

Sentiment Analysis of Telegram App Reviews on Google Play Store Using the Support Vector Machine (SVM) Algorithm

Nofsa Atia Nevrada ^{1*}, Muhammad Adie Syaputra ^{2*}

* Sistem Informasi, Universitas Dharma Wacana

veronikanevrada@gmail.com ¹, adie.syaputra@dharmawacana.ac.id ²

Article Info

Article history:

Received 2024-11-08

Revised 2024-11-28

Accepted 2025-01-15

Keyword:

*Data pre-processing,
Positive and negative sentiment
classification,
Sentiment analysis,
Support Vector Machine,
Telegram application reviews.*

ABSTRACT

This study aims to analyze the sentiment of Telegram application reviews on the Google Play Store using the Support Vector Machine (SVM) algorithm. From a total of 14,700,000 initial reviews, a sampling technique was carried out to obtain 400 review data, which then went through the pre-processing stage to produce 345 review data to be classified. The SVM model used showed good performance with an accuracy of 81.16%, precision in the positive class reached 93%, recall in the negative class of 94%, and an average f1-score of around 81%. However, there was a discrepancy between the high rating and the content of the review, which highlighted the existence of high-rated reviews that contained criticism or vice versa. The confusion matrix analysis also showed some misclassification, where reviews should be categorized as positive sentiment but detected as negative, and vice versa. This research is expected to provide valuable feedback for Telegram application developers to improve the quality of service, although the results of this analysis have not been fully discussed in practice. The limitation of this study is that it only tested reviews that used Indonesian, which limited the scope of the findings to the context of local users.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

I. PENDAHULUAN

Telegram adalah salah satu aplikasi pesan instan berbasis cloud yang diluncurkan pada tahun 2013 oleh Pavel Durov dan saudaranya, Nikolai Durov [1]. Pertumbuhan penggunaan aplikasi pesan instan seperti Telegram telah mendorong pentingnya memahami pandangan serta pengalaman pengguna secara lebih mendalam. Telegram mengumumkan bahwa jumlah pengguna aktif bulanan yang mereka miliki kini sudah mencapai 950 juta. Sang CEO, Pavel Durov, menargetkan angka tersebut untuk naik hingga mencapai 1 miliar pengguna sebelum tahun 2024 ini berakhir [2].

Banyaknya pengguna platform komunikasi menjadikan aplikasi Telegram sebagai salah satu tujuan utama masyarakat untuk berkomunikasi dan berbagi informasi. Dari data Google Play Store, Telegram menduduki peringkat ke-4 pada jejaring sosial yang sering digunakan di Indonesia. Berdasarkan data tersebut, penting untuk memperhatikan kualitas layanan Telegram, terutama karena popularitasnya

masih kalah dengan pesaing lain pada jejaring sosial. Untuk memahami mengapa Telegram berada di bawah pesaingnya, analisis sentimen dapat digunakan untuk mengidentifikasi tanggapan pengguna, termasuk kekuatan dan kelemahan aplikasi. Analisis sentiment bertujuan untuk mengklasifikasikan ulasan pengguna ke dalam kategori positif dan negatif [3]. Dengan demikian, penyedia layanan dapat melakukan perbaikan berdasarkan feedback pengguna, sehingga metode yang akurat diperlukan dalam mengklasifikasikan ulasan-ulasan tersebut.

Algoritma Support Vector Machine (SVM) merupakan metode yang efektif karena kemampuannya yang kuat dalam memisahkan data pada kategori yang berbeda secara optimal [4]. SVM juga merupakan teknik machine learning yang didasarkan pada prinsip kerja menggunakan hyperplane untuk memisahkan dua kelas dari data input secara akurat [5]. Dalam beberapa penelitian, SVM terbukti memiliki akurasi yang lebih tinggi daripada metode lain. Sebagai contoh, dalam penelitian, [6] Dari hasil pengujian yang dilakukan terhadap 1500 data komentar pengguna, evaluasi model

menggunakan 10 Fold Cross Validation menunjukkan bahwa tingkat keakurasian untuk kepuasan pengguna aplikasi WhatsApp berdasarkan algoritma Naïve Bayes adalah sebesar 70,40% dan Support Vector Machine sebesar 77,00%, sedangkan nilai AUC Naïve Bayes sebesar 0,585 dan Support Vector Machine adalah 0,876. Dari hasil tersebut algoritma Support Vector Machine dapat digunakan untuk penelitian dengan karakteristik data yang sama. Lalu penelitian berjudul “Using Support Vector Machine For Sentiment Analysis Of Truecaller And Getcontact App Reviews” Penelitian ini menghasilkan 3148 baris data ulasan pada aplikasi Truecaller dan aplikasi Getcontact 3148 baris data. Hasil penelitian menunjukkan bahwa aplikasi Truecaller pada 10-fold cross validation memiliki rata-rata akurasi sebesar 88,20% dan aplikasi Getcontact memiliki akurasi rata-rata sebesar 87,90%. Sementara itu, aspek sentimen pada aplikasi Truecaller memiliki nilai rata-rata akurasi sebesar 60,20%, sedangkan aplikasi Getcontact memiliki akurasi rata-rata 63,30% [7].

Berdasarkan informasi yang telah disampaikan, Metode Support Vector Machine (SVM) menghasilkan akurasi yang lebih tinggi dibanding metode lainnya [8] dalam mengklasifikasikan sentimen pada ulasan aplikasi. Penelitian ini memutuskan untuk menggunakan algoritma Support Vector Machine (SVM) guna mengklasifikasikan ulasan aplikasi Telegram berdasarkan sentiment positif dan negatif. Dengan menganalisis ulasan di Google Play Store, penelitian ini bertujuan untuk memahami persepsi pengguna terhadap aplikasi Telegram dan mengevaluasi performa algoritma SVM dalam memprediksi sentimen tersebut. Pentingnya penelitian ini muncul karena ulasan dan penilaian (rating) yang terlihat langsung di Google Play Store tidak selalu menggambarkan sentimen sebenarnya. Ada ulasan dengan rating rendah tetapi berisi kata-kata positif, dan sebaliknya, ulasan dengan rating tinggi tetapi mengandung ungkapan negatif. Hasil penelitian ini diharapkan dapat memberikan gambaran yang lebih mendalam mengenai persepsi pengguna, yang pada akhirnya bisa menjadi masukan berharga bagi pengembang dalam meningkatkan kualitas layanan aplikasi Telegram.

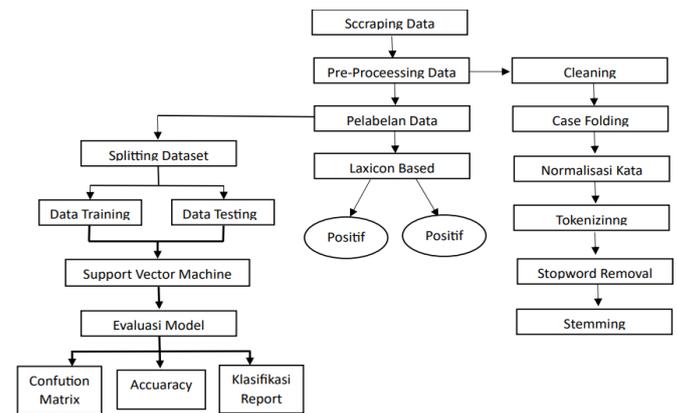
II. METODE

Metode penelitian digunakan untuk menguraikan rencana dan Langkah-langkah yang akan diambil dalam penelitian untuk mencapai tujuan yang ditentukan [9]. Penelitian ini menggunakan metode Support Vector Machine untuk analisis sentimen. Diagram alur penelitian diilustrasikan pada Gambar 1.

A. Scrapping Data

Penelitian ini menggunakan data ulasan pengguna, yang dikumpulkan menggunakan pustaka seperti *google-play-scraper* [10] untuk mendapatkan data ulasan secara langsung dari Google Play Store, Google Colab dipilih karena fleksibilitasnya dalam menangani proses scraping [11], serta *pandas* untuk mengelola dan menyimpan data yang diambil

dalam format tabel. Data yang dikumpulkan difilter untuk memastikan hanya ulasan dalam bahasa Indonesia yang sesuai dengan kebutuhan analisis sentimen. Setelah scraping, proses *pre-processing* dilakukan untuk membersihkan data dari ulasan yang tidak relevan dan duplikat. Langkah ini bertujuan memastikan bahwa data akhir benar-benar mencerminkan persepsi pengguna secara akurat terhadap aplikasi Telegram, sehingga hasil analisis menjadi lebih valid dan relevan.



Gambar 1 Alur Penelitian

Dalam penelitian ini, sebanyak 400 sampel ulasan terbaru telah dipilih menggunakan teknik sampling berdasarkan rumus Slovin. Metode ini memastikan bahwa ukuran sampel cukup representatif untuk memberikan gambaran yang akurat mengenai persepsi pengguna terhadap aplikasi yang diteliti.

$$n = \frac{N}{1+N(e)^2} = \frac{14.700.000}{1+14.700.000(0,05)^2} = 399,9989116 = 400 \text{ ulasan}$$

Keterangan:

n = Jumlah Sampel

N = Jumlah Populasi

e = Derajat Kesalahan /Error yang digunakan

B. Pre-Processing Data

Setelah tahap scraping, tahap pre-processing data dilakukan untuk mempersiapkan data ulasan [12] Telegram sebelum analisis sentimen. Berikut adalah tahapan pre-processing data:

- *Cleaning*

Dalam proses pembersihan teks, fungsi *regex* (regular expression) digunakan untuk menghapus karakter yang tidak penting seperti URL, HTML, emoji, simbol, dan angka [13]. Library dalam Python memfasilitasi pencocokan pola untuk mendeteksi elemen-elemen ini. Misalnya, URL dihapus dengan mencocokkan pola `https?://[S+/\www\.\S+]`, sementara tag HTML dihilangkan dengan pola `<.*?>`. Emoji dihapus menggunakan rentang Unicode khusus, dan simbol non-alfanumerik dihilangkan dengan pola `[\^a-zA-Z0-9\s]`. Untuk menghilangkan angka, digunakan pola `\d`. Semua

fungsi ini diterapkan pada kolom teks menggunakan metode *apply()* dari library *pandas*, menghasilkan teks yang lebih bersih dan siap untuk analisis lebih lanjut.

- *Case Folding*

Proses case folding adalah proses mengubah seluruh huruf dalam teks menjadi huruf kecil, menghilangkan perbedaan yang disebabkan oleh variasi kapitalisasi [14]. Proses ini menjaga konsistensi teks, yang penting untuk analisis lebih lanjut. Proses *case folding* dilakukan menggunakan fungsi *.lower()* dari library *pandas* pada Python, sehingga semua huruf dalam teks menjadi kecil untuk memudahkan analisis.

- *Normalisasi Data*

Langkah ini mengoreksi kata-kata yang tidak baku pada ulasan pengguna kedalam bentuk bakunya [15]. peneliti membuat kamus yang didalamnya terdapat kata-kata asli dan kata-kata singkatan. Sebagai contoh, “bgt : banget”, “tdk : tidak”, dan kata singkatan lainnya. Contoh kamus kata yang dinormalisasikan dan koreksinya ditunjukkan pada TABEL I.

TABEL I
NORMALISASI DATA

No	Kata Tidak Baku	Kata Baku
1	woww	wow
2	aminn	amin
3	met	selamat
4	netaas	menetas
5	keberpa	keberapa
6	eeehhhh	eh
7	kata2nyaaa	kata-katanya
8	hallo	halo
9	kaka	kakak
10	ka	kak

- *Tokenization*

Pada Langkah ini teks dipecah menjadi kata-kata individual [16] Tujuannya untuk mempresentasikan teks dengan cara yang mudah dipahami oleh model, sehingga memungkinkan algoritma mengenali pola dalam teks. Proses tokenisasi dilakukan menggunakan fungsi *split()* dalam Python, yang memisahkan teks berdasarkan spasi. Library *pandas* juga digunakan untuk menerapkan fungsi *tokenize* pada setiap elemen teks dalam kolom normalisasi melalui metode *apply()*.

- *Stopword Removal*

Menghapus kata-kata umum yang tidak memberikan kontribusi signifikan [17]. Tujuan penghapusan stopwords adalah untuk menghilangkan kata-kata ini dari teks, yang memungkinkan model untuk fokus pada istilah yang lebih bermakna dan relevan. Dengan menghilangkan stopwords seperti "dan", "atau", atau "yang", model dapat lebih fokus pada kata-kata yang relevan dan bermakna. Teknik ini menggunakan pustaka *nlTK* untuk mengakses daftar stopwords dalam bahasa Indonesia, dan diterapkan menggunakan fungsi *list comprehension* pada kolom teks yang sudah ditokenisasi.

- *Stemming*

Proses ini mengurangi kata-kata yang diimfleksikan atau diturunkan dalam bentuk akarnya [18] seperti mengubah "berlari" menjadi "lari", yang membantu meminimalkan jumlah kata unik yang perlu diproses oleh model [19]. Teknik ini, bersama dengan penghapusan stopwords, meningkatkan efisiensi dan kinerja model. Tools yang digunakan termasuk *Sastrawi*, library stemming untuk Bahasa Indonesia, dan *Pandas* untuk manipulasi data. Fungsi *apply()* bersama lambda digunakan untuk menerapkan fungsi stemming pada setiap teks dalam kolom dataframe, sementara *join()* menggabungkan kata-kata hasil stemming menjadi satu string.

C. Pelabelan Data

Setelah tahap pre-processing data, dilakukan pelabelan data menggunakan pendekatan berbasis lexicon based, yaitu dengan memanfaatkan kamus atau leksikon untuk mengidentifikasi kata-kata yang memiliki makna positive atau negative [20].

D. Splitting Dataset

Splitting dataset adalah proses membagi data menjadi dua bagian utama, yaitu data training dan data testing, yang bertujuan untuk melatih dan menguji model. Dalam penelitian ini, pembagian data dilakukan dengan teknik *train-test split*, menggunakan fungsi *train_test_split* dari pustaka *sklearn.model_selection*, yang mengalokasikan 80% data sebagai data *training* dan 20% data sebagai data *testing* [21]. Untuk visualisasi, digunakan teknik *bar plot* dengan pustaka *matplotlib.pyplot*, yang menampilkan grafik batang jumlah data *training* dan data *testing*, dilengkapi label persentase untuk memperlihatkan proporsi masing-masing jenis data dari total dataset. Visualisasi ini membantu memahami distribusi data antara bagian pelatihan dan pengujian, memberikan gambaran jelas mengenai pembagian data dalam model.

E. Klasifikasi Menggunakan Algoritma SVM

Dalam penelitian ini, algoritma Support Vector Machine (SVM) digunakan untuk klasifikasi biner terhadap ulasan pengguna aplikasi Telegram, dengan menggunakan *SVC* dari *sklearn.svm* untuk membangun model SVM. Metode ini diterapkan untuk mengklasifikasikan sentimen (positif atau negatif) dari teks ulasan dengan fitur yang diekstraksi. SVM bekerja dengan menemukan *hyperplane* yang memaksimalkan margin antara dua kelas, yang membuatnya sangat cocok untuk klasifikasi teks berdimensi tinggi. Model SVM ini menggunakan kernel tertentu, seperti *linear* atau *rbf*, yang membantu memetakan data ke dalam ruang fitur yang sesuai agar dapat dipisahkan dengan baik. Proses optimasi dilakukan dengan mencari parameter terbaik, seperti *C* (regularisasi) dan tipe kernel, untuk meningkatkan performa klasifikasi. Optimasi dilakukan dengan teknik *GridSearchCV* atau *RandomizedSearchCV*, yang membantu menemukan kombinasi parameter terbaik untuk

meningkatkan akurasi model pada data validasi [22], membuatnya sangat cocok untuk klasifikasi teks, terutama dalam analisis sentimen di mana data cenderung memiliki dimensi yang tinggi. Pemilihan SVM juga didasarkan pada kelebihanannya dalam menangani data teks dengan fitur berdimensi tinggi, yang umumnya ditemukan dalam analisis sentimen berbasis teks.

Untuk representasi data teks, *TF-IDF (Term Frequency - Inverse Document Frequency)* digunakan melalui *TfidfVectorizer* dari pustaka *sklearn*. Teknik TF-IDF memberi bobot lebih pada kata yang unik dalam setiap ulasan [23], membantu model untuk menangkap pentingnya kata-kata tertentu dalam menentukan sentimen. Dalam implementasi ini, data pelatihan diubah menjadi representasi numerik menggunakan TF-IDF yang mempelajari kosakata dari ulasan, dan kosakata tersebut diterapkan juga pada data pengujian untuk menjaga konsistensi.

Setelah data diubah menjadi bentuk vektor, evaluasi performa model dilakukan menggunakan beberapa metrik seperti *confusion matrix*, *classification report*, dan *accuracy score* dari *sklearn.metrics*. *Confusion matrix* divisualisasikan menggunakan *Seaborn* untuk memberikan gambaran tentang prediksi model yang benar dan salah pada setiap kelas sentimen. Teknik visualisasi ini mempermudah interpretasi kinerja model dalam memprediksi kelas sentimen positif dan negatif.

Dengan pendekatan ini, algoritma SVM diterapkan untuk mengklasifikasikan sentimen ulasan, dan visualisasi *confusion matrix* memberikan wawasan tambahan tentang efektivitas model dalam klasifikasi sentimen dari data tekstual.

F. Confusion Matrix

Confusion matrix merupakan metode evaluasi model dari klasifikasi. *Confusion matrix* berbentuk sebuah tabel yang menunjukkan jumlah data yang diklasifikasikan dengan benar dan salah dalam masing-masing kategori atau kelas target [24]. Contoh *confusion matrix* [25]. Tabel *confusion matrix* disajikan pada TABEL II.

TABEL II
CONFUSION MATRIX

Actual Class	Prediction Class	
	Positive	Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Negative (FN)	True Positive (TP)

Confusion matrix digunakan untuk menghitung akurasi, presisi, recall, spesififikasi, dan F1 Score. Dibawah ini: Rumus untuk menghitung akurasi, presisi, recall, spesififikasi, dan F1 Score adalah sebagai berikut.

1) Akurasi

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN}$$

Akurasi mengukur seberapa sering model membuat prediksi yang benar. Nilai akurasi diperoleh dengan

menjumlahkan jumlah True Positives (TP) dan True Negatives (TN), kemudian dibagi dengan total prediksi, yaitu jumlah TP, TN, False Positives (FP), dan False Negatives (FN).

2) Presisi

$$\text{Presisi} = \frac{TP}{TP+FP}$$

Presisi menunjukkan seberapa akurat prediksi positif yang dibuat oleh model. Ini adalah rasio antara jumlah True Positives (TP) dengan total prediksi positif (TP + FP). Presisi yang tinggi berarti model jarang memberikan hasil positif palsu.

3) Recall

$$\text{Recall} = \frac{TP}{TP+FN}$$

Recall mengukur seberapa baik model mendeteksi semua data positif yang sebenarnya. Ini adalah rasio antara jumlah True Positives (TP) dan jumlah semua data aktual positif (TP + FN). Recall yang tinggi berarti model jarang melewatkan data positif.

4) Spesififikasi

$$\text{Spesifisitas} = \frac{TN}{TN+FP}$$

Spesifisitas menunjukkan seberapa baik model dalam mendeteksi data negatif yang benar. Ini adalah rasio antara jumlah True Negatives (TN) dengan jumlah semua data aktual negatif (TN + FP). Spesifisitas yang tinggi berarti model jarang memberikan hasil negatif palsu.

5) F1 Score

$$\text{F1 Score} = 2 \frac{\text{Recall} \times \text{Presisi}}{\text{Recall} + \text{Presisi}}$$

F1 Score adalah rata-rata harmonik dari presisi dan recall. F1 Score memberikan gambaran seimbang antara presisi dan recall, terutama jika ada ketidakseimbangan antara jumlah data positif dan negatif. Nilai F1 yang tinggi menunjukkan bahwa model memiliki presisi dan recall yang baik.

G. Visualisasi Data

Untuk memvisualisasikan hasil, peneliti menggunakan *WordCloud*, yaitu alat visual yang menampilkan kata-kata yang sering muncul dalam ulasan pengguna, memberikan gambaran yang jelas tentang frekuensi kata dalam ulasan pengguna [26]. Data diolah menggunakan *pandas* untuk memisahkan kategori sentimen, dan setiap kategori digabungkan ke dalam teks panjang. Teknik ini memungkinkan pemisahan data yang efisien berdasarkan sentimen positif dan negatif. Visualisasi kemudian dibuat dengan library *WordCloud*, yang menghasilkan grafik dengan ukuran kata sesuai frekuensinya, dan menggunakan *matplotlib* untuk tampilan yang lebih mudah dipahami.

III. HASIL DAN PEMBAHASAN

Hasil dan Pembahasan digunakan untuk menyajikan data dari temuan penelitian untuk menunjukkan keberhasilan penelitian ini

A. Scrapping Data

Pengambilan data dilakukan dengan metode scrapping menggunakan Google Colab dengan menginstall *google-play-scraper* untuk mendapatkan data ulasan Telegram pada google playstore. Data yang didapatkan berupa *review ID*, *username*, *rating*, *review text*, waktu dan tanggal sebanyak 400 ulasan dari jumlah total ulasan 14.700.000, hal ini dikarenakan pada penelitian ini menggunakan sampel yang sudah dihitung dengan rumus slovin. Hasil scrapping data yang telah dikumpulkan dapat dilihat pada TABEL III.

TABEL III
HASIL SCRAPPING DATA

Review ID	Username	Rating	Review Text	Date
0a235fd7-2a10-4cef-be37-3348aa8d7e1a	Arif Firmansah	5	Good	03/11/2024 12:07
61c4ac2a-58db-4625-8dea-706437f7ec6e	apik Rafik	3	Mantap	03/11/2024 11:51
5abbcf23-bf92-4445-841a-64fafa3e54c9	Enggal Prakoso	2	Telegram, nomorku baru tapi kenapa ga bisa buat daftar? tolonglah perbaiki kayak dulu lagi. Aku kesusahan mau daftar	03/11/2024 11:46
ae4168e6-0259-4834-bda1-e76d1dbec7ef	Lukman hakim	1	Ndk ada kede masuk	03/11/2024 11:45
af393cba-5995-43ac-b07e-e0185b0597f4	Uddi Ajjah	1	Keluar masuk sendiri dan sudah dipengaturan juga susah	03/11/2024 11:40
8fb297b5-a34f-412c-a563-ca871560f2a0	deden efendi	5	Ok 🤝	03/11/2024 11:30

B. Pre-Processing Data

Pre-processing data merupakan tahapan selanjutnya setelah tahap scrapping, pada tahapan ini dilakukan pemrosesan data dengan *pre processing* yang melalui enam tahapan yaitu Cleaning, Case Folding, Normalisasi Kata, Tokenization, Stopword Removal, dan Stemming.

1) Cleaning

Memanggil fungsi untuk menghilangkan karakter yang tidak dibutuhkan pada teks di kolom ulasan dalam proses klasifikasi seperti url, symbol, hashtag, dan emoji. Hasil cleaning dari data ulasan aplikasi Telegram dapat dilihat pada TABEL IV.

TABEL IV
HASIL TAHAP CLEANING

Date	Username	Rating	Review Text	Cleaning
03/11/2024 12:07	Arif Firmansah	5	Good	Good
03/11/2024 11:51	apik Rafik	3	Mantap	Mantap
03/11/2024 11:46	Enggal Prakoso	2	Telegram, nomorku baru tapi kenapa ga bisa buat daftar? tolonglah perbaiki kayak dulu lagi. Aku kesusahan mau daftar	Telegram nomorku baru tapi kenapa ga bisa buat daftar tolonglah perbaiki kayak dulu lagi. Aku kesusahan mau daftar
03/11/2024 11:45	Lukman hakim	1	Ndk ada kede masuk	Ndk ada kede masuk
03/11/2024 11:40	Uddi Ajjah	1	Keluar masuk sendiri dan sudah dipengaturan juga susah	Keluar masuk sendiri dan sudah dipengaturan juga susah
03/11/2024 11:30	deden efendi	5	Ok 🤝	Ok

2) Case Folding

Memanggil fungsi *case folding* untuk merubah huruf kapital pada teks dalam ulasan menjadi huruf kecil atau non-kapital. Hasil case folding dari data ulasan aplikasi Telegram dapat dilihat pada TABEL V.

TABEL V
HASIL TAHAP CASE FOLDING

Review Text	Cleaning	Case Folding
Good	Good	good
Mantap	Mantap	mantap
Telegram, nomorku baru tapi kenapa ga bisa buat daftar? tolonglah perbaiki kayak dulu lagi. Aku kesusahan mau daftar	Telegram nomorku baru tapi kenapa ga bisa buat daftar tolonglah perbaiki kayak dulu lagi Aku kesusahan mau daftar	telegram nomorku baru tapi kenapa ga bisa buat daftar tolonglah perbaiki kayak dulu lagi aku kesusahan mau daftar
Ndk ada kede masuk	Ndk ada kede masuk	ndk ada kede masuk
Keluar masuk sendiri dan sudah dipengaturan juga susah	Keluar masuk sendiri dan sudah dipengaturan juga susah	keluar masuk sendiri dan sudah dipengaturan juga susah
Ok 👍	Ok	ok

3) Normalisasi Kata

Melakukan pemuatan data berupa kamus singkatan kata dalam Bahasa Indonesia yang kemudian dilakukan proses normalisasi terhadap kata-kata yang berupa singkatan tertentu pada kolom ulasan. Hasil case normalisasi kata dari data ulasan aplikasi Telegram dapat dilihat pada tabel VI.

TABEL VI
HASIL TAHAP NORMALISASI KATA

Cleaning	Case Folding	Normalisasi
Good	good	good
Mantap	mantap	mantap
Telegram nomorku baru tapi kenapa ga bisa buat daftar tolonglah perbaiki kayak dulu lagi Aku kesusahan mau daftar	telegram nomorku baru tapi kenapa ga bisa buat daftar tolonglah perbaiki kayak dulu lagi aku kesusahan mau daftar	telegram nomorku baru tapi kenapa tidak bisa buat daftar tolonglah perbaiki kayak dulu lagi aku kesusahan mau daftar
Ndk ada kede masuk	ndk ada kede masuk	ndk ada kede masuk
Keluar masuk sendiri dan sudah dipengaturan juga susah	keluar masuk sendiri dan sudah dipengaturan juga susah	keluar masuk sendiri dan sudah dipengaturan juga susah
Ok	ok	ok

4) Tokenization

Memanggil fungsi tokenizing untuk memisahkan kata atau memberikan token terhadap kata dalam kalimat yang ada pada kolom ulasan. Hasil tokenization dari data ulasan aplikasi Telegram dapat dilihat pada TABEL VII.

TABEL VII
HASIL TAHAP TOKONIZATION

Case Folding	Normalisasi	Tokenize
good	good	['good']
mantap	mantap	['mantap']
telegram nomorku baru tapi kenapa ga bisa buat daftar tolonglah perbaiki kayak dulu lagi aku kesusahan mau daftar	telegram nomorku baru tapi kenapa tidak bisa buat daftar tolonglah perbaiki kayak dulu lagi aku kesusahan mau daftar	['telegram', 'nomorku', 'baru', 'tapi', 'kenapa', 'tidak', 'bisa', 'buat', 'daftar', 'tolonglah', 'perbaiki', 'kayak', 'dulu', 'lagi', 'aku', 'kesusahan', 'mau', 'daftar']
ndk ada kede masuk	ndk ada kede masuk	['ndk', 'ada', 'kede', 'masuk']
keluar masuk sendiri dan sudah dipengaturan juga susah	keluar masuk sendiri dan sudah dipengaturan juga susah	['keluar', 'masuk', 'sendiri', 'dan', 'sudah', 'dipengaturan', 'juga', 'susah']
ok	ok	['ok']

5) Stopword Removal

Memanggil fungsi untuk menghapus kata yang tidak memiliki makna pada kalimat yang ada dalam kolom ulasan yang biasanya merupakan kata sambung seperti kata “di”, “dan”, “sangat” yang terdapat pada *library stopwords* Indonesia *NLTK*. Hasil stopwords removal dari data ulasan aplikasi Telegram dapat dilihat pada tabel VIII.

TABEL VIII
HASIL TAHAP STOPWORD REMOVAL

Normalisasi	Tokenize	Stopword Removal
good	['good']	['good']
mantap	['mantap']	['mantap']
telegram nomorku baru tapi kenapa tidak bisa buat daftar tolonglah perbaiki kayak dulu lagi aku kesusahan mau daftar	['telegram', 'nomorku', 'baru', 'tapi', 'kenapa', 'tidak', 'bisa', 'buat', 'daftar', 'tolonglah', 'perbaiki', 'kayak', 'dulu', 'lagi', 'aku', 'kesusahan', 'mau', 'daftar']	['telegram', 'nomorku', 'daftar', 'tolonglah', 'perbaiki', 'kayak', 'kesusahan', 'daftar']
ndk ada kede masuk	['ndk', 'ada', 'kede', 'masuk']	['ndk', 'kede', 'masuk']
keluar masuk sendiri dan sudah dipengaturan juga susah	['keluar', 'masuk', 'sendiri', 'dan', 'sudah', 'dipengaturan', 'juga', 'susah']	['masuk', 'dipengaturan', 'susah']
ok	['ok']	['ok']

6) Stemming

Memanggil fungsi Stemming untuk menghapus imbuhan pada kata-kata sehingga kata tersebut berubah menjadi bentuk seperti kata “berlari” menjadi “lari”. Hasil stemming dari data ulasan aplikasi Telegram dapat dilihat pada tabel IX.

TABEL IX
HASIL TAHAP STEMMING

Tokenize	Stopword Removal	Stemming
['good']	['good']	good
['mantap']	['mantap']	mantap
['telegram', 'nomorku', 'baru', 'tapi', 'kenapa', 'tidak', 'bisa', 'buat', 'daftar', 'tolonglah', 'perbaiki', 'kayak', 'dulu', 'lagi', 'aku', 'kesusahan', 'mau', 'daftar']	['telegram', 'nomorku', 'daftar', 'tolonglah', 'perbaiki', 'kayak', 'kesusahan', 'daftar']	telegram nomor daftar tolong baik kayak susah daftar
['ndk', 'ada', 'kede', 'masuk']	['ndk', 'kede', 'masuk']	ndk kede masuk

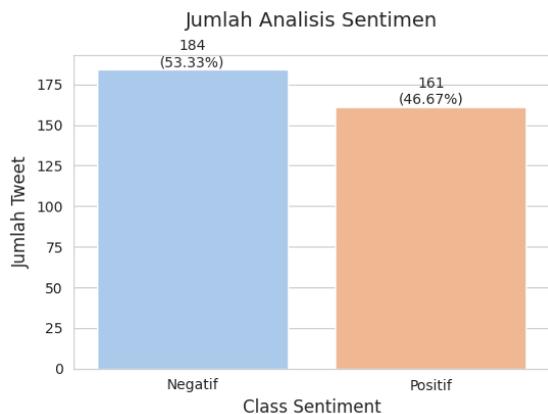
['keluar', 'masuk', 'sendiri', 'dan', 'sudah', 'dipengaturan', 'juga', 'susah']	['masuk', 'dipengaturan', 'susah']	masuk atur susah
['ok']	['ok']	ok

C. Pelabellan Data Lexicon Based

Setelah dilakukan tahapan *pre-processing* data, selanjutnya dilakukan dengan pencocokan kata pada dataset lexicon berbahasa Indonesia yang telah disiapkan untuk menghasilkan nilai *polarity* kalimat yang akan digunakan untuk penentuan sentimen positif dan negatif. Hasil pelabellan data menggunakan kamus *lexicon* dapat dilihat pada TABEL X.

TABEL X
HASIL TAHAP STEMMING

Cleaning	Case_Folding	Normalisasi	Tokenize	Stopword Removal	Steming_Data	Sentiment
Good	good	good	['good']	['good']	good	Positif
Mantap	mantap	mantap	['mantap']	['mantap']	mantap	Positif
Telegram nomorku baru tapi kenapa ga bisa buat daftar tolonglah perbaiki kayak dulu lagi Aku kesusahan mau daftar	telegram nomorku baru tapi kenapa ga bisa buat daftar tolonglah perbaiki kayak dulu lagi aku kesusahan mau daftar	telegram nomorku baru tapi kenapa tidak bisa buat daftar tolonglah perbaiki kayak dulu lagi aku kesusahan mau daftar	['telegram', 'nomorku', 'baru', 'tapi', 'kenapa', 'tidak', 'bisa', 'buat', 'daftar', 'tolonglah', 'perbaiki', 'kayak', 'dulu', 'lagi', 'aku', 'kesusahan', 'mau', 'daftar']	['telegram', 'nomorku', 'daftar', 'tolonglah', 'perbaiki', 'kayak', 'kesusahan', 'daftar']	telegram nomor daftar tolong baik kayak susah daftar	Negatif
Ndk ada kede masuk	ndk ada kede masuk	ndk ada kede masuk	['ndk', 'ada', 'kede', 'masuk']	['ndk', 'kede', 'masuk']	ndk kede masuk	Negatif
Keluar masuk sendiri dan sudah dipengaturan juga susah	keluar masuk sendiri dan sudah dipengaturan juga susah	keluar masuk sendiri dan sudah dipengaturan juga susah	['keluar', 'masuk', 'sendiri', 'dan', 'sudah', 'dipengaturan', 'juga', 'susah']	['masuk', 'dipengaturan', 'susah']	masuk atur susah	Negatif
Ok	ok	ok	['ok']	['ok']	ok	Positif



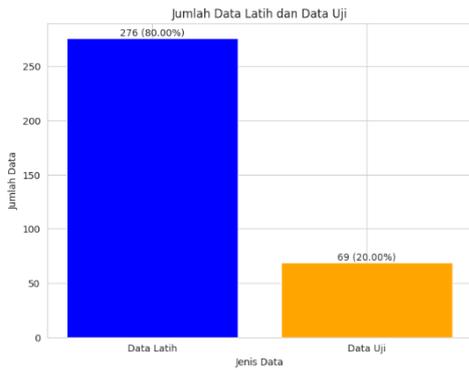
Gambar 2 Jumlah Analisis Sentimen

Adapun perbandingan jumlah dataset setelah dilakukan preprocessing dan lexicon dapat dilihat pada Gambar 2.

Berdasarkan Jumlah Analisis Sentimen, dapat diketahui bahwa perbandingan jumlah dataset setelah dilakukan preprocessing dan lexicon yakni sentimen negatif sebanyak 184 (53.33%) dan sentimen positif sebanyak 161 (46.67%) dengan total keseluruhan 345 data ulasan sesuai hasil penghapusan baris dan kolom.

D. Splitting Dataset

Splitting dataset adalah proses membagi data menjadi dua bagian utama, yaitu data training dan data testing, yang bertujuan untuk melatih dan menguji model. Hasil Splitting dataset dapat dilihat pada Gambar 3.



Gambar 3 Jumlah Data Latih Dan Data Uji

Hasil pembagian Data Latih dan Data Uji, dapat diketahui bahwa data latih berjumlah 276 (80.00%) dan data uji berjumlah 69 (20.00%) dengan total keseluruhan 345 data ulasan sesuai hasil penghapusan baris dan kolom.

Setelah proses pemanggilan data training dan data testing selesai dilakukan, maka didapatkan data yang telah siap untuk diolah untuk melakukan pengujian model matrix Support Vector Machine. Hasil *confusion matrix* dapat dilihat pada Gambar 4.

SVM Accuracy: 0.8115942028985508

SVM Accuracy: 81.16%

SVM Classification Report:

	precision	recall	f1-score	support
Negatif	0.73	0.94	0.82	32
Positif	0.93	0.70	0.80	37
accuracy			0.81	69
macro avg	0.83	0.82	0.81	69
weighted avg	0.84	0.81	0.81	69

Gambar 4 Hasil Confusion Matrix SVM

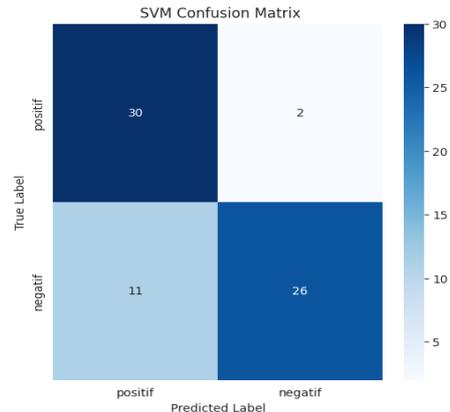
Berdasarkan *confusion matrix*, dapat diketahui bahwa data ulasan negatif yang terprediksi negatif (benar) sebanyak 30 dan yang terprediksi positif (salah) sebanyak 2. Sedangkan data ulasan positif yang terprediksi negatif (salah) sebanyak 11 dan yang terprediksi positif (benar) sebanyak 26.

E. Evaluasi Akurasi Support Vector Machine

Pada klasifikasi Support Vector Machine dalam menentukan tingkat kemiripan antara hasil pengukuran dengan nilai yang sebenarnya diukur menggunakan tingkat akurasi dari hasil perhitungan dataset yang digunakan, akurasi ditetapkan untuk mengetahui seberapa besar tingkat kesalahan pengukuran yang dapat terjadi pada saat perhitungan dataset yang digunakan.

Dalam penelitian ini peneliti menggunakan 20% data testing setelah data review dibagi menjadi 2 untuk mendapatkan hasil akurasi model. Hasil evaluasi

performa model support vector machine dapat dilihat pada Gambar 5.



Gambar 5 Hasil Evaluasi Performa Model SVM

F. Wordcloud

Tahapan selanjutnya yaitu adalah tahap visualisasi data menggunakan wordcloud untuk mengetahui kata yang sering muncul pada setiap sentimen. Hasil *wordcloud* dapat dilihat pada Gambar 6 dan 7.



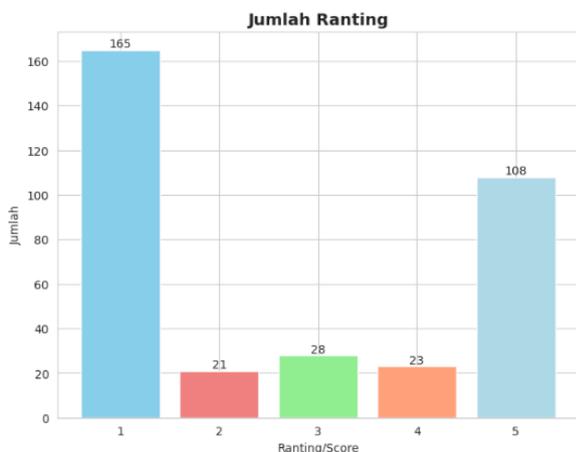
Gambar 6 Hasil Wordclod Sentimrn Negatif



Gambar 7 Hasil Wordclod Sentimrn Positif

G. Ranting

Tahapan terakhir yaitu adalah tahap melihat rating dalam dataset dengan menampilkan grafik batang (*bar chart*) untuk menggambarkan distribusi jumlah tiap nilai. Hasil ranting yang disajikan dalam bentuk grafik batang (*bar chart*) dapat dilihat pada Gambar 8.



Gambar 8 Jumlah Ranting

Berdasarkan Gambar 8, terdapat total 345 ulasan sesuai hasil penghapusan baris dan kolom, dengan rincian: 165 ulasan memberi ranting 1, 21 ulasan memberi ranting 2, 28 ulasan memberi ranting 3, 23 ulasan memberi ranting 4, dan 108 ulasan memberi ranting 5, menunjukkan bahwa sebagian besar ulasan cenderung negatif.

IV. KESIMPULAN

Penelitian ini menganalisis sentimen ulasan aplikasi Telegram di Google Play Store dengan total data ulasan awal sebanyak 14.700.000. Setelah dilakukan teknik sampling, diperoleh 400 data ulasan yang, setelah tahap pre-processing, menghasilkan 345 data ulasan untuk diklasifikasikan menggunakan algoritma Support Vector Machine. Hasil analisis menunjukkan bahwa model ini memiliki performa baik, dengan akurasi sebesar 81,16%. Precision pada kelas positif mencapai 93%, sementara recall pada kelas negatif sebesar 94%, dan f1-score rata-rata sekitar 81%. Meskipun model memiliki performa yang baik, terdapat ketidaksesuaian antara rating tinggi dengan isi ulasan, yang menyoroti adanya ulasan dengan rating tinggi tetapi berisi kritik, atau sebaliknya.

Berdasarkan analisis confusion matrix, ditemukan beberapa ulasan yang seharusnya diklasifikasikan sebagai sentimen positif tetapi terdeteksi sebagai negatif, dan sebaliknya. Temuan ini mengindikasikan bahwa, meskipun akurat, model masih membutuhkan evaluasi lebih lanjut untuk menyempurnakan klasifikasi agar lebih konsisten. Penelitian ini diharapkan memberikan umpan balik berharga bagi pengembang aplikasi Telegram untuk meningkatkan kualitas layanan. Namun, agar hasil analisis ini benar-benar bermanfaat secara praktis, pengembang perlu memahami area spesifik yang memerlukan perbaikan, misalnya, stabilitas saat video call atau kemudahan akses ke fitur keamanan, guna meningkatkan pengalaman pengguna secara keseluruhan.

DAFTAR PUSTAKA

- [1] N. Andre Saputra, J. Alexandra, Dan I. Budi Trisno, "Analisis Sentimen Pemanfaatan Obrolan Grup Telegram Berbagi Informasi Lowongan Kerja Menggunakan Metode Naïve Bayes Classifier," *Jati*, Vol. 7, No. 2, Hlm. 1321–1327, Sep 2023, Doi: 10.36040/Jati.V7i2.6693.
- [2] Universiti Pendidikan Sultan Idris, N. Abu Bakar, F. N. Mohd Rofizi, Universiti Pendidikan Sultan Idris, N. F. Mohd Rusli, Dan Universiti Pendidikan Sultan Idris, "Telegram As An Alternative Medium In The Teaching And Learning Process At Home (Pdpr)," *Oj-Tp*, Vol. 7, No. 2, Sep 2022, Doi: 10.30880/Ojtp.2022.07.02.008.
- [3] D. T. Lukmana, S. Subanti, Dan Y. Susanti, "Analisis Sentimen Terhadap Calon Presiden 2019 Dengan Support Vector Machine Di Twitter," 2019.
- [4] A. P. Natasuwarna, "Seleksi Fitur Support Vector Machine Pada Analisis Sentimen Keberlanjutan Pembelajaran Daring," *Tc*, Vol. 19, No. 4, Hlm. 437–448, Nov 2020, Doi: 10.33633/Tc.V19i4.4044.
- [5] K. A. Rokhman, B. Berlilana, Dan P. Arsi, "Perbandingan Metode Support Vector Machine Dan Decision Tree Untuk Analisis Sentimen Review Komentar Pada Aplikasi Transportasi Online," *Joism*, Vol. 3, No. 1, Hlm. 1–7, Jan 2021, Doi: 10.24076/Joism.2021v3i1.341.
- [6] A. Saepulrohman, S. Saepudin, Dan D. Gustian, "Analisis Sentimen Kepuasan Pengguna Aplikasi Whatsapp Menggunakan Algoritma Naïve Bayes Dan Support Vector Machine," *Aisthebest*, Vol. 6, No. 2, Hlm. 91–105, Des 2021, Doi: 10.34010/Aisthebest.V6i2.4919.
- [7] S. Rita, D. Indrayana, Dan A. Pambudi, "Penggunaan Support Vector Machine Untuk Analisis Sentimen Ulasan Aplikasi Truecaller Dan Getcontact," *Bit*, Vol. 20, No. 2, Hlm. 131, Sep 2023, Doi: 10.36080/Bit.V20i2.2493.
- [8] S. Fide, S. Suparti, Dan S. Sudarno, "Analisis Sentimen Ulasan Aplikasi Tiktok Di Google Play Menggunakan Metode Support Vector Machine (Svm) Dan Asosiasi," *J.Gauss*, Vol. 10, No. 3, Hlm. 346–358, Des 2021, Doi: 10.14710/J.Gauss.V10i3.32786.
- [9] R. Maulana, A. Voutama, Dan T. Ridwan, "Analisis Sentimen Ulasan Aplikasi My Pertamina Pada Google Play Store Menggunakan Algoritma Nbc," *J. Teknologi Terpadu*, Vol. 9, No. 1, Hlm. 42–48, Jul 2023, Doi: 10.54914/Jtt.V9i1.609.
- [10] A. F. Putri, G. Manik, F. Nabila, Dan N. Chamidah, "Implementasi Scraping Google Scholar Menggunakan Html Dom Untuk Pengumpulan Data Artikel Dosen Upn Veteran Jakarta Berbasis Web".
- [11] L. Hidayati, L. P. Kusuma, D. Agustini, Dan V. Y. P. Ardhana, "Implementasi Web Scraping Untuk Pengumpulan Data Media Sosial Lingkup Pemerintah Provinsi Ntb," *J.Sist.Inf.Inform.*, Vol. 7, No. 1, Hlm. 63–72, Mar 2024, Doi: 10.47080/Simika.V7i1.3200.
- [12] D. A. C. Rachman, R. Goejantoro, Dan F. D. T. Amijaya, "Implementasi Text Mining Pengelompokan Dokumen Skripsi Menggunakan Metode K-Means Clustering," *J. Ekspansional*, Vol. 11, No. 2, Hlm. 167, Jan 2021, Doi: 10.30872/Ekspansional.V11i2.660.
- [13] S. Wahyu Handani, D. Intan Surya Saputra, Hasirun, R. Mega Arino, Dan G. Fiza Asyofi Ramadhan, "Sentiment Analysis For Go-Jek On Google Play Store," *J. Phys.: Conf. Ser.*, Vol. 1196, Hlm. 012032, Mar 2019, Doi: 10.1088/1742-6596/1196/1/012032.
- [14] N. Herlinawati, Y. Yuliani, S. Faizah, W. Gata, Dan S. Samudi, "Analisis Sentimen Zoom Cloud Meetings Di Play Store Menggunakan Naïve Bayes Dan Support Vector Machine," *Com, Engine, Sys, Sci*, Vol. 5, No. 2, Hlm. 293, Jul 2020, Doi: 10.24114/Cess.V5i2.18186.
- [15] T. Safitri, Y. Umidah, Dan I. Maulana, "Analisis Sentimen Pengguna Twitter Terhadap Bts Menggunakan Algoritma Support Vector Machine," Vol. 7, No. 1.
- [16] A. D. Adhi Putra, "Analisis Sentimen Pada Ulasan Pengguna Aplikasi Bibit Dan Bareksa Dengan Algoritma Knn," *Jatisi*, Vol.

- 8, No. 2, Hlm. 636–646, Jun 2021, Doi: 10.35957/Jatisi.V8i2.962.
- [17] M. I. Ahmadi, F. Apriani, M. Kurniasari, S. Handayani, Dan D. Gustian, “Sentiment Analysis Online Shop On The Play Store Using Method Support Vector Machine (Svm),” 2020.
- [18] R. Mahendrajaya, G. A. Buntoro, Dan M. B. Setyawan, “Analisis Sentimen Pengguna Gopay Menggunakan Metode Lexicon Based Dan Support Vector Machine,” *Jkt*, Vol. 3, No. 2, Hlm. 52, Okt 2019, Doi: 10.24269/Jkt.V3i2.270.
- [19] D. Safryda Putri Dan T. Ridwan, “Analisis Sentimen Ulasan Aplikasi Pospay Dengan Algoritma Support Vector Machine,” *Oai*, Vol. 11, No. 01, Hlm. 32–40, Mar 2023, Doi: 10.33884/Jif.V11i01.6611.
- [20] N. Cahyono Dan Anggista Oktavia Praneswara, “Analisis Sentimen Ulasan Aplikasi Tiktok Shop Seller Center Di Google Playstore Menggunakan Algoritma Naive Bayes,” *Ijcs*, Vol. 12, No. 6, Des 2023, Doi: 10.33022/Ijcs.V12i6.3473.
- [21] T. Turki Dan S. S. Roy, “Novel Hate Speech Detection Using Word Cloud Visualization And Ensemble Learning Coupled With Count Vectorizer,” *Applied Sciences*, Vol. 12, No. 13, Hlm. 6611, Jun 2022, Doi: 10.3390/App12136611.
- [22] A. Kusuma Dan H. N. Irmanda, “Analisis Sentimen Pada Ulasan Aplikasi Indodax Di Google Play Store Menggunakan Metode Support Vector Machine”.
- [23] N. P. Husain Dan A. F. Syam, “Analisis Sentimen Ulasan Pengguna Tiktok Pada Google Play Store Berbasis Tf-Idf Dan Support Vector Machine,” Vol. 5, No. 1, 2024.
- [24] Gilbert, Syariful Alam, Dan M. Imam Sulisty, “Analisis Sentimen Berdasarkan Ulasan Pengguna Aplikasi Mypertamina Pada Google Playstore Menggunakan Metode Naive Bayes,” *Storage: Jurnal Ilmiah Teknik Dan Ilmu Komputer*, Vol. 2, No. 3, Hlm. 100–108, Agu 2023, Doi: 10.55123/Storage.V2i3.2333.
- [25] D. Normawati Dan S. A. Prayogi, “Implementasi Naive Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter,” Vol. 5, 2021.
- [26] B. F. S. Supriyanto Dan S. Rosalin, “Analisis Sentimen Program Merdeka Belajar Dengan Text Analysis Wordcloud & Word Frequency,” *Jmp*, Vol. 12, No. 1, Hlm. 25–32, Mar 2023, Doi: 10.33395/Jmp.V12i1.12312.