

## Analisis Sentimen Ulasan Pengguna Aplikasi YouCut Pada Google Play Store Menggunakan Metode Logistic Regression Classifier

Athifa Dihyan Calysta, Budi Prasetyo, Regina Ayumi Ulayyaa

Program Studi Sistem Informasi, FMIPA, Universitas Negeri Semarang, Indonesia,  
Program Studi Teknik Informatika, FMIPA, Universitas Negeri Semarang, Indonesia

Gedung D5 Lt.2, Kampus Sekaran Gunungpati, Semarang 50229

E-mail: athifakbn@students.unnes.ac.id, bprasetyo@mail.unnes.ac.id,

reginaayumi@students.unnes.ac.id

Diterima 30 Juli 2025      Disetujui 01 April 2026

### Abstrak

Perkembangan teknologi digital yang pesat mendorong masyarakat semakin aktif dalam memberikan ulasan terhadap layanan digital, termasuk aplikasi mobile. Ulasan pengguna memainkan peran penting dalam membentuk reputasi dan keberhasilan aplikasi di pasar. Namun, tingginya volume ulasan menyulitkan proses analisis secara manual. Penelitian ini bertujuan untuk menganalisis sentimen ulasan pengguna aplikasi YouCut secara otomatis dengan pendekatan machine learning menggunakan algoritma Logistic Regression. Dataset berupa pengambilan 1.000 ulasan pengguna dalam bahasa Indonesia diambil secara otomatis dari Google Play Store dan diproses melalui tahapan data cleaning, text preprocessing, pelabelan sentimen berbasis lexicon, ekstraksi fitur dengan TF-IDF, serta pelatihan dan evaluasi model. Hasil analisis menunjukkan bahwa model Logistic Regression mampu mengklasifikasikan sentimen dengan akurasi mencapai 89%. Visualisasi distribusi sentimen menunjukkan dominasi sentimen positif sebesar 62,2%. Analisis kata dominan melalui wordcloud mengungkapkan bahwa pengguna mengapresiasi kemudahan penggunaan dan hasil edit video, namun juga menyuarakan keluhan terhadap performa dan gangguan iklan. Temuan ini memberikan wawasan strategis bagi pengembang untuk meningkatkan fitur dan pengalaman pengguna aplikasi YouCut secara berkelanjutan.

Kata kunci: Logistic Regression, Machine Learning, Sentimen, Ulasan, YouCut

### Abstract

*The rapid development of digital technology encourages people to be more active in reviewing digital services, including mobile applications. User reviews play an important role in shaping an app's reputation and success in the market. However, the high volume of reviews makes it difficult to analyze manually. This research aims to automatically analyze the sentiment of YouCut app user reviews with a machine learning approach using the Logistic Regression algorithm. The dataset in the form of 1,000 user reviews in Indonesian language is taken automatically from Google Play Store and processed through the stages of data cleaning, text preprocessing, lexicon-based sentiment labeling, feature extraction with TF-IDF, and model training and evaluation. The analysis results show that the Logistic Regression model is able to classify sentiment with an accuracy of 89%. Visualization of sentiment distribution shows the dominance of positive sentiment by 62.2%. Analysis of dominant words through wordcloud revealed that users appreciated the ease of use and video editing results, but also voiced*

*complaints about performance and ad interruptions. The findings provide strategic insights for developers to continuously improve the features and user experience of the YouCut app.*

Keywords: Logistic Regression, Machine Learning, Reviews, Sentiment, YouCut

#### **How to cite:**

Calysta, A. D., Prasetyo, B., & Ulayyaa, R. A. (2024). Analisis sentimen ulasan pengguna aplikasi YouCut pada Google Play Store menggunakan metode Logistic Regression Classifier. *Indonesian Journal of Mathematics and Natural Sciences*, 47(1). 42-56

#### **PENDAHULUAN**

Dalam ekosistem digital yang terus menerus berkembang di tiap harinya, manusia saat ini menjadi lebih terbiasa dengan penggunaan teknologi digitalnya. Jaringan internet yang semakin meluas pun menjadikan manusia lebih aktif dalam memberikan ulasan sebagai bumbu pelengkap terhadap layanan digital. Aplikasi mobile menjadi salah satu target dalam pemberian ulasan yang diungkapkan oleh khalayak masyarakat. Dengan berbekal pengalaman dan kepuasan dalam penggunaan aplikasi, ulasan-ulasan tersebut pada akhirnya memengaruhi reputasi dan keberhasilan aplikasi di pasarnya. Karenanya, pengembang maupun pengelola aplikasi diharapkan dapat memahami ulasan sentimen dari para pengguna sehingga dapat meningkatkan kualitas layanan aplikasinya. Namun, terkadang terdapat aplikasi yang mana jumlah ulasan masuk dapat bertambah terus-menerus sehingga sulit untuk memprosesnya secara manual. Hal tersebut mendorong pendekatan langsung yang berbasis machine learning dan Natural Language Processing (NLP).

Penelitian ini mengangkat permasalahan mengenai klasifikasi sentimen ulasan aplikasi secara langsung atau otomatis dan akurat. Sejumlah pendekatan telah digunakan untuk menyelesaikan masalah tersebut, mulai dari analisis berbasis kamus (lexicon-based), penggunaan metode Naïve Bayes, dan Support Vector Machine (SVM) hingga pendekatan lebih kompleks seperti neural networks. Beberapa studi yang telah ada sebelumnya menunjukkan bahwa metode Logistic Regression juga efektif dalam pemecahan masalah klasifikasi teks, terutama dalam konteks binary classification seperti pada penelitian yang telah dilakukan oleh Pulungan et al. (2024). Oleh karena itu, penelitian ini menggunakan pendekatan label sentimen yang diperoleh dari metode lexicon-based sebagai tahap awal, dan hasil target klasifikasinya dalam supervised learning menggunakan metode Logistic Regression karena kemampuannya yang tinggi dalam memodelkan data linier serta kemudahan interpretasi pada hasilnya.

Analisis sentimen ulasan aplikasi telah menjadi topik yang dilakukan pada banyak penelitian sebelumnya. Penelitian ini memiliki rujukan utama yang dilakukan oleh Wahyudi et al. (2021), analisis sentimen juga telah diterapkan pada ulasan aplikasi Grab menggunakan metode Support Vector Machine (SVM). Pada penelitian yang telah dilakukan oleh Putri and Kharisudin (2022) menunjukkan bahwa Logistic Regression dengan pendekatan Synthetic Minority Oversampling Technique (SMOTE) dapat menghasilkan (Area UnderCurve) AUC tinggi dalam klasifikasi review pada aplikasi Tokopedia. Hal ini didukung dengan penelitian yang telah dilakukan oleh Kisma et al. (2023) yang telah membandingkan algoritma Naïve Bayes dengan Logistic Regression untuk mengklasifikasikan aplikasi berdasarkan rating dan jenisnya. Penelitian tersebut mendapatkan hasil berupa metode Logistic Regression yang memiliki nilai hampir mirip dengan metode Naïve Bayes meskipun tidak lebih tinggi darinya. Sehingga berdasarkan pada tingkat keberhasilan penggunaan metode Logistic Regression dalam penelitian sebelumnya, penelitian ini menggunakan algoritma tersebut untuk menganalisis sentimen ulasan pada aplikasi YouCut.

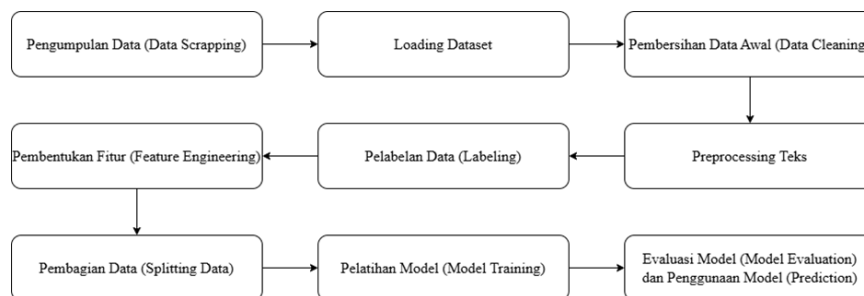
Dalam beberapa penelitian sebelumnya cenderung pada pembahasan mengenai klasifikasi berdasarkan sentimen biner (positif-negatif) dan belum banyak yang menggabungkan analisis teks menggunakan TF-IDF dengan evaluasi model komprehensif yang menggunakan confusion matrix dan metrik lain seperti precision, recall, dan F1-score.

Selain itu, algoritma Logistic Regression yang digunakan pada analisis dataset ulasan aplikasi setelah melalui proses preprocessing secara mendalam, seperti penggunaan stopwords removal, tokenisasi, dan balancing data tanpa SMOTE, serta evaluasi model murni berdasarkan performa klasifikasi aktual tanpa overfitting belum banyak digunakan.

Maka dari itu, penelitian ini akan menganalisis dan membahas mengenai sentimen ulasan pengguna pada aplikasi YouCut. YouCut merupakan aplikasi mobile yang digunakan untuk mengedit berbagai jenis video yang tersedia di Google Play Store (Fuad et al., 2020). Dataset yang digunakan berupa data mentah pada ulasan pengguna aplikasi YouCut yang kemudian dibersihkan dari kolom atau baris yang memiliki nilai hilang (Not a Number (NaN)). Selanjutnya, dataset mentah yang telah menjadi dataset bersih diproses melalui tahap preprocessing, seperti pembersihan teks dari angka maupun tanda baca, penghapusan duplikat yang mungkin terjadi, stopwords removal, tokenisasi, dan konversi menjadi bentuk numerik menggunakan metode TF-IDF. Label sentimen diperoleh melalui pendekatan lexicon-based secara langsung. Karena hasil evaluasi menunjukkan bahwa penggunaan algoritma Logistic Regression memiliki nilai akurasi tertinggi pada uji pemodelan, sehingga algoritma tersebut dipilih menjadi metode yang digunakan pada analisis sentimen pada ulasan aplikasi YouCut. Evaluasi menggunakan algoritma Logistic Regression dilakukan menggunakan metrik akurasi, precision, recall, dan F1-score. Seluruh rangkaian proses tersebut dilakukan menggunakan pustaka Python seperti pandas, scikit-learn, dan matplotlib pada Google Colab. Sehingga diharapkan analisis sentimen ulasan pada aplikasi YouCut dengan menggunakan metode Logistic Regression dapat membantu pengembang aplikasi YouCut memahami persepsi pengguna dalam menggunakan layanan aplikasi dan mendukung peningkatan layanan serta fitur pada aplikasi.

## METODE

Dataset dalam penelitian ini berupa ulasan dari komentar yang terdapat pada aplikasi YouCut yang didapatkan dari situs Google Play Store. Dengan percobaan pengambilan jumlah data hingga sebanyak 1.000 ulasan, kumpulan data komentar tersebut diambil secara otomatis menggunakan pustaka Python google-play-scraper dengan bahasa yang diatur ke dalam Bahasa Indonesia dan negara asal ulasan dari Indonesia. Tahapan yang dilakukan pada penelitian ini dapat dilihat pada gambar 1 sebagai berikut:



Gambar 1. Flowchart Tahapan

Analisis sentimen terhadap sentimen ulasan pengguna aplikasi ini dilakukan melalui tahapan yang dimulai dari pengumpulan data ulasan secara langsung dari Google Play Store (Fahlevvi, 2022). Dataset mentah yang diperoleh kemudian melalui proses pembersihan dan pelabelan sentimen berdasarkan skor ulasan. Selanjutnya, teks ulasan diproses melalui serangkaian tahap preprocessing untuk mempersiapkannya sebelum diubah menjadi representasi numerik melalui pembentukan fitur. Representasi numerik ini kemudian akan digunakan untuk melatih dan mengevaluasi model klasifikasi Logistic Regression yang digunakan untuk memprediksi sentimen ulasan.

### 2.1 Pengumpulan Data (Data Scrapping)

Tahap awal dari proses penelitian ini dilakukan dengan pengumpulan data mentah dari ulasan pengguna aplikasi YouCut. Langkah ini dilakukan dengan penggunaan scrapping atau pengambilan data secara langsung atau otomatis dari Google Play Store (Subagja et al., 2021). Sebagai input, proses ini memerlukan identifikasi unik dari aplikasi YouCut dengan menggunakan id='com.camerasideas.trimmer' yang didapat dari Google Play Store. Proses scrapping yang dilakukan menggunakan fungsi reviews\_all yang telah disediakan oleh google-play-scraper untuk mengambil ulasan berdasarkan parameter yang diberikan. Proses ini nantinya akan menghasilkan output berupa data ulasan mentah dalam format terstruktur.

### 2.2 Loading Dataset

Dataset ulasan mentah yang telah berhasil dikumpulkan pada tahap pengumpulan data perlu diorganisir ke dalam struktur yang mudah untuk diakses dan dimanipulasi. Data ulasan mentah yang tersimpan dalam variabel hasil dari scrapping kemudian diubah struktur datanya menjadi sebuah data Frame. DataFrame ini berupa struktur data berbentuk baris dan kolom yang mirip dengan tabel dalam database atau spreadsheet. Objek DataFrame pandas (app\_reviews\_df) yang merupakan hasil dari proses ini selanjutnya akan dilakukan tahap pembersihan dan analisisnya. Informasi mengenai DataFrame yang berisi data ulasan mentah dapat dilihat pada Gambar 2. berikut

```
# Menampilkan informasi tentang DataFrame app_reviews_df
app_reviews_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 112500 entries, 0 to 112499
Data columns (total 11 columns):
 #   Column                Non-Null Count  Dtype  
---  --
 0   reviewId              112500 non-null object  
 1   userName              112500 non-null object  
 2   userImage             112500 non-null object  
 3   content               112500 non-null object  
 4   score                112500 non-null int64  
 5   thumbsUpCount         112500 non-null int64  
 6   reviewCreatedVersion   87181 non-null  object  
 7   at                   112500 non-null datetime64[ns]
 8   replyContent          3997 non-null  object  
 9   repliedAt             3997 non-null  datetime64[ns]
10  appVersion            87181 non-null  object  
dtypes: datetime64[ns](2), int64(2), object(7)
memory usage: 9.4+ MB
```

Gambar 2. Informasi Data Ulasan Mentah

### 2.3 Pembersihan Data Awal (Data Cleaning)

Data mentah seringkali mengandung nilai yang hilang (missing values) atau duplikasi yang tidak diinginkan (Ali et al., 2021). Sehingga pada tahap ini bertujuan untuk mengatasi masalah tersebut. Seperti pada penelitian yang dilakukan oleh Saputra (2020), DataFrame yang baru saja dimuat dalam app\_reviews\_df dilakukan proses pembersihan dengan dua langkah utama, yaitu menghapus baris yang mengandung setidaknya satu nilai yang hilang dengan menggunakan metode .dropna(), dan menghapus baris yang merupakan duplikat persis dari baris lain dengan menggunakan metode .drop\_duplicates(). DataFrame akan menjadi lebih bersih dengan menggunakan clean\_df, di mana setiap baris mewakili ulasan unik yang tidak memiliki nilai yang hilang pada kolom-kolom penting yang akan digunakan. Informasi mengenai DataFrame yang telah dilakukan proses pembersihan dapat dilihat pada Gambar 3 sebagai berikut.

```
# Menampilkan informasi tentang DataFrame clean_df setelah membersihkan nilai yang hilang (NaN)
clean_df.info()

<class 'pandas.core.frame.DataFrame'>
Index: 3371 entries, 108 to 112431
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   reviewId               3371 non-null   object  
1   userName               3371 non-null   object  
2   userImage              3371 non-null   object  
3   content                3371 non-null   object  
4   score                  3371 non-null   int64   
5   thumbsUpCount          3371 non-null   int64   
6   reviewCreatedVersion    3371 non-null   object  
7   at                     3371 non-null   datetime64[ns]
8   replyContent           3371 non-null   object  
9   repliedAt              3371 non-null   datetime64[ns]
10  appVersion              3371 non-null   object  
dtypes: datetime64[ns](2), int64(2), object(7)
memory usage: 316.0+ KB
```

Gambar 3. Informasi Data Ulasan Final

## 2.4 Preprocessing Teks

Tahap preprocessing ini dilakukan dengan tujuan untuk menstandarisasi dan membersihkan teks karena teks ulasan pengguna sering kali menggunakan bahasa yang tidak baku dan mengandung elemen yang tidak perlu untuk dianalisis (Chai, 2023). Kolom teks dari DataFrame yang sudah diberi label kemudian diproses dengan serangkaian transformasi teks yang dilakukan secara berurutan menggunakan fungsi-fungsi sebagai berikut.

- Cleaning, yaitu fungsi untuk menghapus karakter atau pola yang tidak diinginkan seperti mentions, hashtags, links, angka, maupun tanda baca (Anugrah, 2019).
- Case Folding, berupa fungsi untuk mengubah semua teks menjadi huruf kecil.
- Tokenisasi, fungsi untuk memecah kalimat menjadi kata-kata individu (Sirts & Peekman, 2020).
- Filtering Stop Words, fungsi untuk menghapus kata-kata umum yang tidak memiliki arti penting dalam konteks sentimen.
- Fixing Slang Words, berupa fungsi yang bertujuan untuk mengganti kata-kata slang atau singkatan dengan bentuk bakunya.
- Stemming, untuk mengubah kata menjadi bentuk dasarnya (Qulub et al., 2020).

Tahapan stemming ini biasanya menghasilkan kolom-kolom baru di DataFrame yang menyimpan hasil dari setiap proses transformasi. Dengan menggunakan tool re untuk regular expressions, string, nltk untuk tokenisasi dan stop word, Sastrawi untuk stemming bahasa Indonesia, dan pandas untuk mengelola teks dalam DataFrame, teks ulasan yang bersih dan seimbang sudah siap untuk diubah menjadi fitur numerik. Cuplikan hasil setelah dilakukan proses preprocessing teks terdapat pada Gambar 4 sebagai berikut.

| clean_df               |                                                                                                                                                                                                                                                                                                                                |                     |            |                                                                                                                                         |                                                                                                                                         |                                                                                                                                         |                                                                                                                                                           |                                                                                                               |                                                                                                            |                                                                                                |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| at                     | replyContent                                                                                                                                                                                                                                                                                                                   | repliedAt           | appVersion | text_clean                                                                                                                              | text_casefoldingText                                                                                                                    | text_slangwords                                                                                                                         | text_tokenizingText                                                                                                                                       | text_stopword                                                                                                 | text_stemmed                                                                                               | text_akhir                                                                                     |
| 2025-05-29 05:29:33.59 | Torima kasih atas masukan Andai Kami menghargai waktu yang Anda luangkan untuk berbagi pemikiran Anda. Kami terus berupaya untuk meningkatkan aplikasi dan menghadirkan lebih banyak fitur yang bermanfaat. Saran Anda terkait teks otomatis akan kami terima. Silakan hubungi kami di youcut@inshot.com jika Anda tidak dapat | 2025-05-29 07:35:29 | 1.673.1205 | Fitur teks otomatisperubah teks ke audioekstrak video to video tidak ada seperti capcut harusnya dilengkapi lagi fiturnyaselebihnya oke | fitur teks otomatisperubah teks ke audioekstrak video to video tidak ada seperti capcut harusnya dilengkapi lagi fiturnyaselebihnya oke | fitur teks otomatisperubah teks ke audioekstrak video to video tidak ada seperti capcut harusnya dilengkapi lagi fiturnyaselebihnya oke | [fitur, teks, otomatisperubah, teks, ke, audioekstrak, video, to, video, tidak ada, seperti, capcut, harusnya, dilengkapi, lagi, fiturnyaselebihnya, oke] | [fitur, teks, otomatisperubah, teks, audioekstrak, video, video, capcut, dilengkapi, fiturnyaselebihnya, oke] | [fitur, teks, otomatisperubah, teks, audioekstrak, video, video, capcut, lengkap, fiturnyaselebihnya, oke] | fitur teks otomatisperubah teks audioekstrak video video capcut lengkap fiturnyaselebihnya oke |

Gambar 4. Hasil Preprocessing

```

# Membersihkan teks dan menyimpannya di kolom 'text_clean'
clean_df['text_clean'] = clean_df['content'].apply(cleaningText)

# Mengubah huruf dalam teks menjadi huruf kecil dan menyimpannya di 'text_casefoldingText'
clean_df['text_casefoldingText'] = clean_df['text_clean'].apply(casefoldingText)

# Mengganti kata-kata slang dengan kata-kata standar dan menyimpannya di 'text_slangwords'
clean_df['text_slangwords'] = clean_df['text_casefoldingText'].apply(fix_slangwords)

# Memecah teks menjadi token (kata-kata) dan menyimpannya di 'text_tokenizingText'
clean_df['text_tokenizingText'] = clean_df['text_slangwords'].apply(tokenizingText)

# Menghapus kata-kata stop (kata-kata umum) dan menyimpannya di 'text_stopword'
clean_df['text_stopword'] = clean_df['text_tokenizingText'].apply(filteringText)

# Terapkan fungsi yang dimodifikasi (stemmingText_list_of_words)
clean_df['text_stemmed'] = clean_df['text_stopword'].apply(stemmingText_list_of_words)

# Kemudian, gabungkan lagi untuk TF-IDF jika TF-IDF mengharapakan string
clean_df['text_akhir'] = clean_df['text_stemmed'].apply(lambda x: ' '.join(x))

```

Gambar 5. Cuplikan Kode Preprocessing

## 2.5 Pelabelan Data (Labeling)

Karena analisis sentimen yang akan dilakukan adalah supervised learning, setiap ulasan perlu memiliki label sentimen berupa Positif ataupun Negatif. Labeling ini dibuat berdasarkan skor ulasan yang tersedia dalam data menggunakan DataFrame yang sudah dibersihkan menggunakan `clean_df`, khususnya pada kolom yang berisi skor ulasan ('score') (Putra, 2023). Proses pelabelan melibatkan pembuatan aturan sederhana berdasarkan nilai skor ulasan dengan menetapkan label "Positif" untuk skor 4 dan 5, "Negatif" untuk skor 1 dan 2, dan penggunaan "Netral" ataupun pengabaian untuk skor 3. Aturan ini diterapkan pada setiap baris pada DataFrame. Tool pandas untuk memanipulasi DataFrame dan operasi kondisional menghasilkan DataFrame yang sama namun dengan penambahan satu kolom baru berisi label sentimen untuk setiap ulasan.

```

import csv
import requests
from io import StringIO

# Loads positive lexicon data from GitHub
# Membaca data kamus kata-kata positif dari GitHub
lexicon_positive = dict()

response = requests.get('https://raw.githubusercontent.com/angelmetanosaa/dataset/main/lexicon_positive.csv')
# Mengirim permintaan HTTP untuk mendapatkan file CSV dari GitHub

if response.status_code == 200:
    # Jika permintaan berhasil
    reader = csv.reader(StringIO(response.text), delimiter=',')
    # Membaca teks respons sebagai file CSV menggunakan pembaca CSV dengan pemisah koma

    for row in reader:
        # Mengulangi setiap baris dalam file CSV
        lexicon_positive[row[0]] = int(row[1])
        # Menambahkan kata-kata positif dan skornya ke dalam kamus lexicon_positive
else:
    print("Failed to fetch positive lexicon data")

```

Gambar 6. Cuplikan Kode Labeling

```

# Loads negative lexicon data from GitHub
# Membaca data kamus kata-kata negatif dari GitHub
lexicon_negative = dict()

response = requests.get('https://raw.githubusercontent.com/angelmetanosaa/dataset/main/lexicon_negative.csv')
# Mengirim permintaan HTTP untuk mendapatkan file CSV dari GitHub

if response.status_code == 200:
    # Jika permintaan berhasil
    reader = csv.reader(StringIO(response.text), delimiter=',')
    # Membaca teks respons sebagai file CSV menggunakan pembaca CSV dengan pemisah koma

    for row in reader:
        # Mengulangi setiap baris dalam file CSV
        lexicon_negative[row[0]] = int(row[1])
        # Menambahkan kata-kata negatif dan skornya ke dalam kamus lexicon_negative
else:
    print("Failed to fetch negative lexicon data")

```

Gambar 7. Cuplikan Lanjutan Kode Labeling

## 2.6 Pembentukan Fitur (Feature Engineering)

Tahap pembentukan fitur akan mengubah teks ulasan yang telah diproses menjadi vektor angka (Gumilang, 2018). Karena pada algoritma machine learning, termasuk Logistic Regression memerlukan input dalam format numerik. Teks ulasan yang sudah bersih dan distandardisasi selanjutnya dilakukan teknik TF-IDF (Term Frequency-Inverse Document Frequency) yang menghitung seberapa sering sebuah kata muncul dalam sebuah ulasan (Term Frequency) dan memberikan bobot yang lebih tinggi pada kata-kata yang jarang muncul di seluruh kumpulan ulasan (Inverse Document Frequency). Sehingga kata-kata unik atau khas dari kumpulan ulasan tersebut menjadi lebih penting. Penggunaan TF-IDF mengubah korpus teks menjadi bentuk matriks numerik di mana setiap baris mewakili satu ulasan dan setiap kolom mewakili sebuah kata, dengan nilai sel yang menunjukkan bobot TF-IDF kata tersebut dalam ulasan. Tools `TfidfVectorizer` atau `CountVectorizer` dari pustaka `sklearn.feature_extraction.text` menghasilkan matriks numerik yang merepresentasikan fitur dari setiap ulasan.

```
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import TfidfVectorizer

# Set the figure size
plt.figure(figsize=(10, 6))

# Check class distribution
class_dist_plot = sns.countplot(x='polarity', data=clean_df)
plt.title('Class Distribution')

# Add counts above the bars on the plot
for p in class_dist_plot.patches:
    class_dist_plot.annotate(format(p.get_height(), '.0f'),
                             (p.get_x() + p.get_width() / 2., p.get_height()),
                             ha='center', va='center',
                             xytext=(0, 10),
                             textcoords='offset points')

plt.show()
```

Gambar 8. Cuplikan Kode Feature Engineering

## 2.7 Pembagian Data (Splitting Data)

Dataset yang telah memiliki fitur numerik dan label sentimen perlu dibagi menjadi dua set terpisah, yaitu data pelatihan (training set) dan data pengujian (testing set) untuk melatih model machine learning yang kemudian mengevaluasi kinerjanya secara objektif. Matriks fitur dan kolom label sentimen dari seluruh dataset dilakukan proses pembagian yang dilakukan secara acak untuk memastikan distribusi data yang proposional di kedua set (Putro et al., 2024). Tool fungsi `train_test_split` dari pustaka `sklearn.model_selection` digunakan untuk menghasilkan empat subset data berupa matriks fitur untuk pelatihan (`X_train`), vektor label untuk pelatihan (`y_train`), matriks fitur untuk pengujian (`X_test`), dan vektor label untuk pengujian (`y_test`).

## 2.8 Pelatihan Model (Model Training)

Model Logistic Regression yang merupakan algoritma klasifikasi linier untuk memprediksi probabilitas bahwa sebuah instance (ulasan) termasuk ke dalam kelas tertentu berupa sentimen Positif atau Negatif dibangun dengan menggunakan data pelatihan. Proses pelatihan untuk data pelatihan (`X_train` dan `y_train`) melibatkan algoritma Logistic Regression dengan mencari parameter atau bobot terbaik yang dapat memisahkan kelas-kelas sentimen berdasarkan fitur-fitur teks yang ada pada `X_train` dan label sentimen pada `y_train`. Sehingga tool Logistic regression dari pustaka `sklearn.linear_model` dapat digunakan untuk memprediksi sentimen pada data baru.

```
# Bagi data menjadi data latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(X_tfidf, y, test_size=0.2, random_state=42)
```

Gambar 9. Cuplikan Kode Model Training

## 2.9 Evaluasi Model (Model Evaluation) dan Penggunaan Model (Prediction)

Tahap evaluasi menggunakan data pengujian dan model Logistic Regression yang telah dilatih untuk mengetahui seberapa baik model tersebut bekerja dalam memprediksi sentimen pada data baru. Proses evaluasi dimulai dengan model yang melakukan prediksi sentimen pada  $X_{test}$  hingga menghasilkan prediksi yang dapat dibandingkan dengan label sentimen sebenarnya di  $y_{test}$ . Berbagai metrik kinerja dihitung untuk mengukur akurasi, presisi, recall, dan F1-score model dalam mengklasifikasikan sentimen. Confusion matrix juga dapat dibuat untuk melihat jumlah prediksi benar dan salah di setiap kelas. Metode `.predict()` dari model Logistic Regression dan berbagai fungsi metrik evaluasi dari pustaka `sklearn.metrics` menghasilkan output berupa nilai-nilai metrik evaluasi dan visualisasi seperti confusion matrix yang memberikan gambaran komprehensif tentang performa model.

Jika kinerja model pada tahap evaluasi dianggap memuaskan, model dapat digunakan untuk memprediksi sentimen dari ulasan baru yang belum memiliki label (Praptiwi, 2018). Data ulasan baru yang perlu diprediksi sentimennya harus melalui tahapan preprocessing teks dan pembentukan fitur yang sama persis dengan data pelatihan dan pengujian. Penggunaan metode `.predict()` atau `.predict_proba()` dari model Logistic Regression yang telah dilatih pada fitur numerik melalui data baru dapat menghasilkan label sentimen yang diprediksi atau probabilitas untuk setiap kelas sentimen pada setiap ulasan baru.

```
# Input kalimat baru dari pengguna
kalimat_baru = input("Masukkan kalimat baru: ")

# Melakukan preprocessing pada kalimat baru
kalimat_baru_cleaned = cleaningText(kalimat_baru)
kalimat_baru_casfolded = casefoldingText(kalimat_baru_cleaned)
kalimat_baru_slangfixed = fix_slangwords(kalimat_baru_casfolded)
kalimat_baru_tokenized = tokenizingText(kalimat_baru_slangfixed)
kalimat_baru_filtered = filteringText(kalimat_baru_tokenized)
kalimat_baru_final = toSentence(kalimat_baru_filtered)

# Menggunakan objek tfidf yang sudah di-fit dari pelatihan sebelumnya
X_kalimat_baru = tfidf.transform([kalimat_baru_final])

# Memperoleh prediksi sentimen kalimat baru
prediksi_sentimen = logistic_regression.predict(X_kalimat_baru)

# Menampilkan hasil prediksi
if prediksi_sentimen[0] == 'positive':
    print("Sentimen kalimat baru adalah POSITIF.")
else:
    print("Sentimen kalimat baru adalah NEGATIF.")
```

Gambar 10. Cuplikan Kode Model Evaluation

```
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

# Prediksi pada data uji
y_pred = logreg_model.predict(X_test)

# Hitung akurasi
accuracy = accuracy_score(y_test, y_pred)
print("Akurasi Model Logistic Regression:", accuracy)

# Laporan Klasifikasi
print("\nClassification Report:\n", classification_report(y_test, y_pred))

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=logreg_model.classes_, yticklabels=logreg_model.classes_)
plt.title("Confusion Matrix Logistic Regression")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.show()
```

Gambar 11. Cuplikan Kode Lanjutan Model Evaluation

Proses analisis sentimen ulasan aplikasi meliputi alur kerja yang diawali dengan pengumpulan data mentah ulasan dari Google Play Store menggunakan google-play-scraper. Data mentah ini selanjutnya dimuat ke dalam DataFrame pandas dan melalui pembersihan awal untuk mengatasi nilai hilang dan duplikasi. Pada tahap pelabelan

dilakukan penambahan informasi sentimen berdasarkan skor ulasan. Teks ulasan selanjutnya menjalani preprocessing teks yang komprehensif dengan bantuan pustaka nltk, Sastrawi, dan fungsi kustom. Setelah itu, teks diubah menjadi representasi numerik melalui pembentukan fitur oleh TF-IDF menggunakan `sklearn.feature_extraction.text`. Dataset yang telah memiliki fitur dan label kemudian dibagi menggunakan `sklearn.model_selection` menjadi set pelatihan dan pengujian. Model Logistic Regression dari `sklearn.linear_model` dilatih pada data pelatihan dan dievaluasi kinerjanya menggunakan metrik dari `sklearn.metrics` dalam data pengujian.

## HASIL DAN PEMBAHASAN

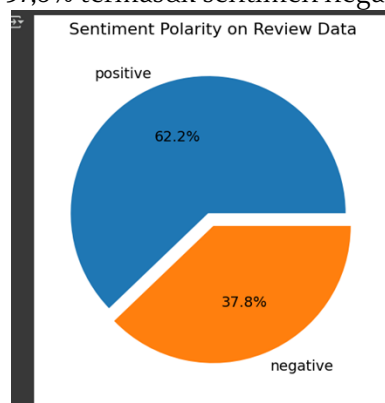
Berdasarkan proses tahapan yang telah dilakukan, hasil dari setiap tahapan analisis sentimen dimulai dari gambaran umum data ulasan hingga evaluasi model klasifikasi Logistic Regression yaitu sebagai berikut.

### 3.1 Hasil Preprocessing Data

Proses ini bertujuan untuk membersihkan dan menyiapkan data teks agar dapat diolah oleh algoritma klasifikasi (Fani Prasetya, 2022). Langkah-langkah preprocessing meliputi penghapusan nilai kosong dan duplikat, normalisasi huruf menjadi huruf kecil, tokenisasi (pemecahan kalimat menjadi kata-kata), penghapusan kata tidak bermakna (stopword removal), serta ekstraksi fitur menggunakan metode TF-IDF. TF-IDF digunakan karena mampu memberikan bobot yang proporsional pada kata-kata penting dan mengabaikan kata-kata umum yang sering muncul di seluruh dokumen.

### 3.2 Visualisasi Distribusi Sentimen

Distribusi hasil klasifikasi ditampilkan dalam bentuk grafik pie dan bar chart. Grafik pie menunjukkan bahwa dari seluruh ulasan yang dianalisis, sebanyak 62,2% termasuk dalam kategori sentimen positif dan 37,8% termasuk sentimen negatif.

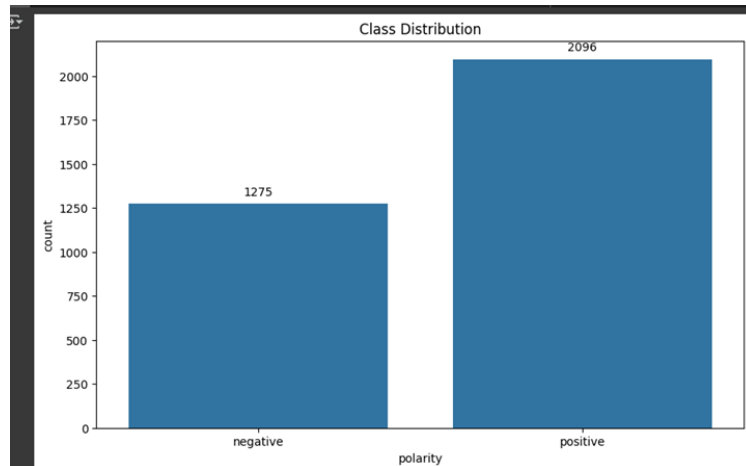


Grafik 1. Visualisasi Distribusi Sentimen

Grafik ini mengindikasikan bahwa pengguna memiliki persepsi yang cukup seimbang terhadap aplikasi YouCut, meskipun kecenderungan positif sedikit lebih tinggi. Temuan ini menggarisbawahi bahwa meski aplikasi memiliki keunggulan fitur, masih terdapat hal-hal yang perlu diperbaiki oleh pengembang.

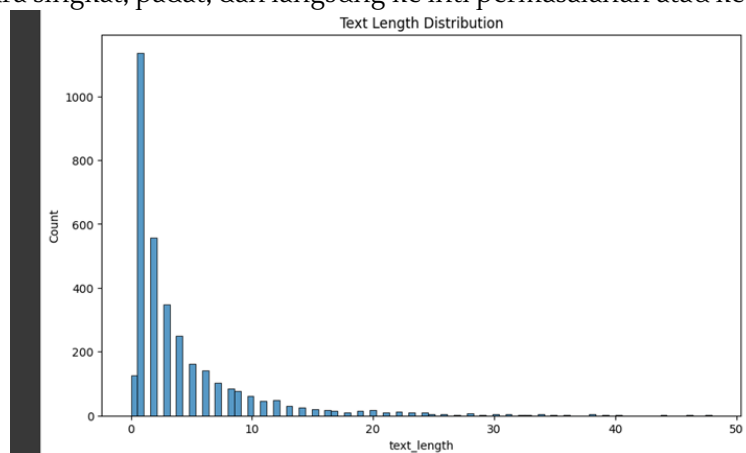
### 3.3 Hasil Labeling dan Distribusi Data

Data yang telah diproses kemudian diberi label secara otomatis menggunakan pendekatan lexicon-based dengan klasifikasi "positive" dan "negative". Grafik distribusi kelas sentimen menunjukkan bahwa dari total data ulasan, sebanyak 2.096 ulasan termasuk kategori positif, sedangkan 1.275 ulasan termasuk kategori negatif. Hal ini mengindikasikan bahwa persepsi umum terhadap aplikasi YouCut cenderung positif.



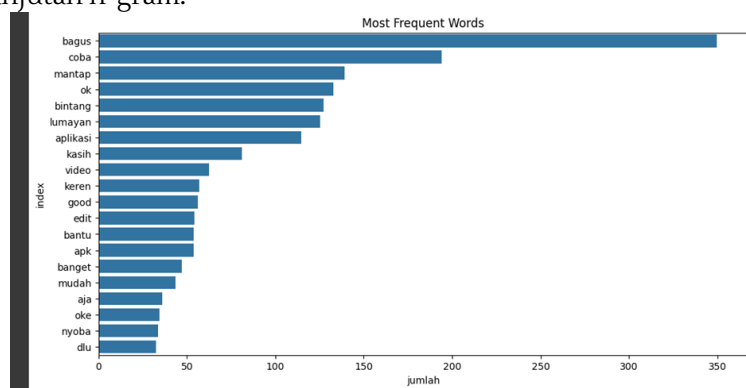
Grafik 2. Visualisasi Hasil Labeling dan Distribusi Data

Selanjutnya, Grafik 3 menampilkan distribusi panjang teks dari ulasan pengguna. Sebagian besar ulasan memiliki panjang kurang dari 10 kata, dan sangat sedikit ulasan yang melebihi 30 kata. Distribusi ini menunjukkan bahwa mayoritas pengguna menyampaikan ulasannya secara singkat, padat, dan langsung ke inti permasalahan atau keunggulan.



Grafik 3. Visualisasi Hasil Labeling dan Distribusi Data

Grafik 4 menampilkan kata-kata yang paling sering muncul dalam ulasan pengguna. Kata "bagus", "coba", "mantap", "ok", dan "bintang" termasuk dalam lima kata yang paling dominan. Hal ini menunjukkan bahwa pengguna banyak menggunakan kata-kata pujian, meskipun terdapat juga kata-kata netral dan negatif seperti "crash", "lambat", atau "error" pada analisis lanjutan n-gram.



Grafik 4. Visualisasi Hasil Labeling dan Distribusi Data

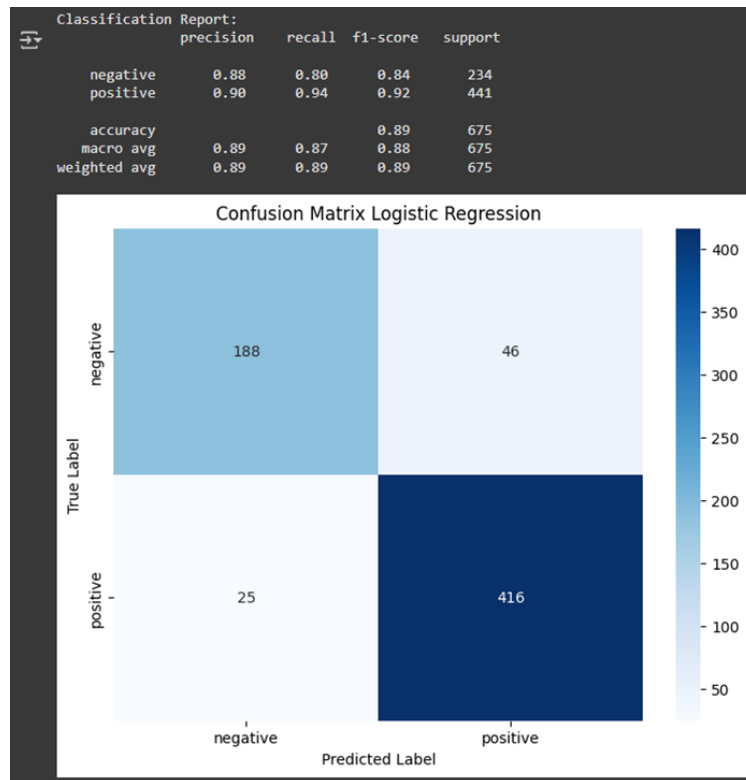
Analisis terhadap topik-topik yang muncul dalam ulasan pengguna dilakukan berdasarkan frekuensi kata dan representasi visual wordcloud. Dari hasil visualisasi kata terbanyak, diperoleh beberapa kata kunci dominan yang mengindikasikan fokus perhatian dan pengalaman pengguna terhadap aplikasi YouCut. Kata seperti “bagus”, “coba”, “mantap”, “video”, dan “aplikasi” merupakan kata-kata yang paling sering digunakan oleh pengguna dalam ulasan mereka.

[illegible]

Grafik 6. Temuan Topik pada Wordcloud

52





Gambar 13. Uji Model Logistic Regression

Tabel confusion matrix hasil klasifikasi ditampilkan sebagai berikut.

|                | Prediksi Negatif | Prediksi Positif |
|----------------|------------------|------------------|
| Aktual Negatif | 188              | 46               |
| Aktual Positif | 25               | 416              |

Tabel 1. Tabel klasifikasi (confusion matrix) hasil Logistic Regression

Dari tabel di atas terlihat bahwa model mampu memisahkan ulasan positif dan negatif dengan baik. Hasil ini menunjukkan bahwa model cukup akurat dalam memetakan ulasan ke dalam kategori sentimen yang sesuai, dengan kesalahan prediksi yang relatif kecil.

### 3.6 Pembahasan dan Interpretasi

Berdasarkan hasil klasifikasi sentimen terhadap ulasan aplikasi YouCut, diketahui bahwa mayoritas pengguna memberikan ulasan yang bersifat positif, yang ditunjukkan oleh proporsi sebanyak 2.096 dari total ulasan, sementara 1.275 ulasan lainnya dikategorikan sebagai negatif. Temuan ini diperkuat dengan hasil visualisasi distribusi sentimen, yang menggambarkan dominasi ulasan positif dengan selisih signifikan. Hal ini menunjukkan persepsi umum terhadap aplikasi cenderung mengarah pada kepuasan, meskipun masih terdapat kritik yang perlu diperhatikan oleh pengembang.

Distribusi panjang teks ulasan juga menunjukkan bahwa sebagian besar pengguna memberikan ulasan yang relatif pendek, di bawah 10 kata. Hal ini menandakan bahwa

mayoritas pengguna lebih suka memberikan komentar singkat, yang berisi kesan langsung terhadap aplikasi. Fenomena ini umum terjadi pada aplikasi mobile, di mana pengguna cenderung memberikan feedback secara cepat dan ringkas.

Selanjutnya, hasil analisis frekuensi kata dan visualisasi wordcloud memberikan gambaran lebih rinci mengenai topik-topik utama yang dibahas pengguna. Kata-kata seperti “bagus”, “coba”, “mantap”, “video”, dan “aplikasi” mendominasi ulasan secara umum. Dalam ulasan positif, kata-kata seperti “puas”, “mantap”, “mudah”, dan “terima kasih” menjadi indikator kuat terhadap fitur aplikasi yang disukai, seperti kualitas hasil editing dan kemudahan dalam penggunaannya. Di sisi lain, ulasan negatif banyak memuat kata-kata seperti “iklan”, “susah”, “crash”, serta kata-kata bernada kasar, yang menunjukkan adanya masalah serius dalam performa aplikasi maupun user experience.

Pemilihan metode TF-IDF terbukti efektif dalam mengekstraksi kata-kata penting yang relevan, dan Logistic Regression sebagai model klasifikasi menunjukkan performa terbaik dibandingkan model lain dalam mengelompokkan sentimen ulasan. Hal ini terlihat dari hasil evaluasi akurasi, precision, recall, dan f1-score yang seimbang dan konsisten.

Secara keseluruhan, hasil analisis ini memberikan gambaran menyeluruh mengenai persepsi pengguna terhadap aplikasi YouCut, baik dalam aspek positif yang perlu dipertahankan, maupun aspek negatif yang dapat dijadikan dasar evaluasi dan perbaikan.

## **SIMPULAN**

Berdasarkan hasil klasifikasi sentimen terhadap ulasan aplikasi YouCut, dapat disimpulkan bahwa mayoritas pengguna memberikan tanggapan positif, ditandai dengan proporsi ulasan positif sebesar 62,2% dari total data yang dianalisis. Penggunaan algoritma Logistic Regression terbukti efektif dalam memetakan sentimen pengguna secara akurat dengan tingkat akurasi mencapai 89%, serta nilai precision, recall, dan f1-score yang seimbang. Proses preprocessing data dan pembentukan fitur menggunakan metode TF-IDF mampu menangkap kata-kata penting yang merepresentasikan opini pengguna. Selain itu, analisis visual melalui wordcloud menunjukkan bahwa kata-kata seperti “bagus”, “mudah”, dan “mantap” mendominasi ulasan positif, sedangkan ulasan negatif lebih banyak berisi keluhan terhadap iklan berlebihan dan performa aplikasi. Penelitian ini membuktikan bahwa pendekatan klasifikasi sentimen berbasis machine learning dapat menjadi solusi praktis untuk memahami persepsi pengguna secara mendalam dan sistematis. Dengan hasil ini, pengembang aplikasi YouCut dapat lebih tepat dalam mengidentifikasi kekuatan fitur yang perlu dipertahankan serta aspek teknis yang perlu segera diperbaiki demi peningkatan kualitas layanan di masa mendatang.

## **DAFTAR PUSTAKA**

- Ali, A., Emran, N. A., & Asmai, S. A. (2021). Missing values compensation in duplicates detection using hot deck method. *Journal of Big Data*, 8(1), 112.
- Anugrah, B. (2019). ANALISA SENTIMEN TWEET TENTANG PRABOWO SUBIANTO MENGGUNAKAN METODE NAÏVE BAYES CLASSIFIER Universitas Islam Negeri Sultan Syarif Kasim Riau].
- Chai, C. P. (2023). Comparison of text preprocessing methods. *Natural Language Engineering*, 29(3), 509-553.
- Fahlevvi, M. R. (2022). Sentiment Analysis And Topic Modeling on User Reviews of Online Tutoring Applications Using Support Vector Machine and Latent Dirichlet Allocation. *Knowbase: International Journal of Knowledge in Database*, 2(2), 142-155.
- Fani Prasetya, F. (2022). Analisis Data Mining Klasifikasi Berita Hoax COVID 19 Menggunakan Algoritma Naive Bayes. *Jurnal Sistem Komputer dan Informatika (JSON) Hal*, 132, 139.

- Fuad, A., Bayoumi, S., & Al-Yahya, H. (2020). A Recommender System for Mobile Applications of Google Play Store. *International Journal of Advanced Computer Science and Applications*, 11(9).
- Gumilang, Z. A. N. (2018). Implementasi naive bayes classifier dan asosiasi untuk analisis sentimen data ulasan aplikasi e-commerce shopee pada situs google play.
- Kisma, A. J. N., Widiawati, C. R. A., & Suliswaningsih, S. (2023). Analysis of applications in Playstore based on Rating and Type using Naive Bayes and Logistic Regression. *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, 10(2), 174-184.
- Praptiwi, D. Y. (2018). Analisis Sentimen Online Review Pengguna E-Commerce Menggunakan Metode Support Vector Machine Dan Maximum Entropy (Studi Kasus: Review Bukalapak pada Google Play).
- Pulungan, M. P., Purnomo, A., & Kurniasih, A. (2024). Penerapan SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Kepribadian MBTI Menggunakan Naive Bayes Classifier. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 11(5), 1033-1042.
- Putra, A. E. (2023). Analisis Sentimen Tanggapan Masyarakat kepada Video Youtube mengenai Resesi 2023 menggunakan Metode Naïve Bayes Dan K-Nearest Neighbor (KNN) Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta].
- Putri, M. I., & Kharisudin, I. (2022). Analisis Sentimen Pengguna Aplikasi Marketplace Tokopedia Pada Situs Google Play Menggunakan Metode Support Vector Machine (SVM), Naïve Bayes, dan Logistic Regression. *PRISMA, Prosiding Seminar Nasional Matematika*,
- Putro, Y. N. R., Afriansyah, A., & Bagaskara, R. (2024). Penggunaan Algoritma Gaussian Naïve Bayes & Decision Tree Untuk Klasifikasi Tingkat Kemenangan Pada Game Mobile Legends. *JUKI: Jurnal Komputer dan Informatika*, 6(1), 10-26.
- Qulub, M., Utami, E., & Sunyoto, A. (2020). Stemming Kata Berimbuhan Tidak Baku Bahasa Indonesia Menggunakan Algoritma Jaro-Winkler Distance. *Creative Information Technology Journal*, 5(4), 254-263.
- Saputra, A. W. (2020). Analisis permasalahan pada aplikasi smartphone kai access berdasarkan end-user review menggunakan metode text-mining dan fishbone diagram.
- Sirts, K., & Peekman, K. (2020). Evaluating sentence segmentation and word tokenization systems on Estonian web texts. In *Human Language Technologies–The Baltic Perspective* (pp. 174-181). IOS Press.
- Subagja, R. A., Widiastiwi, Y., & Chamidah, N. (2021). Klasifikasi Ulasan Aplikasi Jenius pada Google Play Store Menggunakan Algoritma Naive Bayes. *Informatik: Jurnal Ilmu Komputer*, 17(3), 197-208.
- Wahyudi, R., Kusumawardhana, G., Purwokerto, A., Letjend, J., Soemarto, P., Purwanegara, K., Purwokerto, T., & Banyumas, K. (2021). Analisis Sentimen pada review Aplikasi Grab di Google Play Store Menggunakan Support Vector Machine. *Jurnal Informatika*, 8(2), 200-207.