

Klasifikasi Teks Ulasan Aplikasi Netflix Pada *Google Play Store* Menggunakan Algoritma Naive Bayes dan SVM

Nabilah Khoirunnisaa¹, Kaylista Nabila Nastiti Kesuma²,
Septiyanthi Setiawan³, Ajif Yunizar Pratama Yusuf^{4*}

^{1,2,3,4} Ilmu Komputer, Informatika, Universitas Bhayangkara Jakarta Raya, Bekasi, Indonesia

E-mail: ¹202010225270@mhs.ubharajaya.ac.id, ²202010225160@mhs.ubharajaya.ac.id,

³202010225055@mhs.ubharajaya.ac.id, ^{4*}ajif.yunizar@dsn.ubharajaya.ac.id

(* : corresponding author)

Abstrak

Netflix ialah platform *streaming* berlangganan yang menyajikan beragam acara seperti tv series, dokumenter, serta film yang terhubung pada sebuah perangkat yang terkoneksi dengan internet. Netflix salah satu platform aplikasi *streaming* video yang digunakan paling banyak di seluruh dunia dan kini mulai menerapkan teknologi analisis data dan *machine learning* untuk meningkatkan layanan penggunaannya. Melalui *Google Play Store*, pengguna dapat menyampaikan berbagai macam ulasan mengenai aplikasi Netflix. Dari banyaknya data ulasan tersebut, dapat dimanfaatkan untuk mengekstrak informasi penting yang tersembunyi dan bermanfaat untuk mengevaluasi kualitas suatu aplikasi. Oleh karena itu, penelitian ini bertujuan guna mengklasifikasikan ulasan teks aplikasi Netflix dengan membandingkan dua algoritma yang diterapkan yakni, *Support Vector Machine* (SVM) serta *Naive Bayes*. Dengan maksud untuk mendapati manakah algoritma yang lebih efektif dari segi akurasi. Dataset diperoleh melalui *Google Play Store* dan menerapkan metode *scraping*, sebanyak 1000 ulasan dan diolah menggunakan bahasa pemrograman Python. Kemudian data ulasan aplikasi Netflix yang telah didapatkan dipecah menjadi *train data* 70% dan *test data* 30%. Penerapan algoritma *Naive Bayes* mendapatkan hasil *accuracy* senilai 82%, lalu *Support Vector Machine* (SVM) memperoleh hasil *accuracy* senilai 85%. Sehingga, menunjukkan hasil penerapan algoritma *Naive Bayes* tidak lebih efektif dibandingkan *Support Vector Machine* (SVM).

Kata kunci: Klasifikasi Teks, *Google Play Store*, *Naive Bayes*, Netflix, *Support Vector Machine* (SVM)

Abstract

Netflix is a subscription streaming platform that presents various shows, such as TV series, documentaries, and films, connected to a device connected to the internet. One of the most popular sites for streaming videos is Netflix, throughout the world and is now starting to apply data analysis and machine learning technology to improve its user services. Through the *Google Play Store*, users can submit various reviews about the Netflix application. It is possible to extract significant hidden information from this vast quantity of review data that is helpful for assessing an application's quality. Therefore this research aims to classify text reviews of the Netflix application by comparing the two algorithms applied, that is, *Support Vector Machine* (SVM) and *Naive Bayes*. With the aim of finding out which algorithm performs better in terms of accuracy. The dataset was obtained through the *Google Play Store* and applied to the *scraping* method, totaling 1000 reviews, and processed utilizing the Python programming language. Then the Netflix application review data that was obtained was divided into 70% *train data* and 30% *test data*. 82% of the accuracy results were obtained using the *Naive Bayes* approach., while the support vector machine (SVM) yielded 85% accuracy. It therefore demonstrates that support vector machines (SVM) are no more successful than the outcomes of applying the *Naive Bayes* method.

Keywords: Text Classification, *Google Play Store*, *Naive Bayes*, Netflix, *Support Vector Machine* (SVM)

1. PENDAHULUAN

Perkembangan era digital yang begitu pesat, aplikasi perangkat lunak menjadi bagian utama dari kehidupan sehari-hari yang tidak terpisahkan bagi para penggunanya. *Google Play Store* menyediakan ratusan ribu aplikasi yang bisa di-*download* di perangkat *mobile* dengan *system* android. Seiring dengan popularitas aplikasi, ulasan pengguna di platform distribusi seperti *Google Play Store* menjadi penilaian utama untuk mengevaluasi kualitas dan kepuasan pengguna terhadap suatu aplikasi.

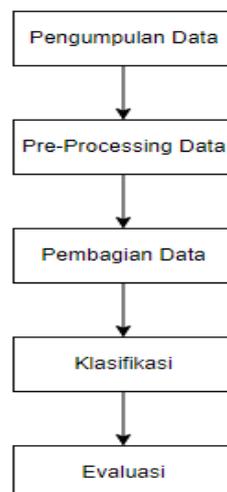
Salah satu aplikasi yang mendapatkan perhatian besar dalam hal ulasan pengguna adalah *Netflix*, platform penyedia konten *streaming* terkemuka. *Netflix* didirikan pada tahun 1997 dan kini sudah menyebar ke lebih dari 190 negara termasuk Indonesia. Aplikasi *Netflix* menawarkan berbagai macam acara, seperti film dokumenter, serial televisi, dan film dari berbagai bahasa serta gaya. Aplikasinya memungkinkan pengguna melakukan *streaming* setiap saat dengan syarat terhubung ke jaringan internet. Ini dapat dilakukan melalui smartphone, TV pintar, PC, atau laptop [1].

Pada *Google Play Store* hingga bulan Desember 2023, lebih dari 1 Miliar kali unduhan dan tercatat 14,4 juta ulasan pengguna di kolom komentar untuk aplikasi *Netflix*. Ulasan dari pengguna pada *Google Play Store*, umumnya terdiri dari ulasan yang mengandung sentimen positif dan negatif antara lain berupa saran atau kritik serta keluhan yang ditujukan pada aplikasi *Netflix*. Ulasan-ulasan tersebut memuat pendapat dari pengguna aplikasi *Netflix* sehingga pengguna baru akan melihat *review* dari aplikasi *Netflix* sebagai bagian dari menimbang apakah akan mengunduh dan menggunakannya atau tidak [2].

Ulasan yang terdapat pada *Google Play Store* sendiri sangat banyak, sangat beragam serta tidak terstruktur. Untuk memudahkan pengguna mencari *review* mengenai suatu program yang tersedia di *Netflix*, maka penelitian ini bertujuan untuk mengklasifikasikan teks ulasan yang terdapat pada *Google Play Store* terhadap aplikasi *Netflix*. Dalam *data mining*, klasifikasi teks diperuntukan untuk menemukan atau menata *class* di dalam suatu dokumen. Klasifikasi ini dibedakan berdasarkan berbagai karakteristik sehingga membuat model dapat diterapkan pada data yang akan diujiakan[3]. Diterapkan dua algoritma guna klasifikasi, yakni *Support Vector Machine* (SVM) dan *Naive Bayes*. Perbandingan antara keduanya ini diharapkan mampu memberikan pemahaman yang kian mendalam mengenai keunggulan dan keterbatasan masing-masing algoritma. Penelitian ini menghasilkan keluaran hasil algoritma mana untuk diketahui yang bekerja lebih baik dari segi tingkat akurasi dalam klasifikasi teks ulasan aplikasi *Netflix* pada *Google Play Store*.

2. METODE PENELITIAN

Salah satu yang terdapat dalam *data mining* ialah pengolahan *text mining*. *Text mining* bertujuan untuk mendapatkan informasi dari sekumpulan dokumen yang akan dipergunakan dalam proses klasifikasi itu sendiri. *Text mining* dapat diolah memanfaatkan bagian klasifikasi, yaitu pengkategorian teks (*text categorization*) dan pengelompokan teks (*text clustering*) [4]. Pada proses *text mining*, menggunakan data yang bersumber dari sekumpulan teks dalam bentuk yang tidak terstruktur atau setidaknya sebagian terstruktur. Gambar 1 menampilkan urutan alur yang akan digunakan dalam penelitian ini.



Gambar 1. Tahap-tahap Penelitian

2.1. Pengumpulan Data

Selanjutnya, ialah mengumpulkan data guna pengolahan lebih lanjut. Dalam hal ini, dikumpulkan data yang berbentuk teks ulasan mengenai aplikasi *Netflix* pada *Google Play Store* menggunakan teknik *scraping* dengan perintah *google-play-scrapers*. Data ulasan aplikasi *Netflix* yang dikumpulkan berjumlah sebanyak 1000 ulasan dan *rating* dari bintang 1 sampai bintang 5.

2.2. Pre-Processing Data

Pada tahapan selanjutnya ialah *pre-processing data*. Proses yang begitu mendalam dalam menentukan kualitas untuk tahapan selanjutnya yaitu pada tahap klasifikasi ialah tahapan *pre-processing* ini. Pada tahap *pre-processing*, data yang telah didapatkan sebelumnya akan diproses untuk memperoleh data yang dibutuhkan lalu dilakukan *cleaning data* atau membersihkan data-data yang tidak diperlukan seperti *missing value* dan *whitespace*[5]. Tujuannya adalah untuk membuang *noise* serta memperbaiki data menjadi lebih terstruktur.

2.3. Pembagian Data

Pada tahapan berikutnya ialah pembagian / *split data*. Pada dasarnya pembagian data dipisahkan ke dalam dua bagian, yakni data uji dan data latih. Data latih (*training*) membantu pembuatan model, sementara data pengujian menakar hasil kerja model pada data yang sebelumnya belum sempat terlihat. Split data dapat dipecah menjadi perbandingan 8:2, 7:3, dan 6:4.

2.4. Klasifikasi Teks

Machine learning terdapat dua tipe yaitu *supervised* dan *unsupervised*. *Supervised* adalah model *machine learning* yang menggunakan data berlabel. Sementara *unsupervised* adalah model *machine learning* yang mempelajari pola data tanpa adanya target / label data. Klasifikasi teks termasuk kedalam tipe *supervised learning* yaitu proses mengklasifikasikan dokumen-dokumen yang telah diinterpretasikan sebelumnya ke dalam setidaknya satu atau lebih kelompok dari kelas-kelas yang sama[6].

Klasifikasi teks adalah suatu metode dalam bidang pemrosesan bahasa alami (NLP) dan *machine learning* yang bertujuan untuk mengelompokkan atau menetapkan label kepada teks berdasarkan karakteristik tertentu. Mengelompokkan data ke dalam kategori yang telah ditetapkan sebelumnya juga digunakan dalam klasifikasi teks untuk memprediksi data dari kelas yang sebelumnya belum diketahui[7].

Teknik klasifikasi melalui penerapan *Naive Bayes Algorithm* dan *Support Vector Machine* (SVM) termasuk ke dalam *supervised learning* yang diterapkan dalam penelitian ini. Oleh sebab itu, memungkinkan setiap data *training* model yang dipergunakan adalah yang telah dilabeli.

2.5. Naive Bayes

Dalam metode *Machine Learning* salah satu algoritmanya ialah *Naive Bayes*, yang banyak digunakan dalam pemrosesan bahasa alami (NLP). Seorang ilmuwan asal Inggris yang mempublikasikan teori Bayes bernama Thomas Bayes, merupakan suatu teori guna memperkirakan kemungkinan di masa depan yang berdasar pada kejadian di masa lalu. Penerapan teori ini, yaitu pengalaman di masa lalu diibaratkan sebagai pelatihan data, dan pengibaratan masa depan sebagai penilaian data[8].

Dengan asumsi bahwa atribut-atributnya memiliki sifat yang bebas (*independent*), pengklasifikasian *Naive Bayes* disederhanakan melalui penggunaan model probabilitas dan statistik[7]. Menurut klasifikasi *Naive Bayes*, ada atau tidaknya suatu karakteristik kelas tertentu tidak terkait dengan karakteristik kelas lainnya. Keunggulan dari *Naive Bayes* adalah paling simpel, tangkas, dan memiliki ketepatan yang cukup tinggi. *Naive Bayes* sering kali menghasilkan hasil yang baik meski sederhana, terutama pada tugas klasifikasi teks.

Naive Bayes memiliki rumus :

$$P(H|X) = \frac{P(X|H) P(H)}{P(X)} \dots \dots (1)$$

Keterangan:

X = Data yang memiliki kelas yang tidak diketahui

H = Asumsi data X melambangkan kelas tertentu atau spesifik

$P(H/X)$ = Peluang asumsi H berlandaskan keadaan X (*posterior probability*)

$P(H)$ = Peluang asumsi H (*prior probability*)

$P(X/H)$ = Peluang asumsi X berlandaskan keadaan H (*likelihood*)

$P(X)$ = Peluang dari X (*predictor prior probability*)

2.6. Support Vector Machine (SVM)

SVM memperkirakan permasalahan klasifikasi dan regresi. Dasar prinsip SVM ialah sebagai *linear classification*, dengan artian kasus *linear classification* mampu dipisahkan. Namun, SVM telah dibangun guna memecahkan permasalahan *non-linier*, ke dalam ruang kerja yang luas yakni melalui memasukkannya konsepsi kernel[9]. Jika *hyperplane* mengklasifikasikan dataset secara *linier* maka algoritmanya kita sebut sebagai *Support Vector Classifier (SVC)* dan algoritma yang memisahkan dataset dengan pendekatan *non-linier* maka kita sebut sebagai *Support Vector Machine (SVM)*. *Hyperplane* merupakan bagian dari tugas yang dipergunakan sebagai partisi antar kelas yang ada.

Pada karakteristik ruang, SVM dapat dilihat sebagai *hyperplane*. Ini karena titik-titik yang menunjukkan jenis positif dan negatif terpisah satu sama lain. Prinsip kerja SVM adalah untuk membuat *hyperplane* dengan ukuran margin yang sama dan kecenderungan untuk tidak sebanding area kelas tertentu. Hal ini dapat dicapai dengan mengukur margin dan kemudian melakukan pencarian titik maksimalnya. [7].

Berbeda dengan SVM, SVC sendiri mencari *hyperplane* yang dapat memartisi kelas dengan menerapkan kriteria kesalahan minimal. Sedangkan SVM mencari *hyperplane* yang memaksimalkan margin di antara kelasnya. SVC sendiri hanya berusaha dalam meminimalkan kekeliruan klasifikasi dan tidak memaksimalkan margin, menggunakan teknik optimasi matematis.

Rumus dalam menentukan *hyperline* adalah sebagai berikut:

$$w \cdot x + b = 0 \rightarrow w_0 + w_1 x_1 + w_2 x_2 \dots \dots \dots (2)$$

Keterangan:

$W = \{w_1, w_2, \dots, w_n\} \rightarrow$ bobot

n = Jumlah atribut

b = Skalar yang digunakan sebagai bias

x = Nilai atribut

Dalam penelitian ini, ialah menerapkan model pendekatan kernel linier atau sama saja dengan *Support Vector Classifier (SVC)*. Rumusnya adalah sebagai berikut:

$$k(x, z) = \varphi(x)^t \varphi(z) \dots \dots \dots (3)$$

Keterangan:

k = Fungsi kernel untuk semua vektor *input* dan a

φ = Fungsi pemetaan dari area *input* ke area fitur

2.7. Evaluasi

Setelah model dibuat menggunakan klasifikasi Naive Bayes dan SVM, selanjutnya ialah tahap evaluasi akan dilakukan. Penelitian ini menggunakan evaluasi untuk mengevaluasi keakuratan metode klasifikasi teks. Evaluasi juga mengevaluasi kinerja atau performa sistem. Penelitian ini menerapkan evaluasi dengan menggunakan *confusion matix*. Metode ini dapat menemukan *f1-score*, akurasi, presisi, dan *recall* [10].

Metode *confusion matrix* diterapkan guna menjalankan akurasi perhitungan pada gagasan *data mining* [11]. *Confusion matrix* juga digunakan untuk memprediksi kata guna membandingkan dari data input kelas yang asli. *Confusion matrix* ini terdiri dari kelas yang diprediksi dan kelas yang sebenarnya. Nilai sebenarnya yang dimiliki adalah data aktual, sedangkan data prediksi adalah nilai yang diperoleh dari hasil pemodelan pembelajaran mesin yang terlihat pada Tabel 1 berikut:

Tabel 1. *Confusion Matrix*

		<i>Actual Class</i>	
		<i>Positive</i>	<i>Negative</i>
<i>Predicted Class</i>	<i>Positive</i>	<i>True Positive</i>	<i>False Negative</i>
	<i>Negative</i>	<i>False Positive</i>	<i>True Negative</i>

Keterangan:

- True Positive (TP)* : Total *sample* yang diklasifikasikan dengan benar menjadi positif
- True Negative (TN)* : Total *sample* yang diklasifikasikan dengan benar menjadi negatif
- False Positif (FP)* : Total *sample* yang seharusnya diklasifikasikan menjadi negatif, tetapi salah
- False Negative (FN)* : Total *sample* yang seharusnya diklasifikasikan menjadi positif, tetapi salah diklasifikasikan sebagai negatif

Berdasarkan *confusion matrix*, didapatkan berbagai parameter dari pengukuran kinerja suatu algoritma, yaitu *dan f1-score, accuracy, precision, serta recall*[12]. *Accuracy* adalah persentase keseluruhan sampel yang diklasifikasikan dengan benar oleh suatu model, *Precision* adalah proporsi dari prediksi positif yang kenyataannya benar, *Recall* ialah proporsi dari hal positif hingga diklasifikasikan dengan benar oleh model, dan *f1-score* ialah ukuran kombinasi dari presisi dan *recall*.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \dots \dots (4)$$

$$Precision = \frac{TP}{TP + FP} \dots \dots (5)$$

$$Recall = \frac{TP}{TP + FN} \dots \dots (6)$$

$$F1 - score = \frac{2 \times precision \times recall}{precision + recall} \dots \dots (7)$$

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Data

Tahap pengumpulan ulasan menggunakan cara *web scraping* menerapkan perintah *google-play-scraping* pada *Google Play Store* di bahasa pemrograman python. Data yang terkumpul merupakan data teks berupa *review* (komentar) dan *rating* yang diberikan oleh pengguna aplikasi *Netflix* dengan jumlah sebanyak 1000 ulasan dan *rating* dari bintang 1 sampai bintang 5. Dataset tersebut terdiri dari 11 atribut yaitu *userImage, appVersion, reviewCreatedVersion, userName, score, reviewId, thumbsUpCount, content, at, dan repliedAt, replyContent*. Gambar 2 menunjukkan dataset yang akan digunakan.

reviewId	userName	userImage	content	score	thumbsUpCount	reviewCreatedVersion	at	replyContent	repliedAt	app\
4ca988b0-3d40-471c-be2e-7565e813963	gilang Bossq	lh.googleusercontent.com/a/ACg8oc...	Kenapa dicemirkan ke tv digital tidak keluar ...	3	309	8.96.1 build 16 50568	2023-12-13 20:05:01	None	None	8.96 16
27b8abd0-9ad0-45e0-b9aa-lb6c08b58379	Binta Andreina	lh.googleusercontent.com/a-/ALV-U...	Tolong dong DIPERHATIKAN pihak netflix, ini ud...	1	1165	8.94.0 build 10 50546	2023-11-10 19:48:12	None	None	8.94 10
bbea55ca-70a8-4148-a836-1a08581329d4	Humanity	lh.googleusercontent.com/a-/ALV-U...	Dari saya sendiri ini aplikasi netflix menupak...	4	130	8.96.1 build 16 50568	2023-12-20 18:28:35	None	None	8.96 16
1f367e58-dd0e-4f6c-ab33-16a304418ac0	Simpan File	lh.googleusercontent.com/a/ACg8oc...	Kenapa ya,, belakangan ini film di netflix ada...	3	1	8.96.1 build 16 50568	2023-12-23 17:17:06	None	None	8.96 16
fabaf084-ddb0-4fa4-ba48-19b25421c21a	Aulia Kusuma Putri	lh.googleusercontent.com/a-/ALV-U...	habis di update bukannya makin bagus malah mak...	1	180	8.95.2 build 17 50558	2023-12-02 10:21:04	None	None	8.95 17

Gambar 2. Pengumpulan Data

3.2 Pre-Processing Data

Setelah data diperoleh, tahap selanjutnya ialah dilakukannya proses untuk menyiapkan data atau tahap *pre-processing*. Tahapan *pre-processing data* ialah bertujuan untuk menghapus beberapa atribut yang tidak digunakan. Sehingga, data yang dipilih untuk penelitian ini hanya terdiri dari 2 atribut yaitu *content* dan *score* dengan proses *filtering*. *Content* merupakan ulasan pengguna *Netflix* dan *score* merupakan *rating* yang diberikan oleh pengguna *Netflix*. Pada Gambar 3 ditampilkan data yang telah dilakukan proses *filtering*.

	content	score
148	Netflix parah banget, sudah berlangganan malah...	1
415	parah banget padahal yang buka cuma gua doang ...	1
185	Audio bahasa indonesia banyak tidak ada	1
19	Baru aja bayar bulan ke 2 udah begini ga bisa ...	1
3	Kenapa ya,, belakangan ini film di netflix ada...	3

Gambar 3. Filtering Data

Selanjutnya akan dilakukan pelabelan data yang ditunjukkan oleh Gambar 4. Pelabelan data ini dilakukan agar data siap diklasifikasikan. Pelabelan data biasanya berupa kategori atau kelas yang menunjukkan keanggotaan suatu entitas data ke dalam kelompok tertentu. *Rating* atau komentar yang dituangkan pengguna kepada aplikasi *Netflix* digunakan untuk melabelkan data dalam penelitian ini. Semua label dibagi menjadi tiga kelompok, kelompok dengan *rating* 1-2 menunjukkan hasil negatif, kelompok dengan *rating* 3 menunjukkan hasil ambigu (akan dihilangkan), dan kelompok dengan *rating* 4-5 menunjukkan hasil positif. *Rating* 3 akan dihilangkan karena mencakup ulasan yang tidak termasuk positif maupun negatif sehingga tidak akan digunakan dalam penelitian ini.

	content	score	Label
0	Netflix parah banget, sudah berlangganan malah...	1	Negatif
1	parah banget padahal yang buka cuma gua doang ...	1	Negatif
2	Audio bahasa indonesia banyak tidak ada	1	Negatif
3	Baru aja bayar bulan ke 2 udah begini ga bisa ...	1	Negatif
4	Kenapa ya,, belakangan ini film di netflix ada...	3	None
5	Tidak bisa mengatur bahasa saya masuk dengan m...	2	Negatif
6	Aplikasi sampah g guna, g bisa daftar, blokir ...	1	Negatif
7	Aplikasi ini sangat bagus dan menarik untuk pe...	5	Positif
8	Susah benar daftar nya disini, dari awal nya s...	1	Negatif
9	Netflix lg kenapa ya ko gabisa muter film nya ...	1	Negatif

Gambar 4. Pelabelan Data

Selanjutnya akan dilakukan *cleaning data* atau pembersihan data. *Cleaning data* dikerjakan guna membersihkan data-data yang tidak berguna seperti *missing value* dan *whitespace* yang bertujuan untuk membuang *noise* dan membuat data lebih terstruktur. Sebelum melakukan *cleaning data*, perintah *info()* akan menunjukkan informasi lengkap mengenai *dataframe*, seperti jumlah baris data, nama-nama kolom, jumlah data, dan tipe data. Gambar 5 menampilkan data sebelum dilakukan proses *cleaning*.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   content    1000 non-null   object
1   score      1000 non-null   int64
2   Label      902 non-null    object
dtypes: int64(1), object(2)
memory usage: 23.6+ KB
```

Gambar 5. Sebelum *Cleaning Data*

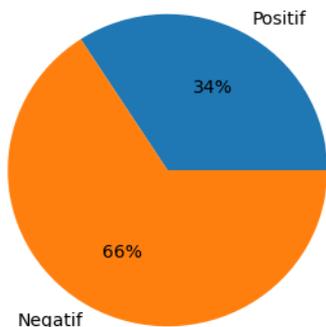
Selanjutnya dilakukan pengecekan berapa banyak terdapat *missing value*. Lalu, diketahui terdapat 98 *record* yang berisi nilai *null* (*missing value*). Maka harus dilakukan *cleaning data* untuk menghapus *missing value* tersebut. Sehingga diperoleh 902 *record* data dari sebelumnya 1000 *record* data, yang ditunjukkan oleh Gambar 6.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 902 entries, 0 to 901
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   content    902 non-null    object
1   score      902 non-null    int64
2   Label      902 non-null    object
dtypes: int64(1), object(2)
memory usage: 21.3+ KB
```

Gambar 6. Setelah *Cleaning Data*

Dikarenakan *class* untuk *rating* 3 (ambigu) telah dihilangkan, sehingga dari 902 data didapatkan hasil sebanyak 308 data yang tergolong *class* positif, dan sebanyak 594 data *class* negatif. Persentase *class* ditunjukkan oleh Gambar 7.

Persentase Text Analysis Netflix Reviews



Gambar 7. Persentase *Class* Positif dan Negatif

3.3 Pembagian Data

Data ulasan aplikasi *Netflix* dipecah ke dalam dua bagian, *training data* (data pelatihan) serta *testing data* (data uji). Untuk penelitian ini, modul *sklearn* yakni *train_test_split* digunakan pada python. Dari 902 data bersih yang diterapkan, 631 adalah data pelatihan dan 271 adalah data pengujian.

3.4 Evaluasi Klasifikasi

Kedua algoritma yang diterapkan ialah dipergunakan sebagai algoritma untuk klasifikasi, SVM serta *Naive Bayes* akan diterapkan guna melaksanakan klasifikasi dalam penelitian ini. Setelah itu, tahap evaluasi pada model dilakukan melalui penerapan *confusion matrix*. Hasil pengujian pada model yang telah didapat melalui penerapan algoritma *Naive Bayes* menunjukkan akurasi sebesar 82%, seperti dalam Tabel 2:

Tabel 2. Hasil Pengujian Algoritma *Naive Bayes*

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
Negatif	0.79	1.00	0.88	184
Positif	1.00	0.45	0.62	87
<i>accuracy</i>			0.82	271
<i>macro avg</i>	0.90	0.72	0.75	271
<i>weighted avg</i>	0.86	0.82	0.80	271

Sedangkan, hasil pengujian dengan algoritma SVM didapatkan nilai akurasi sebesar 85%. Tabel 3 menunjukkan hasil dari algoritma SVM yang telah diterapkan pengujiannya pada model:

Tabel 3. Hasil Pengujian Algoritma SVM

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
Negatif	0.85	0.94	0.89	184
Positif	0.84	0.66	0.74	87
<i>accuracy</i>			0.85	271
<i>macro avg</i>	0.85	0.80	0.81	271
<i>weighted avg</i>	0.85	0.85	0.84	271

4. KESIMPULAN

Menurut hasil, diperoleh dalam penelitian ini dengan melakukan klasifikasi teks ulasan terhadap aplikasi *Netflix*, maka kesimpulan yang didapat yaitu, penerapan metode *scraping* untuk memperoleh data pada *Google Play Store* dengan jumlah sebanyak 1000 data, yang kemudian diterapkan *pre-processing data* dengan menghilangkan atribut-atribut yang tidak diperlukan, lalu dilakukan pelabelan data, serta dilakukan *cleaning data* (pembersihan data) untuk menghapus data yang berisi nilai null (*missing value*). Sehingga setelah dilakukan tahap *preprocessing data*, dihasilkan 902 data yang terbagi atas dua *class* yaitu, *class* positif sejumlah 308 data, dan *class* negatif sejumlah 594 data. Selanjutnya, 30 % data dibagi untuk data uji yang berjumlah 271 data, dan 70% untuk data latih yang berjumlah 631 data. Dilanjutkan dengan tahap evaluasi atau pengujian model dengan menggunakan *confusion matrix*,

Pengujian model dengan membandingkan algoritma *Naive Bayes* dan SVM untuk mengklasifikasikan ulasan teks aplikasi *Netflix* pada *Google Play Store*, terbukti bahwa model klasifikasi SVM menunjukkan hasil terbaik yaitu *Support Vector Machine* (SVM) menghasilkan akurasi 0,85 (85%) dan *Naive Bayes* menghasilkan akurasi 0,82 (82%). Dengan demikian, dapat disimpulkan bahwa algoritma *Support Vector Machine* (SVM) lebih baik dan efektif daripada *Naive Bayes Algorithm* untuk melakukan klasifikasi ulasan teks aplikasi *Netflix* pada *Google Play Store*.

DAFTAR PUSTAKA

- [1] A. Anggono, A. S. Y. Irawan, and P. Purwanto, "Klasifikasi Minat Masyarakat Dalam Berlangganan Netflix di Masa Pandemi Covid-19 dengan Algoritma Naive Bayes (Studi Kasus: Kecamatan Karawang Timur)," *Jurnal Ilmiah Wahana Pendidika.*, vol. 9, no. 8, pp. 31–44, 2023.
- [2] S. A. Aputra, et al, "Sentiment Analysis Analysis of E-Wallet Sentiments on Google Play Using the Naive Bayes Algorithm Based on Particle Swarm Optimization," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 3, no. 3, pp. 377–382, 2019.
- [3] R. A. Subagja, Y. Widiastiwi, and N. Chamidah, "Klasifikasi Ulasan Aplikasi Jenius pada Google Play Store Menggunakan Algoritma Naive Bayes," *Informatik Jurnal Ilmu Komputer*, vol. 17, no. 3, p. 197, 2021.
- [4] I. S. Milal, et al., "Klasifikasi Teks Review Pada E-Commerce Tokopedia Menggunakan Algoritma Svm," *Naratif Jurnal Nasional Riset Aplikasi dan Teknik Informatika*, vol. 5, no. 1, pp. 34–45, 2023..
- [5] R. Wahyudi and G. Kusumawardana, "Analisis Sentimen pada Aplikasi Grab di Google Play Store Menggunakan Support Vector Machine," *Jurnal Informatika.*, vol. 8, no. 2, pp. 200–207, 2021.
- [6] L. Mutawalli, M. T. A. Zaen, and W. Bagye, "Klasifikasi Teks Sosial Media Twitter Menggunakan Support Vector Machine (Studi Kasus Penusukan Wiranto)," *Jurnal Informatika dan Rekayasa Elektronika*, vol. 2, no. 2, p. 43, 2019.
- [7] M. I. Fikri, T. S. Sabrila, and Y. Azhar, "Perbandingan Metode Naive Bayes dan Support Vector Machine pada Analisis Sentimen Twitter," *Smatika Jurnal*, vol. 10, no. 02, pp. 71–76, 2020.
- [8] F. Nurwanda, J. R. Rizkiani "Perbandingan Metode Naive Bayes Classifier dan Support Vector Machine pada Analisis Sentimen Twitter Topik Lifestyle," *Jurnal Ilmiah Wahana Pendidikan*, vol. 9, no. 21, pp. 314–323, 2023.
- [9] U. Riyanto, "Analisis Perbandingan Algoritma Naive Bayes Dan Support Vector Machine Dalam Mengklasifikasikan Jumlah Pembaca Artikel Online," *JIKA (Jurnal Informatika)*, vol. 2, no. 2, pp. 62–72, 2018.
- [10] A. J. Firdausi, W. Astuti, Adiwijaya "Perbandingan Algoritma Klasifikasi SVM dan Naive Bayes Dalam Analisis Sentimen Pembelajaran Daring di Masa Pandemi COVID-19 di

- Twitter,” *e-Proceedings of Engineering*, vol. 9, no. 3, pp. 2065–2083, 2022.
- [11] Y. Pratiwi, A. Yaqin, S. Informasi, F. I. Komputer, F. I. Komputer, and N. B. Classifier, “Klasifikasi Tweet Tidak Senonoh Twitter dengan Naive Bayes Classifier,” *e-Jurnal JUSITI (Jurnal Sistem Informasi dan Teknologi Informasi)*, vol. 11, no. 01, pp. 70–80, 2022.
- [12] A. C. Saputra and A. S. Saragih, “Klasifikasi Rating Aplikasi Android Di Google Play Store Menggunakan Algoritma Gradient Boost,” *Jurnal Teknika (Jurnal Teoritis dan Terapan Bidang Keteknikan)*, vol. 6, no. 1, pp. 18–29, 2022.