



## Analisis Sentimen Aplikasi Liputan6.Com pada Ulasan Pengguna di Google Playstore dengan Menggunakan Algoritma Support Vector Machine (Svm) dan Naïve Bayes

Yayang Tika Robiatush Sholiha<sup>1\*</sup>, Lubna Asjad Muhda Nabilah<sup>2</sup>, Imron<sup>3</sup>

<sup>1-3</sup>Fakultas Teknik dan Informatika, Sistem Informasi, Universitas Bina Sarana Informatika, Kota Tangerang, Indonesia

E-mail: [tikasolia27@gmail.com](mailto:tikasolia27@gmail.com)<sup>1</sup>, [lubnanabilah23@gmail.com](mailto:lubnanabilah23@gmail.com)<sup>2</sup>, [imron.imr@bsi.ac.id](mailto:imron.imr@bsi.ac.id)<sup>3</sup>

Jl. Gatot Subroto No.8, Cimone, Kec. Karawaci, Kota Tangerang, Banten 15114

Korespondensi penulis : [tikasolia27@gmail.com](mailto:tikasolia27@gmail.com) \*

**Abstract:** This study aims to evaluate user sentiment toward the Liputan6.com application available on the Google Play Store. In the digital era, user reviews serve as a significant indicator in assessing the quality of an application. However, the inconsistency between rating scores and review content renders manual analysis less objective. To address this issue, a machine learning approach was adopted by comparing two algorithms, namely Support Vector Machine (SVM) and Naïve Bayes (NB). A total of 2,500 reviews were collected through a web scraping process and automatically labeled based on the rating (positive if  $\geq 3$ , negative if  $< 3$ ). The data preprocessing stages included cleaning, case folding, tokenizing, stopword removal, and token filtering. Subsequently, word weighting was carried out using the TF-IDF method, followed by classification using 10-Fold Cross Validation in RapidMiner. The evaluation results indicate that, in the positive class, NB demonstrated superior precision (89.47%), whereas SVM achieved higher recall (98.94%) and F1-score (90.96%). In the negative class, SVM performed better in terms of precision (66.15%), while NB attained higher recall (65.65%) and F1-score (36.34%). Further evaluation based on AUC and accuracy positioned SVM in the good category (AUC 0.842; accuracy 83.82%), while NB was categorized as fail (AUC 0.505; accuracy 60.87%). Overall, SVM is considered to be more effective than NB.

**Keywords:** Sentiment Analysis, Google Play Store, Support Vector Machine, Naïve Bayes, TF-IDF, Application Review, Web Scraping

**Abstrak:** Penelitian ini bertujuan mengevaluasi sentimen pengguna terhadap aplikasi [Liputan6.com](#) di Google Play Store. Di era digital, ulasan pengguna menjadi indikator penting dalam menilai kualitas aplikasi. Namun, ketidaksesuaian antara rating dan komentar membuat analisis manual kurang objektif. Oleh karena itu, digunakan pendekatan machine learning dengan membandingkan algoritma Support Vector Machine (SVM) dan Naïve Bayes (NB). Sebanyak 2.500 ulasan dikumpulkan melalui web scraping dan diberi label otomatis berdasarkan rating (positif  $\geq 3$ , negatif  $< 3$ ). Proses pengolahan data mencakup cleaning, case folding, tokenizing, stopword removal, dan filter token. Setelah pembobotan menggunakan TF-IDF, data diklasifikasi dengan 10-Folds Cross Validation di RapidMiner. Hasil evaluasi menunjukkan bahwa pada kelas positif, NB unggul dalam precision (89,47%), tetapi SVM memiliki recall (98,94%) dan F1-score (90,96%) yang lebih tinggi. Pada kelas negatif, SVM lebih baik dalam precision (66,15%), sedangkan NB unggul dalam recall (65,65%) dan F1-score (36,34%). Evaluasi lanjutan berdasarkan nilai AUC dan akurasi menempatkan SVM dalam kategori baik (AUC 0,842 akurasi 83,82%), sedangkan NB tergolong gagal (AUC 0,505 akurasi 60,87%). Secara keseluruhan, SVM dinilai lebih unggul dibandingkan NB.

**Kata Kunci:** Analisis Sentimen, Google Play Store, Support Vector Machine, Naïve Bayes, TF-IDF, Ulasan Aplikasi, Web Scraping

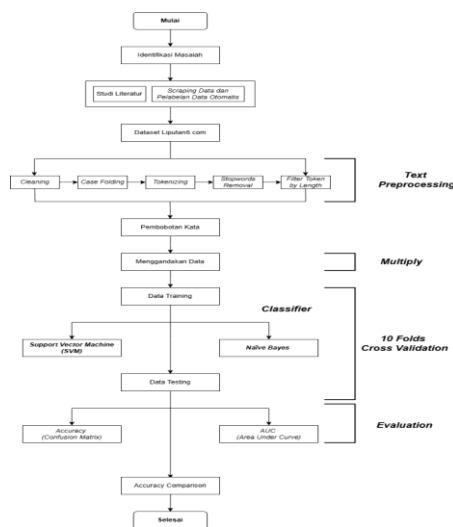
### 1. PENDAHULUAN

Pola konsumsi informasi masyarakat telah berubah karena kemajuan teknologi informasi di era digital. Dari media cetak ke media daring, aksesibilitas perangkat digital seperti smartphone dan komputer telah mendorong penggunaan media berita online [1]. Liputan6.com, salah satu portal berita online terkenal di Indonesia, telah memiliki aplikasi

mobile sejak 2015. Aplikasi tersebut telah mendapatkan lebih dari satu juta unduhan dan menerima rata-rata 4,2 dari sekitar 76.300 ulasan. Namun, hasil pengamatan terhadap ulasan di aplikasi tersebut menunjukkan ketidaksesuaian antara peringkat numerik dan isi komentar; beberapa ulasan bernada negatif disertai dengan peringkat tinggi, dan sebaliknya [2]. Karena ketidakkonsistenan ini, sulit untuk menentukan tingkat kepuasan pengguna hanya berdasarkan rating. Analisis sentimen diperlukan untuk mendapatkan pemahaman yang lebih mendalam tentang pendapat pengguna. Ini memungkinkan interpretasi data ulasan secara lebih objektif dan terstruktur; metode ini dapat menentukan aspek positif dan negatif dari ulasan, yang menghasilkan penilaian yang lebih akurat tentang kepuasan pengguna [3]. Support Vector Machine (SVM) adalah salah satu metode yang dapat digunakan dalam analisis sentimen karena berfokus pada pencarian hyperplane optimal yang memisahkan data ke dalam dua kategori utama, sentimen positif dan negatif. SVM adalah teknik pembelajaran supervisi yang efektif dalam mengklasifikasi opini[4]. Sebaliknya, algoritma Naïve Bayes—yang menggunakan prinsip probabilistik untuk menentukan kecenderungan suatu teks terhadap kategori tertentu—sangat populer untuk klasifikasi teks. Algoritma ini sangat efisien terutama setelah penerapan pembobotan TF-IDF [5]. Atas dasar tersebut, penelitian ini dilakukan dengan tujuan untuk membandingkan kinerja kedua algoritma dalam mengklasifikasikan persepsi pengguna terhadap aplikasi Liputan6.com. Temuan dari penelitian ini diharapkan dapat memberikan masukan bagi pengembang aplikasi dalam memahami pandangan pengguna serta mendukung upaya peningkatan kualitas layanan secara berkelanjutan.

## 2. METODE PENELITIAN

Penelitian ini dilakukan melalui beberapa tahapan yang diuraikan melalui gambar berikut:



Gambar 1. Tahapan Penelitian

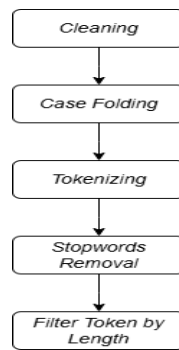
Keterangan:

### A. Pengumpulan Data

Tahap pengumpulan data mencakup studi literatur dan proses scraping disertai pelabelan otomatis. Kajian pustaka dilakukan untuk memahami konsep analisis sentimen serta penerapan algoritma SVM dan Naïve Bayes menggunakan Python. Ulasan pengguna aplikasi Liputan6.com kemudian diambil dari Google Play Store melalui Google Colab. Data yang diperoleh diberi label otomatis berdasarkan rating, dengan nilai  $\geq 3$  dikategorikan positif dan  $< 3$  sebagai negatif. Hasilnya berupa dataset CSV yang siap untuk dianalisis lebih lanjut.

### B. Preprocessing Data

Setelah tahap pelabelan, data diproses melalui tahap preprocessing menggunakan perangkat lunak RapidMiner untuk membersihkan dan menyusun data agar lebih terstruktur, dengan menghapus elemen yang tidak relevan [6], sebagaimana ditunjukkan pada Gambar



Gambar 2. Preprocessing Data

### C. Pembobotan Kata

Setelah tahap preprocessing selesai, dilakukan pembobotan kata menggunakan metode TF-IDF untuk menghitung bobot setiap kata berdasarkan frekuensinya dalam dokumen dan kelangkaannya di seluruh data, dengan tujuan menghasilkan representasi numerik yang mencerminkan relevansi kata, di mana proses ini dilakukan melalui operator *Process Documents from Data* pada ekstensi *Text Processing* di RapidMiner.

### D. Menggandakan Data

Setelah tahap pembobotan kata, data digandakan menggunakan operator *Multiply*. Data yang telah diduplikasi selanjutnya digunakan secara bersamaan dalam proses pengelompokkan dengan mengimplementasikan pendekatan algoritmik *Support Vector Machine (SVM)* dan metode probabilistik *Naïve Bayes*, proses klasifikasi dapat dilakukan secara sistematis untuk mendapatkan hasil analisis yang lebih terarah dan terukur di RapidMiner.

### E. 10-Fold Cross Validation

Setelah proses duplikasi data, dataset dibagi menjadi data pelatihan dan pengujian menggunakan metode 10-Fold Cross Validation, di mana 2.500 data dibagi secara acak dan seimbang ke dalam 10 fold, masing-masing berisi 250 data. Proses pelatihan dan pengujian dilakukan sebanyak 10 kali, dengan 1 fold (250 data) sebagai data uji dan 9 fold (2.250 data) sebagai data latih pada setiap iterasi. Pada tahap ini, klasifikasi sentimen dilakukan ke dalam dua kelas, yaitu positif dan negatif, dengan menggunakan algoritma Support Vector Machine (SVM) dan Naïve Bayes.

### F. Evaluation

Tahap akhir penelitian ini berupa evaluasi model untuk mengukur akurasi klasifikasi algoritma SVM dan Naïve Bayes. Penilaian dilakukan berdasarkan akurasi dan nilai Area Under Curve (AUC), dengan menggunakan confusion matrix sebagai alat ukur. Matriks ini menyajikan perbandingan antara prediksi model dan data aktual melalui metrik akurasi, presisi, dan recall, sehingga memungkinkan penilaian efektivitas model dalam mengklasifikasi sentimen secara tepat. [7]. Berikut adalah tabel *Confusion Matrix* :

Tabel I  
Confession Matrix

Data		Aktual	
		<i>True Positive (+)</i>	<i>False Negative (-)</i>
Prediksi	<i>True Positive (+)</i>	<i>True Positive (TP)</i>	<i>False Positive (FP)</i>
	<i>False Negative (-)</i>	<i>False Negative (FN)</i>	<i>True Negative (TN)</i>

Sumber : [8]

Dari *Confusion Matrix* dapat disimpulkan *matrix* evaluasi [7], sebagai berikut:

- a. *True Positive (TP)*: Kondisi ketika data aktual adalah "Good" dan model berhasil memprediksi dengan benar sebagai "Good".
- b. *True Negative (TN)*: Terjadi ketika data aktual adalah "Bad" dan diprediksi sebagai "Bad".
- c. *False Positive (FP)*: Situasi di mana data yang seharusnya "Bad" tetapi diprediksi sebagai "Good".
- d. *False Negative (FN)*: Situasi di mana data yang sebenarnya "Good" tetapi diprediksi sebagai "Bad".

Berdasarkan Tabel I, evaluasi menggunakan metode *Confusion Matrix* menghasilkan nilai-nilai untuk metrik seperti *accuracy*, *precision*, dan *recall* [7], berikut adalah rumus-rumus perhitungannya :

- a. *Accuracy*

Akurasi yaitu persentase prediksi yang benar.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

- b. *Precision*

Presisi merupakan tingkat ketepatan prediksi positif.

$$Precision = \frac{TP}{TP+FP}$$

- c. *Recall*

*Recall* merupakan kemampuan model mendeteksi data positif.

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

- d. *F1-Score*

Nilai *F1-Score* dihitung sebagai rata-rata harmonis dari presisi dan *recall*.

$$F1-Score = 2 \times \frac{precision \times recall}{recall + precision} \times 100\%$$

Setelah evaluasi menggunakan *confusion matrix*, analisis lanjutan dilakukan melalui Receiver Operating Characteristics (ROC) dan Area Under the Curve (AUC) untuk memberikan gambaran yang lebih menyeluruh mengenai performa model klasifikasi.

#### a. Area Under the Curve (AUC)

AUC merupakan indikator evaluasi yang mengukur kemampuan model dalam membedakan dua kelas berdasarkan kurva ROC. Nilainya berkisar antara 0 hingga 1; semakin mendekati 1, semakin baik performa klasifikasi. Nilai 1 menunjukkan klasifikasi sempurna, sedangkan 0,5

setara dengan tebakan acak. AUC juga digunakan sebagai dasar perbandingan kinerja antar model. [9]. Berikut adalah nilai auc dan interpretasinya :

Area under the curve (AUC)	Interpretation
$0.9 \leq \text{AUC}$	Excellent
$0.8 \leq \text{AUC} < 0.9$	Good
$0.7 \leq \text{AUC} < 0.8$	Fair
$0.6 \leq \text{AUC} < 0.7$	Poor
$0.5 \leq \text{AUC} < 0.6$	Fail

Sumber: [9]

Gambar 3. Nilai auc dan interpretasi

Sumber: Nahm, 2022

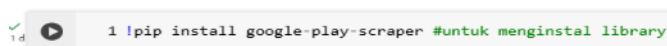
### b. Kurva *Receiver Operating Characteristic (ROC)*

Kurva ROC adalah grafik yang menggambarkan hubungan antara sensitivitas dan 1 – spesifisitas untuk mengevaluasi kemampuan model membedakan dua kelas. Kurva ini menunjukkan performa model pada berbagai ambang klasifikasi, meskipun tidak menampilkan jumlah sampel atau nilai ambang secara eksplisit dan dapat kurang halus jika data terbatas. [9].

## 3. HASIL DAN PEMBAHASAN

### A. PENGUMPULAN DATA

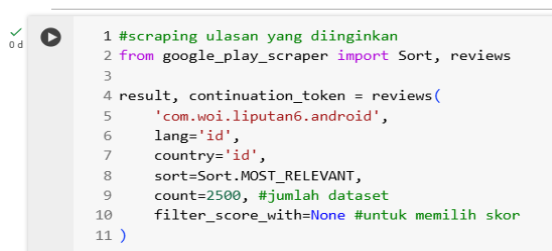
Dalam penelitian ini, data dikumpulkan secara otomatis melalui metode web scraping dengan memanfaatkan library *google-play-scraper*, yang dijalankan pada platform Google Colab, guna memperoleh ulasan pengguna aplikasi Liputan6.com dari Google Play Store.



```
1 !pip install google-play-scraper #untuk menginstal library
```

Gambar 4. Install Library Web Scraping

Data dikumpulkan secara otomatis menggunakan pustaka *google-play-scraper* dengan ID aplikasi *com.woi.liputan6.android* sebagai acuan. Sebanyak 2.500 ulasan berbahasa Indonesia yang paling relevan berhasil diperoleh dari Google Play Store.



```
1 #scraping ulasan yang diinginkan
2 from google_play_scraper import Sort, reviews
3
4 result, continuation_token = reviews(
5     'com.woi.liputan6.android',
6     lang='id',
7     country='id',
8     sort=Sort.MOST_RELEVANT,
9     count=2500, #jumlah dataset
10    filter_score_with=None #untuk memilih skor
11 )
```

Gambar 5. Proses Scraping Data

	userName	score	at	content
0	Pengguna Google	5	2018-11-25 18:28:27	Setelah mencoba di hampir semua aplikasi berit...
1	Pengguna Google	5	2020-04-10 01:07:38	Aplikasi LIPUTAN6 ini bagus bagus beritanya, t...
2	Estu Gentala	4	2023-08-28 02:00:24	isi berita di app tdk selengkap kalau baca di ...
3	FARID NAHUMARURY	5	2023-11-15 12:43:08	Aplikasi ini sangat bermanfaat gue bisa meliha...
4	andi shandra	5	2020-04-29 17:51:20	Beritanya terkini dan dapat dipercaya. Tampil...
...	...	...	...	...
2496	Pengguna Google	5	2020-03-18 05:00:12	Mantap meski baru dicoba
2496	Pengguna Google	5	2020-03-13 10:18:56	Beritanya bagus bagus
2497	ramli zhal	5	2020-05-27 00:30:46	Bokeh Berita nya,update trus
2498	Pengguna Google	5	2019-12-11 04:58:08	Beritanya update. mantap.
2499	RAMBLI TRISNA	5	2022-06-29 03:01:09	Mantap info beritanya.

2500 rows x 4 columns

Gambar 6. Hasil Scraping Data

Setelah data berhasil dikumpulkan, informasi tersebut akan disimpan dalam sebuah file berformat CSV sesuai dengan jumlah ulasan yang dibutuhkan.

```
1 my_df.to_csv("ulasanaplikasi_liputan6.csv", index = False) #untuk menyimpan dataset csv
```

Gambar 7.Penyimpanan Dataset Liputan6.com

Dataset dalam format CSV dibaca untuk mengakses ulasan dan skor penilaian, dilanjutkan dengan pembersihan data dan konversi skor ke tipe numerik. Sentimen kemudian dilabeli otomatis berdasarkan skor:  $\geq 3$  sebagai positif dan  $< 3$  sebagai negatif. Data hasil pelabelan disimpan ulang dalam format Excel untuk analisis lanjutan.

```
1 import pandas as pd
2
3 # Memuat data excel
4 file_path = "/content/sample_data/ulasanaplikasi_liputan6.csv"
5 df = pd.read_csv(file_path)
6
7 # Pastikan kolom yang diperlukan ada dalam dataset
8 if 'content' not in df.columns or 'score' not in df.columns:
9     raise ValueError("Kolom 'content' atau 'score' tidak ditemukan dalam dataset!")
10
11 # Menghapus data kosong pada content dan score
12 df = df.dropna(subset=['content', 'score'])
13
14 # Pastikan score bertipe numerik
15 df['score'] = pd.to_numeric(df['score'], errors='coerce')
16
17 # Menghapus baris dengan score NaN (tidak valid)
18 df = df.dropna(subset=['score'])
19
20 # Membuat label sentimen dari score (>=3 dianggap positif, <3 dianggap negatif)
21 df['sentiment_label'] = df['score'].apply(lambda x: 'positif' if x >= 3 else 'negatif')
22
23 # Ganti nama kolom 'content' menjadi 'text'
24 df = df.rename(columns={'content': 'text'})
25
26 # Simpan hasil ke excel hanya dengan text dan sentiment label
27 df[['text', 'sentiment_label']].to_excel("/content/hasil_sentimen.xlsx", index=False, engine='openpyxl')
28
29 # Menampilkan beberapa hasil
30 print(df[['text', 'sentiment_label']].head(10))
31
```

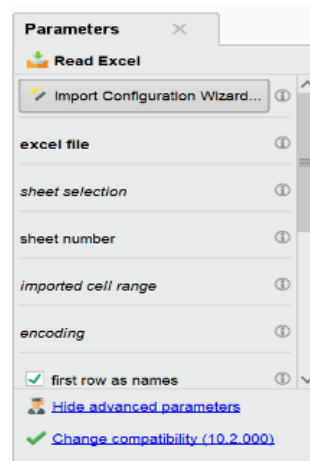
Gambar 8. Proses Scraping Pelabelan Data

	text	sentiment_label
0	Lebih awal diberitakan Mksih swal	positif
1	Cara gampang info nya pun mudah ...	positif
2	Sangat memuaskan aplikasinya dan mudah fitur2 ...	positif
3	Saya menyesal jadi maaf yaa. Maka dari itu say...	positif
4	Tampilan sangat" bagu setelah di update Yang j...	negatif
5	Kembalikan tampilannya seperti dulu, yg sekara...	negatif
6	Tampilan aplikasinya buruk, membaca jadi tidak...	negatif
7	Tampilan layar lebih bagus & nyaman yang dulu,...	negatif
8	Tampilan buruk & tidak nyaman membacanya	negatif
9	Sangat jelek formatnya,tdk bisa koment	negatif

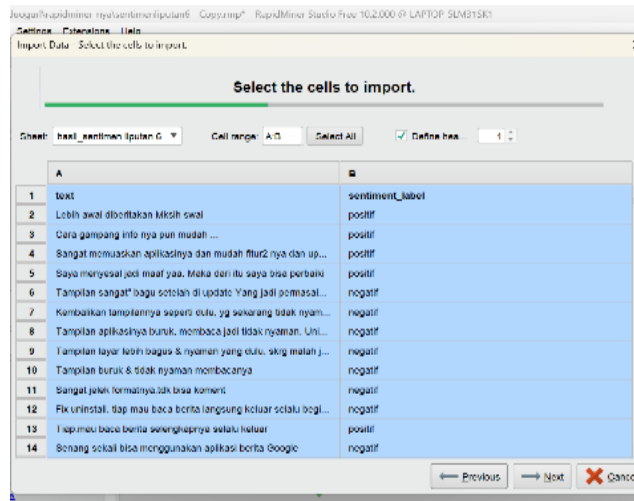
Gambar 9. Hasil Scraping Pelabelan Data

## B. Preprocessing Data

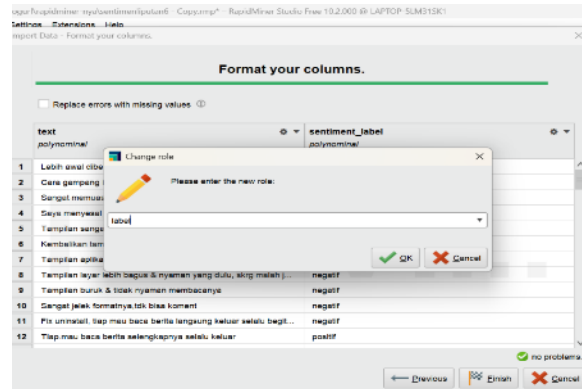
Sebelum tahap *preprocessing*, dilakukan proses awal berupa impor dataset menggunakan operator *Read Excel* melalui menu *Import Configuration Wizard*. Setelah data dimasukkan, kolom sentimen ditetapkan sebagai *label* dan diubah ke tipe *binominal*, karena hanya terdiri dari dua kelas: positif dan negatif.



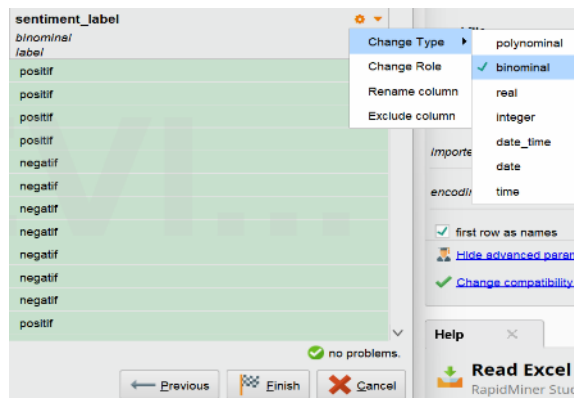
Gambar 10. Parameter Read Excel



Gambar 11. Hasil Import Data

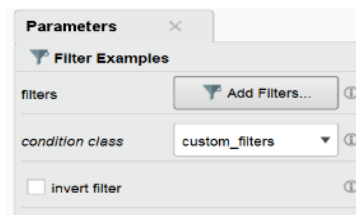


Gambar 12. Penentuan Role

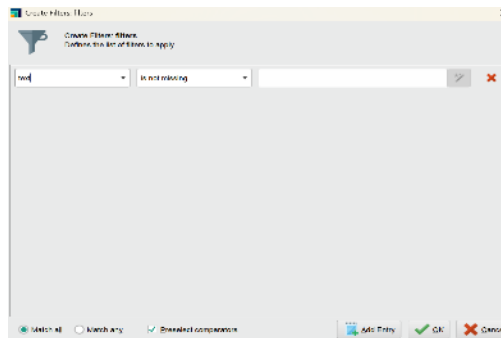


Gambar 13. Mengubah Type

Setelah impor dataset, operator *Filter Examples* digunakan untuk menghapus entri dengan nilai kosong pada atribut *Text*, sehingga hanya data yang valid digunakan dalam analisis.

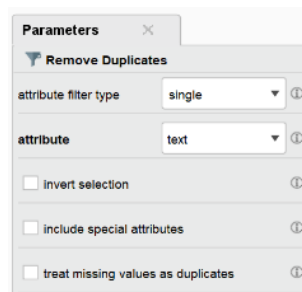


Gambar 14. Parameter Filter Examples



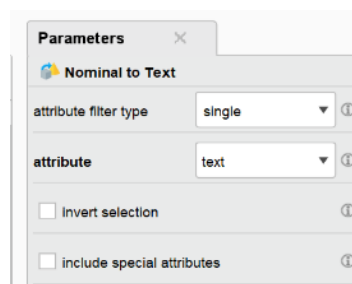
Gambar 15. Add Filter

Data yang telah disaring kemudian diproses dengan operator *Remove Duplicates* untuk menghapus duplikasi pada atribut *Text*, guna memastikan keunikan entri dan meningkatkan kualitas dataset.



Gambar 16. Parameter Remove Duplicates

Tahap selanjutnya adalah penerapan Operator *Nominal to Text* digunakan untuk mengonversi atribut bertipe nominal menjadi teks, sehingga data dapat diproses dalam analisis berbasis teks.

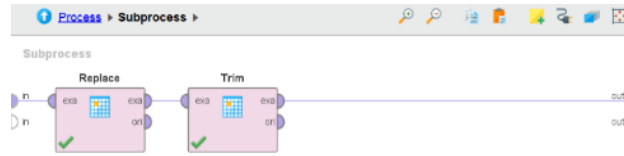


Gambar 17. Parameter Nominal To Text

Setelah melalui tahapan awal, langkah selanjutnya adalah *preprocessing* terhadap dataset, yang terdiri dari lima tahap utama: pembersihan data (*cleansing*), normalisasi huruf menjadi kecil (*case folding*), pemisahan kata dari kalimat (*tokenizing*), penghapusan kata-kata umum yang tidak informatif (*stopwords removal*), dan penyaringan token berdasarkan panjang karakter (*filter tokens by length*).

1) *Cleaning*

Data yang noise, tidak konsisten, atau tidak memiliki keterkaitan dengan tujuan analisis disaring dan dihapus melalui proses cleaning [10]. Proses *cleaning* dilakukan menggunakan operator *Replace* dan *Trim* untuk menghapus simbol tidak relevan pada kolom *Text*, dengan bantuan ekspresi reguler sebagai filter karakter khusus .



Gambar 18. Proses Cleaning

	text	sentiment_label
1	text	sentiment_label
2	Lebih awal diberitakan Mksih swal	positif
3	Cara gampang info nya pun mudah ...	positif
4	Sangat memuaskan aplikasinya dan mudah fitur2 nya dan update warta berita terkini	positif
5	Saya menyesal jadi maaf yaa. Maka dari itu saya bisa perbaiki	positif
6	Tampilan sangat" bagu setelah di update Yang jadi permasalahan setiap kali masuk Pasti keluar sendiri jika ada nilai kepuasan di bawah satu bintang * Saya pribadi akan memberikan penilaian setengah *	negatif

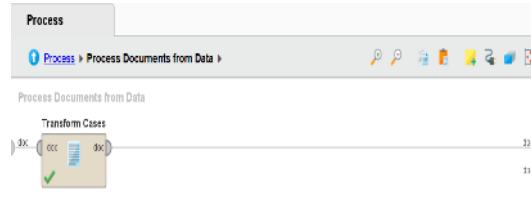
Gambar 19. Teks Sebelum Dilakukan Proses Cleaning

Row No.	sentiment_L...	text
1	positif	Lebih awal diberitakan Mksih swal
2	positif	Cara gampang info nya pun mudah
3	positif	Sangat memuaskan aplikasinya dan mudah fitur2 nya dan update warta berita terkini
4	positif	Saya menyesal jadi maaf yaa Maka dari itu saya bisa perbaiki
5	negatif	Tampilan sangat bagu setelah di update Yang jadi permasalahan setiap kali masuk Pasti keluar sendiri jika ada nilai kepuasa...

Gambar 20. Hasil Teks Sesudah Proses Cleaning

## 2) Case Folding

Case folding dilakukan untuk mengubah seluruh huruf menjadi huruf kecil [11], menggunakan operator *Transform Case* dalam *Process Documents from Data*.



Gambar 21. Proses Case Folding

1	text	
2	Lebih awal diberitakan Mksih swal	positif
3	Cara gampang info nya pun mudah ...	positif
4	Sangat memuaskan aplikasinya dan mudah fitur2 nya dan update warta berita terkini	positif
5	Saya menyesal jadi maaf yaa. Maka dari itu saya bisa perbaiki	positif
6	Tampilan sangat" bagu setelah di update Yang jadi permasalahan setiap kali masuk Pasti keluar sendiri jika ada nilai kepuasan di bawah satu bintang * Saya pribadi akan memberikan penilaian setengah *	negatif

Gambar 22. Teks Sebelum Dilakukan Proses Case Folding

Row No.	sentiment_L	text
1	positif	lebih awal diberitakan mksih swal
2	positif	cara gampang info nya pun mudah
3	positif	sangat memuaskan aplikasinya dan mudah fitur2 nya dan update warta berita terkini
4	positif	saya menyesal jadi maaf yaa maka dari itu saya bisa perbaiki
5	negatif	tampilan sangat bagu setelah di update yang jadi permasalahan setiap kali masuk pasti keluar sendiri jika ada nilai kepuasan di ba

Gambar 23. Hasil Teks Sesudah Dilakukan Proses Case Folding

## 3) Tokenizing

Proses tokenizing bertujuan memisahkan kalimat menjadi satuan kata serta menghapus spasi, tanda baca, dan karakter non-alfabet [12], yang dilakukan menggunakan operator *Transform Case* dengan mode non-letters.



Gambar 24. Proses Tokenizing

1	text	sentiment_label
2	Lebih awal diberitakan Mksih swal	positif
3	Cara gampang info nya pun mudah ...	positif
4	Sangat memuaskan aplikasinya dan mudah fitur2 nya dan update warta berita terkini	positif
5	Saya menyesal jadi maaf yaa. Maka dari itu saya bisa perbaiki	positif
6	Tampilan sangat" bagu setelah di update Yang jadi permasalahan setiap kali masuk Pasti keluar sendiri jika ada nilai kepuasan di bawah satu bintang * Saya pribadi akan memberikan penilaian setengah *	negatif

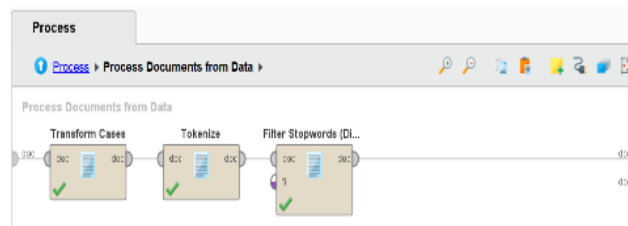
Gambar 25. Teks Sebelum Dilakukan Proses Tokenizing

Row No.	sentiment_label	text
1	positif	lebih awal diberitakan mksih awal
2	positif	cara gampang info nya pun mudah
3	positif	sangat memuaskan aplikasinya dan mudah fitur nya dan update warta berita terkini
4	positif	saya menyesal jadi maaf yaa maka dari itu saya bisa perbaiki
5	negatif	tampilan sangat bagu setelah di .update yang jadi permasalahan setiap kali masuk pasti keluar sendiri jika ada nilai kepuasan d...

Gambar 26. Hasil Teks Sesudah Dilakukan Proses Tokenizing

#### 4) Stopword Removal

Stopwords removal dilakukan untuk menghapus kata-kata umum yang kurang informatif [13], menggunakan operator *Filter Stopwords (Dictionary)* di RapidMiner dengan kamus Bahasa Indonesia yang disusun oleh peneliti.



Gambar 27. Proses Stopwords Removal

	text	sentiment_label
1	text	
2	Lebih awal diberitakan Mksih awal	positif
3	Cara gampang info nya pun mudah ...	positif
4	Sangat memuaskan aplikasinya dan mudah fitur2 nya dan update warta berita terkini	positif
5	Saya menyesal jadi maaf yaa. Maka dari itu saya bisa perbaiki	positif
6	Tampilan sangat" bagu setelah di update Yang jadi permasalahan setiap kali masuk Pasti keluar sendiri jika ada nilai kepuasan di bawah satu bintang * Saya pribadi akan memberikan penilaian setengah *	negatif

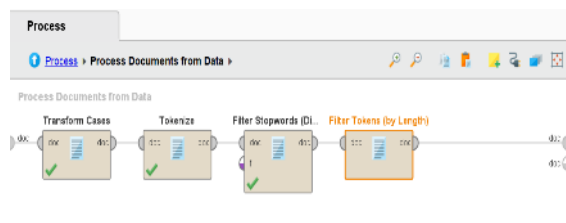
Gambar 28. Teks Sebelum Dilakukan Proses Stopwords Removal

Row No.	sentiment_label	text
1	positif	diberitakan mksih awal
2	positif	gampang info nya mudah
3	positif	memuaskan aplikasinya mudah fitur nya .update warta berita terkini
4	positif	menyesal maaf yaa perbaiki
5	negatif	tampilan bagu .update permasalahan kali masuk nilai kepuasan bintang pribadi penilaian

Gambar 29. Hasil Teks Sesudah Dilakukan Proses Stopwords Removal

### 5) Filter Tokens (by length)

Tahap penyaringan token berdasarkan panjang karakter dilakukan dengan operator *Filter Tokens (by Length)* untuk mengeliminasi kata-kata yang terlalu pendek atau terlalu panjang [14], dengan parameter panjang minimal 4 karakter dan maksimal 25 karakter.



Gambar 30. Proses Filter Tokens (By Length)

text	sentiment_label
1 Lebih awal diberitakan Mksih swal	positif
2 Cara gampang info nya pun mudah ...	positif
3 Sangat memuaskan aplikasinya dan mudah fitur2 nya dan update warta berita terkini	positif
4 Saya menyesal jadi maaf yaa. Maka dari itu saya bisa perbaiki	positif
5 Tampilan sangat" bagu setelah di update Yang jadi permasalahan setiap kali masuk Pasti keluar sendiri jika ada nilai kepuasan di bawah satu bintang * Saya pribadi akan memberikan penilaian setengah *	negatif

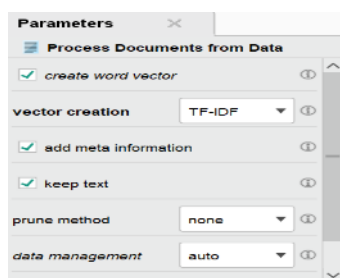
Gambar 31. Teks Sebelum Dilakukan Proses Filter Tokens (By Length)

Row No.	sentiment_label	text
1	positif	di baritakan msh swal
2	positif	gampang info mudah
3	positif	memuaskan aplikasinya mudah fitur update warta berita terkini
4	positif	menyesal maaf perbaiki
5	negatif	tampilan bagu update permasalahan kali masuk nilai kepuasan di bawah penilaian

Gambar 32. Hasil Teks Sesudah Dilakukan Proses Filter Tokens (By Length)

## C. Pembobotan Kata

Setelah tahap *preprocessing* selesai, dilakukan pembobotan kata menggunakan metode TF-IDF melalui operator *Process Documents from Data* dengan opsi *Create Word Vector*, guna memberikan nilai bobot pada setiap kata berdasarkan frekuensi dan relevansinya dalam dokumen [15].



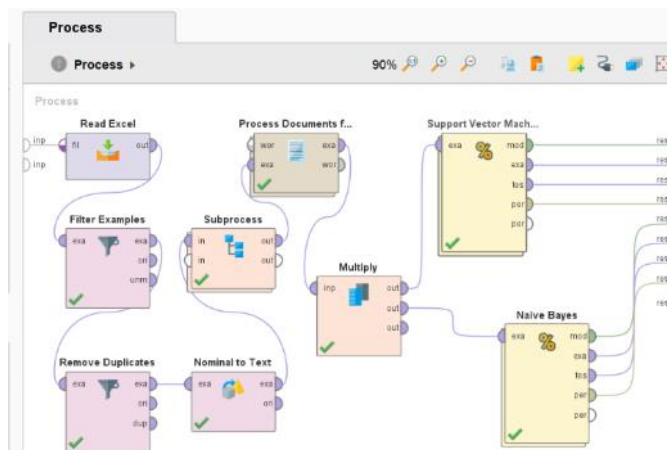
Gambar 33. Parameter TF-IDF

Row No.	sentiment...	text	zakurat	zamin	zamin	zaminx	zaminxy	zamin
1	positif	diw-lakan m...	0	0	0	0	0	0
2	positif	pancang info...	0	0	0	0	0	0
3	positif	memuaskan ...	0	0	0	0	0	0
4	positif	menyesa mt...	0	0	0	0	0	0
5	negatif	tempelan dag...	0	0	0	0	0	0
6	negatif	kembalikan t...	0	0	0	0	0	0
7	negatif	tempelan apl...	0	0	0	0	0	0
8	negatif	tempelan aya...	0	0	0	0	0	0
9	negatif	tempelan cur...	0	0	0	0	0	0
10	negatif	jelek format...	0	0	0	0	0	0
11	negatif	uninstal bac...	0	0	0	0	0	0
12	positif	lempas baca...	0	0	0	0	0	0
13	negatif	panjang aplk...	0	0	0	0	0	0
14	negatif	rukagen ry...	0	0	0	0	0	0
15	positif	stelan gamba...	0	0	0	0	0	0

Gambar 34. Hasil Pembobotan TF-IDF

#### D. Desain Model SVM dan Naïve Bayes

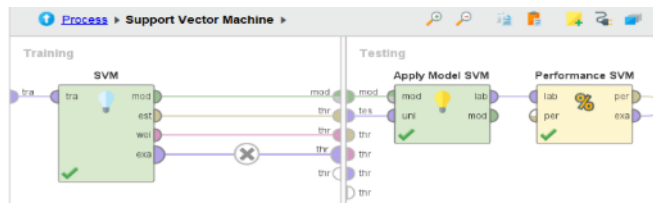
Setelah pembobotan TF-IDF dilakukan, data diduplikasi menggunakan operator *Multiply* agar dapat digunakan secara paralel pada algoritma Support Vector Machine (SVM) dan Naïve Bayes, sehingga memungkinkan dilakukannya perbandingan kinerja kedua metode terhadap data yang sama.



Gambar 35. Desain Model Perbandingan Algoritma SVM Dan Naïve Bayes

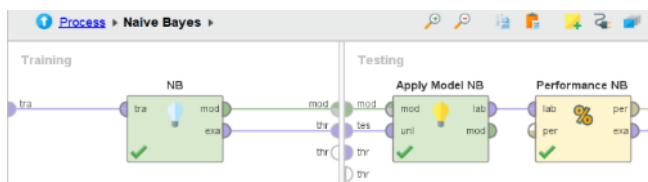
#### E. Desain Proses SVM dan Naïve Bayes

Dalam implementasi algoritma Support Vector Machine (SVM), peneliti menggunakan tiga operator utama, yakni *Support Vector Machine*, *Apply Model*, dan *Performance (Classification)*, yang masing-masing berfungsi untuk membangun model, menerapkannya pada data, serta mengevaluasi performa klasifikasinya.



Gambar 36. Desain Proses 10-Fold Cross Validation Untuk SVM

Penerapan algoritma Naïve Bayes dilakukan dengan menggunakan rangkaian operator *Naïve Bayes*, *Apply Model*, dan *Performance (Classification)*. Konsistensi struktur dan urutan ini memungkinkan evaluasi kinerja dilakukan secara setara terhadap masing-masing algoritma.



Gambar 37. Desain Proses 10-Fold Cross Validation Untuk Naïve Bayes

Setelah seluruh proses klasifikasi selesai, algoritma Support Vector Machine (SVM) dan Naïve Bayes menghasilkan prediksi kelas terhadap data yang telah dipraolah. Evaluasi kinerja dilakukan melalui operator *Apply Model* dan *Performance (Classification)*, yang menghasilkan metrik evaluasi seperti akurasi, presisi, dan recall. Hasil ini mempermudah perbandingan performa antara kedua algoritma.

Table View Plot View

accuracy: 83.82% +/- 1.40% (micro average: 83.82%)

	true positif	true negatif	class precision
pred. positif	2050	382	84.29%
pred. negatif	22	43	66.15%
class recall	98.94%	10.12%	

Gambar 38. Hasil Implementasi SVM

Table View Plot View

accuracy: 60.87% +/- 2.57% (micro average: 60.87%)

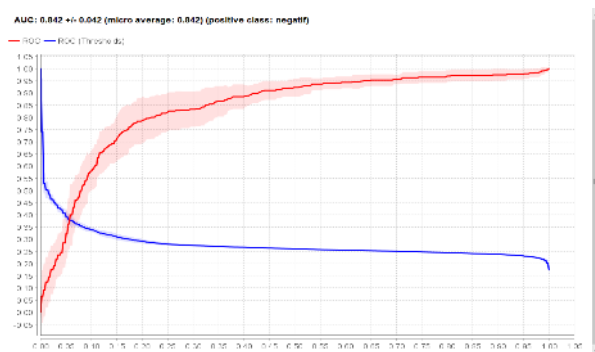
	true positif	true negatif	class precision
pred. positif	1241	146	89.47%
pred. negatif	831	279	25.14%
class recall	59.69%	65.65%	

Gambar 39. Hasil Implementasi Naïve Bayes

Berdasarkan hasil pemodelan sebelumnya, bagian ini menyajikan dan menguraikan Kurva ROC serta Confusion Matrix dari masing-masing algoritma yang diterapkan.

### 1) Kurva ROC Support Vector Machine (SVM)

Gambar di bawah ini menyajikan Kurva ROC yang dihasilkan oleh algoritma *Support Vector Machine (SVM)* :

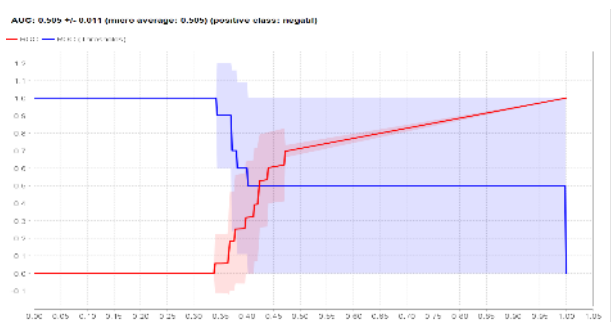


Gambar 40. Kurva ROC SVM

Kurva ROC yang dihasilkan oleh algoritma *Support Vector Machine (SVM)* menunjukkan nilai *AUC (Area Under the Curve)* sebesar 0.842. Berdasarkan nilai tersebut, performa model dikategorikan *Good* dalam klasifikasi

### 2) Kurva ROC Naive Bayes

Kurva ROC ditunjukkan pada gambar berikut dari algoritma *Naïve Bayes*



Gambar 41. Kurva ROC Naïve Bayes

Kurva ROC yang dihasilkan oleh algoritma *Naive Bayes* menghasilkan nilai *AUC (Area Under the Curve)* sebesar 0.505, Berdasarkan nilai tersebut, performa model dikategorikan *Fail* dalam klasifikasi.

## F. Evaluasi Model

Setelah proses klasifikasi menggunakan algoritma Support Vector Machine (SVM) dan Naïve Bayes selesai, tahap selanjutnya adalah evaluasi model melalui perhitungan Confusion Matrix untuk memperoleh metrik evaluatif seperti akurasi, presisi, dan recall. Evaluasi dilakukan pada dataset uji yang terdiri dari 2.500 data, mencakup 2.075 entri berlabel positif

dan 425 entri berlabel negatif. Berdasarkan hasil klasifikasi, algoritma SVM mencapai akurasi sebesar 83,82%, sedangkan Naïve Bayes memperoleh akurasi sebesar 60,87%. Nilai akurasi, presisi, dan recall selengkapnya disajikan berdasarkan Confusion Matrix pada bagian berikut.

1) *Confusion Matrix Support Vector Machine (SVM)*

Tabel II  
Confusion Matrix Support Vector Machine (SVM)

	<i>True Positif</i>	<i>True Negatif</i>	<i>Precision</i>
<b>Pred. Positif</b>	2050	382	84,29%
<b>Pred. Negatif</b>	22	43	66,15%
<i>Class Recall</i>	98,94%	10,12%	

Berikut merupakan rumus *confusion matrix* :

$$a. \text{ Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)} = \frac{(2050+43)}{(2050+43+382+22)} = \frac{2093}{2497} = 0,8382$$

$$b. \text{ Precision Positif} = \frac{TP}{TP+FP} = \frac{2050}{2050+382} = \frac{2050}{2432} = 0,8429$$

$$c. \text{ Precision Negatif} = \frac{TN}{TN+FN} = \frac{43}{43+22} = \frac{43}{65} = 0,6615$$

$$d. \text{ Recall Positif} = \frac{TP}{TP+FN} = \frac{2050}{2050+22} = \frac{2050}{2072} = 0,9894$$

$$e. \text{ Recall Negatif} = \frac{TN}{TN+FP} = \frac{43}{43+382} = \frac{43}{425} = 0,1012$$

$$f. \text{ F1-Score Positif} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{recall} + \text{precision}} \times 100\% = 2 \times \frac{0,8429 \times 0,9894}{0,9894 + 0,8429} \times 100\% = 2 \times \frac{0,8337}{1,8323} \times 100\% = 0,9096$$

$$g. \text{ F1-Score Negatif} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{recall} + \text{precision}} \times 100\% = 2 \times \frac{0,6615 \times 0,1012}{0,1012 + 0,6615} \times 100\% = 2 \times \frac{0,0669}{0,7627} \times 100\% = 0,1754$$

2) *Confusion Matrix Naïve Bayes*Tabel III  
Confusion Matrix Naïve Bayes

	<b>True Positif</b>	<b>True Negatif</b>	<b>Precision</b>
<b>Pred. Positif</b>	1241	146	89,47%
<b>Pred. Negatif</b>	831	279	25,14%
<b>Class Recall</b>	59,89%	65,65%	

Berikut merupakan rumus *confusion matrix*:

$$a. \text{ Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)} = \frac{(1241+279)}{(1241+279+146+831)} = \frac{1520}{2497} = 0,6087$$

$$b. \text{ Precision Positif} = \frac{TP}{TP+FP} = \frac{1241}{1241+146} = \frac{1241}{1387} = 0,8947$$

$$c. \text{ Precision Negatif} = \frac{TN}{TN+FN} = \frac{279}{279+831} = \frac{279}{1110} = 0,2514$$

$$d. \text{ Recall Positif} = \frac{TP}{TP+FN} = \frac{1241}{1241+831} = \frac{1241}{2072} = 0,5989$$

$$e. \text{ Recall Negatif} = \frac{TN}{TN+FP} = \frac{279}{279+146} = \frac{279}{425} = 0,6565$$

$$f. \text{ F1-Score Positif} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{recall} + \text{precision}} \times 100\% = 2 \times \frac{0,8947 \times 0,5989}{0,5989 + 0,8947} \times 100\% = 2 \times \frac{0,5362}{1,4936} \times 100\% = 0,7177$$

$$g. \text{ F1-Score Negatif} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{recall} + \text{precision}} \times 100\% = 2 \times \frac{0,2514 \times 0,6565}{0,6565 + 0,2514} \times 100\% = 2 \times \frac{0,1650}{0,9079} \times 100\% = 0,3634$$

**G. Fase Evaluasi Hasil**

Algoritma Support Vector Machine (SVM) menunjukkan akurasi sebesar 83,82%, lebih tinggi dibandingkan Naïve Bayes yang mencapai 60,87%. Hasil ini mengindikasikan bahwa SVM memiliki kestabilan dan ketepatan klasifikasi yang lebih baik secara keseluruhan.

Berdasarkan kurva ROC, nilai AUC pada SVM mencapai 0,842 yang termasuk dalam kategori *Good*, sedangkan Naïve Bayes hanya memperoleh AUC sebesar 0,505, yang tergolong *Fail*. Hal ini menandakan bahwa SVM lebih efektif dalam membedakan antara kelas positif dan negatif.

#### 1) *Precision, Recall, dan F1-Score*

Untuk kelas positif, Naïve Bayes unggul dalam presisi (89,47% vs 84,29%), namun kalah pada aspek recall (59,89% vs 98,94%), sehingga F1-score SVM lebih tinggi (90,96% vs 71,77%).

Pada kelas negatif, SVM unggul dalam presisi (66,15% vs 25,14%), sementara Naïve Bayes memiliki recall yang lebih tinggi (65,65% vs 10,12%), menghasilkan F1-score negatif yang lebih baik pada Naïve Bayes (36,34% vs 17,54%).

Secara umum, SVM memiliki kinerja yang lebih seimbang terutama dalam klasifikasi data berlabel positif. Sementara itu, Naïve Bayes cenderung lebih sensitif terhadap data negatif, meskipun dengan akurasi yang lebih rendah. Dengan mempertimbangkan seluruh metrik evaluasi, model Support Vector Machine (SVM) terbukti lebih unggul dibandingkan Naïve Bayes dalam analisis sentimen terhadap ulasan pengguna aplikasi Liputan6.com.

#### **4. KESIMPULAN**

Hasil penelitian ini mengindikasikan bahwa algoritma Support Vector Machine (SVM) memiliki keunggulan kinerja yang lebih menonjol dibandingkan dengan algoritma Naïve Bayes dalam proses klasifikasi sentimen terhadap ulasan pengguna aplikasi Liputan6.com. Berdasarkan hasil evaluasi melalui confusion matrix, algoritma SVM mencapai tingkat akurasi sebesar 83,82%, yang secara signifikan lebih tinggi dibandingkan akurasi yang diperoleh oleh Naïve Bayes, yaitu sebesar 60,87%. Selain itu, nilai Area Under the Curve (AUC) yang dicapai oleh SVM sebesar 0,842 menunjukkan bahwa model ini memiliki kemampuan yang baik dalam membedakan antar kelas, sedangkan AUC Naïve Bayes hanya mencapai 0,505, yang mencerminkan performa klasifikasi yang rendah.

Dari sisi kemampuan mengklasifikasikan data dengan sentimen positif, algoritma SVM menunjukkan keunggulan pada nilai recall dan F1-score, yang mencerminkan kemampuannya dalam mengenali sebagian besar data positif secara tepat. Sebaliknya, Naïve Bayes cenderung lebih peka terhadap data negatif, walaupun akurasinya masih rendah akibat nilai presisi yang tidak optimal. Secara keseluruhan, algoritma SVM dinilai lebih seimbang dan konsisten dalam menghasilkan klasifikasi yang akurat, sehingga lebih sesuai untuk digunakan dalam pengolahan data sentimen berbasis teks dari ulasan aplikasi.

## DAFTAR REFERENSI

- Anwar, M. S., Subroto, I. M. I., & Mulyono, S. (2019). Sistem pencarian e-journal menggunakan metode stopword removal dan stemming berbasis Android. *Konferensi Ilmiah Mahasiswa Unissula*, 2, 58–70.
- Audiansyah, D. D., Ratnawati, D. E., & Hanggara, B. T. (2022). Analisis sentimen aplikasi MyXL menggunakan metode Support Vector Machine berdasarkan ulasan pengguna di Google Play Store. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 6(8), 3987–3994. <http://j-ptiik.ub.ac.id>
- Darmawan, G., Alam, S., & Sulistyono, M. I. (2023). Analisis sentimen berdasarkan ulasan pengguna aplikasi MyPertamina pada Google Playstore menggunakan metode Naïve Bayes. *STORAGE – Jurnal Ilmiah Teknologi dan Ilmu Komputer*, 2(3), 100–108.
- Gifari, O. I., Adha, M., Freddy, F., & Durrand, F. F. S. (2022). Film review sentiment analysis using TF-IDF and Support Vector Machine. *Journal of Information Technology*, 2(1), 36–40.
- Kusnia, U., Kurniawan, F., & Artikel, S. (2022). Analisis sentimen review aplikasi media berita online pada Google Play menggunakan metode algoritma Support Vector Machines (SVM) dan Naive Bayes. *Jurnal Keilmuan dan Aplikasi Teknik Informatika*, 14(1), 24–25. <https://doi.org/10.35891/explorit>
- Ma'rifah, H., Wibawa, A. P., & Akbar, M. I. (2020). Klasifikasi artikel ilmiah dengan berbagai skenario preprocessing. *Sains, Aplikasi Komputasi dan Teknologi Informasi*, 2(2), 70. <https://doi.org/10.30872/jsakti.v2i2.2681>
- Nahm, F. S. (2022). Receiver operating characteristic curve: Overview and practical use for clinicians. *Korean Journal of Anesthesiology*, 75(1), 25–36. <https://doi.org/10.4097/kja.21209>
- Nurian, A. (2023). Analisis sentimen ulasan pengguna aplikasi Google Play menggunakan Naïve Bayes. *Jurnal Informatika dan Teknik Elektro Terapan*, 11(3s1), 829–835. <https://doi.org/10.23960/jitet.v11i3s1.3348>
- Putra, F., Tahiyat, H. F., Ihsan, R. M., Rahmaddeni, R., & Efrizoni, L. (2024). Penerapan algoritma K-Nearest Neighbor menggunakan wrapper sebagai preprocessing untuk penentuan keterangan berat badan manusia. *MALCOM – Indonesian Journal of Machine Learning and Computer Science*, 4(1), 273–281. <https://doi.org/10.57152/malcom.v4i1.1085>
- Ridwansyah, T. (2022). Implementasi text mining terhadap analisis sentimen masyarakat dunia di Twitter terhadap kota Medan menggunakan K-Fold Cross Validation dan Naïve Bayes Classifier. *KLIK – Kajian Ilmiah Informatika dan Komputer*, 2(5), 178–185. <https://doi.org/10.30865/klik.v2i5.362>
- Romadoni, F., Umaidah, Y., & Sari, B. N. (2020). Text mining untuk analisis sentimen pelanggan terhadap layanan uang elektronik menggunakan algoritma Support Vector Machine. *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, 9(2), 247–253. <https://doi.org/10.32736/sisfokom.v9i2.903>

- Siti Aisah, I., Irawan, B., & Suprapti, T. (2024). Algoritma Support Vector Machine (SVM) untuk analisis sentimen ulasan aplikasi Al Qur'an digital. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(6), 3759–3765. <https://doi.org/10.36040/jati.v7i6.8263>
- Suryawati, I., & Alam, S. (2022). Transformasi media cetak ke platform digital (Analisis mediamorfosis Harian SOLOPOS). *Jurnal Signal*, 10(2), 190. <https://doi.org/10.33603/signal.v10i2.7240>
- Valerian, F. R., et al. (2025). Klasifikasi tingkat obesitas menggunakan metode GBM dan confusion matrix. *Jurnal*, 9(2), 2242–2249.
- Wardhani, D., Astuti, R., & Saputra, D. D. (2024). Optimasi feature selection text mining: Stemming dan stopword. *Innovation: Journal of Social Science Research*, 4, 7537–7548.