

ANALISIS SENTIMEN ULASAN APLIKASI ALFAGIFT PADA PLAYSTORE MENGUNAKAN METODE SUPPORT VECTOR MACHINE

Raden Raihan Nur Barkah Ramadhan¹, Elah Nurlelah²

Fakultas Teknik Dan Informatika Universitas Bina Sarana Informatika Jakarta ^{1,2}

Email: radenraihan16@gmail.com

Informasi	Abstract
Volume : 3	<i>As smartphone usage continues to increase in Indonesia, understanding user perceptions through application reviews becomes essential to capture real user experiences. This study aims to analyze user sentiment toward the Alfagift application available on the Google Play Store and classify the sentiment using the Support Vector Machine (SVM) method. A total of 1,887 user reviews were collected between November 2025 and January 2026 through web scraping techniques. The research stages included data collection, followed by preprocessing steps such as data cleaning, case folding, word normalization, tokenization, stopword removal, and stemming. Term weighting was performed using the TF-IDF (Term Frequency-Inverse Document Frequency) method, and sentiment was classified into two categories: positive (ratings above 3) and negative (ratings equal to or below 3). The results showed that 64.10% of reviews were positive and 35.90% were negative. A linear kernel SVM model was applied using an 80:20 training-testing split, achieving an accuracy of 84.92%, precision of 83.84%, recall of 82.02%, and F1-score of 82.79%. The findings indicate that SVM is effective for sentiment classification and provides valuable insights for application improvement.</i>
Nomor : 2	
Bulan : Februari	
Tahun : 2026	
E-ISSN : 3062-9624	
Keyword: Sentiment Analysis, Support Vector Machine (SVM), Alfagift, App Reviews, Text Mining, TF-IDF, Play Store	

Abstrak

Seiring meningkatnya penggunaan smartphone di Indonesia, memahami perasaan pengguna melalui ulasan aplikasi menjadi hal penting untuk mengetahui pengalaman mereka secara nyata. Penelitian ini bertujuan menganalisis sentimen pengguna aplikasi Alfagift yang tersedia di Play Store dan mengelompokkan sentimen tersebut ke dalam kategori tertentu menggunakan metode Support Vector Machine (SVM). Data yang digunakan berjumlah 1.887 ulasan pengguna yang dikumpulkan pada periode November 2025 hingga Januari 2026 melalui teknik web scraping. Tahapan penelitian dimulai dari pengumpulan data, kemudian dilanjutkan dengan proses pra-pemrosesan yang meliputi pembersihan data, pengubahan huruf menjadi kecil (case folding), normalisasi kata, tokenisasi, penghapusan stopwords, serta stemming. Selanjutnya dilakukan pembobotan kata menggunakan metode TF-IDF (Term Frequency-Inverse Document Frequency) dan pengelompokan sentimen ke dalam dua kelas, yaitu positif (rating lebih dari 3) dan negatif (rating kurang dari atau sama dengan 3). Dari data yang telah diproses, diperoleh 64,10% ulasan bersentimen positif dan 35,90% bersentimen negatif. Model SVM dengan kernel linear diterapkan pada data yang dibagi menjadi 80% data latih dan 20% data uji, dengan hasil evaluasi menunjukkan kinerja yang sangat baik, yaitu akurasi 84,92%, presisi 83,84%, recall 82,02%, dan F1-Score 82,79%. Hasil penelitian menunjukkan bahwa sentimen positif umumnya berkaitan dengan kemudahan penggunaan fitur dan manfaat aplikasi, sedangkan sentimen negatif lebih banyak dipengaruhi oleh kendala teknis seperti masalah login dan proses verifikasi akun, sehingga metode SVM terbukti efektif dalam mengklasifikasikan sentimen ulasan pengguna sekaligus memberikan masukan yang bermanfaat bagi pengembang

aplikasi dalam meningkatkan kualitas layanan.

Kata Kunci: Analisis Sentimen, Support Vector Machine (SVM), Alfagift, Ulasan Aplikasi, Text Mining, TF-IDF, Play Store.

A. PENDAHULUAN

Teknologi informasi (TI) kini sudah menjadi bagian penting dalam kehidupan kita. Setiap hari, kita menggunakan berbagai perangkat digital, aplikasi, dan layanan daring karena TI membantu kita mengakses informasi, berkomunikasi, bekerja, serta memperoleh layanan dengan lebih cepat dan efisien. Dalam perkembangan teknologi yang semakin pesat, berbagai teknologi seperti machine learning, big data, dan pemrosesan bahasa alami semakin banyak digunakan untuk memahami bagaimana masyarakat merespons layanan digital, tidak hanya dari angka penilaian, tetapi juga dari kata-kata yang mereka gunakan.

Pada Zaman digital saat ini, pemakaian aplikasi mobile telah menjadi bagian dari cara hidup masyarakat. Hampir semua kebutuhan bisa dipenuhi melalui aplikasi, mulai dari transportasi, komunikasi, hingga pembelian kebutuhan sehari-hari. Salah satu perusahaan ritel terkemuka di Indonesia, Alfamart, turut serta dalam tren ini dengan meluncurkan aplikasi Alfagift. Aplikasi ini mempermudah konsumen untuk berbelanja, mendapatkan penawaran, dan mengakses layanan loyalitas secara online.

Semakin banyaknya pengguna, ulasan yang ditulis di Play Store juga semakin meningkat. Ulasan-ulasan ini sangat krusial karena dapat mencerminkan tingkat kepuasan pengguna terhadap aplikasi itu. Menteri Komunikasi dan Informatika (Menkominfo) Johnny G Plate mengatakan bahwa jumlah pemilik smartphone atau ponsel pintar di Indonesia mencapai 345,3 juta, lebih banyak dari total penduduk yang mencapai 271,35 juta jiwa (Nabawi, 2023). Hal ini menandakan bahwa platform digital, seperti Playstore, memiliki peran krusial dalam hubungan antara pengguna dan penyedia layanan.

Analisis sentimen adalah teknik yang mengidentifikasi bagaimana perasaan diekspresikan dalam teks dan dikategorikan sebagai sentimen positif atau negatif (Enjeli & Wijaya, 2024). Analisis sentimen memfokuskan pada identifikasi nada emosional dalam data teks, serta bagaimana frasa, kata, atau ekspresi digunakan untuk menyampaikan kepuasan atau ketidakpuasan. Metode Support Vector Machine (SVM) menunjukkan nilai akurasi sebesar 87.95 %. Dari 1080 data uji, dan Metode Naive bayes menunjukkan nilai akurasi sebesar 81.65%. Dari 1081 data uji (Sarimole, 2024).

Beberapa penelitian terdahulu digunakan sebagai perbandingan metodologi dalam analisis sentimen. Penelitian Buana dan Arianto (2024) menunjukkan bahwa algoritma Support Vector Machine (SVM) mampu memisahkan sentimen positif dan negatif pada ulasan aplikasi ZenPro dengan akurasi sekitar 90%. Sementara itu, penelitian Rhomaningtias, Khairunisa, May Wara, dan Hindrayani (2024) membandingkan metode Naïve Bayes dan SVM pada ulasan aplikasi SMILE Indonesia, dengan hasil bahwa Naïve Bayes sedikit lebih unggul berdasarkan metrik akurasi dan recall. Secara umum, SVM merupakan metode berbasis teori pembelajaran statistik yang mampu menentukan hyperplane optimal untuk memisahkan data ke dalam beberapa kelas sentimen. Keunggulan SVM terletak pada kemampuannya mengolah data berdimensi tinggi, sehingga efektif digunakan dalam analisis teks ulasan aplikasi seperti Alfagift.

Berdasarkan latar belakang tersebut, terdapat beberapa permasalahan yang dapat diidentifikasi, yaitu ulasan pengguna di Play Store belum dimanfaatkan secara maksimal untuk memahami kepuasan maupun keluhan terhadap aplikasi Alfagift. Selain itu, gaya bahasa pengguna yang beragam, termasuk penggunaan bahasa santai dan istilah sehari-hari, membuat proses analisis sentimen otomatis menjadi lebih kompleks. Data ulasan yang terus meningkat juga menghasilkan teks yang bervariasi, sehingga diperlukan metode yang mampu menganalisis opini secara akurat, bukan hanya berdasarkan penilaian angka. Oleh karena itu, perumusan masalah penelitian ini berfokus pada efektivitas metode SVM dengan pembobotan TF-IDF dalam menghasilkan model klasifikasi sentimen yang akurat, kemampuan SVM dalam mengklasifikasikan perasaan pengguna, serta pemanfaatan hasil analisis sebagai bahan pertimbangan dalam pengembangan layanan aplikasi.

Tujuan penelitian ini adalah untuk mengenali dan mengelompokkan kepuasan pengguna aplikasi Alfagift ke dalam kategori sentimen positif atau negatif, sehingga dapat memberikan gambaran umum mengenai persepsi pengguna. Selain itu, penelitian ini bertujuan menguji penggunaan metode SVM dalam proses klasifikasi teks ulasan berbahasa Indonesia untuk mengetahui tingkat efektivitasnya. Manfaat yang diharapkan dari penelitian ini adalah memberikan wawasan dan rekomendasi yang dapat dimanfaatkan oleh pengembang aplikasi sebagai dasar pengambilan keputusan dalam meningkatkan kualitas layanan dan pengalaman pengguna.

B. METODE PENELITIAN

Metodologi penelitian ini dilakukan melalui beberapa tahapan yang terstruktur agar

tujuan analisis sentimen dapat tercapai secara sistematis. Proses dimulai dari pengumpulan data ulasan pengguna aplikasi Alfagift di Play Store selama periode November 2025 hingga Januari 2026 melalui teknik web scraping yang menghasilkan data berupa teks komentar dan rating bintang. Setelah data diperoleh, tahap preprocessing dilakukan untuk membersihkan dan menyiapkan teks agar siap dianalisis, meliputi cleaning, case-folding, normalisasi, tokenisasi, stopword removal, dan stemming. Selanjutnya, data diberi label sentimen berdasarkan rating pengguna, yaitu kategori negatif untuk rating ≤ 3 dan positif untuk rating > 3 . Dataset kemudian dibagi menjadi data pelatihan sebesar 80% dan data pengujian sebesar 20% guna memastikan model mampu belajar sekaligus diuji pada data baru yang belum pernah digunakan sebelumnya.

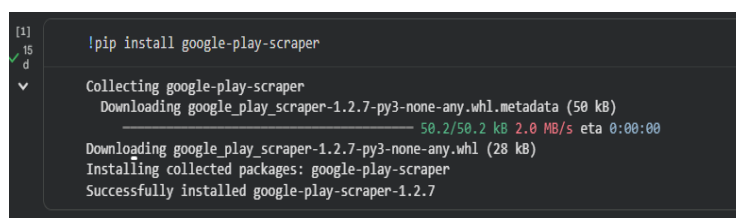
Tahap berikutnya adalah pembobotan kata menggunakan metode TF-IDF untuk mengubah teks menjadi representasi numerik sehingga dapat diproses oleh algoritma Support Vector Machine (SVM). Model SVM dengan kernel linier diterapkan untuk mengklasifikasikan sentimen ulasan menjadi positif atau negatif berdasarkan pola yang dipelajari dari data pelatihan. Setelah proses pelatihan selesai, model dievaluasi menggunakan metrik akurasi, presisi, recall, dan F1-Score, serta confusion matrix dan validasi silang 5-fold guna memperoleh hasil yang stabil dan andal. Instrumen penelitian meliputi Play Store API atau web scraping tools untuk pengambilan data, Google Colab sebagai lingkungan pengolahan, serta library Python seperti NLTK, Pandas, NumPy, dan Scikit-learn untuk preprocessing, pemodelan, dan evaluasi kinerja model.

C. HASIL DAN PEMBAHASAN

Hasil Penelitian

Pengumpulan Data

Install Google Play Scraper



```
[1] !pip install google-play-scraper
15
d
Collecting google-play-scraper
  Downloading google_play_scraper-1.2.7-py3-none-any.whl.metadata (50 kB)
    50.2/50.2 kB 2.0 MB/s eta 0:00:00
  Downloading google_play_scraper-1.2.7-py3-none-any.whl (28 kB)
Installing collected packages: google-play-scraper
Successfully installed google-play-scraper-1.2.7
```

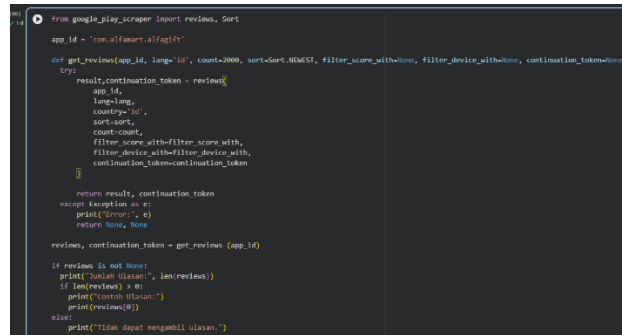
Gambar 1 Google Play Scraper

Sumber : Dokumentasi Pribadi

Library yang digunakan dalam penelitian ini antara lain `google_play_scraper`, yang berperan untuk mengambil informasi aplikasi dan ulasan pengguna dari Google Play Store,

pandas yang digunakan untuk memproses dan mengelola data dalam bentuk DataFrame, serta datetime yang digunakan untuk menentukan rentang waktu pengambilan data, yaitu start_date dan end_date, sehingga proses pengumpulan dan pengolahan data dapat berjalan secara terstruktur dan efisien.

Scrapping data menggunakan Google Play Scrape



```

from google_play_scraper import reviews, Sort

app_id = 'com.alfamart.alfagift'

def get_reviews(app_id, lang='id', count=2000, sort=Sort.NEWEST, filter_score=None, filter_device=None, continuation_token=None):
    try:
        result, continuation_token = reviews(
            app_id,
            lang=lang,
            country='id',
            sort=sort,
            count=count,
            filter_score=filter_score,
            filter_device=filter_device,
            continuation_token=continuation_token
        )
        return result, continuation_token
    except Exception as e:
        print(f"Error: {e}")
        return None, None

reviews, continuation_token = get_reviews(app_id)

if reviews is not None:
    print(f"Jumlah ulasan: {len(reviews)}")
    if len(reviews) > 0:
        print("Contoh ulasan:")
        print(reviews[0])
    else:
        print("Tidak dapat mengambil ulasan.")

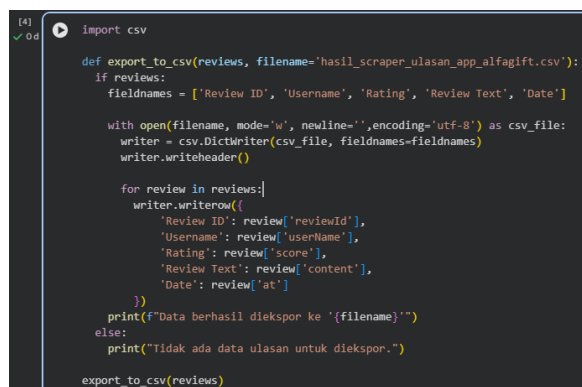
```

Gambar 2 Proses Scrapping Data

Sumber : Dokumentasi Pribadi

Pengambilan data dilakukan dengan terlebih dahulu mengimpor fungsi reviews dan konstanta Sort dari library google play scraper. Aplikasi yang dianalisis ditentukan melalui app id sebagai identitas unik di Google Play Store, yaitu “com.alfamart.alfagift”, yang menjadi sumber ulasan dalam penelitian ini. Parameter lang diatur ke Bahasa Indonesia agar ulasan yang diperoleh sesuai dengan konteks pengguna lokal. Jumlah ulasan yang diambil dibatasi hingga maksimal 2.000 data melalui parameter count, sementara pengurutan data dilakukan berdasarkan ulasan terbaru menggunakan Sort.NEWEST. Seluruh ulasan dikumpulkan tanpa penyaringan skor dengan mengatur filter score with ke None, meskipun opsi ini tetap tersedia jika diperlukan untuk mengambil ulasan dengan penilaian tertentu, seperti bintang satu. Apabila jumlah ulasan melebihi batas pengambilan dalam satu proses, continuation token digunakan untuk melanjutkan pengambilan data secara bertahap.

Ekspor hasil scrapping data ke file CSV



```

import csv

def export_to_csv(reviews, filename='hasil_scraper_ulasan_app_alfagift.csv'):
    if reviews:
        fieldnames = ['Review ID', 'Username', 'Rating', 'Review Text', 'Date']

        with open(filename, mode='w', newline='', encoding='utf-8') as csv_file:
            writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
            writer.writeheader()

            for review in reviews:
                writer.writerow({
                    'Review ID': review['reviewId'],
                    'Username': review['userName'],
                    'Rating': review['score'],
                    'Review Text': review['content'],
                    'Date': review['at']
                })
            print(f"Data berhasil diekspor ke '{filename}'")
    else:
        print("Tidak ada data ulasan untuk diekspor.")

export_to_csv(reviews)

```

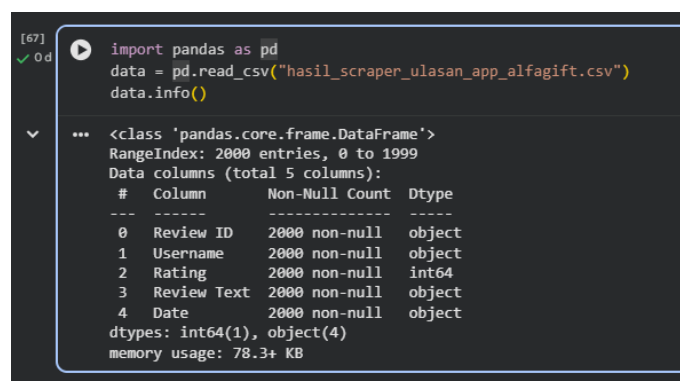
Gambar 3 Ekspor Hasil Scrapping Data

Setelah proses scraping selesai, dibuat fungsi export to csv dengan dua parameter, yaitu reviews sebagai daftar ulasan hasil scraping dan file name sebagai nama file keluaran, hasil scraper ulasan app alfragift.csv. Data yang dikumpulkan mencakup beberapa informasi penting, antara lain review id sebagai identitas unik setiap ulasan, username sebagai nama pengguna yang memberikan ulasan, content yang berisi teks ulasan dan menjadi fokus utama analisis sentimen, score yang menunjukkan rating pengguna dari satu hingga lima bintang, serta at yang mencatat tanggal dan waktu ulasan diberikan. Informasi ini memungkinkan penelitian untuk menganalisis kepuasan pengguna serta tren sentimen seiring waktu dengan lebih terstruktur.

Preprocessing

Upload data ke Google Collabs

Langkah pertama yaitu harus mengupload dataset ke dalam Google Collabs



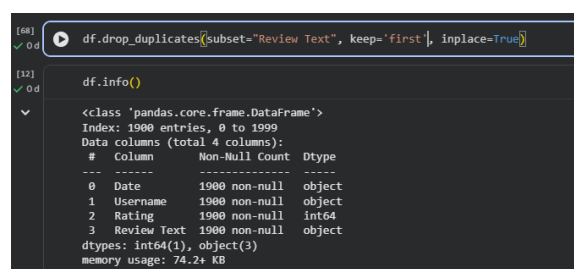
```
[67]
✓ 0 d import pandas as pd
data = pd.read_csv("hasil_scraper_ulasan_app_alfagift.csv")
data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Review ID    2000 non-null    object
1   Username     2000 non-null    object
2   Rating       2000 non-null    int64
3   Review Text  2000 non-null    object
4   Date         2000 non-null    object
dtypes: int64(1), object(4)
memory usage: 78.3+ KB
```

Gambar 4 Upload Dataset

Sumber : Dokumentasi Pribadi

Menghapus Duplikat



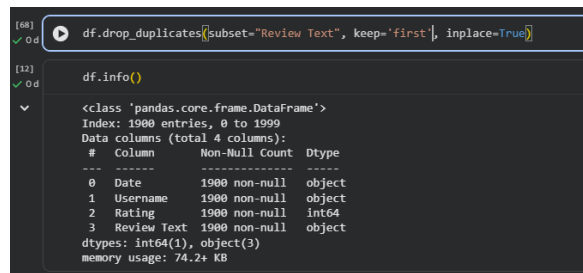
```
[68]
✓ 0 d df.drop_duplicates(subset="Review Text", keep='first', inplace=True)

[12]
✓ 0 d df.info()

<class 'pandas.core.frame.DataFrame'>
Index: 1900 entries, 0 to 1999
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Date         1900 non-null    object
1   Username     1900 non-null    object
2   Rating       1900 non-null    int64
3   Review Text  1900 non-null    object
dtypes: int64(1), object(3)
memory usage: 74.2+ KB
```

Gambar 5 Menghapus Duplikat

Menampilkan Visualisasi Wordcloud sebelum Preprocessing



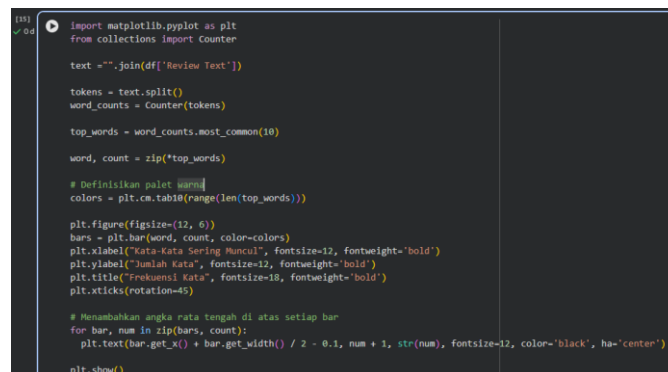
Gambar 6 Menampilkan Wordcloud sebelum Preprocessing

Sebelum melakukan preprocessing, terlebih dahulu dibuat visualisasi wordcloud dari seluruh ulasan. Dengan visualisasi ini, dapat dilihat kata-kata yang paling sering muncul, seperti "aku", "dan", "bagus", "saya", "yang", "aplikasi", "alfagift", "tidak", "bisa", "gak", dan "banget". Hal ini menunjukkan bahwa stopwords bawaan perlu diperluas dengan kata-kata umum dalam bahasa Indonesia agar hasil analisis menjadi lebih akurat.



Gambar 7 Wordcloud sebelum Preprocessing

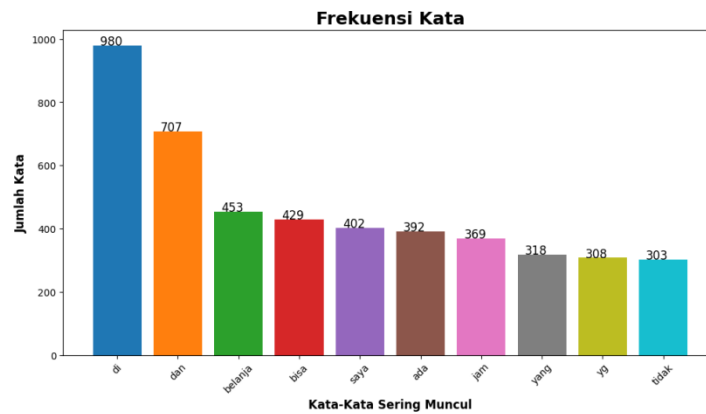
Menampilkan visualisasi diagram batang frekuensi kata



Gambar 8 Menampilkan Frekuensi Kata

Sumber : Dokumentasi Pribadi

Visualisasi ini bertujuan untuk menampilkan kata-kata yang paling sering muncul dalam ulasan. Grafik tersebut memperlihatkan kata-kata yang umum digunakan oleh pengguna saat menulis komentar, sekaligus membantu dalam memilih fitur yang tepat untuk analisis. Informasi dari visualisasi ini nantinya dapat digunakan untuk membuat representasi vektor menggunakan metode TF-IDF.



Gambar 1 Visualisasi Frekuensi Kata

Sumber : Dokumentasi Pribadi

Proses Cleaning

```

2  import re
3  import string
4  import nltk
5
6  # Fungsi untuk menghapus URL
7  def remove_URL(tweet):
8  if tweet is not None and isinstance(tweet, str):
9  url = re.compile(r'https?://\S+|www\.\S+')
10 return url.sub(r'', tweet)
11 else:
12 return tweet
13
14 #Fungsi untuk Menghapus HTML
15 def remove_html(tweet):
16 if tweet is not None and isinstance(tweet, str):
17 html = re.compile(r'<.*?>')
18 return html.sub(r'', tweet)
19 else:
20 return tweet
21
22 # Fungsi untuk menghapus emoji
23 def remove_emoji(tweet):

```

```

24 if tweet is not None and isinstance(tweet, str):
25     emoji_pattern = re.compile("[
26     u"\U0001F600-\U0001F64F" # emoticons
27     u"\U0001F300-\U0001F5FF" # symbol & pictographs
28     u"\U0001F680-\U0001F6FF" # transport & map symbols
29     u"\U0001F700-\U0001F77F" # alchemical symbols
30     u"\U0001F780-\U0001F7FF" # Geometric Shapes Extended
31     u"\U0001F800-\U0001F8FF" # Supplemental Arrows-C
32     u"\U0001F900-\U0001F9FF" # Supplemental Symbols and Pictographs
33     u"\U0001FA00-\U0001FA6F" # Chess Symbols
34     u"\U0001FA70-\U0001FAFF" # Symbols and Pictographs Extended-A
35     u"\U0001F004-\U0001F0CF" # Additional Emoticons
36     u"\U0001F1E0-\U0001F1FF" # Flags
37     "]" + "", flags=re.UNICODE)
38     return emoji_pattern.sub(r'', tweet)
39 else:
40     return tweet
41
42 # Fungsi untuk menghapus simbol
43 def remove_symbols(tweet):
44     if tweet is not None and isinstance(tweet, str):
45         cleaned_tweet = re.sub(r'^A-Za-Z0-9\s', '', tweet)
46         return cleaned_tweet
47     else:
48         return tweet
49
50 # Fungsi untuk menghapus angka
51 def remove_numbers(tweet):
52     if tweet is not None and isinstance(tweet, str):
53         cleaned_tweet = re.sub(r'\d', '', tweet)
54         return cleaned_tweet
55     else:
56         return tweet

```

57

```
58 df['cleaning'] = df['Review Text'].apply(lambda x: remove_URL(x))
```

```
59 df['cleaning'] = df['cleaning'].apply(lambda x: remove_html(x))
```

```
60 df['cleaning'] = df['cleaning'].apply(lambda x: remove_emoji(x))
```

```
61 df['cleaning'] = df['cleaning'].apply(lambda x: remove_symbols(x))
```

```
62 df['cleaning'] = df['cleaning'].apply(lambda x: remove_numbers(x))
```

63

```
64 df.head(11)
```

Pada tahap ini, dilakukan pembersihan data pada kolom konten yang berisi komentar pengguna aplikasi. Langkah ini menjadi tahap awal yang penting dalam pemrosesan teks, terutama untuk analisis sentimen. Dengan menghapus elemen-elemen seperti URL, HTML, emoji, tanda baca, dan angka, data menjadi lebih bersih dan siap dianalisis. Pembersihan ini membantu meningkatkan kualitas data, mengurangi risiko kesalahan saat klasifikasi, serta memudahkan model machine learning, seperti SVM, dalam memahami makna yang terkandung dalam ulasan pengguna.

[illegible]

Gambar 10 Data setelah dilakukan Cleaning

Sumber : Dokumentasi Pribadi

Tahap Casefolding

```
[17]
✓ 0 d
def case_folding(text):
    if isinstance(text, str):
        lowercase_text = text.lower()
        return lowercase_text
    else:
        return text

df['case_folding'] = df['cleaning'].apply(case_folding)
df.head(10)
```

Gambar 11 Casefolding

Sumber : Dokumentasi Pribadi

Tabel 1 Hasil Casefolding

Sebelum	Sesudah
---------	---------

Top, tapi ada juga struk atau receipt gak di kirim ke akun Alfacif saya setelah saya belanja, tapi koinnya saya bertambah, tolong di perbaiki lagi ya, terima kasih.	top tapi ada juga struk atau receipt gak di kirim ke akun alfacif saya setelah saya belanja tapi koinnya saya bertambah tolong di perbaiki lagi ya terima kasih
Masa tinggal di Area DS Cibubur gak bisa tuker ALFA STAMP PUFFY BUDDIES AST - dah kayak tinggal di Hutan	masa tinggal di area ds cibubur gak bisa tuker alfa stamp puffy buddies ast dah kayak tinggal di hutan

Sumber : Dokumentasi Pribadi

Menormalisasi Kata

```

import pandas as pd

# Fungsi penggantian kata tidak baku
def replace_taboo_words(text, kamus_tidak_baku):
    if isinstance(text, str):
        words = text.split()
        replaced_words = []
        kalimat_baku = []
        kata_diganti = []
        kata_tidak_baku_hash = []

        for word in words:
            if word in kamus_tidak_baku:
                baku_word = kamus_tidak_baku[word]
                if isinstance(baku_word, str) and all(char.isalpha() for char in baku_word):
                    replaced_words.append(baku_word)
                    kalimat_baku.append(baku_word)
                    kata_diganti.append(word)
                    kata_tidak_baku_hash.append(word)
                else:
                    # If baku_word is not a valid string, keep the original word
                    replaced_words.append(word)
            else:
                replaced_words.append(word)

        replaced_text = ' '.join(replaced_words)
        return replaced_text, kalimat_baku, kata_diganti, kata_tidak_baku_hash
    else:
        # If text is not a string, return default empty values
        replaced_text = ''
        kalimat_baku = []
        kata_diganti = []
        kata_tidak_baku_hash = []
        return replaced_text, kalimat_baku, kata_diganti, kata_tidak_baku_hash

```

Gambar 12 Melakukan Normalisasi Kata

Sumber : Dokumentasi Pribadi

Setelah fungsi normalisasi dibuat, langkah selanjutnya adalah membaca kamus kata tidak baku yang tersimpan dalam file Excel. Kamus tersebut kemudian diubah ke dalam bentuk dictionary agar dapat digunakan dalam proses normalisasi kata.

```

[73] ✓ 1d # Baca Kamus kata tidak baku
      kamus_data = pd.read_excel("kamuskatabaku.xlsx")
      kamus_tidak_baku = dict(zip(kamus_data['tidak_baku'], kamus_data['kata_baku']))

```

Gambar 13 Membaca Kamus Kata Tidak Baku

Sumber : Dokumentasi Pribadi

Selanjutnya, fungsi penggantian kata tidak baku akan diterapkan pada data teks. Proses ini bertujuan untuk mengganti kata-kata yang tidak baku dengan bentuk kata yang sesuai standar.

```

import pandas as pd

# Terapkan fungsi penggantian kata tidak baku
df[['normalisasi', 'kata_baku', 'kata_tidak_baku', 'kata_tidak_baku_hash']] = df['case_folding'].apply(lambda x: pd.Series(replace_taboo_words(x, kamus_tidak_baku)))
# df.head(100)

df = df[['Data', 'Username', 'Rating', 'Review Text', 'Cleaning', 'case_folding', 'normalisasi']]
df.head(50)

```

Gambar 14 Penggantian Kata Tidak Baku

Sumber : Dokumentasi Pribadi

Tabel 2 Hasil Normalisasi Kata

Sebelum	Sesudah
---------	---------

kecewa banget sama respon pelayanan CS mau d aplikasi mau d dm ig sama2 slowres smpe berhari2 ga ada solusi	kecewa banget sama respon pelayanan cs mau di aplikasi mau di dm ig sama slowres sampai berhari tidak ada solusi
ga jelas anjir gua pesen air minum gua chat tokonya gada konfirmasi tar tiba tiba dateng ga konfirmasi cuman lewat chat di apk gada notif,ga chat di wa tar di chat lagi ga dijawab jawab susah banget emang	tidak jelas anjir gua pesan air minum gua chat tokonya gada konfirmasi entar tiba tiba datang tidak konfirmasi cuman lewat chat di apk gada notifga chat di wa entar di chat lagi tidak dijawab jawab susah banget memang

Sumber : Dokumentasi Pribadi

Tahap Tokenisasi

```
[79]
✓ 0 d def tokenize(text):
      tokens = text.split()
      return tokens

      df['tokenize'] = df['normalisasi'].apply(tokenize)

      df.head(10)
```

Gambar 15 Proses Tokenisasi

Sumber : Dokumentasi Pribadi

Tabel 3 Hasil Tokenisasi

Sebelum	Sesudah
lama pengiriman nya	[lama, pengiriman, ya]
sangat membantu	[sangat, membantu]
barang tidak dikirim	[barang, tidak, dikirim]
sangat jelek pengantaran ditaroh diluar	[sangat, jelek, pengantaran, ditaroh, diluar]

Sumber : Dokumentasi Pribadi

Tahap Stopword Removal

```
[25]
✓ 0 d from nltk.corpus import stopwords
      nltk.download('stopwords')
      stop_words = stopwords.words('indonesian')

      [nltk_data] Downloading package stopwords to /root/nltk_data...
      [nltk_data] Unzipping corpora/stopwords.zip.
```

Gambar 16 Mengambil Data

Sumber : Dokumentasi Pribadi

Setelah daftar stopwords diperoleh, tahap selanjutnya adalah menerapkan proses penghapusan stopwords pada data teks. Pada proses ini, kata-kata yang dianggap tidak penting akan dihilangkan sehingga teks menjadi lebih fokus pada informasi utama.

```
[26]
✓ 0 d def remove_stopwords(text):
      return[word for word in text if word not in stop_words]
      df['stopword removal'] = df['tokenize'].apply(lambda x: remove_stopwords(x))

      df.head(10)
```

Gambar 17 Melakukan Stopword Removal

Sumber : Dokumentasi Pribadi

Tabel 4 Hasil Stopword Removal

Sebelum	Sesudah
[pesanan, saya, kurang, dan, tidak, ada, respon]	[pesanan, respon]
[sangat, membantu]	[membantu]
[sekarang, kalau, pesan, alfagift, lama, banget]	[pesan, alfagift, banget]

Sumber : Dokumentasi Pribadi

Tahap Stemming Data

```
[84]
✓ 6 d !pip install Sastrawi

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from nltk.stem import PorterStemmer
from nltk.stem.snowball import SnowballStemmer
```

Gambar 18 Install Library Sastrawi

Sumber : Dokumentasi Pribadi

Setelah library Sastrawi siap, langkah berikutnya adalah menerapkan stemming pada data. Proses ini akan mengubah kata-kata menjadi bentuk dasarnya agar analisis teks menjadi lebih konsisten dan mudah dipahami

```
factory = StemmerFactory()
stemmer = factory.create_stemmer()

def stem_text(text):
    return [stemmer.stem(word) for word in text]

df['stemming_data'] = df['stopword removal'].apply(lambda x: ' '.join(stem_text(x)))
df.head(10)
```

Gambar 19 Proses Stemming Data

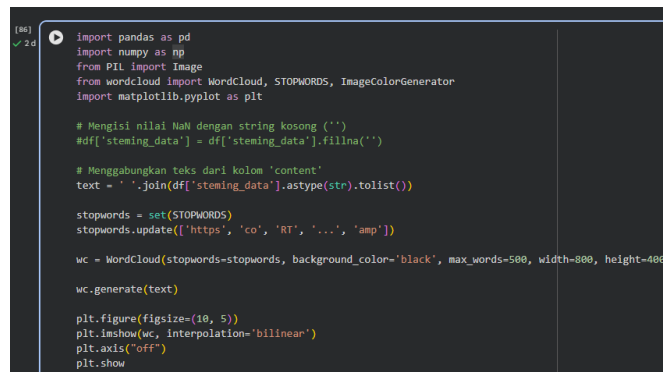
Sumber : Dokumentasi Pribadi

Tabel 5 Hasil Stemming Data

Sebelum	Sesudah
[saya, pesan, susu, bayi, dari, jam, di, anter, sore]	pesan susu bayi jam anter jam sore
[lama, pengiriman, ya]	kirim ya
[tolong, banget, hadirkan, cs, manusia, bukan, bot]	tolong banget hadir cs manusia bot

Sumber : Dokumentasi Pribadi

Menampilkan Wordcloud setelah preprocessing



Gambar 23 Import yang sudah melewati Preprocessing

Sumber : Dokumentasi Pribadi

	Date	Username	Rating	Review Text	cleaning	case_folding	normalisasi	tokenize	stopword removal	stemming_data
0	2026-01-28 03:51:56	Santa Balga	3	mungkin harus ada pembaharuan di aplikasi, say...	mungkin harus ada pembaharuan di aplikasi saya...	mungkin harus ada pembaharuan di aplikasi saya...	mungkin harus ada pembaharuan di aplikasi saya...	[pembaharuan, 'harus', 'ada', 'pembaharuan', 'di']	[pembaharuan, 'aplikasi', 'pembaharuan', 'di']	baharu aplikasi dengan serta belaka ataght...
1	2026-01-27 22:22:42	DGR	1	Tolong kasih pelayananya sesuai arahan saya...	Tolong kasih pelayananya sesuai arahan saya...	Tolong kasih pelayananya sesuai arahan saya...	Tolong kasih pelayananya sesuai arahan saya...	[tolong, 'kasih', 'pelayananya', 'saya', 'arah']	[tolong, 'kasih', 'pelayananya', 'saya', 'arah']	tolong kasih layanan sesuai arahan yang serb...
2	2026-01-27 11:22:02	Rana Mardani	1	wahai gajaca dibukaka	wahai gajaca dibukaka	wahai gajaca dibukaka	wahai gajaca dibukaka	[wahai, 'gajaca', 'dibukaka']	[wahai, 'gajaca', 'dibukaka']	wahai gajaca dibukaka
3	2026-01-27 09:32:43	Syafid Nurhasan	1	barang tidak dikirim	barang tidak dikirim	barang tidak dikirim	barang tidak dikirim	[barang, 'tidak', 'dikirim']	[barang, 'tidak', 'dikirim']	barang kirim
4	2026-01-27 09:32:41	Aasyah Dahi	1	sangat peka pengantaran barang dikur kehapus	sangat peka pengantaran barang dikur kehapus	sangat peka pengantaran barang dikur kehapus	sangat peka pengantaran barang dikur kehapus	[sangat, 'peka', 'pengantaran', 'barang', 'dikur', 'kehapus']	[peka, 'pengantaran', 'barang', 'dikur', 'kehapus']	peka antar barang kur hap...
5	2026-01-27 09:14:01	Eng Hayati	4	saya kicewa mau top up di atfamer memang yg baik...	saya kicewa mau top up di atfamer memang yg baik...	saya kicewa mau top up di atfamer memang yg baik...	saya kicewa mau top up di atfamer memang yg baik...	[saya, 'kicewa', 'mau', 'top', 'up', 'di', 'atfamer', 'memang', 'yg', 'baik']	[kicewa, 'top', 'up', 'atfamer', 'memang', 'yg']	kicewa top up atfamer memang yg baik benar di...
6	2026-01-27 09:06:00	Mia Gati	1	pesan dari jam pagi digiya jam sore padahal...	pesan dari jam pagi digiya jam sore padahal...	pesan dari jam pagi digiya jam sore padahal...	pesan dari jam pagi digiya jam sore padahal...	[pesan, 'dari', 'jam', 'pagi', 'digiya', 'jam', 'sore', 'padahal']	[pesan, 'jam', 'pagi', 'digiya', 'jam', 'sore', 'padahal']	pesan jam pagi digiya jam sore padahal...
7	2026-01-27 07:42:44	Rendi Achmad	2	status pengiriman yang awalnya sedang dikirim...	status pengiriman yang awalnya sedang dikirim...	status pengiriman yang awalnya sedang dikirim...	status pengiriman yang awalnya sedang dikirim...	[status, 'pengiriman', 'yang', 'awalnya', 'sedang', 'dikirim']	[status, 'pengiriman', 'yang', 'awalnya', 'sedang', 'dikirim']	status kirim barang jam kati baik
8	2026-01-27 06:44:40	Ani Dennyani	1	tolong dikirim lagi thru hutang pengirim...	tolong dikirim lagi thru hutang pengirim...	tolong dikirim lagi thru hutang pengirim...	tolong dikirim lagi thru hutang pengirim...	[tolong, 'dikirim', 'lagi', 'thru', 'hutang', 'pengirim']	[tolong, 'dikirim', 'lagi', 'thru', 'hutang', 'pengirim']	tolong kirim thru hutang pengirim...
9	2026-01-27 06:30:34	ISI NIKHY RBBH	1	ALFAGET DS PONDOK NDAH MAKIN KETIR MAKIN B...	ALFAGET DS PONDOK NDAH MAKIN KETIR MAKIN B...	ALFAGET DS PONDOK NDAH MAKIN KETIR MAKIN B...	ALFAGET DS PONDOK NDAH MAKIN KETIR MAKIN B...	[alfaget, 'ds', 'pondok', 'ndah', 'makin', 'ketir', 'makin', 'b']	[alfaget, 'ds', 'pondok', 'ndah', 'makin', 'ketir', 'makin', 'b']	alfaget ds pondok ndah makin ketir makin b...

Gambar 24Tampilan Dataset

Sumber : Dokumentasi Pribadi

Menghapus Data yang kosong

```

[93] df = data.dropna()
[94] df.info()

<class 'pandas.core.frame.DataFrame'>
Index: 1886 entries, 0 to 1899
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype  
---  -
 0   Date                1886 non-null  object 
 1   Username            1886 non-null  object 
 2   Rating              1886 non-null  int64  
 3   Review Text         1886 non-null  object 
 4   cleaning             1886 non-null  object 
 5   case_folding         1886 non-null  object 
 6   normalisasi         1886 non-null  object 
 7   tokenize            1886 non-null  object 
 8   stopwords removal   1886 non-null  object 
 9   stemming_data       1886 non-null  object 
dtypes: int64(1), object(9)
memory usage: 162.1+ KB

```

Gambar 25 Menghapus Data Kosong

Sumber : Dokumentasi Pribadi

Melakukan Pelabelan Sentimen sesuai Nilai Score

```

[17] import pandas as pd

#Fungsi untuk menentukan sentimen
def determine_sentiment(text):
    positive_count = sum(1 for word in text.split() if word in positive_lexicon)
    negative_count = sum(1 for word in text.split() if word in negative_lexicon)
    if positive_count > negative_count:
        return "Positif"
    elif positive_count < negative_count:
        return "Negatif"

#Baca kamus leksikon positif dan negatif
positive_lexicon = set(pd.read_csv("positive.tsv", sep="\t", header=None)[0])
negative_lexicon = set(pd.read_csv("negative.tsv", sep="\t", header=None)[0])

def replace_none_sentiment(sentiment):
    replace_flag = "Positif"
    for i in range(len(sentiment)):
        if sentiment[i] is None:
            sentiment[i] = replace_flag
            replace_flag = "Negatif" if replace_flag == "Positif" else "Positif"
    return sentiment

df['Sentiment'] = df['stemming_data'].apply(determine_sentiment)
df['Sentiment'] = replace_none_sentiment(df['Sentiment'].tolist())

df.head()

```

Gambar 26 Pelabelan Sentimen

Sumber : Dokumentasi Pribadi

Date	Username	Rating	Review Text	cleaning	case_folding	normalisasi	tokenize	stopword removal	stemming data	Sentiment
2026-01-28 03:51:56	Sentia Thalys	3	mungkin harus ada pembaharuan di aplikasi saya...	mungkin harus ada pembaharuan di aplikasi saya...	mungkin harus ada pembaharuan di aplikasi saya...	mungkin harus ada pembaharuan di aplikasi saya...	['mungkin', 'harus', 'ada', 'pembaharuan', 'di', 'aplikasi', 'saya', '...']	['pembaharuan', 'aplikasi', 'pelanggan', 'seti...']	baharu aplikasi langgan setia belanja alaglit.	Positif
2026-01-27 23:29:42	DdR	1	Tolong kasih pelayananya sesuai anntan ada o...	Tolong kasih pelayananya sesuai anntan ada o...	tolong kasih pelayananya sesuai anntan ada o...	tolong kasih pelayananya sesuai anntan ada o...	['tolong', 'kasih', 'pelayananny', 'sesuai', '...']	['tolong', 'kasih', 'pelayananny', 'sesuai', '...']	tolong kasih layan sesuai annti orang serobot.	Positif
2026-01-27 11:22:02	Raka Mandagi	1	woll gabisa dibukaaa	woll gabisa dibukaaa	woll gabisa dibukaaa	woll gabisa dibukaaa	['woll', 'gabisa', 'dibukaaa']	['woll', 'gabisa', 'dibukaaa']	woll gabisa dibukaaa	Negatif
2026-01-27 09:32:43	Syafid Nurhusan	1	barang tidak dikirim	barang tidak dikirim	barang tidak dikirim	barang tidak dikirim	['barang', 'tidak', 'dikirim']	['barang', 'dikirim']	barang kirim	Positif
2026-01-27 09:32:41	Anandya Dian	1	sangat jelek pengantaran ditaruh diluar kehuja...	sangat jelek pengantaran ditaruh diluar kehuja...	sangat jelek pengantaran ditaruh diluar kehuja...	sangat jelek pengantaran ditaruh diluar kehuja...	['sangat', 'jelek', 'pengantaran', 'ditaruh', 'diluar', '...']	['jelek', 'pengantaran', 'ditaruh', 'diluar', '...']	jelek antar ditaruh luar huan	Negatif

Gambar 27 Hasil Pelabelan Sentimen

Sumber : Dokumentasi Pribadi

Visualisasi jumlah Ulasan berdasarkan Sentimen

```

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Functions for sentiment analysis, re-defined for robustness
def determine_sentiment(text):
    positive_count = sum(1 for word in text.split() if word in positive_lexicon)
    negative_count = sum(1 for word in text.split() if word in negative_lexicon)
    if positive_count > negative_count:
        return "Positif"
    elif positive_count < negative_count:
        return "Negatif"
    else:
        return None # handled by replace_none_sentiment

def replace_none_sentiment(sentiment_list):
    processed_sentiment = []
    replace_flag = "Positif"
    for s in sentiment_list:
        if s is None:
            processed_sentiment.append(replace_flag)
            replace_flag = "Negatif" if replace_flag == "Positif" else "Positif"
        else:
            processed_sentiment.append(s)
    return processed_sentiment

# Load lexicons (ensure these files are in the working directory)
positive_lexicon = set(pd.read_csv('positive.txt', sep='\\t', header=None)[0])
negative_lexicon = set(pd.read_csv('negative.txt', sep='\\t', header=None)[0])

# Ensure 'sentiment' column is created/updated on the current df
df['sentiment'] = df['cleaning_data'].apply(determine_sentiment)
df['sentiment'] = replace_none_sentiment(df['sentiment'].tolist())

sentiment_count = df['sentiment'].value_counts()
sns.set_style('whitegrid')

fig, ax = plt.subplots(figsize=(8, 4))
ax = sns.barplot(x=sentiment_count.index, y=sentiment_count.values, palette='pastel')
plt.title('Sentimen Analisis Sentimen', fontsize=12)
plt.xlabel('Class Sentiment', fontsize=12)
plt.ylabel('Jumlah Tweet', fontsize=12)
plt.xticks(rotation=45, fontsize=12)

total = len(df['sentiment'])

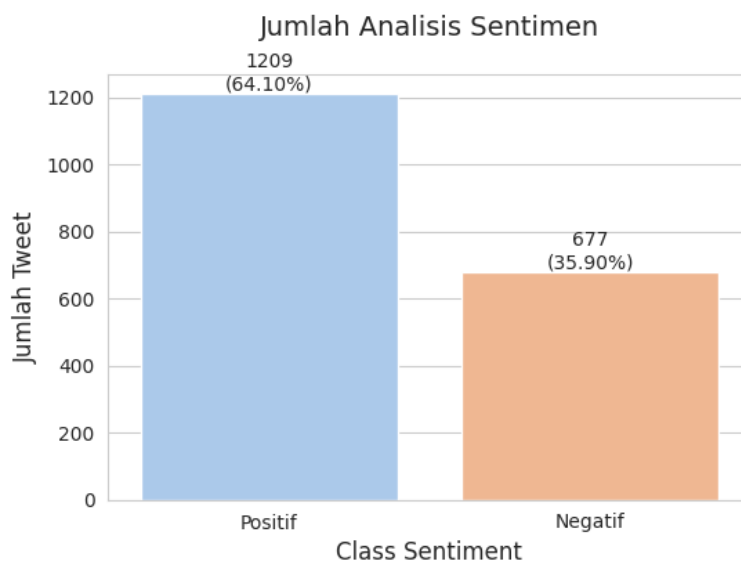
for i, count in enumerate(sentiment_count.values):
    percentage = f'{int(count / total * 100)}%'
    ax.text(i, count + 0.18, f'{count} ({percentage})', ha='center', va='bottom')

plt.show()

```

Gambar 28 Proses Visualisasi ulasan berdasarkan Sentimen

Sumber : Dokumentasi Pribadi

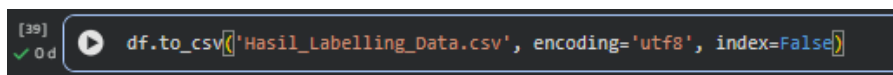


Gambar 29 Jumlah Analisis Sentimen

Sumber : Dokumentasi Pribadi

Pada Gambar 29 terlihat bahwa dataset ulasan aplikasi terbagi menjadi dua kategori sentimen. Ulasan dengan sentimen positif berjumlah 1.029 ulasan atau sekitar 64,10%, sedangkan ulasan dengan sentimen negatif berjumlah 677 ulasan atau sekitar 35,90%.

Ekspor Data yang sudah di labeli menjadi CSV

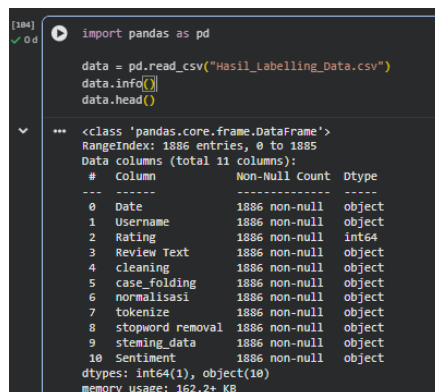


Gambar 30 Ekspor Data

Sumber : Dokumentasi Pribadi

Splitting Data

Import Dataset yang telah dilabeli



Gambar 31 Import Dataset

Sumber : Dokumentasi Pribadi

Date	Username	Rating	Review Text	cleaning	case_folding	normalisasi	tokenize	stopword_removal	stemming_data	Sentiment
2016-01-18 03:51:54	Sasha Dahan	5	mengharu kan ada pembunuhannya di aplikasi saya	mengharu kan ada pembunuhannya di aplikasi saya	mengharu kan ada pembunuhannya di aplikasi saya	mengharu kan ada pembunuhannya di aplikasi saya	[mengharu, 'kan', ada, pembunuhannya, di, aplikasi, saya]	[mengharu, 'kan', ada, pembunuhannya, di, aplikasi, saya]	mengharu kan ada pembunuhannya di aplikasi saya	Positif
2016-01-27 22:29:42	Dadi	1	Tolong kasih pelayanan yang sesuai untuk saya	Tolong kasih pelayanan yang sesuai untuk saya	Tolong kasih pelayanan yang sesuai untuk saya	Tolong kasih pelayanan yang sesuai untuk saya	[tolong, kasih, pelayanan, yang, sesuai, untuk, saya]	[tolong, kasih, pelayanan, yang, sesuai, untuk, saya]	Tolong kasih pelayanan yang sesuai untuk saya	Negatif
2016-01-27 11:42:35	Rana Mardiyah	1	versi aplikasi dibatasi	versi aplikasi dibatasi	versi aplikasi dibatasi	versi aplikasi dibatasi	[versi, aplikasi, dibatasi]	[versi, aplikasi, dibatasi]	versi aplikasi dibatasi	Negatif
2016-01-27 09:32:43	Syafiq Nurhasan	1	barang tidak dikirim	barang tidak dikirim	barang tidak dikirim	barang tidak dikirim	[barang, tidak, dikirim]	[barang, tidak, dikirim]	barang tidak dikirim	Negatif
2016-01-27 08:32:43	Andriyus Dini	1	sangat baik pelayanan dan barang dikirim	sangat baik pelayanan dan barang dikirim	sangat baik pelayanan dan barang dikirim	sangat baik pelayanan dan barang dikirim	[sangat, baik, pelayanan, dan, barang, dikirim]	[sangat, baik, pelayanan, dan, barang, dikirim]	sangat baik pelayanan dan barang dikirim	Positif

Gambar 32 Dataset setelah dilabeli

Sumber : Dokumentasi Pribadi

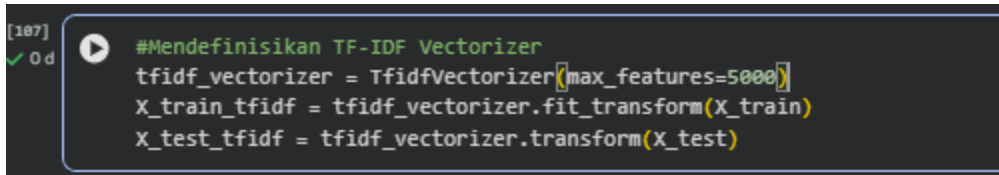
Pembagian Data Latih dan Data Uji



Gambar 33 Pembagian Data dan Hasil Pembagian Data

Pembobotan TF-IDF

Mendefinisikan TF-IDF



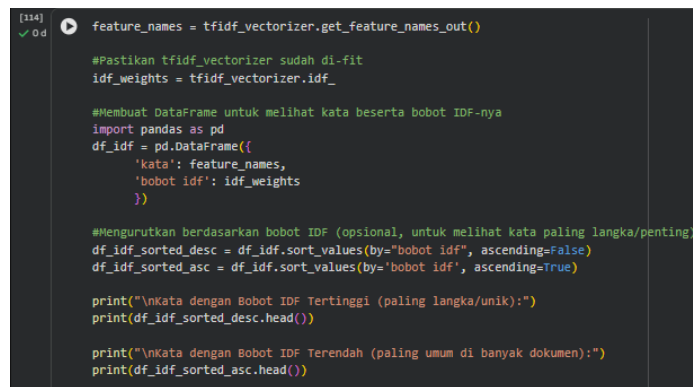
```
[107]
✓ 0 d #Mendefinisikan TF-IDF Vectorizer
tfidf_vectorizer = TfidfVectorizer(max_features=5000)
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)
X_test_tfidf = tfidf_vectorizer.transform(X_test)
```

Gambar 34 Mendefinisikan TF-IDF

Sumber : Dokumentasi Pribadi

Kode ini digunakan untuk mengubah data ulasan dari bentuk teks menjadi bentuk numerik menggunakan metode TF-IDF. Langkah ini sangat penting sebelum melatih model klasifikasi, karena model seperti *Support Vector Machine* hanya dapat memproses data dalam bentuk angka, bukan teks mentah.

Menganalisis bobot Inverse Document Frequency (IDF)



```
[114]
✓ 0 d feature_names = tfidf_vectorizer.get_feature_names_out()

#Pastikan tfidf_vectorizer sudah di-fit
idf_weights = tfidf_vectorizer.idf_

#Membuat DataFrame untuk melihat kata beserta bobot IDF-nya
import pandas as pd
df_idf = pd.DataFrame({
    'kata': feature_names,
    'bobot idf': idf_weights
})

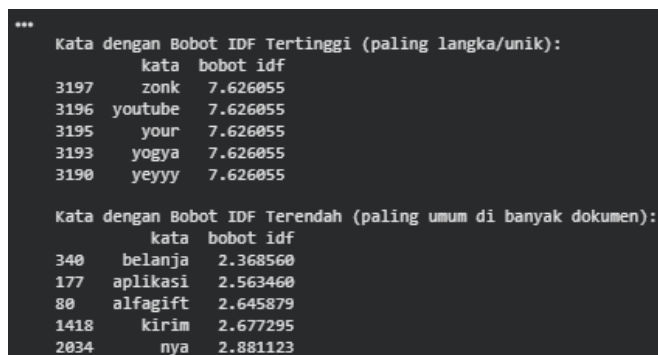
#Mengurutkan berdasarkan bobot IDF (opsional, untuk melihat kata paling langka/penting)
df_idf_sorted_desc = df_idf.sort_values(by="bobot idf", ascending=False)
df_idf_sorted_asc = df_idf.sort_values(by="bobot idf", ascending=True)

print("\nKata dengan Bobot IDF Tertinggi (paling langka/unik):")
print(df_idf_sorted_desc.head())

print("\nKata dengan Bobot IDF Terendah (paling umum di banyak dokumen):")
print(df_idf_sorted_asc.head())
```

Gambar 35 Analisa bobot IDF

Sumber : Dokumentasi Pribadi



```
***
Kata dengan Bobot IDF Tertinggi (paling langka/unik):
   kata  bobot idf
3197  zonnk    7.626055
3196  youtube    7.626055
3195   your    7.626055
3193  yogya    7.626055
3190  yeyyy    7.626055

Kata dengan Bobot IDF Terendah (paling umum di banyak dokumen):
   kata  bobot idf
340  belanja    2.368560
177  aplikasi    2.563460
80   alfagift    2.645879
1418 kirim    2.677295
2034  nya    2.881123
```

Gambar 4. 2 Output Analisa bobot IDF

Sumber : Dokumentasi Pribadi

Pada Gambar 37 terlihat bahwa kata-kata dengan nilai IDF tertinggi antara lain zonnk, youtube, your, yogya, dan yeyyy, yang menunjukkan kata-kata tersebut muncul sangat jarang, mungkin hanya dalam satu ulasan, dan bisa jadi merupakan typo atau kata yang tidak umum digunakan. Sebaliknya, kata-kata dengan bobot IDF rendah seperti bagus, ya, aplikasi, dan cari muncul sangat sering dalam berbagai ulasan, menandakan kata-kata ini umum digunakan

oleh pengguna. Berikut ini adalah contoh perhitungan manual untuk salah satu kata dalam analisis TF-IDF :

Menghitung Term Frequency

$$TF(zonk) = \frac{1}{9} = 0.111$$

Menghitung Document Frequency

$$IDF(zonk) = \log\left(\frac{1508 + 1}{1 + 1}\right) + 1$$

$$IDF(zonk) = \log\left(\frac{1509}{2}\right) + 1$$

$$IDF(zonk) = \log(754,5) + 1$$

$$IDF(zonk) = 6.62655 + 1 = 7.62655$$

Menghitung bobot akhir TF-IDF

$$TF - IDF = TF \times IDF$$

$$TF - IDF(zonk) = 0.111 \times 7.62655 = 0.846$$

Penerapan Support Vector Machine

Mencari parameter menggunakan GridSearchCV

```
from sklearn.model_selection import GridSearchCV

#Mencari parameter terbaik menggunakan GridSearchCV
param_grid = {'C': [0.1, 1, 10], 'kernel': ['linear', 'rbf']}
grid_search = GridSearchCV(SVC(), param_grid, cv=5)
grid_search.fit(X_train_tfidf, y_train)

print("Best Parameters:", grid_search.best_params_)

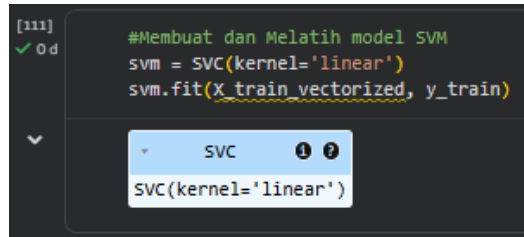
Best Parameters: {'C': 1, 'kernel': 'linear'}
```

Gambar 37 GridSearchCV

Sumber : Dokumentasi Pribadi

Model *Support Vector Machine* (SVM) terbaik diperoleh dengan menggunakan kernel linier dan nilai parameter C sebesar 1. Hasil ini menunjukkan bahwa pola data sentimen pada ulasan aplikasi Alfagift dapat dipisahkan secara linier dengan cukup baik. Oleh karena itu, penggunaan kernel yang lebih kompleks seperti *Radial Basis Function* (RBF) tidak diperlukan, karena selain tidak memberikan peningkatan kinerja yang signifikan, juga membutuhkan daya komputasi yang lebih besar.

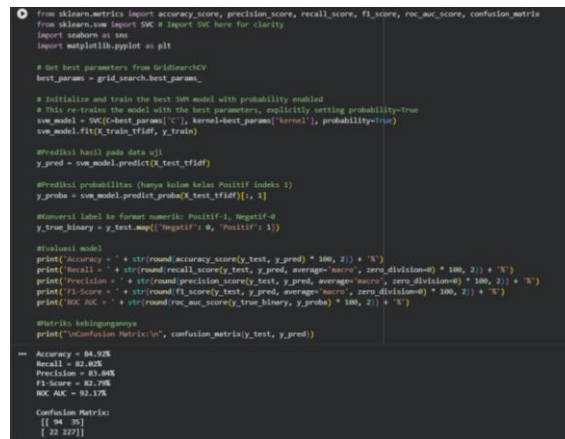
Membuat dan Melatih Support Vector Machin (SVM)



Gambar 38 Membuat dan Melatih model SVM

Evaluasi Model

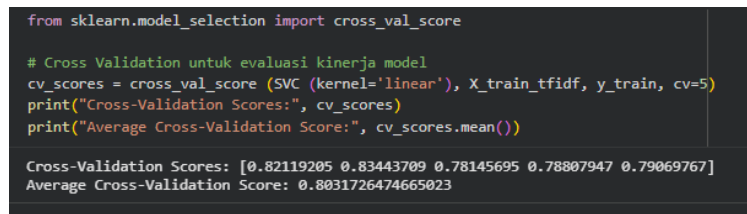
Hasil Matrix Evaluasi



Gambar 39 Hasil Matrix Evaluasi

Sumber : Dokumentasi Pribadi

Melakukan Cross Validation



Gambar 40 Kode dan Hasil Cross Validation

Sumber : Dokumentasi Pribadi

Berdasarkan hasil 5-fold cross-validation yang ditunjukkan pada Gambar 40, diperoleh nilai akurasi rata-rata sebesar 80,31%. Perbedaan kinerja antar setiap fold terlihat relatif kecil, yang menandakan bahwa model memiliki performa yang stabil dan konsisten pada berbagai pembagian data. Dengan demikian, model Support Vector Machine (SVM) dengan kernel linier dapat dikatakan cukup andal dan layak digunakan untuk melakukan prediksi sentimen pada data uji.

Membuat Visualisasi Confusion Matrix

```
[132]
✓ 0 d # Fungsi untuk plot confusion matrix

def plot_confusion_matrix(model_name, y_true, y_pred):
    cm = confusion_matrix(y_true, y_pred)

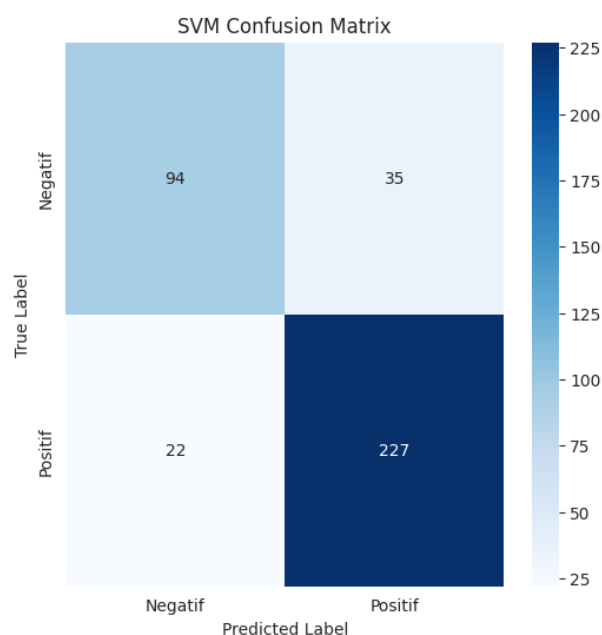
    plt.figure(figsize=(6, 6))
    sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
                xticklabels=['Negatif', 'Positif'],
                yticklabels=['Negatif', 'Positif'])

    plt.title(f"{model_name} Confusion Matrix")
    plt.xlabel("Predicted Label")
    plt.ylabel("True Label")
    plt.show()

plot_confusion_matrix("SVM", y_test, y_pred)
```

Gambar 41 Membuat Visualisasi Confus Matrix

Sumber : Dokumentasi Pribadi



Gambar 42 Visualisasi Confus Matrix

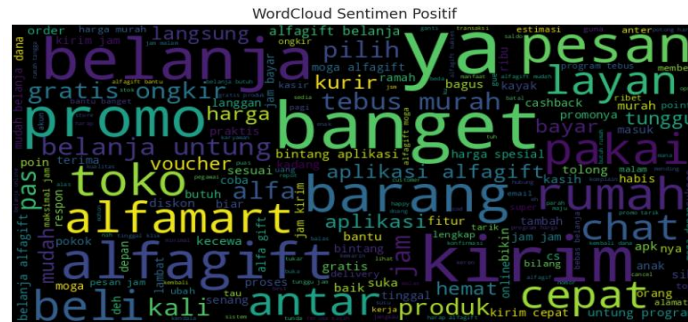
Sumber : Dokumentasi Pribadi

Tabel confusion matrix juga dapat dimanfaatkan untuk menghitung berbagai metrik evaluasi kinerja model, seperti akurasi, presisi, dan recall. Selain itu, nilai F1-Score turut dihitung sebagai ukuran keseimbangan antara presisi dan recall.

Visualisasi

Visualisasi pada tahap ini dirancang untuk mengekstraksi informasi penting dari kumpulan ulasan yang tersedia dengan mengelompokkan kata-kata berdasarkan topik yang paling sering dibahas oleh pengguna aplikasi Alfabeta. Dalam penelitian ini, digunakan word cloud dan diagram batang untuk menyajikan hasil analisis klasifikasi sentimen. Word cloud bertujuan menampilkan data teks secara visual, di mana ukuran kata mencerminkan frekuensi kemunculannya dalam ulasan—semakin besar ukuran kata, semakin sering kata

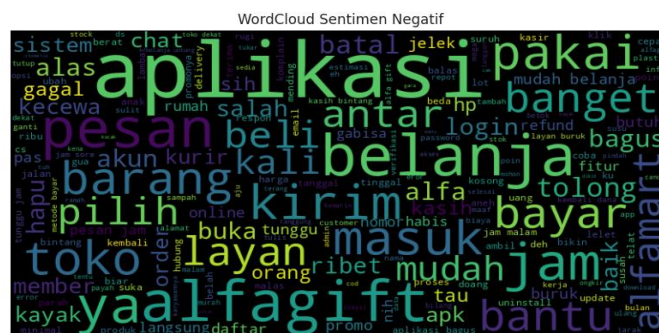
tersebut muncul. Sementara itu, diagram batang digunakan untuk menunjukkan jumlah kemunculan kata dalam bentuk grafik.



Gambar 43 Wordcloud Sentimen Positif

Sumber : Dokumentasi Pribadi

Berdasarkan Gambar 43, word cloud yang merepresentasikan sentimen positif menunjukkan bahwa pengguna yang merasa puas dengan aplikasi cenderung menggunakan kata-kata seperti “promo”, “alfagift”, “produk”, dan “cepat” dalam menyampaikan pendapat mereka. Hal ini mengindikasikan bahwa aspek promo, ketersediaan produk, serta kecepatan layanan menjadi faktor yang paling banyak diapresiasi oleh pengguna. Selain itu, perhatian pengguna juga banyak tertuju pada aktivitas pencarian gambar, pengumpulan foto, serta konten yang bersifat inspiratif. Munculnya ungkapan seperti “terima kasih” dan “beri bintang” menunjukkan bahwa aplikasi tidak hanya mampu memenuhi kebutuhan fungsional pengguna, tetapi juga berhasil menciptakan pengalaman yang positif.



Gambar 44 Wordcloud Sentimen Negatif

Sumber : Dokumentasi Pribadi

Berdasarkan Gambar 44, word cloud untuk sentimen negatif memperlihatkan bahwa banyak pengguna menyampaikan keluhan yang berkaitan dengan masalah teknis, seperti kendala pada proses login, pengisian tanggal lahir, serta fitur pencarian gambar. Selain itu, kata-kata seperti “gagal”, “sulit”, dan “jelek” sering digunakan untuk mengekspresikan ketidakpuasan terhadap kinerja beberapa fitur dalam aplikasi. Temuan ini memperkuat hasil

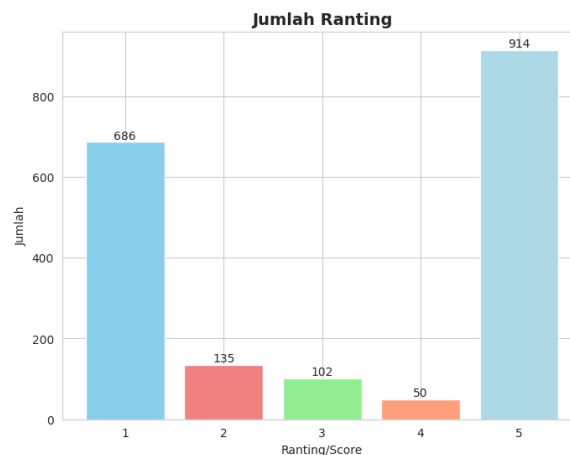
penelitian bahwa proses analisis sentimen perlu mempertimbangkan konteks kalimat secara menyeluruh, mengingat terdapat kata-kata yang umumnya bernilai positif, seperti “bagus”, namun muncul dalam kalimat bernada keluhan



Gambar 45 Wordcloud Sentimen Positif dan Negatif

Sumber : Dokumentasi Pribadi

Berdasarkan Gambar 45, terlihat bahwa kata-kata seperti “bagus”, “banget”, dan “alfagift” paling sering muncul, yang menunjukkan bahwa secara umum aplikasi ini mendapatkan penilaian positif dari para pengguna. Meskipun demikian, kemunculan kata-kata seperti “gagal”, “akun”, “tanggal lahir”, dan “tolong” mengindikasikan masih adanya sejumlah keluhan teknis yang dialami oleh pengguna. Selain itu, fitur-fitur seperti pencarian gambar, penandaan pin, dan proses unduhan menjadi topik yang cukup sering dibahas, yang menunjukkan bahwa pengalaman pengguna sangat dipengaruhi oleh aspek visual dan kemudahan penggunaan aplikasi.



Gambar 46 Visualisasi Distribusi Rating

Berdasarkan Gambar 46, terlihat bahwa sebagian besar ulasan pengguna memberikan skor tertinggi, yaitu 5, dengan total sebanyak 1.887 ulasan. Di sisi lain, skor terendah, yaitu 1, juga muncul dalam jumlah yang cukup signifikan, yakni sebanyak 686 ulasan. Sementara itu, ulasan dengan skor menengah (2, 3, dan 4) relatif lebih sedikit. Pola distribusi ini menunjukkan bahwa pendapat pengguna terhadap aplikasi cenderung terpolarisasi, di mana

sebagian besar pengguna merasa sangat puas, namun terdapat pula sejumlah pengguna yang merasa sangat kecewa. Oleh karena itu, dalam proses pembangunan dan evaluasi model klasifikasi sentimen, perlu diperhatikan ketidakseimbangan jumlah data antar kelas agar

D. KESIMPULAN

Penelitian ini menunjukkan bahwa penerapan algoritma Support Vector Machine (SVM) dengan kernel linier berhasil digunakan untuk menganalisis sentimen ulasan pengguna aplikasi Alfagift di platform Play Store. Model SVM mampu mengklasifikasikan sentimen dengan baik, yang ditunjukkan oleh nilai akurasi sebesar 84,92%, sehingga sebagian besar prediksi pada data uji telah sesuai dengan label sentimen yang sebenarnya. Hasil ini diperoleh melalui beberapa tahapan penelitian, mulai dari pengumpulan data, pra-pemrosesan teks, pemberian bobot kata menggunakan metode TF-IDF, hingga proses pelatihan dan evaluasi model. Nilai recall sebesar 82,02% menunjukkan kemampuan model dalam mengenali sebagian besar ulasan positif, sedangkan nilai presisi sebesar 83,84% menunjukkan bahwa mayoritas prediksi positif yang dihasilkan sudah tepat, dengan nilai F1-Score sebesar 82,79% yang mencerminkan keseimbangan antara keduanya. Dari keseluruhan ulasan, sebanyak 64,10% diklasifikasikan sebagai sentimen positif dan 35,90% sebagai sentimen negatif, di mana ulasan negatif umumnya berkaitan dengan permasalahan teknis seperti kesulitan login dan verifikasi akun, sedangkan ulasan positif lebih banyak menyoroti desain visual yang menarik serta kemudahan penggunaan aplikasi. Oleh karena itu, dapat disimpulkan bahwa metode SVM dengan kernel linier mampu memberikan hasil klasifikasi sentimen yang akurat dan dapat diandalkan sebagai alat bantu bagi pengembang dalam memahami opini pengguna dan meningkatkan kualitas layanan aplikasi Alfagift.

E. DAFTAR PUSTAKA

- Adinugroho, R. (2022). PERBANDINGAN RASIO SPLIT DATA TRAINING DAN DATA TESTING MENGGUNAKAN METODE LSTM.
- Al Hafizh, M. R., Aldi Daim Fauzan, Woro Isti Rayahu, Kiki Mustaqim, & Rahma Hanum. (2025). Web Scraping Data Ulasan Pelanggan untuk Kemajuan Bisnis E-Commerce pada Official Store dan Non-Official Store dengan Pendekatan Natural Language Processing. *Data Sciences Indonesia (DSI)*, 5(1), 1–9. <https://doi.org/10.47709/dsi.v5i1.5748>
- Alzahra, D. A., Enri, U., Yuyun, U., Informatika, P., & Karawang, U. S. (2023). Analisis Sentimen Ulasan Pengguna Klik Indomaret Pada Google Play Menggunakan Support Vector

Machine. 3(4), 2173–2185.

- Arsianto, J. L., & Gautama, T. K. (2025). Pengembangan Aplikasi Web Scraping untuk Crawling Web Data dari Situs E-Commerce Properti. *Jurnal Strategi*, 7(1), 151–163. <https://strategi.it.maranatha.edu/index.php/strategi/article/view/578/402>
- Buana, M. I., & Arianto, D. B. (2024). Analisis Sentimen Ulasan Pengguna Aplikasi ZenPro dengan Implementasi Algoritma Support Vector Machine (SVM). 3(1), 45–52.
- Dariato, E., Laple, R., & Putra, S. (2025). Analisis Sentimen Komentar Pengguna Aplikasi Alfagift Di Google Playstore Menggunakan Metode Support Vector Machine (SVM). *JATILIMA*, 07(03), 685–696.
- Endah Wiji Lestari M.Kom. (2022). WORKSHOP PELATIHAN DASAR PEMROGRAMAN PYTHON.
- Enjeli, M., & Wijaya, A. (2024). Analisis Sentimen Penggunaan Aplikasi Kinemaster Menggunakan Metode Naive Bayes. *JURNAL ILMIAH COMPUTER SCIENCE (JICS)*, 2, 89–98.
- Faadilah, A. (2020). ANALISIS SENTIMEN PADA ULASAN APLIKASI TOKOPEDIA DI GOOGLE PLAY STORE MENGGUNAKAN METODE LONG SHORT TERM MEMORY.
- Firdaus, R. Z., Wijoyo, S. H., & Purnomo, W. (2025). Analisis Sentimen Berbasis Aspek Ulasan Pengguna Aplikasi Alfagift Menggunakan Metode Random Forest dan Pemodelan Topik Latent Dirichlet Allocation. 9(2), 1–10.
- Hale, Y. H. U. (2024). Analisis Sentimen Ulasan Pengguna Aplikasi Shopee Menggunakan Metode Naive Bayes Dan K-Fold Cross Validation. *Bina Sarana Informatika*, 3(3), 80.
- Handika. (2024). Pemanfaatan Python dan Google Colab Dalam Pembelajaran Statistika Deskriptif. *Edumatnesia*, 379–389.
- Hendrawan, A., & Utomo, R. (2020). Penggunaan JavaScript untuk Interaksi Dinamis pada Situs Web. In Penerbit Tahta Media.
- Hendri Cipta Husada, A. S. (2021). Analisis Sentimen Pada Maskapai Penerbangan di Platform Twitter Menggunakan Algoritma Support Vector Machine (SVM) Sentiment Analysis of Airline on Twitter Platform Using Support Vector Machine (SVM) Algorithm. 10(1), 18–26. <https://doi.org/10.34148/teknika.v10i1.311>
- Irma, S. K. I., Yasin, A. M., & Irvan, A. S. (2023). Analisis Sentimen Terhadap Penggunaan Aplikasi Shopee Menggunakan Algoritma Support Vector Machine (SVM). *Jambura Journal of Electrical and Electronics Engineering*, 5(1), 32–35.
- Ismunandar, D. (2021). HYPERPARAMETER TUNING BERBASIS GRIDSEARCHCV PADA

KLASIFIKASI HARGA TELEPON GENGAM. 167–186.

- Kamal, W. W. (2021). Analisis Sentimen Ulasan Produk Skincare Menggunakan Metode Support Vector Machine (Studi Kasus: Forum Female Daily). Universitas Islam Indonesia, 65. <https://dspace.uui.ac.id/handle/123456789/34153>
- Kusuma, I. H., & Cahyono, N. (2023). Analisis Sentimen Masyarakat Terhadap Penggunaan E-Commerce Menggunakan Algoritma K-Nearest Neighbor. Jurnal Informatika: Jurnal Pengembangan IT, 8(3), 302–307. <https://doi.org/10.30591/jpit.v8i3.5734>
- Mardiana, M. (2023). SENTIMEN ANALISIS TERHADAP APLIKASI ALFAGIFT PADA GOOGLE PLAY STORE DENGAN ALGORITMA SUPPORT VECTOR MACHINE (SVM).
- Maulana, B. A., Fahmi, M. J., Imran, A. M., & Hidayati, N. (2024). Analisis Sentimen Terhadap Aplikasi Pluang Menggunakan Algoritma Naive Bayes dan Support Vector Machine (SVM). MALCOM: Indonesian Journal of Machine Learning and Computer Science, 4(2), 375–384. <https://doi.org/10.57152/malcom.v4i2.1206>
- Meilina, L., Kumara, I. N. S., & Setiawan, I. N. (2021). Literature Review Klasifikasi Data Menggunakan Metode Cosine Similarity dan Artificial Neural Network. Majalah Ilmiah Teknologi Elektro, 20(2), 307. <https://doi.org/10.24843/mite.2021.v20i02.p15>
- Nabawi, I. (2023). Klastering Spesifikasi Smartphone Menggunakan Algoritma Self Organizing Map. FAHMA – Jurnal Informatika Komputer, Bisnis Dan Manajemen, 21(1), 35–44.
- Pradana, A. I., & Atina, V. (2024). Teknik K-Fold Cross Validation untuk Mengevaluasi Kinerja Mahasiswa. Institut Teknologi Garut, 239–248. <https://doi.org/10.33364/algoritma/v.21-1.1618>
- Pratama, M. A. (2024). Perbandingan Metode Naïve Bayes Dan Support Vector Machine Pada Analisis Sentimen Ulasan Game Genshin Impact Skripsi. 87. <https://genshin.hoyoverse.com/id/game>
- Prayogi, M. B., & Masitoh, G. (2025). Analisis Sentimen Ulasan Pengguna Aplikasi Alfagift Menggunakan Random Forest. 10(2), 158–170.
- Rabbani, S., Safitri, D., Rahmadhani, N., Sani, A. A. F., & Anam, M. K. (2023). Perbandingan Evaluasi Kernel SVM untuk Klasifikasi Sentimen dalam Analisis Kenaikan Harga BBM: Comparative Evaluation of SVM Kernels for Sentiment Classification in Fuel Price Increase Analysis. MALCOM: Indonesian Journal of Machine Learning and Computer Science, 3(2), 153–160. <https://journal.irpi.or.id/index.php/malcom/article/view/897%0Ahttps://journal.irpi.or.id/index.php/malcom/article/download/897/421>

- Rhomaningtias, L., Khairunisa, A., Shella, S., Wara, M., & Hindrayani, K. M. (2025). ANALISIS SENTIMEN ULASAN APLIKASI SMILE INDONESIA MENGGUNAKAN METODE NAIVE BAYES DAN SUPPORT VECTOR MACHINE (SVM). HOAQ: JURNAL TEKNOLOGI INFORMASI, 16(c), 79–91.
- Riski, A., Pratama, I., Latipah, S. A., & Sari, B. N. (2022). OPTIMASI KLASIFIKASI CURAH HUJAN MENGGUNAKAN SUPPORT VECTOR MACHINE (SVM) DAN RECURSIVE FEATURE ELIMINATION (RFE). JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika), 07(2), 314–324.
- Rizki Wahyudi, G. K. (2021). Analisis Sentimen Review Aplikasi LinkedIn di Google Play Store Menggunakan Support Vector Machine. JURNAL INFORMATIKA, 8(2), 454. <https://doi.org/10.35889/progresif.v20i1.1614>
- Sarimole, F. M. (2024). Analisis Sentimen Terhadap Aplikasi Satu Sehat Pada Twitter Menggunakan Algoritma Naive Bayes Dan Support Vector Machine. Jurnal Sains Dan Teknologi, 5(3), 783–790.
- Styawati, S., Hendrastuty, N., & Isnain, A. R. (2021). Analisis Sentimen Masyarakat Terhadap Program Kartu Prakerja Pada Twitter Dengan Metode Support Vector Machine. Jurnal Informatika: Jurnal Pengembangan IT, 6(3), 150–155. <https://doi.org/10.30591/jpit.v6i3.2870>
- Valerian, F. R., Syarief, M., Fatah, D. A., Informasi, S., Madura, U. T., Kamal, K., & Timur, J. (2025). Klasifikasi tingkat obesitas menggunakan metode gbm dan confusion matrix. Jurnal Mahasiswa Teknik Informatika (JATI), 9(2), 2242–2249.
- Wilyani, F., Nuryan Arif, Q., & Aslimar, F. (2024). Pengenalan Dasar Pemrograman Python Dengan Google Colaboratory Basic Introduction to Python Programming With Google Colaboratory. Jurnal Pengabdian Pada Masyarakat Indonesia, 3(1), 788–796. <https://doi.org/10.55606/jppmi.v3i1.1087>
- Woro Isti Rahayu, Cahyo Prianto, E. A. N. (2021). PERBANDINGAN ALGORITMA K-MEANS DAN NAÏVE BAYES UNTUK MEMPREDIKSI PRIORITAS PEMBAYARAN TAGIHAN RUMAH SAKIT BERDASARKAN TINGKAT KEPENTINGAN PADA PT. PERTAMINA (PERSERO) Woro. Jurnal Teknik Informatika, 13(2), 1–8.