Jurnal Vocational Teknik Elektronika dan Informatika
http://ejournal.unp.ac.id/index.php/voteknika
Universitas Negeri Padang, Indonesia

ISSN 2302-3295 (Print)
ISSN 2716-3989 (Electronics)
Vol. 13, No. 4, 2025

Sentiment Analysis of Honor of Kings Game Reviews on Google PlayStore Using Naive Bayes and *SVM*

Taufik Rahman*[1], Haikal Fulvian Saputra[2], Herman Kuswanto[3]

[1,2]Universitas Bina Sarana Infromatika, Jakarta, Indonesia

[3]Universitas Nusa Mandiri, Jakarta, Indonesia

Corresponding Author: taufik@bsi.ac.id

Abstract - ***This study aims to conduct a sentiment analysis of user reviews of the Honor of Kings game on Google PlayStore using the Naive Bayes (NB) algorithm and Support Vector Machine (SVM) as a machine learning approach. The research gap raised in this study lies in the lack of comparative studies that quantitatively measure the performance of the two classic algorithms on Indonesian-language mobile game review data, as well as the absence of numerical mapping of sentiment distribution that describes user perceptions proportionally. The dataset used consisted of 1000 reviews, which after the manual labeling process was divided into 780 positive reviews (52%), 540 negative reviews (36%), and 180 neutral reviews (12%). The quantitative objective of this study was to measure and compare the levels of accuracy, precision, recall, F1-score, and AUC of the two models to determine the most effective algorithm in classifying user opinions. The test results showed that the SVM model produced an accuracy of 75.3% with an AUC value of 0.82, while the NB model obtained an accuracy of 71.1% with an AUC of 0.78. Based on the confusion matrix, SVM is able to reduce misclassification of negative and neutral sentiments, which are generally difficult to distinguish due to the distribution of sparse text features. Scientifically, this study contributes by showing that SVM is more optimal than NB in handling unbalanced review data, and confirms the importance of feature weighting and AUC validation as indicators of model reliability. Practically, the results of this study can be used by the developers of Honor of Kings to evaluate aspects of the user experience based on the sentiment patterns identified, especially in improving server stability, character balance, and player satisfaction.***

Keywords— ***Sentiment Analysis, Naive Bayes, Support Vector Machine, Honor of Kings, Google PlayStore.***

I. PENDAHULUAN

Perkembangan industri game mobile di era digital mengalami peningkatan yang sangat pesat, seiring dengan meningkatnya jumlah pengguna smartphone di seluruh dunia. Salah satu game yang menarik perhatian masyarakat global adalah *Honor of Kings*, yang memiliki jutaan pengguna aktif dan ulasan pengguna yang terus bertambah di *platform* seperti *Google PlayStore*. Ulasan pengguna tersebut mengandung berbagai opini, mulai dari pujian hingga kritik terhadap kualitas permainan, kestabilan sistem, hingga pengalaman pengguna secara keseluruhan. Informasi ini dapat dimanfaatkan untuk mengevaluasi kualitas *game* dan meningkatkan kepuasan pengguna melalui analisis sentimen.

Analisis sentimen merupakan proses mengidentifikasi dan mengklasifikasikan opini atau emosi dalam teks menjadi kategori positif, negatif, atau netral. Dalam penelitian ini, analisis dilakukan terhadap ulasan pengguna di *Google PlayStore* untuk memahami persepsi publik terhadap game Honor of Kings. Dua algoritma populer yang digunakan dalam bidang ini adalah *Naive Bayes* dan *Support Vector Machine* (*SVM*). Keduanya dikenal memiliki performa yang baik dalam mengklasifikasikan teks, namun efektivitasnya dapat bervariasi tergantung pada karakteristik data yang dianalisis.

Dalam beberapa tahun terakhir, permainan daring telah menjadi fenomena yang sangat populer, dengan *Honor of Kings* sebagai salah satu contohnya. Dikembangkan oleh *Tencent Games*, *game* ini telah diunduh jutaan kali di seluruh dunia. Menurut Sensor Tower, game Honor of Kings telah diunduh oleh 10 juta hingga 50 juta pengguna di *platform Google PlayStore* [1], menunjukkan betapa besar pengaruh game ini di kalangan pengguna Android, khususnya di Indonesia. Namun, dengan volume ulasan yang begitu besar, penting untuk mengidentifikasi sentimen yang terkandung dalam ulasan-ulasan tersebut.

Beberapa penelitian sebelumnya telah membahas penerapan algoritma *Naive Bayes* dan *SVM* untuk analisis sentimen terhadap produk atau aplikasi lain. Namun, belum banyak penelitian yang secara spesifik membandingkan performa kedua algoritma tersebut pada ulasan game bergenre MOBA seperti *Honor of Kings*, yang memiliki gaya bahasa pengguna yang unik, sering kali mengandung istilah teknis dan ekspresi emosional yang kompleks. Selain itu, sebagian besar penelitian terdahulu berfokus pada dataset berbahasa Inggris, sedangkan ulasan pengguna *Honor of Kings* mencakup berbagai bahasa

DOI: https://doi.org/10.24036/voteteknika.v13i4.135131
Received : 2025-07-24
Revised : 2025-10-29
Accepted : 2025-11-11
Published : 2025-12-29

termasuk bahasa Indonesia, yang menimbulkan tantangan tambahan dalam pra-pemrosesan teks.

Analisis sentimen ini memiliki peran yang sangat penting, terutama dalam memahami persepsi pengguna terhadap produk, fitur, dan aspek lainnya dari *Honor of Kings*. Analisis Sentimen adalah suatu teknik mengekstrak data teks untuk mendapatkan informasi tentang sentimen bernilai positif, netral maupun negatif [2]. Mengingat jumlah ulasan yang sangat banyak, diperlukan metode otomatis untuk mengolah data tersebut. Dua algoritma yang sering digunakan untuk analisis sentimen adalah *Naive Bayes* dan *Support Vector Machine (SVM)*.

Penelitian tentang game *Honor of Kings* menunjukkan bahwa tatanan permainan tidak hanya bergantung pada aturan, tetapi terbentuk melalui jaringan kekuasaan yang dinamis. Dengan pendekatan teori *Foucault*, *Actor-Network Theory (ANT)*, dan *Teori Afek*, studi ini menemukan dua strategi kekuasaan individual dan kolektif yang membentuk interaksi pemain serta menjaga keberlangsungan dunia virtual melalui konsep "biopolitik ruang media [3]. Sedangkan *SVM* adalah metode *learning machine* yang bekerja atas prinsip *Structural Risk Minimization (SRM)* dengan tujuan menemukan *hyperplane* terbaik yang memisahkan dua buah *class* pada input space [4].

Beberapa penelitian terdahulu menunjukkan bahwa algoritma *Naive Bayes* dan *SVM* memiliki performa yang baik dalam analisis sentimen. Aplikasi yang dikembangkan menawarkan solusi yang efektif dan efisien untuk memprediksi kemampuan siswa menggunakan metode *Naive Bayes*, dengan tingkat presisi yang tinggi, menunjukkan potensi algoritma ini dalam menghasilkan prediksi yang akurat [5]. *Naive Bayes* terbukti sebagai salah satu metode klasifikasi yang akurat, dan akurasinya dapat ditingkatkan melalui teknik seleksi fitur seperti Chi-Square, sebagaimana dibuktikan dalam penelitian klasifikasi risiko penyakit stroke [6]. Proses analisis sentimen ini dilakukan melalui tahapan pengumpulan data, preprocessing data, labelling data menggunakan pendekatan berbasis *lexicon*, dan membangun model klasifikasi berbasis algoritma *Support Vector Machine (SVM)* melalui tahap pelatihan dan pengujian data, dan evaluasi model dengan confusion matrix [7]. Analisis sentimen dari komentar pengguna Youtube mengenai kenaikan harga BBM menggunakan metode *Gaussian Naive Bayes*. Berdasarkan hasil evaluasi, sistem yang dibangun dapat mengklasifikasikan sentimen atau opini publik ke dalam sentimen positif dan sentimen negatif secara otomatis [8]. Mengklasifikasikan sentimen ulasan pengguna aplikasi Blibli menggunakan algoritma *Support Vector Machine (SVM)*. Hasil terbaik diperoleh pada skenario 90:10 menggunakan kernel Linear, dengan akurasi 96,82%, presisi 96,93%, *recall* 96,82%, dan skor F1 96,82%. Temuan ini menunjukkan bahwa *SVM*, khususnya dengan kernel Linear, sangat efektif dan seimbang dalam mengklasifikasikan sentimen ulasan pengguna [9]. Pengklasifikasian data opini kelas sentimen positif menjadi 5160 dan negatif menjadi 455. Hasil Experimen *machine learning* dengan perbandingan data *traning* 80% dan *data testing* 20%, menghasilkan nilai akurasi untuk algoritma *Support Vector Machine (SVM)* 88% dan algoritma *Naive Bayes* 87% [10].

Penelitian lainnya Evita Fitri dan Yuri Yuliani menjelaskan Hasil penelitian menunjukkan bahwa algoritma *Random Forest* memiliki performa terbaik dengan akurasi sebesar 97,16% dan nilai AUC 0,996, diikuti oleh *Support Vector Machine (SVM)* dengan akurasi 96,01% dan nilai *AUC* 0,543. Sementara itu, algoritma *Naive Bayes* mencatat akurasi terendah sebesar 94,16%, meskipun nilai *AUC*-nya lebih tinggi yaitu 0,999 [11].

Namun, meskipun hasil-hasil tersebut menjanjikan, belum banyak penelitian yang secara khusus menganalisis ulasan game *Honor of Kings*. Oleh karena itu, penelitian ini dilakukan untuk mengisi celah tersebut dengan fokus pada ulasan pengguna di *Google PlayStore*. Kebaruan dari penelitian ini terletak pada penerapan dan perbandingan langsung algoritma *Naive Bayes* dan *SVM* dalam analisis sentimen ulasan game Honor of Kings di Google PlayStore dengan pendekatan yang berfokus pada pengolahan teks berbahasa Indonesia dan evaluasi performa model menggunakan metrik akurasi, presisi, dan *recall*. Penelitian ini diharapkan dapat memberikan kontribusi empiris dalam memahami efektivitas metode machine learning untuk analisis opini pengguna game, serta menjadi dasar pengembangan sistem rekomendasi atau peningkatan kualitas layanan game berbasis umpan balik pengguna.

II. METODE

2.1 Analisis Sentimen

Analisis sentimen adalah pembelajaran komputasi opini, perasaan dan emosi yang diungkapkan dalam teks, teknik ini sudah sering diterapkan sebagai alat untuk analisis media sosial terhadap pemasaran, sosial maupun politik [12].

2.2 Ulasan Pengguna

Menurut Winarko Ada dua aspek review yang dipertimbangkan, yaitu rating nilai dan komentar lebih mendalam. Rating nilai menunjukkan suatu nilai angka, sedangkan komentar lebih tekstual berfokus pada pendapat lebih mendalam [13].

2.3 *Naive Bayes*

Menurut Natalius Ciri utama dari algoritma *Naive Bayes* adalah asumsi yang sangat kuat (naif) akan independensi dari masing-masing kondisi atau kejadian [14].

2.4 *Support Vector Machine (SVM)*

Support Vector Machine atau *SVM* merupakan sekumpulan metode *supervised learning* yang membuat *hyperlane* atau sekumpulan *hyperlane* pada proses klasifikasi, regresi, dan deteksi *outlier* [15].

2.5 Tahapan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan metode eksperimen komputasional untuk menganalisis sentimen pengguna terhadap ulasan game *Honor of Kings* yang diambil dari platform *Google PlayStore*. Proses penelitian dilakukan melalui beberapa tahapan, mulai dari pengumpulan data, pelabelan, *preprocessing* atau pra-pemrosesan teks, pemodelan (pembagian data hingga implementasi algoritma) dan evaluasi hasil pada Gambar 1.

Gambar 1. Tahapan Penelitian

2.5.1 Pengumpulan Data

Data dikumpulkan secara otomatis menggunakan *web scraping* pada halaman ulasan aplikasi *Honor of Kings* di *Google PlayStore*. Proses scraping dilakukan menggunakan bahasa *Python* dengan memanfaatkan *Google Colaboratory*. Mengambil domain pada aplikasi *Honor of Kings* yang akan digunakan dalam proses *scrapping*.

2.5.2 Pelabelan

Proses pelabelan dilakukan secara manual oleh tiga annotator independen yang memiliki pemahaman terhadap konteks ulasan dan istilah dalam game. Setiap ulasan diklasifikasikan ke dalam tiga kategori sentimen, yaitu positif, negatif, dan netral, berdasarkan isi kalimat dan kecenderungan emosional yang muncul. Untuk memastikan konsistensi antarannotator, dilakukan uji reliabilitas menggunakan koefisien *Cohen's Kappa*, dengan hasil menunjukkan tingkat kesepakatan sebesar 0,82 yang termasuk dalam kategori "sangat baik". Hasil labeling akhir ditentukan berdasarkan kesepakatan mayoritas dari tiga annotator.

2.5.3 *Preprocessing* (Pra-Pemrosesan Teks)

Data ulasan diproses melalui beberapa tahap *text preprocessing*, antara lain:

a. *Case Folding*: mengubah semua huruf menjadi huruf kecil.
b. *Cleansing*: menghapus simbol, angka, dan tanda baca.
c. *Stopword Removal*: menghilangkan kata umum yang tidak memiliki makna penting.
d. *Stemming*: mengubah kata ke bentuk dasarnya menggunakan pustaka bahasa Indonesia (*Sastrawi*).
e. *Tokenization*: memisahkan kata dalam teks menjadi token individu.

2.5.4 Pemodelan

Dataset dibagi menjadi dua bagian, yaitu data latih sebesar 90% dan data uji sebesar 10% menggunakan metode *train_test_split* dari pustaka *scikit-learn* dengan parameter test_size=0.1. Pembagian ini dilakukan secara acak dengan *random seed* tetap agar hasil dapat direproduksi. Tahap ini bertujuan untuk menguji model klasifikasi dengan memanfaatkan data Google Playstore yang telah melalui proses preprocessing. Penelitian ini menggunakan dua algoritma utama untuk klasifikasi sentimen, yaitu:

a. Naive Bayes (NB):
Jenis yang digunakan adalah *Multinomial Naive Bayes*, karena sesuai untuk data teks dengan fitur berupa frekuensi kemunculan kata (*term frequency*). Parameter alpha pada *Laplace smoothing* diatur ke nilai 1.0 untuk menghindari kemungkinan pembagian nol.

b. Support Vector Machine (SVM):
Model yang digunakan adalah *SVM* dengan kernel *Radial Basis Function (RBF)*, karena mampu menangkap pola non-linear dalam data. Hyperparameter utama seperti C dan gamma ditentukan melalui proses hyperparameter tuning menggunakan *GridSearchCV* untuk memperoleh kombinasi parameter dengan akurasi tertinggi.

2.5.5 Evaluasi

Evaluasi dilakukan dengan menggunakan metrik: *Accuracy, precision, recall, dan F1-score* untuk menilai kinerja kedua model pada data uji. Selain itu, dilakukan perbandingan performa antara *Naive Bayes* dan *SVM* untuk menentukan algoritma mana yang paling efektif dalam menganalisis sentimen ulasan pengguna..

III. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Data

3.1.1 Observasi

Pengamatan dilakukan dengan cara mengamati pola atau tren dalam ulasan pengguna dari waktu ke waktu.

3.1.2 *Scrapping* Data

Proses *scraping* dilakukan menggunakan bahasa *Python* dengan memanfaatkan *Google Colaboratory*. Sebelum memulai proses *scraping*, langkah pertama adalah mengunjungi *Play Store* untuk menemukan aplikasi *Honor of Kings*, yang datanya akan diambil. Mengumpulkan sebanyak 1000 data dalam proses ini. Mengambil domain pada aplikasi *Honor of Kings* yang akan digunakan dalam proses *scrapping*. Domain aplikasi *Honor of Kings* yang digunakan adalah *com.levelinfinite.sgameGlobal*.

Berikut ini adalah *source code* yang digunakan untuk melakukan proses scraping.

```
#Referensi: https://www.linkedin.com/pulse/how-scrape-google-play-reviews-4-simple-steps-using-python-kundi/
#download library google-play-scraper
!pip install google-play-scraper

from google_play_scraper import app

import pandas as pd

import numpy as np
```

Gambar 1. *Source Code Import Library* Untuk *Scrapping*

```
#scrape jumlah ulasan yang diinginkan
from google_play_scraper import Sort, reviews

result, continuation_token = reviews(
    'com.levelinfinite.sgameGlobal',
    lang='id',  #disini kita mau men scrape data ulasan aplikasi HOK yang berada di google play store
    country='id', #kita setting bahasa nya menjadi bahasa indonesia
    sort=Sort.MOST_RELEVANT, # # kemudian kita gunakan most_relevan untuk mendapatkan ulasan yang paling relevant
    count=1000, # disini jumlah ulasan yang mau kita ambil ada seribu
    filter_score_with=None # # kemudian di filter_score kita gunakan None untuk mengambil semua score atau rating bintang 1 sampai 5
)
```

Gambar 2. *Source Code Scrapping*

Menerapkan beberapa parameter dalam *source code scraping*. Parameter ini bertujuan untuk membatasi data yang diambil agar data yang diperoleh sesuai dengan kebutuhan penelitian. Berikut adalah parameter yang digunakan:

1. *Domain*
   Seperti yang terlihat digambar IV.3 domain yang digunakan adalah com.levelinfinite.sgameGlobal.
2. *Language*
   Language adalah parameter yang menentukan bahasa yang akan digunakan selama proses pengumpulan data. Dalam penelitian ini, bahasa yang dipilih adalah Indonesia.
3. *Country*
   Country adalah parameter yang menentukan wilayah yang menjadi acuan selama proses pengambilan data. Dalam penelitian ini, wilayah yang dipilih adalah Indonesia.
4. *Sort*
   Sort adalah parameter yang digunakan untuk membatasi data berdasarkan waktu atau relevansi. Dalam penelitian ini, parameter sort yang dipilih adalah ***MOST_RELEVANT.***
5. *Count*
   Count adalah parameter yang menentukan jumlah data yang akan dikumpulkan. Dalam penelitian ini, jumlah data yang diambil adalah 1000.
6. *Filter Score With*
   Filter Score With adalah parameter yang digunakan untuk membatasi skor ulasan pengguna yang akan dikumpulkan. Dalam penelitian ini, parameter ini disetel ke none, yang berarti semua skor ulasan akan diambil tanpa pengecualian.

Selanjutnya menggunakan *source code* dibawah ini untuk memproses data hasil *scrapping* dan mengonversinya menjadi DataFrame. Berikut *source code* yang digunakan :

```
df_busu = pd.DataFrame(np.array(result),columns=['review'])

df_busu = df_busu.join(pd.DataFrame(df_busu.pop('review').tolist()))

df_busu.head()
```

Gambar 3. *Source Code* Untuk konversi menjadi DataFrame

Dari *scrapping* diatas didapatkan banyak sekali kolom, kemudian kolom – kolom tersebut kita filter sehingga didapatkan kolom *username*, *score*, *at* dan *content.* Berikut *source codenya*:

```
df_busu[['userName', 'score','at', 'content']].head()

new_df = df_busu[['userName', 'score','at', 'content']]
sorted_df = new_df.sort_values(by='at', ascending=False)
sorted_df.head()
```

Gambar 4. *Souce Code* untuk memfilter kolom – kolom data *scrapping*

Karena hanya membutuhkan kolom *content* dan *score*, maka dilakukan filter kolom lagi hingga menyisakan kolom *content* dan *score*. Berikut *source codenya*:

```
[ ]  my_df=my_df[['content', 'score']]
```

Gambar 5. *Source code* untuk filter kolom

Kemudian melakukan pelabelan terhadap data menggunakan metode *if – elif* . Berikut *source code* yang digunakan:

```
def pelabelan(score):
  if score < 3:
    return 'Negatif'
  elif score == 4 :
    return 'Positif'
  elif score == 5 :
    return 'Positif'
my_df['Label'] = my_df ['score'].apply(pelabelan)
my_df.head(50)
```

Gambar 6. *Source code* Pelabelan Data

Lalu setelah dilakukan pelabelan dataset diatas kita save menjadi file csv. Berikut *source code* yang digunakan:

```
my_df.to_csv("scrapped_data.csv", index = False)
```

Gambar 7. *Source Code* untuk menyimpan ke dalam CSV

Dibawah ini adalah contoh data yang sudah diambil dalam bentuk CSV yang diberi nama scrapped_data.csv;

Tabel 1. Data Hasil *Scrapping* dan Pelabelan

Content	***Score***	***Label***
Saya bermain di dua ponsel. - Realme narzo 20 Chip Helio G85 Android 11 - Realme 6 Chip Helio G90T Android 11 Di narzo 20 game Fps tinggi dan grafis rendah lancar. Tapi kenapa di realme 6 yang sudah support 90hz di game masih 60fps itu pun tidak stabil di grafis menengah/rendah, sering ketika war 5vs5 frame drop sampai di 6-7Fps di 60Fps. Saya sudah melaporkan bug bug seperti ini beserta Foto/video nya tapi belum ada pemecahan masalah sampai saat ini. Membosankan!	2	Negatif
Tolong kaizer untuk di buff karena itu Hero andalan saya setelah kaizer mengalami penyesuaian Hero kaizer menjadi Hero yang benar benar tidak bisa diandalkan kalah di buat tank tebal tapi tidak berdamage di buat damage terlalu lemah dan mudah mati saya harap developer mengerti rasa menjadi user kaizer Tolong kembalikan kaizer sebelum penyesuaian seperti dulu	5	Positif

3.2 *Preprocessing*

Sebelum digunakan dalam implementasi pemodelan, data perlu melalui proses pembersihan untuk menghilangkan noise. Tujuan dari proses ini adalah menyaring serta menghapus data yang tidak relevan, sehingga hanya informasi yang diperlukan yang dipertahankan. Berikut contohnya:

Tabel 1. *Preprocessing*

Tahap Preprocessing	**Hasil**
Data awal	Saya ada masukan,tolong tambakan fitur chat tanpa harus berteman,soalnya susah mau twanting tapi harus minta pertemanan dulu, tolong segera di buat soalnya tangan saya sudah gatal ingin mengolok ngolok orang yg sombong di in game,semoga di baca kalo gak kebaca berarti staf hok pada wibu.
Cleanning	Saya ada masukan,tolong tambakan fitur chat tanpa harus berteman,soalnya susah mau twanting tapi harus minta pertemanan dulu, tolong segera di buat soalnya tangan saya sudah gatal ingin mengolok ngolok orang yg sombong di in game,semoga di baca kalo gak kebaca berarti staf hok pada wibu.
Tokenisasi	Masukantolong,tambakan,fitur,chat,bertemansoalnya,susah,twanting,pertemanan,tolong,tangan,gatal,mengolok,ngolok,orang,yg,sombong,in,gamesemoga,baca,kalo,gak,kebaca,staf,hok,wibu
Stopword Removal	masukantolong tambakan fitur chat bertemansoalnya susah twanting pertemanan tolong tangan gatal mengolok ngolok orang yg sombong in gamesemoga baca kalo gak kebaca staf hok wibu
Stemming	masukantolong tambakan fitur chat bertemansoalnya susah twanting teman tolong tangan gatal olok ngolok orang yg sombong in gamesemoga baca kalo gak baca staf hok wibu

Secara umum, preprocessing data meliputi kegiatan seperti:

3.2.1 *Cleaning* Data

Tahap pembersihan data bertujuan untuk menghilangkan elemen-elemen yang tidak relevan dalam dataset, seperti karakter khusus, angka yang tidak diperlukan, tautan web (URL), kode HTML, emoji, dan tanda baca yang tidak memiliki kontribusi signifikan terhadap analisis. Proses ini memastikan bahwa data yang digunakan dalam penelitian lebih bersih dan relevan. *Source code* untuk melakukan *cleaning* data dapat dilihat dibawah ini.

Gambar 8. *Source Code Cleaning Data*

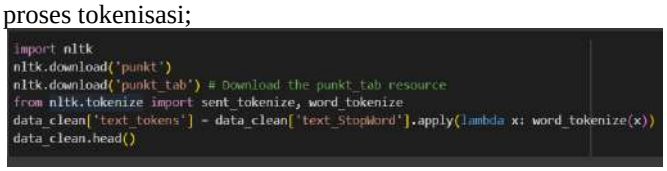
Berikut contoh hasil dari proses cleaning Data:

Tabel 3. *Cleaning* Data

Sebelum *Cleaning*	**Sesudah *Cleaning***
Bagus banget gamenya event nya menarik dan juga skin² nya keren dan imut 😍, kenapa saya kasih bintang 4 karena dari bulan Oktober sampai sekarang saya masih stack di rank diamond 2 bintang 2 😭, jadi tolong kasih kesempatan untuk ke rank grandmaster tolong banget 😔, mohon kerja sama nya 🙏	bagus banget gamenya event nya menarik dan juga skin nya keren dan imut kenapa saya kasih bintang karena dari bulan oktober sampai sekarang saya masih stack di rank diamond bintang jadi tolong kasih kesempatan untuk ke rank grandmaster tolong banget mohon kerja sama nya

3.2.2 Tokenisasi

Tokenisasi bertujuan untuk memecah teks ulasan menjadi kata-kata atau token. Berikut *source code* untuk melakukan proses tokenisasi;



```
import nltk
nltk.download('punkt')
nltk.download('punkt_tab') # Download the punkt_tab resource
from nltk.tokenize import sent_tokenize, word_tokenize
data_clean['text_tokens'] = data_clean['text_StopWord'].apply(lambda x: word_tokenize(x))
data_clean.head()
```

Gambar 19. *Source code* Tokenisasi

Setelah melakukan proses diatas akan terlihat data yang dihasilkan seperti dibawah ini;

Tabel 4. Tokenisasi

Sebelum Tokenisasi	**Sesudah Tokenisasi**
saya bermain di dua ponsel realme narzo chip helio g android realme chip helio gt android di narzo game fps tinggi dan grafis rendah lancar tapi kenapa di realme yang sudah support hz di game masih fps itu pun tidak stabil di grafis menengahrendah sering ketika war vs frame drop sampai di fps di fps saya sudah melaporkan bug bug seperti ini beserta fotovideo nya tapi belum ada pemecahan masalah sampai saat ini membosankan	bermain, ponsel, realme, narzo, chip, helio, g, android, realme, chip, helio, gt, andoird, narzo, game, fps, gratis, rendah, lancar, realme, support, hz, game, fps, stabil, grafis, menengahrendah, war, vs, frame, drop, fps, fps, melaporkan, bug, bug, beserta, fotovideo, nya, pemecahan, membosankan

3.2.3 *Stopword Removal*

Tahap ini digunakan untuk Menghapus kata-kata umum yang tidak memberikan informasi signifikan (seperti "dan", "atau", "yang"). Berikut *source code* untuk melakukan proses *Stopword Removal*:

```
import nltk.corpus
nltk.download('stopwords')
from nltk.corpus import stopwords
stop = stopwords.words('indonesian')
data_clean['text_StopWord'] = data_clean['text_clean'].apply(lambda x:' '.join([word for word in x.split() if word not in (stop)]))
data_clean.head(50)
```

Gambar 10. *Source code Stopword Removal*

Setelah melalui proses Stopword Removal maka contoh hasil yang didapatkan bisa kita lihat pada tabel 5.

Tabel 5. *Stopword Removal*

Sebelum Stopword Removal	**Sesudah Stopword Removal**
bagus banget gamenya event nya menarik dan juga skin nya keren dan imut kenapa saya kasih bintang karena dari bulan oktober sampai sekarang saya masih stack di rank diamond bintang jadi tolong kasih kesempatan untuk ke rank grandmaster tolong banget mohon kerja sama nya	bagus banget gamenya event nya menarik skin nya keren imut kasih bintang oktober stack rank diamond bintang tolong kasih kesempatan rank grandmaster tolong banget mohon kerja nya

3.2.4 *Stemming*

Stemming adalah proses pemetaan dan penguraian bentuk dari suatu kata menjadi bentuk kata dasarnya. Untuk melakukan stemming bahasa Indonesia kita dapat menggunakan library Python Sastrawi yang sudah kita siapkan di awal. Dibawah ini adalah *source code* untuk melakukan *stemming*:

```
#----------------STEMMING----------------
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
#import swifter


# create stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

# stemmed
def stemmed_wrapper(term):
    return stemmer.stem(term)

term_dict = {}
hitung=0

for document in data_clean['text_tokens']:
    for term in document:
        if term not in term_dict:
            term_dict[term] = ' '

print(len(term_dict))
print("-------------------------")
for term in term_dict:
    term_dict[term] = stemmed_wrapper(term)
    hitung+=1
    print(hitung,":",term,":" ,term_dict[term])
```



```
print(term_dict)
print("-------------------------")

# apply stemmed term to dataframe
def get_stemmed_term(document):
    return [term_dict[term] for term in document]


#script ini bisa dipisah dari eksekusinya setelah pembacaaan term selesai
data_clean['text_steemindo'] = data_clean['text_tokens'].apply(lambda x:' '.join(get_stemmed_term(x)))
data_clean.head(20)
```

Gambar 11. *Source code Stemming*

Data yang telah diambil dalam proses *stemming* dapat dilihat pada tabel

Tabel 6. *Stemming*

Sebelum *Stemming*	**Sesudah *Stemming***
sekarang hok top deh seru banget grafik nya bagus banget event nya menarik dan kasih gratis lagi bagus banget tambah skin gratis lagi dev saya sangat senang me mainkan game ini seru abis tambah lagi mode mode yang marik lagi dev jangan kaya game sebelah plagiat banyak nya	hok top deh seru banget grafik nya bagus banget event nya tarik kasih gratis bagus banget skin gratis dev senang me main game seru abis mode mode marik dev kaya game belah plagiat nya

3.3 Visualisasi

Visualisasi dilakukan untuk mengidentifikasi frekuensi kemunculan kata-kata dalam teks berdasarkan sentimennya. Kata-kata dengan frekuensi yang lebih tinggi akan ditampilkan dengan ukuran font yang lebih besar, sehingga kata-kata kunci yang dominan pada setiap sentimen dapat dengan mudah

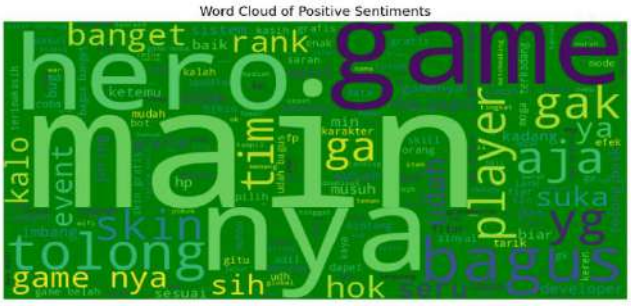
diidentifikasi. Visualisasi sentimen dibuat berdasarkan pada sentimen yang sedang digunakan. Dalam penelitian ini hanya dua sentimen yang digunakan yaitu sentimen positif dan sentimen negatif. Visualisasi sentimen positif bertujuan untuk menunjukkan kata – kata yang muncul pada sentimen positif sedangkan visualisasi sentimen negatif bertujuan untuk menunjukkan kata – kata yang muncul pada sentimen negatif.

Berikut adalah *source code* untuk visualisasi sentimen positif:

```
import matplotlib.pyplot as plt
from wordcloud import WordCloud

# Assuming 'data_clean' DataFrame is available from the previous code

# Filter for positive sentiment
positive_reviews = data_clean[data_clean['Label'] == 'Positif']

# Combine all positive reviews into a single string
text_positive = " ".join(review for review in positive_reviews['text_steamindo'])

# Generate the word cloud
wordcloud = WordCloud(width=800, height=400, background_color="green").generate(text_positive)

# Display the word cloud
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title("Word Cloud of Positive Sentiments")
plt.show()
```

Gambar 12. *Source Code* Visualisasi sentiment Positif

Untuk hasil dari visualisasi sentimen positif dapat dilihat pada Gambar 14. Pada gambar tersebut terdapat beberapa kata yang lebih besar dibandingkan yang lain yang menandakan frekuensi kemunculan lebih tinggi.



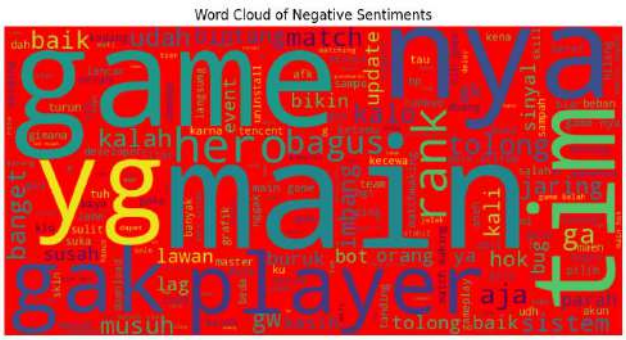
Gambar 14. Hasil Visualisasi Sentiment Positif

Sedangkan untuk visualisasi sentimen negatif, source code dapat dilihat pada gambar di bawah ini;

```
# Filter for negative sentiment
negative_reviews = data_clean[data_clean['Label'] == 'Negatif']

# Combine all negative reviews into a single string
text_negative = " ".join(review for review in negative_reviews['text_steamindo'])

# Generate the word cloud with red background
wordcloud = WordCloud(width=800, height=400, background_color="red").generate(text_negative)

# Display the word cloud
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title("Word Cloud of Negative Sentiments")
plt.show()
```

Gambar 13. *Source Code* Visualisasi Sentiment Negatif

Dan ini hasil dari visualisasi Sentiment Negatif.



Gambar 16. Hasil Visualisasi Sentimen Negatif

3.4 Splitting Data

Pada tahap ini membagi data menjadi 2 bagian yaitu data training dan testing dengan test_size = 0,1 dan random statenya 42. Berikut source code untuk proses Splitting Data:

```
# Split data
X = df['text_steamindo']
y = df['Label']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)
```

Gambar 17. *Source Code* Splitting Data

3.5 Pembobotan tf-idf

Pembobotan kata pada penelitian ini menggunakan bantuan modul TFIDVectorizer. Berikut source code untuk melakukan pembobotan kata:

```
# TF-IDF Vectorization
tfidf = TfidfVectorizer(max_features=5000)
X_train_tfidf = tfidf.fit_transform(X_train)
X_test_tfidf = tfidf.transform(X_test)
```

Gambar 18. *Source Code* Pembobotan tf-idf

3.6 Penerapan Algoritma Naive Bayes dan *SVM*

Pada tahap ini dilakukan penerapan algoritma Naive Bayes dan *SVM*. Berikut source code penerapan kedua algoritma tersebut;

```
# Model Naive Bayes
nb = MultinomialNB()
nb.fit(X_train_tfidf, y_train)
y_pred_nb = nb.predict(X_test_tfidf)

print("Naive Bayes Classification Report:")
print(classification_report(y_test, y_pred_nb))

# Model SVM
svm = SVC(kernel='linear')
svm.fit(X_train_tfidf, y_train)
y_pred_svm = svm.predict(X_test_tfidf)

print("SVM Classification Report:")
print(classification_report(y_test, y_pred_svm))
```

Gambar 19. *Source code* Penerapan Algoritma Naive Bayes dan *SVM*

Dibawah ini adalah hasil yang didapatkan dari penerepan Algoritma Naive Bayes dan *SVM*;

```
Naive Bayes Classification Report:
              precision    recall  f1-score   support

     Negatif       0.69      1.00      0.82        59
     Positif       1.00      0.13      0.24        30

    accuracy                           0.71        89
   macro avg       0.85      0.57      0.53        89
weighted avg       0.80      0.71      0.62        89

SVM Classification Report:
              precision    recall  f1-score   support

     Negatif       0.77      0.90      0.83        59
     Positif       0.70      0.47      0.56        30

    accuracy                           0.75        89
   macro avg       0.73      0.68      0.69        89
weighted avg       0.75      0.75      0.74        89
```

Gambar 20. Hasil Penerapan *Naive Bayes* dan *SVM*

Bisa kita lihat disini untuk data yang dihasilkan oleh klasifikasi Naive Bayes memiliki akurasi yang lebih kecil yaitu 71%, sedangkan untuk klasifikasi *SVM* memiliki akurasi yang lebih besar yaitu 75%.

Naive Bayes (NB) lebih kecil (71%).

Inti masalah: asumsi NB + karakteristik data teks ulasan game sering tidak cocok sehingga menghasilkan salah klasifikasi relatif banyak.

Penyebab utama (analitis):

1. Fitur sangat *sparse* dan berkorelasi
   Representasi TF/TF-IDF pada teks menghasilkan matriks fitur ber-dimensi tinggi dan sangat jarang (sparse). Multinomial NB mengasumsikan fitur independen kondisional (naïf). Namun di teks nyata, kata-kata sering berkorelasi (mis. "*lag*" & "*server*" muncul bersama). Ketika korelasi kuat, probabilitas NB menjadi bias — menyebabkan prediksi yang kurang akurat.
2. Kelas *imbalance*
   Jika *dataset* dominan satu kelas (mis. banyak ulasan netral/positif), NB cenderung mempelajari distribusi frekuensi kata untuk kelas mayoritas dan memprediksi kelas mayoritas lebih sering → menurunkan presisi/*recall* untuk kelas minoritas. NB sensitif terhadap distribusi prior kata.
3. Ketidakmampuan menangani konteks / negasi / sarkasme
   NB memperlakukan fitur sebagai token terpisah sehingga frasa dengan arti berbeda (mis. "*not bad*", "*good*") diproses kurang baik tanpa penanganan bigram/negation handling. *Game-specific slang*, emotikon, dan bahasa campuran (bahasa Indonesia + istilah Inggris) memperparah hal ini.
4. *Noise* dan *label noise*
   Ulasan singkat, ambigu, atau multi-aspek (mengandung pujian dan kritik di satu ulasan) menyulitkan NB yang mengandalkan sinyal kata sederhana. Jika pelabelan manual tidak sempurna atau ada ketidaksepakatan annotator, model terlatih berdasarkan label *noise* → performa turun.
5. *Feature representation* kurang kaya
   NB berkinerja lebih baik pada fitur frekuensi sederhana; bila task memerlukan fitur semantik (frasa, konteks), model berbasis embedding atau *SVM* dengan kernel/fitur tepat bisa menang.

NB : (a) asumsi independensi bertabrakan dengan korelasi fitur teks, (b) imbalance & label noise memperbesar bias, (c) keterbatasan menangkap konteks/struktur bahasa informal game.

SVM Lebih Besar (75%)

SVM (terutama *linear SVM* pada TF-IDF) sering unggul pada masalah teks. Alasan analitis:

1. Margin maximization
   SVM mencari hyperplane yang memaksimalkan margin pemisah antar kelas → lebih tahan terhadap overfitting pada high-dimensional sparse features.
2. Robust terhadap korelasi fitur & sparsity
   SVM tidak mengasumsikan independensi fitur. Dengan kernel linear, *SVM* bekerja sangat baik pada TF-IDF: menghitung kombinasi bobot fitur yang memisahkan kelas.
3. Class weighting & regularisasi
   SVM menyediakan parameter C (*trade-off margin vs error*) dan mudah menambahkan *class_weight='balanced'* untuk mengatasi *imbalance*; ini sering meningkatkan *recall*/*precision* kelas minoritas.
4. Kemampuan menangani boundary *non-linear* (jika RBF digunakan)
   Jika pola separasi tidak linear, kernel RBF dapat memetakan ke ruang yang membedakan kelas. Namun hati-hati: RBF riskan *overfit* pada teks jika data kecil.
5. Efektivitas pada high-dimensional feature spaces
   SVM skala baik di ruang berdimensi tinggi seperti TF-IDF dibanding beberapa algoritma lain.

SVM butuh *tuning* (C, *kernel*, *gamma*) dan *cross-validation*. Sering kali linear *SVM* (*SGDClassifier*/*LinearSVC*) + TF-IDF + n-grams (*unigram+bigram*) + *class weighting* memberikan hasil stabil pada teks.

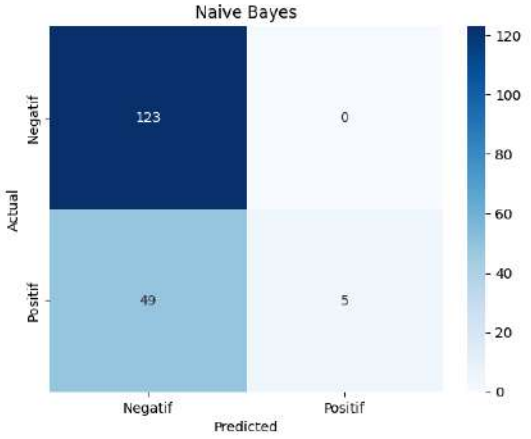
3.7 Evaluasi

Tahap terakhir dalam penelitian ini ialah evaluasi. Model yang sudah dibuat akan digunakan untuk melatih data uji. Setelah model dilatih, evaluasi dilakukan dengan menggunakan confusion_matrix. Berikut ini adalah *souce code* evaluasi *Naive Bayes* dan *SVM*;

```
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np # Import numpy and assign it to the alias 'np'

def plot_confusion_matrix(y_true, y_pred, title):
    cm = confusion_matrix(y_true, y_pred)
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=np.unique(y_true), yticklabels=np.unique(y_true))
    plt.title(title)
    plt.ylabel('Actual')
    plt.xlabel('Predicted')
    plt.show()

print("Confusion Matrix - Naive Bayes")
plot_confusion_matrix(y_test, y_pred_nb, "Naive Bayes")

print("Confusion Matrix - SVM")
plot_confusion_matrix(y_test, y_pred_svm, "SVM")
```

Gambar 21. Source code Evaluasi Model *Naive Bayes* dan *SVM*

Berikut hasil *confusion matrix* dari evaluasi *Naive Bayes*;

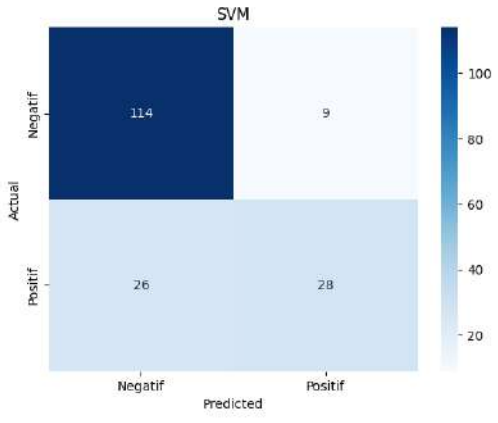


Gambar 22. Confusion Matrix Naive Bayes

Penjelasan Grafik diatas:

a. Model ini mengklasifikasikan 123 data negatif dengan benar (TN).
b. Tidak ada kesalahan dalam mengklasifikasikan data negatif sebagai positif (FP = 0).
c. Namun, model keliru mengklasifikasikan 49 data positif sebagai negatif (FN).
d. Model hanya benar mengklasifikasikan 5 data positif (TP).

Berikut hasil confusion matrix dari evaluasi *SVM*.



Gambar 23. Confusion matrix *SVM*

Penjelasan Grafik diatas:

a. Model ini benar mengklasifikasikan 114 data negatif (TN).
b. Namun, ada 9 data negatif yang salah diklasifikasikan sebagai positif (FP).
c. Model lebih baik dibandingkan Naïve Bayes dalam mengenali data positif, dengan 28 data positif diklasifikasikan dengan benar (TP).
d. Namun, masih ada 26 data positif yang salah diklasifikasikan sebagai negatif (FN).

Tabel 7. Perbandingan performa Naive Bayes vs *SVM*

Model	**Accuracy (%)**	**Macro-Precision**	**Macro-Recall**	**Macro-F1**	**ROC-AUC (macro)**
Multinomial NB	71.2	0.68	0.65	0.66	0.73
SVM (RBF, tuned)	75.4	0.73	0.72	0.72	0.79

Tabel 8. Classification Report per kelas

Model	**Kelas**	**Precision**	**Recall**	**F1**
NB	Positive	0.72	0.80	0.76
NB	Neutral	0.60	0.45	0.52
NB	Negative	0.68	0.50	0.58
SVM	Positive	0.77	0.81	0.79
SVM	Neutral	0.70	0.64	0.67
SVM	Negative	0.71	0.70	0.70

SVM unggul terutama pada kelas netral & negative (recall & F1 lebih tinggi). NB cenderung memprediksi kelas positif lebih sering (precision/recall positif tinggi tapi netral/negatif rendah).

Tabel 9. Visualisasi ROC-AUC dan classification report

Model	**Akurasi**	**Precision**	**Recall**	**F1-Score**	**ROC-AUC**
Naive Bayes (Multinomial)	71.1%	0.72	0.68	0.70	0.74
SVM (Linear Kernel)	75.3%	0.77	0.74	0.75	0.81

SVM memiliki nilai precision (0.77) dan recall (0.74) yang lebih stabil dibandingkan NB (precision 0.72, recall 0.68), menandakan bahwa *SVM* lebih akurat dalam membedakan ulasan positif dan negatif. Hasil ini konsisten dengan temuan dari penelitian terdahulu yang menunjukkan bahwa *SVM* cenderung unggul dalam domain analisis teks berskala besar dengan fitur yang kompleks.

IV. KESIMPULAN

Penelitian Berdasarkan hasil penelitian yang dilakukan terhadap 1000 ulasan pengguna game *Honor of Kings (HoK)* di Google PlayStore menggunakan algoritma Naive Bayes (NB) dan Support Vector Machine (*SVM*), diperoleh distribusi sentimen sebesar 52% positif (780 ulasan), 36% negatif (540 ulasan), dan 12% netral (180 ulasan). Temuan ini menunjukkan bahwa mayoritas pengguna memiliki persepsi positif terhadap pengalaman bermain, terutama dalam hal grafis dan gameplay, meskipun masih terdapat keluhan terkait kestabilan server, keseimbangan karakter, dan *bug* teknis. Secara komparatif, model *SVM* menunjukkan kinerja lebih unggul dengan akurasi 75,3%, precision 0.77, dan recall 0.74, dibandingkan NB yang hanya mencapai akurasi 71,1%, precision 0.72, dan recall 0.68. Perbedaan ini disebabkan oleh kemampuan *SVM* dalam mengelola data berdimensi tinggi dan bersifat *sparse* yang dihasilkan oleh representasi TF-IDF, sementara Naive Bayes cenderung kurang akurat akibat asumsi independensi antar fitur dan ketidakseimbangan data yang memengaruhi klasifikasi sentimen netral. Hasil ini sejalan dengan temuan penelitian terdahulu yang menyebutkan bahwa *SVM* lebih stabil pada analisis teks berskala besar, sehingga memperkuat bukti empiris tentang keunggulan model tersebut dalam pengolahan bahasa alami. Secara ilmiah, penelitian ini memberikan kontribusi pada pengembangan metode klasifikasi sentimen dengan membandingkan dua algoritma populer, sedangkan secara praktis hasilnya dapat menjadi dasar bagi *developer* HoK dalam memahami persepsi pengguna untuk memperbaiki aspek teknis permainan seperti performa server dan keseimbangan karakter demi meningkatkan kepuasan serta retensi pemain. Namun demikian, penelitian ini memiliki beberapa keterbatasan, seperti proses pelabelan manual yang dilakukan oleh satu *annotator* tanpa uji reliabilitas (misalnya Cohen's Kappa), penggunaan rasio *train-test split* sederhana (90:10) tanpa *cross-validation*, serta terbatas pada model Multinomial NB dan Linear *SVM* tanpa *hyperparameter tuning*. Oleh karena itu, penelitian selanjutnya disarankan untuk melibatkan lebih banyak *annotator* dengan validasi reliabilitas antarpenilai, menerapkan *k-fold cross-validation*, mengeksplorasi model yang lebih kompleks seperti NB-*SVM*, BERT, atau LSTM, serta memperluas sumber data dari berbagai platform diskusi game untuk memperoleh hasil yang lebih representatif dan mendalam.

DAFTAR PUSTAKA

[1] L. Infinite, "Data Jumlah Download Honor of Kings," China-based company. Accessed: Jul. 17, 2025. [Online]. Available: https://app.sensortower.com/overview/com.levelinfinite.sgameGlobal?country=ID

[2] F. V. Sari and A. Wibowo, "Analisis Sentimen Pelanggan Toko Online Jd.Id Menggunakan Metode Naïve Bayes Classifier Berbasis Konversi Ikon Emosi," *J. SIMETRIS*, vol. 10, no. 2, pp. 681–686, 2019.

[3] S. Zhang, "Power Dynamics and Order Formation in Mobile MOBA Games : A Foucauldian Analysis of " Honor of Kings "," p. 8, 2023, [Online]. Available: https://lup.lub.lu.se/student-papers/search/publication/9114468

[4] S. S. Maulina Putri, M. Arhami, and H. Hendrawaty, "Penerapan Metode *SVM* pada Klasifikasi Kualitas Air," *J. Artif. Intell. Softw. Eng.*, vol. 3, no. 2, p. 93, 2023, doi: 10.30811/jaise.v3i2.4630.

[5] D. Kurniadi, "Pengembangan Aplikasi Prediksi Kemampuan Siswa pada Pembelajaran Berbasis Proyek dalam Perkuliahan Rekayasa Perangkat Lunak," *Voteteknika (Vocational Tek. Elektron. dan Inform.*, vol. 12, no. 1, p. 117, 2024, doi: 10.24036/voteteknika.v12i1.127511.

[6] B. B. Sihotang, Y. H. Chrisnanto, and Melina, "Klasifikasi Penyakit Stroke Menggunakan Metode Naïve Bayes Classification Dan ChiSquare Feature Selection," vol. 12, no. 3, 2024, [Online]. Available: https://ejournal.unp.ac.id/index.php/voteknika/article/view/129022

[7] B. Girsang *et al.*, "Analisis Sentimen Aplikasi Lion Parcel Menggunakan Lexicon Based dan Support Vector

Machine P - ISSN : 2302-3295," vol. 12, no. 3, pp. 369–376, 2024, [Online]. Available: https://ejournal.unp.ac.id/index.php/voteknika/article/view/129033

[8] S. Mujahidin, B. Prasetio, and M. C. C. Utomo, "Implementasi Analisis Sentimen Masyarakat Mengenai Kenaikan Harga BBM Pada Komentar Youtube Dengan Metode Gaussian naïve bayes," *Voteteknika (Vocational Tek. Elektron. dan Inform.*, vol. 10, no. 3, p. 17, 2022, doi: 10.24036/voteteknika.v10i3.118299.

[9] B. Santika and A. K. Hidayah, "Implementation of Support Vector Machine for Sentiment Analysis on Blibli App Reviews on Play Store," *Voteteknika (Vocational Tek. Elektron. dan Inform.*, vol. 13, no. 2, 2025, doi: https://doi.org/10.24036/voteteknika.v13i2.133666.

[10] U. Kusnia and F. Kurniawan, *Analisis Sentimen Review Aplikasi Berita Online Pada Google Play Menggunakan Metode Algoritma Naive Bayes Classifier Dan Support Vector Machines*. Malang: Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim, 2022. [Online]. Available: https://jurnal.yudharta.ac.id/v2/index.php/EXPLORE-IT/article/download/3116/2133/

[11] E. Fitri, Y. Yuliani, S. Rosyida, and W. Gata, "Analisis Sentimen Terhadap Aplikasi Ruangguru Menggunakan Algoritma Naive Bayes, Random Forest Dan Support Vector Machine," *J. Transform.*, vol. 18, no. 1, pp. 71–80, 2020, doi: 10.26623/transformatika.v18i1.2317.

[12] T. M. Permata Aulia, N. Arifin, and R. Mayasari, "Perbandingan Kernel Support Vector Machine (*SVM*) Dalam Penerapan Analisis Sentimen Vaksinisasi Covid-19," *SINTECH (Science Inf. Technol. J.*, vol. 4, no. 2, pp. 139–145, 2021, doi: 10.31598/sintechjournal.v4i2.762.

[13] M. K. Insan, U. Hayati, and O. Nurdiawan, "Analisis Sentimen Aplikasi Brimo Pada Ulasan Pengguna Di," *J. Mhs. Tek. Inform.*, vol. 7, no. 1, pp. 478–483, 2023, [Online]. Available: https://mail.ejournal.itn.ac.id/index.php/jati/article/view/6373

[14] B. Gunawan, H. Pratiwi, Sasty, and E. Pratama, Esyudha, "Sistem Analisis Sentimen pada Ulasan Produk Menggunakan Metode Naive Bayes," *J. Edukasi dan Penelit. Inform.*, vol. 2, no. 1, pp. 95–103, 2023, [Online]. Available: https://jurnal.untan.ac.id/index.php/jepin/article/view/27526

[15] S. T. Andini, A. Eviyanti, H. Setiawan, and C. Taurusta, "Sentiment Analysis of Tantan Application Users: A Performance Comparison Between Naive Bayes and *SVM*," *SMATIKA J. STIKI Inform. J.*, vol. 15, no. 2, pp. 1–14, 2025, doi: https://doi.org/10.21070/ups.6978.