

# Analisis Sentimen Ulasan Pengguna Aplikasi *Threads* pada Google Play Store Menggunakan Multinomial Naive Bayes dan *Support Vector Machine*

Muhammad Arya Java<sup>1\*</sup>, Mohammad Syafrullah<sup>2</sup>, Windarto<sup>3</sup>, Painem<sup>4</sup>

<sup>1,2,3,4</sup>Fakultas Teknologi Informasi, Teknik Informatika, Universitas Budi Luhur, DKI Jakarta, Indonesia

Jl. Ciledug Raya, Petungkang Utara, Jakarta Selatan, 12260. DKI Jakarta

E-mail: <sup>1\*</sup>muhammad.aryajava@gmail.com, <sup>2</sup>mohammad.syafrullah@budiluhur.ac.id, <sup>3</sup>windarto@budiluhur.ac.id,

<sup>4</sup>painem@budiluhur.ac.id

(\*: corresponding author)

**Abstrak**—Analisis sentimen pada ulasan pengguna aplikasi menjadi topik yang menarik untuk dipelajari karena memberikan wawasan tentang bagaimana pengguna merespons dan mempersepsikan sebuah aplikasi. Penelitian ini bertujuan untuk melakukan analisis sentimen pada ulasan aplikasi *Threads* di *Google Play Store* dengan menggunakan metode Multinomial Naive Bayes dan *Support Vector Machine*. Latar belakang penelitian ini didasarkan pada penggunaan aplikasi jejaring sosial yang semakin populer dan banyaknya ulasan pengguna yang perlu dianalisis secara efisien. Masalah yang dihadapi adalah volume besar ulasan yang sulit untuk dianalisis secara manual. Metode yang digunakan mencakup persiapan data, penanganan ketidakseimbangan kelas dengan penerapan SMOTE, dan ekstraksi fitur dengan TF-IDF. Evaluasi model dilakukan dengan mengukur *accuracy*, *precision*, *recall*, dan *F1-score* untuk mengklasifikasikan sentimen ulasan menjadi positif atau negatif. Dari hasil pengujian menunjukkan kedua algoritma memiliki performa hampir seimbang dengan akurasi sekitar 81% model-model ini mampu mengklasifikasikan sentimen ulasan dengan baik, memiliki *precision* dan *recall* yang tinggi untuk kedua kelas. Kesimpulan penelitian ini adalah metode *Machine Learning* berhasil memberikan solusi efisien dan akurat untuk analisis sentimen pada ulasan aplikasi *Threads* di *Google Play Store*, namun batasan data Bahasa Indonesia menjadi perhatian untuk penelitian selanjutnya.

**Kata Kunci**— Analisis Sentimen, Ulasan Aplikasi, Multinomial Naive Bayes, *Support Vector Machine*, *Google Play Store*.

**Abstract**—*Sentiment analysis of application user reviews is an interesting topic to study because it provides insight into how users respond and perceive an application. This research aims to conduct sentiment analysis on reviews of the Threads application on the Google Play Store using the Multinomial Naive Bayes and Support Vector Machine methods. The background of this research is based on the increasingly popular use of social networking applications and the large number of user reviews that need to be analyzed efficiently. The problem at hand is the large volume of reviews that are difficult to analyze manually. The methods used include data preparation, handling class imbalance by applying SMOTE, and feature extraction with TF-IDF. Model evaluation is carried out by measuring accuracy, precision, recall, and F1-score to classify*

*review sentiment into positive or negative. The test results show that the two algorithms have almost equal performance with an accuracy of around 81%. These models are able to classify review sentiment well, have high precision and recall for both classes. The conclusion of this research is that the Machine Learning method has succeeded in providing an efficient and accurate solution for sentiment analysis in reviews of the Threads application on the Google Play Store, but the limitations of Indonesian language data are a concern for further research.*

**Keyword**—*Sentiment Analysis, Application Reviews, Multinomial Naive Bayes, Support Vector Machine, Google Play Store*

## I. PENDAHULUAN

Penggunaan aplikasi *mobile*, terutama jejaring sosial, telah menjadi bagian penting dalam kehidupan sehari-hari banyak orang. Aplikasi *Threads*, yang dikembangkan oleh perusahaan teknologi terkemuka, adalah salah satu aplikasi jejaring sosial yang populer. *Threads* adalah aplikasi pesan berbasis jejaring sosial yang dikembangkan oleh META, menawarkan fitur seperti pengiriman pesan teks, berbagi foto, dan video dengan pengguna lainnya, serta membagikan status atau *story* sebagai alat interaksi dengan anggota komunitas. Popularitas aplikasi *Threads* yang terus meningkat memungkinkan pengguna untuk meninggalkan ulasan dan penilaian tentang pengalaman mereka di *Google Play Store*, mencerminkan sentimen pengguna yang bisa berupa positif, atau negatif. Karena volume ulasan yang besar, tugas manual dalam menganalisis sentimen dari setiap ulasan menjadi tidak efisien. Oleh karena itu, diperlukan pendekatan otomatis yang efisien dan akurat, di mana analisis sentimen menggunakan metode *Machine Learning* menjadi solusi yang efektif.

Penelitian ini difokuskan pada analisis sentimen dari ulasan pengguna aplikasi *Threads* di *Google Play Store*. Dengan membandingkan dua metode *Machine Learning*, yaitu Metode Multinomial Naive Bayes dan *Support Vector Machine*, diharapkan dapat mencari metode dengan tingkat akurasi yang lebih baik dalam mengklasifikasikan sentimen ulasan sebagai positif atau negatif. Metode Multinomial Naive Bayes adalah

algoritma berdasarkan teorema Bayes dengan asumsi bahwa semua fitur independen satu sama lain, sementara *Support Vector Machine* adalah teknik pembelajaran mesin efektif untuk klasifikasi data linear dan non-linear. Selain itu, penelitian ini juga bertujuan untuk mengukur performa metode Multinomial Naive Bayes dan *Support Vector Machine* dalam analisis sentimen pada data *review* aplikasi *Threads* di Google Play Store.

Diharapkan dengan penelitian ini dapat meningkatkan pemahaman tentang penggunaan metode Multinomial Naive Bayes dan *Support Vector Machine* dalam analisis sentimen pada data tekstual. Hasil klasifikasi akan memberikan informasi mengenai kualitas pembelajaran dari data dan akurasi yang dapat digunakan untuk pengembangan masa depan.

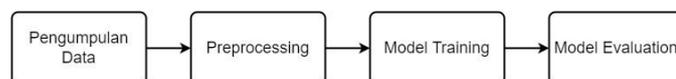
## II. METODE PENELITIAN

### A. Data Penelitian

Dataset atau data yang digunakan dalam penelitian ini bersumber dari ulasan aplikasi *Threads* di *Google Play Store* pada 23 Juli 2023, sejumlah 4352 data berhasil didapatkan dengan *sortir* ulasan terbaru. Data tersebut diperoleh menggunakan modul *google-play-scraper* yang merupakan *package* python.

### B. Tahapan Metode

Penerapan metode yang digunakan untuk analisis sentimen pada penelitian ini, terdapat beberapa tahapan. Tahapan tersebut merepresentasikan setiap proses dan rancangan dalam penelitian, dari awal hingga akhir. Tahapan yang dilakukan dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Metode

### C. Pengumpulan Data

Pengumpulan data dilakukan menggunakan teknik scraping dengan modul *google-play-scraper* yang merupakan *package* python. Pada penelitian ini akan menggunakan data yang bersumber dari ulasan aplikasi *Threads* di *Google Play Store*. Proses *scraping* dilakukan pada 23 Juli 2023 sejumlah 4352 data berhasil didapatkan dengan *sortir* ulasan terbaru.

### D. Preprocessing

Dataset kemudian dibersihkan melalui tahap *preprocessing*. Langkah ini perlu dilakukan agar dataset yang berupa teks dapat dibersihkan dari simbol yang tidak dibutuhkan, data yang tidak lengkap, data yang tidak valid, dan data dengan atribut yang tidak relevan. Tahap *preprocessing* yang akan diterapkan pada penelitian ini dapat dijelaskan sebagai berikut.

1) *Text Cleaning*: *Text Cleaning* merupakan proses untuk menghapus karakter selain a sampai z atau karakter selain huruf, karakter tersebut dihilangkan, seperti contoh 8266, / (garis miring), dan seterusnya sehingga hanya menyisakan karakter huruf saja[1]. Berikut pembersihan yang dilakukan

pada penelitian ini adalah sebagai berikut: Menghapus data teks selain huruf alphabet, Menghapus tab, *new line* dan *backspace*.

2) *Casefolding*: *Casefolding* adalah salah satu bentuk teks *preprocessing* yang paling sederhana dan efektif. Tujuan dari *casefolding* adalah untuk mengubah semua huruf kapital dalam teks menjadi huruf kecil.

3) *Tokenization*: Tahap berikutnya yaitu penerapan tokenisasi, teks yang sebelumnya berbentuk kalimat akan dipenggal per kata untuk menjadi satuan kata atau token. Dengan memberikan token dapat menghitung frekuensi distribusinya. Frekuensi distribusi token adalah jumlah kemunculan setiap token dalam teks atau dokumen, dengan mengetahui frekuensinya kita akan dapat menghitung seberapa sering token tersebut muncul dalam teks.

4) *Mengganti Slangword*: *Slangword* adalah kata yang tidak mengikuti sistem ejaan bahasa Indonesia yang baku (EYD), biasa ditemui berupa singkatan, kata gaul atau modern, dan kesalahan dalam ejaan[2]. Pada proses tersebut akan melibatkan kamus milik[3] yang bisa diunduh pada halaman github (<https://github.com/nasalsabila/kamus-alay>).

5) *Stopword Removal*: Proses ini akan menghapus kata-kata yang tidak memiliki makna atau arti pada saat kata tersebut berdiri sendiri dengan menggunakan algoritma *stoplist* (membuang kata kurang penting) atau *wordlist* (menyimpan kata penting)[4]. Pada penelitian ini akan menggunakan *stoplist* dengan kamus *stopword* dari Fadillah Z Tala[5]. Kamus tersebut yang akan digunakan sebagai pembanding dengan teks, jika terdapat kata yang termasuk dalam kamus maka kata tersebut akan dihapus[2].

6) *Stemming*: Kemudian proses terakhir *preprocessing* pada penelitian ini yaitu *stemming*. *Stemming* adalah proses yang bertujuan untuk mengubah kata-kata menjadi bentuk dasar (*lemma*) dengan menghapus imbuhan (*affixes*) seperti awalan (*prefixes*), akhiran (*suffixes*), dan infleksi lainnya. Tujuan utama dari *stemming* adalah untuk mengurangi variasi bentuk kata yang berbeda menjadi bentuk dasar yang sama.

7) *Mengganti Label*: Pada proses ini label bawaan yang dihasilkan dari metode *scraping* menggunakan *google-play-scraper* akan diganti yang awalnya terdapat label 1, 2, 3, 4, dan 5 menjadi dua label yaitu 1 untuk sentimen positif dan 0 untuk sentimen negatif. Label 1 ditentukan jika nilai sama atau lebih dari 4 dan label 0 ditentukan jika nilai sama atau kurang dari 3. Ini dilakukan untuk mempermudah melakukan sentimen analisis.

8) *TF-IDF*: TF-IDF adalah kependekan dari *Term Frequency-Inverse Document Frequency*. Sejatinnya, TF-IDF adalah gabungan dari 2 proses yaitu *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF). TF-IDF digunakan untuk mengubah data teks yang tidak dipahami komputer menjadi vector agar teks tersebut dapat dipahami komputer. Selain itu, TF-IDF tetap memperhatikan apakah sebuah kata bersifat informatif atau tidak. TF-IDF bekerja dengan memberikan nilai yang cenderung kecil kepada kata yang frekuensinya tinggi, sedangkan untuk kata yang frekuensinya rendah akan memiliki

nilai yang besar. Kata yang sering muncul disebut juga *Stopwords*, di mana kata tersebut tidak terlalu berpengaruh pada makna teks[6]. TF-IDF dirumuskan pada persamaan 1.

$$tf_{v,d} = \frac{n_{v,d}}{\text{jumlah total terms di dokumen}}$$

$$idf_d = \log\left(\frac{\text{jumlah dokumen}}{\text{jumlah dokumen dengan term } v}\right)$$

$$tfidf_{v,d} = tf_{v,d} \times idf_d \quad (1)$$

Dimana  $tf$  adalah frekuensi kemunculan kata pada sebuah dokumen dan  $idf$  merupakan *invers* dokumen frekuensi.

9) *SMOTE*: Data yang diambil seringkali memiliki ketimpangan antar kelasnya. Untuk mengakali data yang terdapat ketidakseimbangan maka dilakukan penerapan *SMOTE* yaitu *Synthetic Minority Over-Sampling Technique*. Teknik *SMOTE* ini digunakan untuk menyeimbangkan data antara *review* positif dan negatif. Pertama, dataset yang sudah diolah akan dibagi menjadi data latih dan data uji terlebih dahulu kemudian dilakukan penerapan *SMOTE* untuk data latih. Hasilnya berupa data latih awal yang menjadi minoritas ditambah dengan hasil *oversampling* dengan *SMOTE* [7].

#### E. Training Model

Setelah melalui proses *preprocessing*, langkah selanjutnya adalah melakukan *training* model. Pada tahap ini, seluruh dataset akan dilatih menggunakan metode *Multinomial Naive Bayes* dan *Support Vector Machine*.

1) *Multinomial Naive Bayes*: Merupakan metode klasifikasi dengan pembelajaran *supervised* menggunakan model probabilistik. *Multinomial Naive Bayes* dipengaruhi oleh serangkaian *term*, dengan kata lain jumlah *term* diperhitungkan. Peluang antara *term* satu dengan yang lain adalah independen (tidak bergantung)[8]. Proses dimulai dengan memasukkan data yang digunakan untuk pembelajaran kemudian dilakukan perhitungan peluang kemunculan suatu kelas pada data latih yang dilakukan menggunakan Persamaan 2 [1].

$$P(c) = \frac{N_c}{N_{doc}} \quad (2)$$

Keterangan:

C : Kategori atau kelas

doc : Dokumen

$N_c$  : Banyaknya kategori c pada dokumen latih

$N_{doc}$  : Banyaknya keseluruhan dokumen latih yang digunakan.

Perhitungan selanjutnya dari probabilitas bahwa kata  $i$  termasuk dalam kategori atau kelas tertentu dapat dilakukan dengan menggunakan Persamaan 3[1].

$$P(w_i, c) = \frac{\text{count}(w_i, c) + 1}{\sum_w \text{count}(w_i, c) + |V|} \quad (3)$$

Keterangan:

$w_i$ : Kata ke- $i$  dalam seluruh dokumen yang berkategori c

$\text{count}(w_i, c)$  : Jumlah kata tertentu yang muncul dalam suatu kategori atau kelas

$\sum_w \text{count}(w_i, c)$ : Jumlah seluruh kata pada kelas

$|V|$ : Merupakan jumlah seluruh kata unik pada kelas

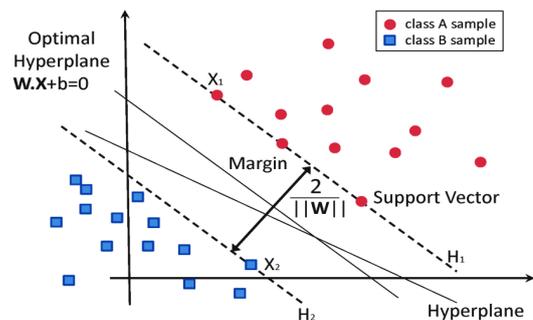
Penambahan nilai satu berfungsi untuk mencegah hasil probabilitas bernilai 0, manipulasi ini disebut dengan *laplace smoothing*. Setelah tahap pembelajaran selesai dilakukan maka selanjutnya dilakukan tahap klasifikasi data baru berdasarkan hasil pembelajaran. Untuk klasifikasi data baru dilakukan perhitungan menggunakan Persamaan 4 [1].

$$P(c, d) = P(c) \prod_{1 \leq i \leq \text{count}(v, d)} P(w_i, c) \quad (4)$$

Keterangan:

$\text{count}(v, d)$ : Jumlah kata unik pada dokumen

2) *Support Vector Machine*: *Support Vector Machine* merupakan satu di antara banyak algoritma yang digunakan untuk klasifikasi dan termasuk dalam kategori *supervised learning*. Konsep kerja *Support Vector Machine* yaitu dengan mencari *hyperplane* atau garis pembatas paling optimal yang berfungsi untuk memisahkan dua kelas[9]. Konsep kerja dari algoritma *Support Vector Machine* dapat dilihat pada Gambar 2.



Gambar 2. Konsep Kerja Support Vector Machine

*Support Vector Machine* menggunakan rumus matematis untuk menentukan *hyperplane* terbaik. Dalam kasus linear, rumusnya adalah sebagai berikut.

$w^T * x + b \geq 1$ , jika  $y = 1$  (positif)

$w^T * x + b \leq -1$ , jika  $y = -1$  (negatif) (5)

Dimana  $w$  adalah vektor bobot,  $x$  adalah vektor fitur,  $b$  adalah bias, dan  $y$  adalah label data.

#### F. Evaluasi Model

Terdapat dua metode untuk mengevaluasi model yaitu dengan *Confusion Matrix* dan *K-Fold Cross Validation*.

1) *Confusion Matrix*: *Confusion Matrix* adalah sebuah tabel yang menunjukkan jumlah prediksi yang benar dan salah yang dilakukan oleh model. *Confusion Matrix* terdiri dari empat angka yang ditunjukkan pada Gambar 3. *True Positive (TP)*: Jumlah prediksi positif yang benar, yaitu ketika model memprediksi sentimen positif dan memang benar. *True Negative (TN)*: Jumlah prediksi negatif yang benar, yaitu ketika model memprediksi sentimen negatif dan memang benar. *False Positive (FP)*: Jumlah prediksi positif yang salah, yaitu ketika model memprediksi sentimen positif padahal sebenarnya negatif. *False Negative (FN)*: Jumlah prediksi negatif yang

salah, yaitu ketika -model memprediksi sentimen negatif padahal sebenarnya positif.

		Prediksi	
		0	1
Aktual	0	True Negative (TN)	False Positive (FP)
	1	False Negative (FN)	True Positive (TP)

Gambar 3. Confusion Matrix

2) *K-Fold Cross Validation*: *Cross-validation* adalah sebuah teknik validasi model untuk menilai bagaimana hasil statistik analisis akan menggeneralisasi kumpulan data independen. Teknik ini utamanya digunakan untuk melakukan prediksi model dan memperkirakan seberapa akurat sebuah model prediktif ketika dijalankan dalam praktiknya. *k-fold cross validation* memecah data menjadi k bagian set data dengan ukuran yang sama. Penggunaan *k-fold cross validation* untuk menghilangkan bias pada data. Pelatihan dan pengujian dilakukan sebanyak k kali[10].

Pengujian Ke	Dataset									
1	█									
2		█								
3			█							
4				█						
5					█					
6						█				
7							█			
8								█		
9									█	
10										█

Gambar 4. K-Fold Cross Validation

Pada Gambar 4 merupakan penggunaan 10-fold cross validation. Dimana setiap data akan di eksekusi sebanyak 10 kali dan setiap subset data akan mempunyai kesempatan sebagai data *testing* atau data *training*. Model pengujian seperti berikut dengan diasumsikan nama setiap pembagian data yaitu D1, hingga D10.

Percobaan pertama data D1 sebagai data *testing* sedangkan D2 hingga D10 sebagai *data training*, kemudian percobaan kedua data D2 sebagai data *testing* sedangkan data D1, D3 hingga D10 sebagai *data training*. Percobaan akan dilakukan terus sampai data *testing* mencapai D10 dan *data training* D1 hingga D9.

### III. HASIL DAN PEMBAHASAN

Hasil dan pembahasan dari implementasi metode dalam penelitian ini dilakukan dengan berbagai tahapan, berikut adalah tahapan-tahapan yang digunakan.

#### A. Tahap Pengumpulan Data

Dataset penelitian bersumber dari aplikasi *Google Play Store* dan pada halaman aplikasi *Threads* yang berupa *review* atau ulasan dari pengguna aplikasi *Threads*. Dataset diperoleh dengan proses *scraping* dan menggunakan *package google-play-scraper* ditunjukkan pada Gambar 5. Terdapat 4352 data berhasil didapatkan berisi *rating* dan *review* pengguna yang diambil pada 23 Juli 2023.

	rating	review
1	4	pengalaman hari kedua setelah unduh threads, masih kurang handal untuk lampirkan foto atau video dengan
2	2	Aku menemukan 2 bug. Yang pertama saat unggah threads dengan lampiran foto resolusi tinggi, threads tidak
3	4	Sedikit saran beberapa fitur buat aplikasi ini... mungkin bisa di optimalkan fitur zoom foto. Soalnya kadang sus
4	5	Pemulaan bagus ada saingan aplikasi burung. Namun masih ada beberapa bug seperti mau nulis sesuatu
5	3	Aplikasi udah baik. Namun, saya menemukan bug ketika akan membuat thread. Dimana ketika saya pindahki
6	4	Aplikasi baru rilis, masih ada banyak bug dan kendala server seperti buffering homepage yang tertula lama pi
7	4	Cukup menarik sih sebagai alternatif twitter, login nya jg sangat simple... Animasi saat open close foto/video
8	2	setelah di upgrade ke Android 10 baru instal lagi nih aplikasi dan lancar... tapi pas ngekit balasan kolomny
9	3	sumpah nih apk bagus banget, cuma tampilannya aja kurang oke menurut ku. 1. Tombol Like koment dan shd
10	1	Masih banyak kurangnya... gk ada translate... Terus yg lewat di beranda kebanyakan centang biru semua... gk
11	4	Aplikasi nya udh bagus tapi ada beberapa fitur yang ditambahkan, seperi fitur untuk save foto, fitur translate, c
12	4	selama saya coba, aplikasinya bagus sih gampang dipakai karena Ultra simpel dan juga datarnya pake aku
13	3	Sejauh ini cukup bagus. Untuk algoritma feeds mungkin bisa dipertahankan seperi ini tapi dengan mengurur
14	3	Aku kasih bintang 3 dulu semoga bisa diperbaiki kedepannya, catatanku: belum ada fitur translate, beranda n
15	5	Bagus sih... Tapi kalau bisa tingkalkan lagi bisa untuk upload dari Instagram langsung bisa masuk ke Thread
16	4	Pengalaman saya selama ini di hp oppo realme c2 OS 9.0 pas mode gelap bawaan dari realme nya, aplikasi
17	4	Aplikasinya bagus, saya nyaman pakainya. Saya prediksi aplikasi ini bakal jadi app yang rame sih kedepann
18	2	Ketika di scroll semua content terulang seperi ketika anda berhasil memenangkan permainan solitaire pada
19	3	kecewa karena ada bug... saat saya mengscroll kebawah dan keatas, tampilan beranda Julian, teks teks, pot
20	3	Okeee apk nya udah lumayan bagus tapi mohon banget buat ditingkalkan fitur-fitur lain seperi - translate karen

Gambar 5. Dataset hasil scraping

#### B. Tahap Preprocessing

Tahap selanjutnya adalah *preprocessing* yang meliputi *text cleaning*, *casefolding*, *tokenization*, mengganti *slangword*, *stopword removal*, *stemming*, mengganti label, *tf-idf*, *SMOTE*. Hasil sebelum dan sesudah tahap *preprocessing* dapat dilihat pada Tabel 1, Gambar 6 menunjukkan sebelum tahap *SMOTE* dan Gambar 7 setelah tahap *SMOTE*.

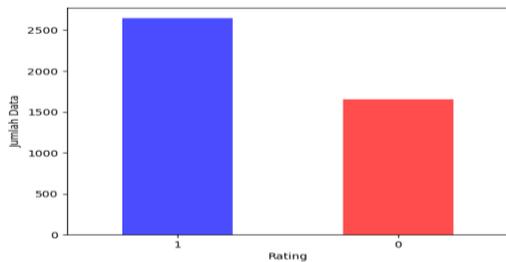
TABEL 1  
HASIL SEBELUM DAN SESUDAH TAHAP PREPROCESSING

Sebelum		Sesudah	
Rating	Review	Rating	Review
2	tolong diperbaiki masi ngebug parah	0	baik mas ngebug parah
5	Asik semoga makin lama gak makin aneh ya aplikasi yang ini	1	asik moga gak aneh aplikasi
4	Gw beri rating 4 dulu. Kalau makin jelek yaudah kasih rating 2. Klo makin bagus ya kasih 5	1	gw rating jelek yaudah kasih rating klo bagus kasih
3	Bintang 3 dulu, kalau bagus keknya ntar update. Kalau ndak lupa...	0	bintang bagus kek ntar update ndak lupa
1	Sayang sekali belum bisa multi akun seperti Instagram.	0	sayang multi akun instagram

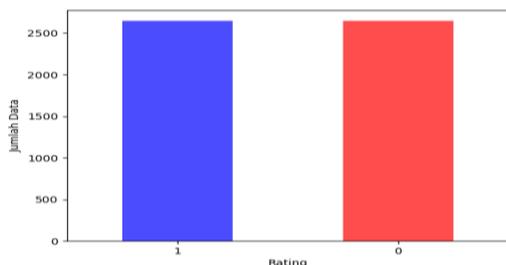
#### C. Tahap Training Model

Pada tahap *training model* dataset yang telah melakukan *preprocessing* kemudian akan dilatih dengan dua model

klasifikasi, yaitu Multinomial Naive Bayes dan *Support Vector Machine*.



Gambar 6. Sebelum Tahap SMOTE



Gambar 7. Sesudah Tahap SMOTE

#### D. Tahap Evaluasi Model

Setelah model dilatih akan didapatkan *confusion matrix* dari setiap metode yang digunakan.

1) *Confusion Matrix* untuk *Multinomial Naive Bayes* adalah sebagai berikut: *True Positive* (TP) adalah 2131, yang berarti model benar-benar memprediksi 2131 ulasan sebagai sentimen positif. *True Negative* (TN) adalah 2138, yang berarti model benar-benar memprediksi 2131 ulasan sebagai sentimen negatif. *False Positive* (FP) adalah 507, yang berarti model salah memprediksi 507 ulasan sebagai sentimen positif padahal sebenarnya negatif. *False Negative* (FN) adalah 514, yang berarti model salah memprediksi 514 ulasan sebagai sentimen negatif padahal sebenarnya positif. *Confusion Matrix* untuk *Multinomial Naive Bayes* ditunjukkan Tabel 2.

TABEL 2  
CONFUSION MATRIX MULTINOMIAL NAIVE BAYES

	Prediksi Kelas 0	Prediksi Kelas 1
Kelas 0 (Aktual)	2138	507
Kelas 1 (Aktual)	514	2131

2) *Confusion Matrix* untuk *Support Vector Machine* adalah sebagai berikut: *True Positive* (TP) adalah 2049, yang berarti model benar-benar memprediksi 2224 ulasan sebagai sentimen positif. *True Negative* (TN) adalah 2224, yang berarti model benar-benar memprediksi 2224 ulasan sebagai sentimen negatif. *False Positive* (FP) adalah 421, yang berarti model salah memprediksi 421 ulasan sebagai sentimen positif padahal sebenarnya negatif. *False Negative* (FN) adalah 596, yang berarti model salah memprediksi 596 ulasan sebagai sentimen negatif padahal sebenarnya positif. *Confusion Matrix* untuk *Support Vector Machine* ditunjukkan pada Tabel 3.

TABEL 3.  
CONFUSION MATRIX SUPPORT VECTOR MACHINE

	Prediksi Kelas 0	Prediksi Kelas 1
Kelas 0 (Aktual)	2224	421
Kelas 1 (Aktual)	596	2049

Kemudian akan diuji dengan *K-Fold Cross Validation* membagi data pelatihan menjadi 10 subset dengan ukuran yang sama, di mana dalam setiap iterasinya, salah satu subset dijadikan sebagai data pengujian (*holdout set*), sedangkan sisanya menjadi data pelatihan. Kemudian, model dilatih menggunakan data pelatihan dan dievaluasi pada data pengujian. Proses ini diulangi sebanyak 10 kali, sehingga setiap subset akan menjadi data pengujian satu kali dan data pelatihan sembilan kali. Hasil evaluasi pada setiap iterasi diambil rata-ratanya untuk mendapatkan kinerja keseluruhan model.

TABEL 4.  
HASIL PENGUJIAN MODEL

Algoritma	Akurasi	Presi si Kelas 0	Presi si Kelas 1	Reca ll Kela s 0	Reca ll Kela s 1	F1- scor e Kela s 0	F1- scor e Kela s 1
<i>Multinomial Naive Bayes</i>	81%	0.81	0.81	0.81	0.81	0.81	0.81
<i>Support Vector Machine</i>	81%	0.79	0.83	0.84	0.77	0.81	0.80

Hasil pengujian model pada Tabel 4, dengan menggunakan dua algoritma, yaitu *Multinomial Naive Bayes* dan *Support Vector Machine*, menunjukkan hasil yang cukup baik dengan tingkat akurasi yang sama, yaitu sebesar 81%. Dalam hal presisi, keduanya memiliki performa yang cukup tinggi. Untuk kelas 0, presisi klasifikasi dari kedua model adalah 0.81, yang berarti 81% dari data yang diprediksi sebagai kelas 0 adalah benar. Sedangkan untuk kelas 1, presisi *Multinomial Naive Bayes* adalah 0.81, dan presisi *Support Vector Machine* adalah 0.83, yang berarti 81% dan 83% dari data yang diprediksi sebagai kelas 1 adalah benar. Pada pengukuran *recall*, kelas 0 *Multinomial Naive Bayes* memiliki nilai 0.81 sedangkan pada model *Support Vector Machine* lebih tinggi dengan nilai 0.84 yang berarti 81% dan 84% dari seluruh data kelas 0 berhasil ditemukan oleh kedua model. Untuk kelas 1, juga terdapat perbedaan antara kedua model. *Recall* dari kelas 1 pada model *Multinomial Naive Bayes* adalah 0.81, sedangkan pada model *Support Vector Machine* adalah 0.77, yang berarti 81% dan 77% dari seluruh data kelas 1 berhasil ditemukan oleh masing-masing model. Hasil evaluasi *F1-score* menunjukkan keseimbangan antara presisi dan *recall* untuk kedua kelas pada kedua model. Keduanya memiliki *F1-score* yang sama, yaitu 0.81 untuk kelas 0 dan 0.80 untuk kelas 1.

#### IV. KESIMPULAN

Hasil evaluasi menunjukkan bahwa kedua algoritma memiliki performa hampir seimbang dengan akurasi 81%. Presisi untuk kelas 0 pada kedua algoritma adalah 0.81, sedangkan presisi untuk kelas 1 pada *Multinomial Naive Bayes*

adalah 0.81 dan pada *Support Vector Machine* adalah 0.83. *Recall* untuk kelas 0 adalah 0.81 untuk Multinomial Naive Bayes dan 0.84 untuk *Support Vector Machine*. Pada kelas 1 juga terdapat perbedaan, yaitu *recall* kelas 1 pada Multinomial Naive Bayes adalah 0.81, sedangkan pada *Support Vector Machine* adalah 0.77. Keduanya memiliki *F1-score* yang sama, yaitu 0.81 untuk kelas 0 sedangkan kelas 1 pada Multinomial Naive Bayes adalah 0.81 dan kelas 1 pada *Support Vector Machine* adalah 0.80. Dengan hasil tersebut dapat disimpulkan bahwa tahap *preprocessing* sangat berpengaruh terhadap performa yang dihasilkan. Tahapan mengganti *slangwords* akan berpengaruh terhadap fitur yang nantinya akan dihasilkan dari tahapan ekstraksi fitur. Kamus kata yang banyak dapat mengurangi redundansi fitur yang memiliki makna atau kata baku yang sama. Untuk penelitian selanjutnya, penulis memberi saran untuk memperbanyak kamus kata pada *slangwords* dan *stopword* agar hasil dari tahap *preprocessing* bisa menjadi lebih baik lagi.

#### REFERENSI

- [1] F. Hadaina and U. Budiyo, "Implementasi Metode Multinomial Naive Bayes Untuk Sentiment Analysis Terhadap Data Ulasan Produk Colearn Pada Google Play Store," in *Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI) Jakarta-Indonesia*, pp. 660–666, 2022.
- [2] V. R. A. Sejati, P. Painem, F. Ferdiansyah, and W. Pramusinto, "Analisis Sentimen Ulasan Pengguna Aplikasi Ipusnas pada Google Play Store dengan Multinomial Naive Bayes," in *2nd Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI)*, pp. 182–190, 2023.
- [3] N. A. Salsabila, Y. A. Winatmoko, and A. A. Septiandri, "Colloquial Indonesian Lexicon," in *2018 International Conference on Asian Language Processing (IALP)*, pp. 236–239, 2018.
- [4] K. Kelvin, J. Banjarnaho, E. Indra, and S. H. Simurat, "Analisis Perbandingan Sentimen Corona Virus Disease-2019 (Covid19) pada Twitter Menggunakan Metode Logistic Regression dan Support Vector Machine (SVM)," *JUSIKOM PRIMA (Jurnal Sistem Informasi dan Ilmu Komputer Prima)*, vol. 5, no. 2, pp. 47–52, 2022.
- [5] F. Z. Tala, "A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia."
- [6] M. Zaki Anbari, M. Zaki Anbari, and B. Sugiantoro, "Studi Komparasi Metode Analisis Sentimen Naive Bayes, SVM, dan Logistic Regression pada Piala Dunia 2022," *Jurnal Media Informatika Budidarma*, vol. 7, no. 2, pp. 688–695, 2023.
- [7] Q. A. N. Prakoso, A. Muliawati, and I. N. Isnainiyah, "Analisis Sentimen terhadap Produk Skin Game di Forum Review Female Daily Menggunakan Metode Multinomial Naive Bayes dan TF-IDF," *JURNAL INFORMATIK*, vol. 18, no. 3, pp. 198–207, 2022.
- [8] F. S. D. Kartika and T. P. Aji, "Klasifikasi Berita Menggunakan Metode Multinomial Naive Bayes," *SCAN*, vol. XVI, no. 3, pp. 1–8, 2021.
- [9] H. C. Husada and A. S. Paramita, "Analisis Sentimen Pada Maskapai Penerbangan di Platform Twitter Menggunakan Algoritma Support Vector Machine (SVM)," *Teknika*, vol. 10, no. 1, pp. 18–26, Feb. 2021.
- [10] F. Tempola, M. Muhammad, and A. Khairan, "Perbandingan Klasifikasi Antara KNN dan Naive Bayes pada Penentuan Status Gunung Berapi Dengan K-Fold Cross Validation," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 5, no. 5, pp. 577–584, 2018.