan sampel dari beberapa kata output sebagai sampel negatif. Hierarchical Softmax menggunakan representasi pohon biner dari layer output dengan kata-kata sebagai *leaf* dan setiap *node*, secara eksplisit merepresentasikan kemungkinan relatif dari *node* turunannya.

### Implementasi SMOTE dan Split data

Proses penyeimbangan data diperlukan untuk mengatasi kelemahan algoritma SVM dalam menangani *dataset* yang belum seimbang. Solusi untuk mengatasi data yang belum seimbang adalah menggunakan metode SMOTE, salah satu pendekatan *oversampling* yang digunakan dalam *machine learning* untuk menganalisis data yang tidak seimbang. Pendekatan ini menciptakan sampel sintesis dari kelas minoritas dalam upaya untuk mengatasi masalah ketidakseimbangan kelas (Anam et al., 2023). *Split* data dilakukan dengan proporsi pembagian antara data pelatihan dan data pengujian pada penelitian ini yaitu sebesar 80:20 (Styawati et al., 2022).

### Klasifikasi dan Evaluasi Model SVM

Metode SVM adalah pengklasifikasi yang diawasi, *dataset* harus dibagi menjadi dua set untuk pengujian dan pelatihan. Teknik pelatihan membutuhkan pelabelan data terlebih dahulu. Label kategorisasi sentimen terdiri dari positif dan negatif. Setelah prosedur pelabelan data, setiap frasa dipelajari dengan

label yang sebelumnya telah ditetapkan selama fase pelatihan data. Selanjutnya, SVM mencoba untuk menarik garis yang dikenal sebagai *hyperplane* untuk membagi teks ke dalam kategori sentimen positif dan negatif (Kurniawan & Maharani, 2020). Pada penelitian ini menggunakan kernel *linear*. Kernel *linear* merupakan kernel dasar karena menyediakan pengukuran kesamaan *linear* antara dua vektor.

*Confusion Matrix* adalah teknik untuk menilai kinerja suatu model klasifikasi. Terdapat empat kategori yaitu *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)* yang mewakili empat hasil klasifikasi dari model. TP adalah kondisi di mana nilai sebenarnya positif, lalu diprediksi sebagai positif (akurat). TN adalah kondisi di mana nilai sebenarnya negatif, lalu diprediksi sebagai negatif (akurat). FP adalah kondisi di mana nilai sebenarnya negatif, lalu diprediksi sebagai positif (keliru). FN adalah kondisi di mana nilai sebenarnya positif, lalu diprediksi sebagai negatif (keliru) (Dey et al., 2020):

*Accuracy* adalah persentase dari jumlah prediksi yang benar (TP dan TN) dan total jumlah hasil prediksi (TP, TN, FP, dan FN). Persamaan ke 1 adalah rumus dari perhitungan *accuracy*:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

*Precision* adalah rasio dari jumlah TP dengan total prediksi positif (TP dan FP). Persamaan ke 2 adalah rumus dari perhitungan *precision*:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

*Recall* adalah rasio dari jumlah TP dengan total jumlah sampel positif yang sebenarnya (TP dan FN). Persamaan ke 3 adalah rumus dari perhitungan *recall*:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

F1 score adalah rata-rata harmonik dari *precision* dan *recall*. Persamaan ke 4 adalah rumus dari perhitungan f1-score:

$$F1 \text{ score} = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (4)$$

### Analisis dan Perancangan

Pada tahap analisis dan perancangan menjabarkan hasil analisis sesuai dengan metode penelitian yaitu: pengumpulan *dataset*, *preprocessing* data, ekstraksi fitur menggunakan Word2Vec dengan model Skip-gram, implementasi teknik *oversampling* SMOTE,

serta *splitting* data, dan klasifikasi & evaluasi model SVM.

**Pengumpulan dataset**

Total *dataset* mentah yang diperoleh dari *scraping* data berjumlah 3.394 data, kemudian dilakukan proses pelabelan *dataset* menggunakan *library* Textblob. Hasilnya *dataset* menjadi 2.515 data dengan label positif berjumlah 2.068 data atau 82,2% dan label negatif berjumlah 447 data atau 17,8%. Kemudian sampel *record* dari *dataset* ulasan pengguna Alfagift ditunjukkan pada Gambar 4:

1	content	label
2	saya memberi bintang pada alfagift karena selalu memberikan pelayanan terbaik bagi konsumennya	positive
3	alfagift sangat membantu dan memudahkan ketika harus belanja atau mau jajan tapi males ke toko	positive
4	enaknya bisa belanja untuk diantar ke rumah tapi untuk beberapa hal seperti membeli deposit mine	positive
5	kecewa saya menukar voucher stok di toko kosong yang akhirnya hangus tidak ada gunanya	negative
6	aplikasi alfagift sangat membantu memudahkan saya dalam berbelanja kebutuhan bulanan dan kebu	positive
7	kamu berhak mendapatkan bintang belanja melalui alfagift jadi lebih mudah kamu tidak perlu keluar	positive
8	kurir kabupaten tangerang terlalu malas menunggu lebih dari dua jam dan tidak kunjung sampai pad	positive
9	saya suka belanja di alfagift banyak promo dan hadiah menarik juga banyak pilihan menu mulai dari	positive
10	siaapa yang maklum kalau belanja aku suka promo apalagi promo jsm bagus sekali jos yang suka ngasi	positive
11	aplikasi belanja online terbaik yang pernah ada selain gratis ongkir pengirimannya juga satu set alias	positive

**Gambar 4** Sampel *record dataset* ulasan pengguna Alfagift

**Preprocessing data**

Langkah pertama adalah melakukan proses *case folding* atau membuat semua karakter menjadi huruf kecil. Contoh *case folding* ditunjukkan pada Tabel 1:

**Tabel 1** Proses *Case Folding*

Content	Case Folding
Cukup praktis belanja disini tidak perlu repot ke toko free ongkir sangat bermanfaat.	cukup praktis belanja disini tidak perlu repot ke toko free ongkir sangat bermanfaat.
Apk apa ini saya pesan COD tapi tidak datang dibatalkan tidak berfungsi apknya jelek percuma, apknya menipu orang lain anda harus bisa dipercaya jangan curang seperti ini.	apk apa ini saya pesan cod tapi tidak datang dibatalkan tidak berfungsi apknya jelek percuma, apknya menipu orang lain anda harus bisa dipercaya jangan curang seperti ini.

Langkah kedua adalah melakukan proses *filtering* atau menghapus seperti sebutan, tautan, *emoticon*, tagar, dan tanda baca. Contoh *filtering* ditunjukkan pada Tabel 2:

**Tabel 2** Proses *Filtering*

Case Folding	Filtering
cukup praktis belanja disini tidak perlu repot ke toko free ongkir sangat bermanfaat.	cukup praktis belanja disini tidak perlu repot ke toko free ongkir sangat bermanfaat
apk apa ini saya pesan cod tapi tidak datang dibatalkan tidak berfungsi apknya jelek percuma, apknya menipu orang lain anda	apk apa ini saya pesan cod tapi tidak datang dibatalkan tidak berfungsi apknya jelek percuma apknya menipu orang lain anda

harus bisa dipercaya jangan curang seperti ini. harus bisa dipercaya jangan curang seperti ini

Langkah ketiga adalah melakukan proses *tokenization* atau memecah kalimat menjadi *token-token*. Contoh *tokenization* ditunjukkan pada Tabel 3:

**Tabel 3** Proses *Tokenization*

Filtering	Tokenization
cukup praktis belanja disini tidak perlu repot ke toko free ongkir sangat bermanfaat	['cukup', 'praktis', 'belanja', 'disini', 'tidak', 'perlu', 'repot', 'ke', 'toko', 'free', 'ongkir', 'sangat', 'bermanfaat']
apk apa ini saya pesan cod tapi tidak datang dibatalkan tidak berfungsi apknya jelek percuma apknya menipu orang lain anda harus bisa dipercaya jangan curang seperti ini	['apk', 'apa', 'ini', 'saya', 'pesan', 'cod', 'tapi', 'tidak', 'datang', 'dibatalkan', 'tidak', 'berfungsi', 'apknya', 'menipu', 'orang', 'lain', 'kali', 'anda', 'harus', 'dipercaya', 'jangan', 'curang', 'seperti', 'ini']

Langkah keempat adalah melakukan proses *stopwords* atau menghapus kata dalam bahasa Indonesia yang sering muncul. Contoh *stopwords* ditunjukkan pada Tabel 4:

**Tabel 4** Proses *Stopwords*

Tokenization	Stopwords
['cukup', 'praktis', 'belanja', 'disini', 'tidak', 'perlu', 'repot', 'ke', 'toko', 'free', 'ongkir', 'sangat', 'bermanfaat']	['praktis', 'belanja', 'repot', 'toko', 'free', 'ongkir', 'bermanfaat']
['apk', 'apa', 'ini', 'saya', 'pesan', 'cod', 'tapi', 'tidak', 'datang', 'dibatalkan', 'tidak', 'berfungsi', 'apknya', 'menipu', 'orang', 'lain', 'kali', 'anda', 'harus', 'dipercaya', 'jangan', 'curang', 'seperti', 'ini']	['apk', 'pesan', 'cod', 'dibatalkan', 'berfungsi', 'menipu', 'orang', 'dipercaya', 'curang']

Langkah kelima adalah melakukan proses *stemming* atau mengubah setiap kata menjadi kata dasar. Contoh *stemming* ditunjukkan pada Tabel 5:

**Tabel 5** Proses *Stemming*

Stopwords	Stemming
['praktis', 'belanja', 'repot', 'toko', 'free', 'ongkir', 'bermanfaat']	praktis belanja repot toko free ongkir manfaat
['apk', 'pesan', 'cod', 'dibatalkan', 'berfungsi', 'menipu', 'orang', 'dipercaya', 'curang']	apk pesan cod batal fungsi tipu orang percaya curang

**Ekstraksi fitur**

Tahap selanjutnya data telah dilabeli dan diproses, kemudian diekstrak fiturnya menggunakan model Skip-gram. Pada tahap ini data yang semula masih berbentuk teks diubah menjadi sebuah vektor kata.

### Implementasi SMOTE dan *Split* data

Tahap berikutnya adalah menyeimbangkan *dataset* dengan menerapkan teknik SMOTE. Metode *oversampling* digunakan untuk meningkatkan jumlah sampel pada kelas minoritas sehingga sama dengan kelas mayoritas. Kemudian dilakukan *split* data dengan rasio 80:20.

### Klasifikasi dan Evaluasi Model SVM

Algoritma SVM memerlukan data pelatihan untuk membangun model pembelajaran mesin dan data pengujian untuk menilai keefektifan model pembelajaran mesin yang telah dibangun. Proses klasifikasi dengan metode SVM pada penelitian yang dilakukan yaitu menggunakan kernel *linear*. Rancangan dari SVM dengan kernel *linear* dalam memprediksi label menjadi positif dan negatif.

Hasil klasifikasi dengan metode SVM perlu untuk dilakukan evaluasi model, dari penelitian yang dilakukan teknik evaluasi model menggunakan *confusion matrix*. Pada penelitian ini menggunakan empat ukuran evaluasi yaitu: *accuracy*, *precision*, *recall*, dan *f1-score*.

## HASIL DAN PEMBAHASAN

Hasil dari penelitian yang dilakukan menggunakan metode klasifikasi SVM (kernel *linear*) dengan ekstraksi fitur Word2Vec dengan model Skip-gram, serta mengimplementasikan pemilihan parameter menggunakan Grid Search, dan menangani ketidakseimbangan dataset menggunakan teknik *oversampling* SMOTE adalah sebagai berikut ini.

### Klasifikasi dan Evaluasi Model SVM

Proses klasifikasi SVM terbagi menjadi 2 tahap yaitu: klasifikasi dengan *dataset* yang tidak seimbang dan *dataset* yang sudah seimbang dengan menerapkan teknik *oversampling* SMOTE. Kemudian terdapat hasil akurasi dari *dataset* yang tidak seimbang ditunjukkan pada Gambar 5:

```
Accuracy with linear kernel (training data): 0.922962226640159
Accuracy with linear kernel (test data): 0.9005964214711729
Precision with linear kernel: 0.9380952380952381
Recall with linear kernel: 0.9425837320574163
F1-score with linear kernel: 0.9403341288782817
Best parameters: {'C': 10000}
Classification Report:

```

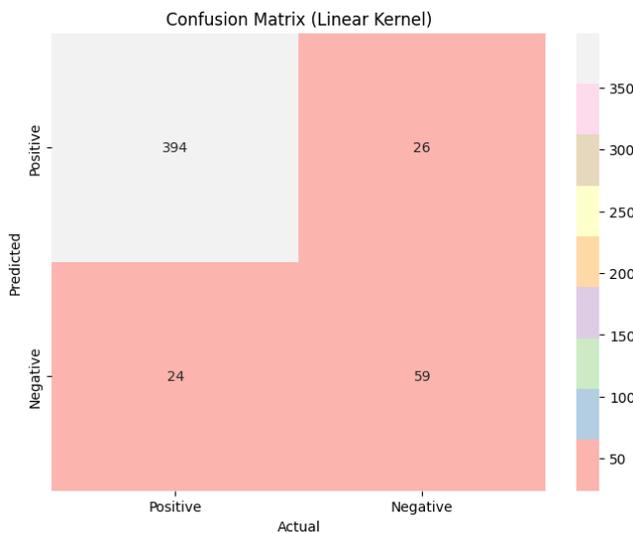
	precision	recall	f1-score	support
negative	0.71	0.69	0.70	85
positive	0.94	0.94	0.94	418
accuracy			0.90	503
macro avg	0.82	0.82	0.82	503
weighted avg	0.90	0.90	0.90	503

**Gambar 5** Hasil Akurasi *Dataset* Tidak Seimbang

Nilai *support* atau jumlah total data *testing* sebanyak 503. Total dataset berjumlah 2.515 data dari *dataset* Non-SMOTE atau *dataset* yang tidak seimbang. Kemudian terdapat pembagian atau *splitting* data sebesar 80:20, sehingga dari total data 2.515 jumlah data *training* sebanyak 2.012 data dan data *testing* sebanyak 503 data. Maka total nilai *support* atau data *testing* sebanyak 503 data.

Nilai akurasi untuk proses pelatihan diperoleh sebesar 92%. Kemudian terdapat nilai *precision* sebesar 94%. *Precision* adalah seberapa baik model dalam mengklasifikasi nilai data positif. Dengan nilai *precision* sebesar 94% artinya menunjukkan bahwa model memiliki ketelitian yang tinggi dalam mengidentifikasi nilai data positif dan 6% adalah yang diprediksi sebagai *false positive*. Selanjutnya nilai *recall* sebesar 94%. *Recall* merupakan seberapa baik model menemukan nilai data positif. Dengan nilai *recall* sebesar 94% adalah model mampu menangkap nilai positif secara akurat dan 6% yang diprediksi sebagai *false negative*. Selain itu nilai *f1-score* merupakan rata-rata dari nilai *precision* dan *recall* atau sebagai acuan keseimbangan dari nilai *precision* dan *recall*. Nilai F1-Score sebesar 94%, sehingga model telah menunjukkan keseimbangan yang sangat baik dalam kinerja memprediksi nilai data positif dan negatif.

Hasil akurasi prediksi dengan kernel *linear* diperoleh sebesar 90% dengan parameter "C" terbaiknya pada nilai : 10000, dalam pencarian parameter terbaik, jarak nilai parameter "C" : [0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000] dengan teknik Grid Search digunakan untuk mencari kombinasi terbaik dari nilai parameter di atas yang akan memberikan kinerja yang paling baik, lalu diperoleh parameter terbaiknya yaitu pada nilai : 10000. Dengan hasil akurasi sebesar 90%, artinya model SVM menunjukkan tingkat akurasi yang tergolong tinggi dan mampu mengklasifikasikan sentimen positif dan negatif dengan sangat baik. Evaluasi hasilnya dapat dilihat pada Gambar 6 yaitu Confusion Matrix:



**Gambar 6** Confusion Matrix dataset tidak seimbang

Nilai *true positive (TP)* sebanyak 394 dan *false negative (FN)* sebanyak 24,  $394 + 24 = 418$  data *positive*. Kemudian nilai *true negative (TN)* sebanyak 59 dan *false positive (FP)* sebanyak 26,  $59 + 26 = 85$  data *negative*. Nilai TP sebanyak 394 data bahwa model berhasil dalam memprediksi nilai positif secara signifikan artinya model sudah sangat baik. Nilai FN sebanyak 24 data bahwa model gagal dalam memprediksi data positif, tetapi jumlahnya sedikit sehingga model sudah sangat baik. Nilai TN sebanyak 59 data bahwa model cukup berhasil dalam memprediksi nilai negatif. Nilai FP sebanyak 26 data bahwa model gagal dalam memprediksi data negatif, dan jumlahnya cukup banyak artinya model cukup baik dalam mengklasifikasikan data negatif. Secara keseluruhan model sangat baik dalam memprediksi data positif, tetapi cukup baik dalam memprediksi data negatif.

Implementasi teknik *oversampling SMOTE* diterapkan dengan *sampling strategy = 0.8*. Kemudian terdapat hasil akurasi *dataset* yang seimbang (*SMOTE*) ditunjukkan pada Gambar 7:

```

Accuracy with linear kernel (training data): 0.9203896540141082
Accuracy with linear kernel (test data): 0.9181208053691275
Precision with linear kernel: 0.9664082687338501
Recall with linear kernel: 0.8862559241706162
F1-score with linear kernel: 0.9245982694684797
Best parameters: {'C': 10000}
Classification Report:

```

	precision	recall	f1-score	support
negative	0.87	0.96	0.91	323
positive	0.97	0.89	0.92	422
accuracy			0.92	745
macro avg	0.92	0.92	0.92	745
weighted avg	0.92	0.92	0.92	745

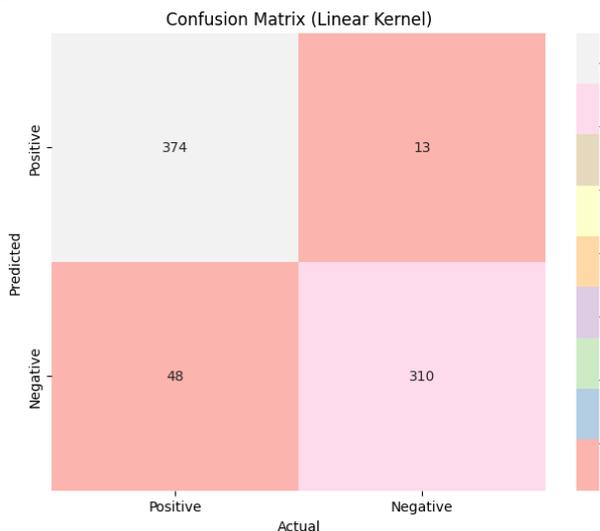
**Gambar 7** Hasil Akurasi *Dataset* Seimbang (*SMOTE*)

*Oversampling SMOTE* adalah teknik di mana menyeimbangkan *dataset* dengan meningkatkan kelas minoritas agar seimbang dengan kelas mayoritas. Kemudian untuk nilai *support* atau data *testing* sebanyak 745. Total *dataset* terdapat 2.515 data dengan kelas positif sebanyak 2.068 data dan kelas negatif sebanyak 447 data. Selanjutnya diterapkan teknik *oversampling SMOTE* dengan *sampling strategy = 0.8*, maka nilai data positif (kelas mayoritas)  $2.068 * 0.8 = 1.654$ . Hasil dari nilai  $1.654 - 447$  (nilai data negatif) = 1.207. Setelah menerapkan *oversampling SMOTE*, nilai data negatif yang bertambah sebanyak 1.207 data negatif. Total data negatif menjadi  $1.207 + 447 = 1.654$  data negatif. Terakhir, nilai *support* sebelumnya atau non-*SMOTE* adalah sebanyak 503, tetapi pada *dataset* yang telah seimbang menjadi 745, yaitu diperoleh dari total data positif + total data negatif yang telah diterapkan *SMOTE* sebanyak 2.068 data + 1.654 data = 3.722 data. Setelah itu membagi atau proses *splitting* data dengan rasio 80:20, maka data pelatihan sebanyak 2.977 data dan data *testing* sebanyak 745 data, sehingga nilai *support*nya menjadi 745 data.

Nilai akurasi proses pelatihan yaitu sebesar 92%. Selanjutnya nilai *precision* sebesar 97%, artinya model yang menerapkan *SMOTE* ini sudah sangat baik dalam mengidentifikasi nilai data positif dan sebanyak 3% diprediksi sebagai *false positive*. Kemudian terdapat nilai *recall* sebesar 89%, artinya model yang menerapkan *SMOTE* ini sudah sangat baik dalam menangkap nilai data positif dan sebanyak 11% diprediksi sebagai *false negative*. Selain itu terdapat nilai *f1-score* sebesar 92%, artinya model telah memiliki nilai keseimbangan antara *precision* dan *recall* yang sangat baik dalam mengklasifikasi nilai data positif dan negatif.

Hasil akurasi prediksi dengan kernel *linear* yang menerapkan teknik *oversampling SMOTE* diperoleh sebesar 92% dengan parameter “C” terbaiknya pada nilai : 10000, dalam pencarian parameter terbaik, jarak nilai parameter “C” : [0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000] dengan teknik Grid Search digunakan untuk mencari kombinasi terbaik dari nilai parameter di atas yang akan memberikan kinerja yang paling baik, lalu diperoleh parameter terbaiknya yaitu pada nilai : 10000. Dengan hasil akurasi sebesar 92%, artinya dibandingkan dengan *dataset* yang tidak seimbang sebelumnya, hasil akurasinya meningkat sekitar 2%. Teknik *SMOTE* meningkatkan nilai akurasi dari

klasifikasi model SVM. Evaluasi hasilnya dapat dilihat pada Gambar 8 yaitu Confusion Matrix:



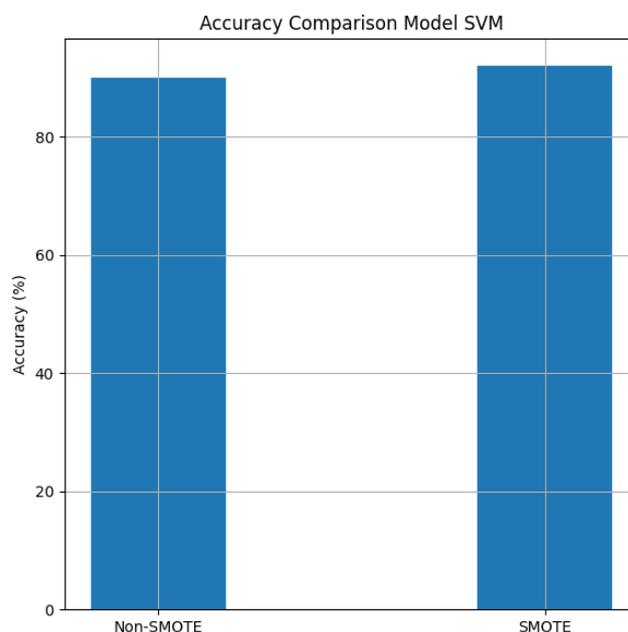
**Gambar 8** Confusion Matrix *Dataset* Seimbang (SMOTE)

Nilai *true positive (TP)* sebanyak 374 dan *false negative (FN)* sebanyak 48,  $374 + 48 = 422$  data *positive*. Kemudian nilai *true negative (TN)* sebanyak 310 dan *false positive (FP)* sebanyak 13,  $310 + 13 = 323$  data *negative*. Nilai TP sebanyak 374 data bahwa model berhasil dalam memprediksi nilai positif secara signifikan artinya model sudah sangat baik. Nilai FN sebanyak 48 data bahwa model gagal dalam memprediksi data positif, tetapi jumlahnya tidak terlalu signifikan artinya model sudah baik. Nilai TN sebanyak 310 data bahwa model berhasil dalam memprediksi nilai negatif secara signifikan, artinya model sudah sangat baik. Nilai FP sebanyak 13 data bahwa model gagal dalam memprediksi data negatif, tetapi jumlahnya sedikit, artinya model sudah sangat baik. Secara keseluruhan model sangat baik dalam memprediksi data positif maupun data negatif.

Hasil perbandingan nilai akurasi dari dataset tidak seimbang dan yang seimbang (SMOTE) ditunjukkan pada Tabel 6:

**Tabel 6** Komparasi hasil akurasi model SVM

	Non-SMOTE	SMOTE
<i>Accuracy</i>	90%	92%



**Gambar 9** Komparasi Hasil Akurasi Model SVM

Perbandingan hasil akurasi dari model SVM yang Non-SMOTE dan *Oversampling* SMOTE tidak memiliki jarak yang signifikan yaitu akurasi Non-SMOTE sebesar 90% dan akurasi SMOTE sebesar 92%, tetapi *dataset* yang diterapkan teknik SMOTE terbukti meningkatkan nilai akurasi dari SVM, artinya mengatasi salah satu kelemahan SVM adalah kecenderungan kinerja yang kurang baik dalam menangani *dataset* yang tidak seimbang. Peningkatan jumlah akurasinya sebesar 2%. Pemilihan parameter “C” pada kernel linear SVM berpengaruh dalam mencari parameter “C” terbaik, dengan mencari kombinasi terbaik untuk memaksimalkan kinerja dari algoritma SVM, sehingga hasil akhir akurasinya *reliable* dan *robust*.

### KESIMPULAN

Hasil dari penelitian yang dilakukan dengan mengusulkan metode penggunaan algoritma Support Vector Machine (SVM) dengan ekstraksi fitur menggunakan Word2Vec dan menggunakan model Skip-gram untuk melakukan klasifikasi sentimen. Kemudian menerapkan teknik pemilihan parameter dengan Grid Search untuk meningkatkan kinerja model. Selain itu, untuk menangani ketidakseimbangan *dataset* yang umum terjadi dalam klasifikasi sentimen, yaitu dengan mengimplementasikan teknik *oversampling* SMOTE. Hasil eksperimen menunjukkan peningkatan dalam kinerja SVM, dengan akurasi yang lebih tinggi setelah diterapkan teknik SMOTE. Penelitian ini memberikan kontribusi penting

dalam pengembangan metode klasifikasi sentimen yang efektif dan dapat diandalkan. Teknik Grid Search berpengaruh dalam membuat kinerja model menjadi lebih *reliable* dan *robust*, hal tersebut diperoleh dari pencarian parameter terbaik untuk memaksimalkan kinerja model SVM. Teknik Oversampling SMOTE mempunyai pengaruh signifikan yaitu peningkatan akurasi sebesar 2%, artinya model SVM bekerja dengan baik pada kasus *dataset* yang telah seimbang. Berdasarkan metode yang diusulkan tersebut bahwa hasil klasifikasi menunjukkan tingkat akurasi yang tinggi, yaitu dari *dataset* tidak seimbang sebesar 90% dan dari *dataset* yang seimbang (SMOTE) sebesar 92%.

Ekstraksi fitur menggunakan Word2Vec dengan model Skip-gram cocok digunakan untuk klasifikasi model SVM. Kemudian pemilihan parameter pada kernel *linear* atau parameter “C” dengan memberikan opsi 0,001 sampai 10000 dengan metode Grid Search, semakin tinggi nilai parameternya maka dapat meningkatkan hasil akurasi SVM. Selanjutnya teknik oversampling SMOTE terbukti mampu meningkatkan hasil akurasi sebesar 2%. Penelitian ini berhasil dalam mengatasi beberapa kelemahan dari algoritma SVM dan mampu menghasilkan nilai akurasi yang baik.

Penelitian selanjutnya dapat mengeksplorasi penggunaan kernel lainnya, seperti Polynomial, RBF, dan Sigmoid untuk memaksimalkan hasil dari klasifikasi SVM.

## UCAPAN TERIMA KASIH

Peneliti berterima kasih kepada seluruh pihak yang terlibat dari Universitas Jenderal Achmad Yani, sehingga peneliti dapat menyelesaikan penelitian ini.

## DAFTAR PUSTAKA

Anam, M. K., Triyani, ;, Fitri, A., Agustin, ;, Lusiana, ;, Muhammad, ;, Firdaus, B., Agus, ;, & Nurhuda, T. (2023). Sentiment Analysis for Online Learning using The Lexicon-Based Method and The Support Vector Machine Algorithm. *ILKOM Jurnal Ilmiah*, *15*(2), 290–302. <http://dx.doi.org/10.33096/ilkom.v15i2.1590.290-302>

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, *408*, 189–215. <https://doi.org/10.1016/j.neucom.2019.10.118>

Chong, K. S., & Shah, N. (2022). Comparison of Naive

Bayes and SVM Classification in Grid-Search Hyperparameter Tuned and Non-Hyperparameter Tuned Healthcare Stock Market Sentiment Analysis. *International Journal of Advanced Computer Science and Applications*, *13*(12), 90–94. <https://doi.org/10.14569/IJACSA.2022.0131213>

- Dey, S., Wasif, S., Tonmoy, D. S., Sultana, S., Sarkar, J., & Dey, M. (2020). A Comparative Study of Support Vector Machine and Naive Bayes Classifier for Sentiment Analysis on Amazon Product Reviews. *2020 International Conference on Contemporary Computing and Applications, IC3A 2020*, 217–220. <https://doi.org/10.1109/IC3A48958.2020.233300>
- Guia, M., Silva, R. R., & Bernardino, J. (2019). Comparison of Naive Bayes, Support Vector Machine, Decision Trees and Random Forest on Sentiment Analysis. *IC3K 2019 - Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, I(Ic3k)*, 525–531. <https://doi.org/10.5220/0008364105250531>
- Kurniawan, F. W., & Maharani, W. (2020). Indonesian Twitter Sentiment Analysis Using Word2Vec. *2020 International Conference on Data Science and Its Applications, ICoDSA 2020*, 31–36. <https://doi.org/10.1109/ICoDSA50139.2020.9212906>
- Mujahid, M., Lee, E., Rustam, F., Washington, P. B., Ullah, S., Reshi, A. A., & Ashraf, I. (2021). Sentiment analysis and topic modeling on tweets about online education during covid-19. *Applied Sciences (Switzerland)*, *11*(18). <https://doi.org/10.3390/app11188438>
- Nawang Sari, R. P., Kusumaningrum, R., & Wibowo, A. (2019). Word2vec for Indonesian sentiment analysis towards hotel reviews: An evaluation study. *Procedia Computer Science*, *157*, 360–366. <https://doi.org/10.1016/j.procs.2019.08.178>
- Octaviani, K., Andayani Komara, M., & Kurniawan, I. (2022). Analisis Kesuksesan Aplikasi Alfagift Menggunakan Model Delon Dan Mclean Studi Kasus Alfa Express Rest Area Km 72B. *Jurnal Informatika, Teknologi Dan Sains*, *4*(3), 173–178. <https://doi.org/10.51401/jinteks.v4i3.1946>
- Sheik Abdullah, A., Akash, K., ShaminThres, J., & Selvakumar, S. (2021). Sentiment Analysis of Movie Reviews Using Support Vector Machine Classifier with Linear Kernel Function. *Advances in Intelligent Systems and Computing*, *1176*, 345–354. [https://doi.org/10.1007/978-981-15-5788-0\\_34](https://doi.org/10.1007/978-981-15-5788-0_34)
- Sohrabi, M. K., & Hemmatian, F. (2019). An efficient preprocessing method for supervised sentiment

analysis by converting sentences to numerical vectors: a twitter case study. *Multimedia Tools and Applications*, 78(17), 24863–24882. <https://doi.org/10.1007/s11042-019-7586-4>

Styawati, S., Nurkholis, A., Aldino, A. A., Samsugi, S., Suryati, E., & Cahyono, R. P. (2022). Sentiment Analysis on Online Transportation Reviews Using Word2Vec Text Embedding Model Feature Extraction and Support Vector Machine (SVM) Algorithm. *2021 International Seminar on Machine Learning, Optimization, and Data Science, ISMODE 2021, February 2023*, 163–167. <https://doi.org/10.1109/ISMODE53584.2022.9742906>

Wahyudi, R., & Kusumawardana, G. (2021). Analisis Sentimen pada Review Aplikasi Grab di Google Play Store Menggunakan Support Vector Machine. *Jurnal Informatika*, 8(2), 200–207. <https://doi.org/10.31294/ji.v8i2.9681>