
Analisis Sentimen Terhadap Aplikasi *Google Meet* Berdasarkan Komentar Pengguna Menggunakan Metode Logistic Regression

Refandi Yusuf¹, Kanzun Bahumatra², Nurul Komaria³, Elyana Azatul Aqma⁴,

Laili Cahyani⁵

¹²³⁴⁵ Pendidikan Informatika, Universitas Trunojoyo Madura, Bangkalan, Indonesia

email: ¹ 200631100057@student.trunojoyo.ac.id, ² 200631100051@student.trunojoyo.ac.id,

³200631100047@student.trunojoyo.ac.id, ⁴ 200631100048@student.trunojoyo.ac.id, ⁵ laili.cahyani@trunojoyo.ac.id.

DOI: <https://doi.org/10.21107/edutic.v11i1.28113>

Abstrak

Google Meet merupakan aplikasi *video conference* yang digunakan untuk kegiatan jarak jauh secara online. *Google Meet* memiliki fitur-fitur yang mudah digunakan dan dapat melakukan panggilan video berkualitas tinggi. Penggunaan aplikasi *Google Meet* dalam sehari bisa mencapai 100 juta pengguna. Namun setelah menggunakan aplikasi *Google Meet* terdapat perbedaan pendapat memberikan rasa kepuasan dan ketidakpuasan terhadap aplikasi *Google Meet*. Hal ini dapat diamati dari ulasan yang diberikan pengguna pada kolom komentar di *Play Store*. Pada penelitian ini menggunakan metode logistic regression dan dilakukan analisis sentimen pengguna aplikasi *Google Meet* yang bertujuan mengelompokkan ulasan pengguna dalam bentuk sentimen positif, negatif dan netral. Sampel data yang digunakan adalah 700 data ulasan pengguna *Google Meet*, pengujian analisis sentimen dilakukan menggunakan metode *logistic regression* menunjukkan nilai akurasi sebesar 79.59%, dengan rincian ulasan negatif memiliki precision 100%, recall 16.67%, dan f1-score 28.57%. Ulasan netral memiliki precision 0%, recall 0%, dan f1-score 0%. Ulasan positif memiliki precision 79.02%, recall 100%, dan f1-score 88.28%. Secara umum didapatkan informasi mengenai kata yang paling sering muncul dalam kelas sentimen positif adalah bagus, mudah, mantap, oke, bantu, gambar, suara jernih, bermanfaat, keren, lancar, baru, coba, kualitas gambar baik, suka dan terimakasih. Sedangkan untuk kelas negatif didapatkan informasi mengenai kata yang paling sering muncul dalam kelas sentimen negatif adalah jelek, hapus, sulit, dan lambat. Dengan penelitian tersebut dapat dijadikan acuan dalam meningkatkan kualitas aplikasi *Google Meet*.

Kata Kunci: *Google Meet, Logistic Regression, Sentimen.*

Abstract

Google Meet is a video conferencing application that is used for online remote activities. *Google Meet* has easy-to-use features and can make high-quality video calls. The use of the *Google Meet* application in a day can reach 100 million users. However, after using the *Google Meet* application there are differences of opinion giving a sense of satisfaction and dissatisfaction with the *Google Meet* application. This can be observed from the reviews given by users in the Comments column on the *Play Store*. In this study used the logistic regression method and conducted a sentiment analysis of the *Google Meet* application users which aims to classify user reviews in the form of positive, negative and neutral sentiments. The sample data used was 700 *Google Meet* user review data, sentiment analysis testing was carried out using the logistic regression method showing an accuracy value of 79.59%, with negative review details having 100% precision, 16.67% recall, and 28.57% f1-score. Neutral reviews have a precision of 0%, recall of 0%, and an f1-score of 0%. Positive reviews have a precision of 79.02%, recall of 100%, and an f1-score of 88.28%. In general, information about the words that most often appear in the positive sentiment class is good, easy, steady, okay, helpful, pictures, clear sound, useful, cool, smooth, new, try, good image quality, like and thank you. As for the negative class, information obtained about the words that most often appear in the negative sentiment class are ugly, delete, difficult, and slow. With this research, it can be used as a reference in improving the quality of the *Google Meet* application.

Keywords : *Google Meet, Logistic Regression, Sentiment.*

PENDAHULUAN

Pada tahun 2020, Pandemi Covid-19 melanda Indonesia hingga dunia. Pandemi Covid-19 ini menyebabkan kelumpuhan di beberapa bidang seperti, Pendidikan. Sehingga para guru dan dosen memilih *e-learning* sebagai alat penunjang pembelajaran. Aplikasi *video conference* merupakan salah satu pendukung *e-learning* yang banyak digunakan oleh guru maupun dosen sebagai pendukung pembelajaran. *Google Meet* merupakan aplikasi *video conference* yang populer saat ini dan digunakan untuk kegiatan jarak jauh secara online. *Google Meet* memiliki fitur-fitur yang mudah digunakan dan dapat melakukan panggilan video berkualitas tinggi. Penggunaan aplikasi *Google Meet* dalam sehari bisa mencapai 100 juta pengguna.

Penelitian terdahulu merupakan bahan rujukan untuk mencari permasalahan terhadap bahasan guna membandingkan kemudian mencari permasalahan baru untuk penelitian selanjutnya di masa depan. Penelitian terdahulu juga digunakan untuk menunjukkan keaslian dan mencegah plagiarisme dalam penelitian. Peneliti melakukan studi literatur dan menemukan rujukan dari penelitian sebelumnya yang relevan dengan penelitian ini. Meskipun terdapat keterkaitan pembahasan, penelitian ini masih sangat berbeda dengan penelitian terdahulu. Berikut ini merupakan penelitian terdahulu yang peneliti temukan.

Penelitian yang dilakukan oleh Andy Novantika dan Sugiman (2022) yang berjudul “Analisis Sentimen Ulasan Pengguna Aplikasi Video Conference Google Meet menggunakan Metode SVM dan Logistic Regression”. Pada penelitian ini dilakukan analisis sentimen data ulasan pengguna aplikasi Google Meet dengan algoritma machine learning yang bertujuan menggolongkan data ulasan dalam bentuk sentimen positif atau negatif. Metode klasifikasi yang digunakan adalah Support Vector Machine dan Logistic Regression dengan 4 kernel SVM yaitu Linear, Radial Basis Function (RBF), Sigmoid dan Polynomial. Nilai akurasi yang didapatkan pada data ulasan aplikasi Google Meet untuk masing-masing kernel berturut-turut adalah 87,02%, 84,59%, 86,63% dan 71,12%, sementara untuk metode Logistic Regression sebesar 85,17%. Diperoleh kesimpulan bahwa algoritma terbaik dengan akurasi tertinggi adalah algoritma SVM dengan kernel Linear yaitu sebesar 87,02%. Persamaan penelitian ini dengan penelitian sebelumnya adalah Objek yang diteliti sama – sama ulasan pengguna Google Meet dan metode yang digunakan sama – sama Logistic Regression. Sedangkan perbedaan penelitian sebelumnya dengan penelitian ini adalah Metode klasifikasi yang digunakan penelitian sebelumnya menggunakan Support Vector Machine sedangkan penelitian ini tidak.

Setiap aplikasi memiliki kekurangan dan kelebihan masing-masing menurut pengguna, hal tersebut dapat memberikan respon tanggapan yang beragam dari pengguna aplikasi seperti rasa kepuasan atau rasa kekecewaan. Kolom komentar review ulasan aplikasi menjadi salah satu tempat untuk memberikan tanggapan kepuasan dan kekecewaan pengguna terhadap aplikasi tersebut (Giovani et al., 2020). Apabila aplikasi tersebut memberi pengaruh positif maka kepuasan pengguna dapat dikatakan sebagai ulasan yang positif. Sebaliknya tanggapan kekecewaan yang berikan pengguna dapat dikatakan sebagai ulasan yang negatif karena pengguna menerima hal yang tidak sesuai dengan apa yang diinginkan. Dengan adanya review dari aplikasi menjadikan informasi tambahan bagi perusahaan dalam mempengaruhi asumsi dan keputusan pengguna dalam menggunakan aplikasi yang diinginkan (Agustina et al., 2018). Pengguna baru menjadikan komentar sebagai tolak ukur untuk menentukan aplikasi yang terbaik dan memuaskan untuk digunakan.

Pada penelitian ini dilakukan klasifikasi analisis sentimen pengguna aplikasi *Google Meet* yang bertujuan mengelompokkan ulasan pengguna dalam bentuk sentimen positif, negatif dan netral. Klasifikasi adalah proses pengkategorian terhadap sekumpulan dokumen ke dalam kategori tertentu. Penelitian ini bertujuan untuk melakukan analisis sentimen terhadap aplikasi *Google Meet*

berdasarkan komentar pengguna menggunakan metode *logistic regression*. Data dikumpulkan dari berbagai platform ulasan online yang mengulas tentang pengalaman pengguna saat menggunakan aplikasi *Google Meet*. Dari hasil klasifikasi akan diperoleh hasil akurasi dalam mengklasifikasi data ulasan aplikasi google meet. Dengan analisis sentimen yang dilakukan akan didapatkan gambaran dari analisis data ulasan pada aplikasi, didapatkan juga nilai persentase ulasan positif, ulasan negatif, ulasan netral dan hasil kata yang paling muncul pada kolom ulasan aplikasi *Google Meet*.

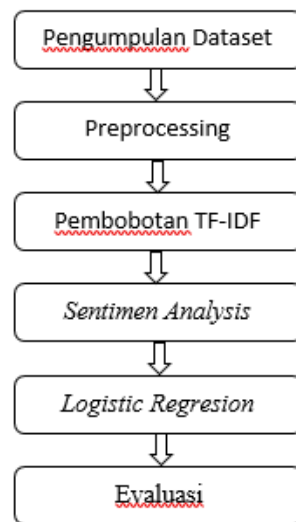
METODE PENELITIAN

Data yang digunakan untuk penelitian ini merupakan ulasan aplikasi *google meet* yang diambil dari web *appfollow.io*. Data yang didapat merupakan data ulasan berupa rating dan komentar pengguna dari *google play store*. Jumlah data yang diperoleh adalah sekitar 700 data ulasan .

	A	B	C	D	E	F	G	H	I
1	Date,AppName,Language,Review	Language,Author,Rating,Review,	kolom						
2	09/02/2023 01.05,Google Meet,id,id,Ifadillah Juzman,1,Gk bisa masukp,								
3	09/02/2023 06.32,Google Meet,id,id,Megawaitibn Zainuddin,5,Suka,								
4	09/02/2023 06.41,Google Meet,id,id,Siswi Yanto,1,Biasany bisa cepet.ini.kok.lambat?,								
5	09/02/2023 07.04,Google Meet,id,it,Dina Yuliana,1,Sulit di update,								
6	09/02/2023 07.37,Google Meet,id,pl,orindo santomi sinaga,4,zczct 8g8t3W9hicgrew64EAW722								
7	09/02/2023 08.31,Google Meet,id,en,Muhamad Nouval,5,Nice,								
8	09/02/2023 08.54,Google Meet,id,ms,Hadi Rosidi,1,Orang baru nih,								
9	09/02/2023 09.03,Google Meet,id,en,Ramadhani Dhani,5,Ok,								
10	09/02/2023 10.04,Google Meet,id,it,Petrus S. Suparjiono,1,Mohon di buka.,								
11	09/02/2023 10.37,Google Meet,id,id,Helmi,5,Bagus,								
12	09/02/2023 11.24,Google Meet,id,id,Debi Rahayu,2,Bguss,								
13	09/02/2023 12.12,Google Meet,id,id,Lukas Riva,5,Amazing,								
14	09/02/2023 12.21,Google Meet,id,en,Moh. Zahron Rois,5,Ok,								
15	09/02/2023 13.02,Google Meet,id,ro,Agung Sulaksono Saputra,5,Mantul pisan,								
16	09/02/2023 13.15,Google Meet,id,id,ryan,5,Bagus,								

Gambar 1. Dataset yang digunakan

Dari data tersebut nantinya akan dilakukan analisis sentimen dengan menggunakan metode *logistic regression* dengan tahapan sebagai berikut :



Gambar 2 . Alur Penelitian

Lima langkah penelitian pada Gambar 2 dimulai dengan pengumpulan data set, yaitu tahap pengumpulan data ulasan pengguna aplikasi google meet yang diambil dari web *appfollow.io* yang digunakan sebagai data uji tahap melakukan identifikasi dan analisis yang lebih spesifik terkait peringkasan teks otomatis dan penerapan metode TFIDF berbasis bobot kata. Langkah kedua yakni preprocessing, Dalam tahap preprocessing ini dilakukan Langkah case folding, Lowercase, spell normalization, filtering, stopword removal, dan tokenizing. Langkah ketiga adalah TF-IDF yaitu pembuatan model data dan peringkasan teks menggunakan metode TFIDF yang diterapkan pada

basis bobot kata. Langkah keempat penerapan metode logistic regression dengan bantuan pemrosesan menggunakan Bahasa Pemrograman Python. Langkah terakhir adalah mengevaluasi hasil penelitian dari segi durasi komputasi, kepuasan pembaca, serta menarik simpulan dan memberi saran atas kendala yang ditemukan dalam pembuatan sistem.

2.1 Text mining

Text mining adalah sebuah proses menambang teks untuk menemukan suatu informasi yang berguna dalam koleksi dokumen teks sehingga diperoleh pola, tren, atau keterhubungan antar teks (Sodik & Kharisudin, 2021). Dalam *text mining* ada beberapa tahapan yang dilalui dengan detail sebagai berikut :

2.1.1 Text preprocessing

Langkah awal dalam preprocessing teks adalah *case folding*, yang mengubah semua huruf dalam dokumen menjadi huruf kecil (*lowercase*). Setelah itu, dilakukan *spell normalization* untuk memperbaiki kata-kata yang disingkat dan tidak baku dengan menggunakan kamus. Tahap berikutnya adalah *tokenizing*, yaitu *filtering* atau membersihkan data dari karakter yang tidak penting seperti spasi atau angka. Lalu dilakukan *stopword removal* untuk menghilangkan kata hubung. Terakhir memecah kalimat menjadi kata-kata secara individu dengan *tokenizing*.

2.1.2 Pembobotan TF-IDF (Term Frequency-Inverse Document Frequency)

TF-IDF adalah sebuah metode yang merupakan integrasi antar *term frequency* (TF), dan *inverse document frequency* (IDF) (Widyasanti et al., 2018). TF-IDF ini adalah sebuah ukuran statistik yang digunakan untuk mengevaluasi seberapa penting sebuah kata di dalam sebuah dokumen. TF adalah frekuensi kemunculan kata pada di tiap dokumen yang diberikan menunjukkan seberapa penting kata itu di dalam tiap dokumen tersebut (Septian et al., 2019).

Term Frequency dihitung menggunakan Persamaan (1) dengan *term frequency* ke-*i* adalah frekuensi kemunculan term ke-*i* dalam dokumen ke-*j*. *Inverse Document Frequency* (IDF) adalah logaritma dari rasio jumlah seluruh dokumen dalam korpus dengan jumlah dokumen yang memiliki term yang dimaksud seperti yang dituliskan secara matematis pada Persamaan (2). Nilai didapatkan dengan mengalikan keduanya yang diformulasikan pada Persamaan (3).

$$tf_i = \frac{freq_i(d_j)}{\sum_{i=1}^k freq_i(d_j)} \quad (1)$$

$$idf_i = \log \frac{|D|}{|\{d:t_i \in d\}|} \quad (2)$$

$$(tf - idf)_{ij} = tf_i(d_j) * idf_i \quad (3)$$

Fungsi metode TFIDF adalah untuk mencari representasi nilai dari tiap-tiap dokumen dari suatu kumpulan data training (*training set*) dimana nantinya dibentuk suatu vektor antara dokumen dengan kata (*documents with terms*) yang kemudian untuk kesamaan antar dokumen dengan cluster akan ditentukan oleh sebuah prototype vektor yang disebut juga dengan *cluster centroid* (Widyasanti et al., 2018)

2.2 Sentimen Analysis

Analisis sentimen adalah sebuah proses untuk menentukan sentimen atau opini dari seseorang yang diwujudkan dalam bentuk teks dan bisa dikategorikan sebagai sentimen positif atau negatif (Hermanto & Noviriandini, 2021). Dengan adanya sentimen analisis, komentar dapat digunakan untuk menilai suatu subjek.

2.3 Logistic Regresion

Logistik Regression atau Regresi logistik merupakan suatu teknik analisis data dalam statistika yang bertujuan untuk mengetahui hubungan antara beberapa variabel dimana variabel responnya adalah bersifat kategorik, baik nominal maupun ordinal dengan variabel penjelasnya dapat bersifat kategorik atau kontinu (Ramli et al., 2013). Metode *logistic regression* dan *linear regression* memiliki teknik dan prosedur yang tidak jauh berbeda. Metode *ordinary least square* (OLS) sering digunakan dalam proses mengestimasi nilai parameter oleh linear, sementara dalam mengestimasi nilai parameter *linear regression* menggunakan metode *maximum likelihood estimation* (MLE).

Rumus :

$$\pi(x) = \frac{e^{\beta_0 + \sum_{j=1}^p \beta_j x_j}}{1 + e^{\beta_0 + \sum_{j=1}^p \beta_j x_j}} \quad (1)$$

Dari persamaan (1) diperoleh $1 - \pi(x)$ sebagai berikut:

$$1 - \pi(x) = 1 - \frac{e^{\beta_0 + \sum_{j=1}^p \beta_j x_j}}{1 + e^{\beta_0 + \sum_{j=1}^p \beta_j x_j}}$$

$$1 - \pi(x) = \frac{1 + e^{\beta_0 + \sum_{j=1}^p \beta_j x_j} - e^{\beta_0 + \sum_{j=1}^p \beta_j x_j}}{1 + e^{\beta_0 + \sum_{j=1}^p \beta_j x_j}}$$

$$= \frac{1}{1 + e^{\beta_0 + \sum_{j=1}^p \beta_j x_j}}$$

Sehingga $\frac{\pi(x)}{1 - \pi(x)}$ sebagai berikut:

$$\frac{\pi(x)}{1 - \pi(x)} = e^{\beta_0 + \sum_{j=1}^p \beta_j x_j} \quad (2)$$

Jadi, persamaan logistiknya adalah:

$$g(x) = \ln\left(\frac{\pi(x)}{1 - \pi(x)}\right)$$

$$= \ln\left(e^{\beta_0 + \sum_{j=1}^p \beta_j x_j}\right)$$

$$= \beta_0 + \sum_{j=1}^p \beta_j x_j$$

HASIL DAN PEMBAHASAN

Data yang telah diperoleh kemudian dilakukan *text preprocessing* sebagai berikut :

- Tahap Preprocessing

```
In [2]: # Menghapus kolom 'a' dan 'Unnamed'
df = df.drop(['kolom', 'Date', 'AppName', 'Language', 'Review Language', 'Author'], axis=1)
df.head()
```

Out[2]:

	Rating	Review
0	1	Gk bisa masukp
1	5	Suka
2	1	Biasany bisa cepet ini.kok.lambat?
3	1	Sulit di update
4	4	zczct 8g8t3W9hicgrew64EAW722Z3TZD4x4uhrz8gugtw...

Gambar 3. Tahap penghilangan kolom

Gambar 3 adalah tahap awal yang dilakukan yaitu menghilangkan kolom yang tidak diperlukan nantinya. Kolom yang tersisa setelah di drop adalah kolom Rating dan kolom Review.

```
In [3]: # Membuat fungsi untuk menentukan nilai sentimen
def sentiment_label(rating):
    if rating <= 2:
        return 'negatif'
    elif rating >= 4:
        return 'positif'
    else:
        return 'netral'

# Menerapkan fungsi pada kolom 'Rating' dan membuat kolom baru 'Label sentimen'
df['label sentimen'] = df['Rating'].apply(sentiment_label)

# Menampilkan DataFrame dengan kolom baru
df.head(10)
```

Out[3]:

	Rating	Review	label sentimen
0	1	Gk bisa masukp	negatif
1	5	Suka	positif
2	1	Biasany bisa cepet.ini.kok.lambat?	negatif
3	1	Sulit di update	negatif
4	4	zccct 8g8t3W9hicgrew64EAW722Z3TZD4x4uhrz8gugtw...	positif
5	5	Nice	positif
6	1	Orang baru nih	negatif
7	5	Ok	positif
8	1	Mohon di buka.	negatif
9	5	Bagus	positif

Gambar 4. Pemberian label

Pada Gambar 4 ini Selanjutnya memberikan label sentimen terhadap Review yang telah ditulis. Label dibuat berdasarkan rating yang diberikan. Apabila rating kurang dari 2 maka mendapat nilai negatif, Jika lebih dari 4 maka mendapat nilai positif dan selain dari itu mendapat nilai netral.

```
In [4]: #Case folding
# mengubah huruf kapital menjadi huruf kecil pada kolom Review
df['Review'] = df['Review'].str.lower()

# menampilkan 10 baris pertama dari dataframe setelah mengubah huruf kapital menjadi huruf kecil pada kolom Review
df.head(10)
```

Out[4]:

	Rating	Review	label sentimen
0	1	gk bisa masukp	negatif
1	5	suka	positif
2	1	biasany bisa cepet.ini.kok.lambat?	negatif
3	1	sulit di update	negatif
4	4	zccct 8g8t3w9hicgrew64eaw722z3tzd4x4uhrz8gugtw...	positif
5	5	nice	positif
6	1	orang baru nih	negatif
7	5	ok	positif
8	1	mohon di buka.	negatif
9	5	bagus	positif

Gambar 5. Case folding

Gambar 5 dilakukan *Case folding* mengubah huruf kapital menjadi huruf kecil pada kolom *Review*.

```
In [5]: #Import Kamus
key_df = pd.read_csv('E:\Kuliah\Semester 6\Text Mining\Kamus Alay\key_norm.csv')

# Membuat dictionary dari kamus dengan key adalah kata tidak baku dan value adalah kata baku
dictionary = dict(zip(key_df['singkat'], key_df['hasil']))

# Fungsi untuk mengganti kata tidak baku dengan kata baku berdasarkan kamus
def replace_words(text):
    words = text.split()
    new_words = []
    for word in words:
        if word in dictionary:
            new_words.append(dictionary[word])
        else:
            new_words.append(word)
    return ' '.join(new_words)

# Mengganti kata tidak baku pada kolom 'Review' dengan kata baku berdasarkan kamus
df['Review'] = df['Review'].apply(replace_words)

df.head()
```

Out[5]:

	Rating	Review	label sentimen
0	1	tidak bisa masukp	negatif
1	5	suka	positif
2	1	biasany bisa cepet.ini.kok.lambat?	negatif
3	1	sulit di update	negatif
4	4	zozot 8g8t3w9hiogrew64eaw722z3tzd4x4uhrz8gugtw...	positif

Gambar 6. Proses memasukkan kamus

Gambar 6 merupakan Proses selanjutnya dengan memasukkan kamus berdasarkan tujuan untuk mengubah pada *review* yang tidak baku menjadi baku sesuai dengan kata kata yang ada di kamus tersebut. hal ini akan memberikan hasil akhir lebih akurat karena kata yang diujikan akan lebih banyak.

```
In [6]: #Filtering
import re

# Menghilangkan karakter selain huruf dan spasi
df['Review'] = df['Review'].apply(lambda x: re.sub('[^a-zA-Z\s]', '', x))

# Menghilangkan angka
df['Review'] = df['Review'].apply(lambda x: re.sub('\d+', '', x))

# Menghilangkan multiple spasi
df['Review'] = df['Review'].apply(lambda x: re.sub('\s+', ' ', x))

df.head()
```

Out[6]:

	Rating	Review	label sentimen
0	1	tidak bisa masukp	negatif
1	5	suka	positif
2	1	biasany bisa cepetinikoklambat	negatif
3	1	sulit di update	negatif
4	4	zozot gtwhiogreweawztzd4uhrzgugtwzfhxtzdw	positif

Gambar 7. Tahap Filtering

Gambar 7 kemudian melakukan tahap *Filtering* yang bertujuan untuk menghilangkan spasi, spasi ganda, angka dan karakter selain huruf.

```
In [7]: from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
# Define the stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

# Apply stemming to the 'Review' column
df['Review'] = df['Review'].apply(stemmer.stem)

# Display the first 5 rows of the resulting data frame
df.head()
```

Out[7]:

	Rating	Review	label sentimen
0	1	tidak bisa masukp	negatif
1	5	suka	positif
2	1	biasany bisa cepetinikoklambat	negatif
3	1	sulit di update	negatif
4	4	zozot gtwhicgreweawztzdxuhrzgugtwzfhxtzdwd	positif

Gambar 8. Tahap *Stemmer*

Gambar 8 adalah tahap melakukan *stemmer* atau merubah kata menjadi bentuk asal.

```
In [8]: #Stopword Removal

from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory

stopword_factory = StopWordRemoverFactory()

stopword_remover = stopword_factory.create_stop_word_remover()

df['Review'] = df['Review'].apply(lambda x: stopword_remover.remove(x))

df.head(100)
```

Out[8]:

	Rating	Review	label sentimen
0	1	masukp	negatif
1	5	suka	positif
2	1	biasany cepetinikoklambat	negatif
3	1	sulit update	negatif
4	4	zozot gtwhicgreweawztzdxuhrzgugtwzfhxtzdwd	positif
...
95	5	wueuey	positif
96	1	mending pakai zoom	negatif
97	4	mudah tida tele tele praktis ganggu ekonomis	positif
98	5	okk	positif
99	5	bgusdsss	positif

100 rows x 3 columns

Gambar 9. Tahap *Stopword Removal*

Pada Gambar 9 merupakan tahap *Stopword Removal* atau menghilangkan kata yang tidak penting seperti kata hubung.

```
In [9]: import nltk
from nltk.tokenize import word_tokenize

df['Review_tokenized'] = df['Review'].apply(lambda x: word_tokenize(x))
df.head()
```

Out[9]:

	Rating	Review	label sentimen	Review_tokenized
0	1	masukp	negatif	[masukp]
1	5	suka	positif	[suka]
2	1	biasany cepetinikoklambat	negatif	[biasany, cepetinikoklambat]
3	1	sulit update	negatif	[sulit, update]
4	4	zozot gtwhiogreawawztzdxuhrzgugtwzfhxtzdw	positif	[zozot, gtwhiogreawawztzdxuhrzgugtwzfhxtzdw]

Gambar 10. Tahap tokenisasi

Berdasarkan Gambar 10 langkah selanjutnya merupakan tahap tokenisasi dengan menggunakan librari NLTK (*Natural Language Toolkit*).

- TF-IDF

```
In [10]: from sklearn.feature_extraction.text import TfidfVectorizer

# Create an instance of the TfidfVectorizer class
tfidf_vectorizer = TfidfVectorizer()

# Fit the vectorizer to the 'Review' column of the DataFrame 'df'
tfidf_matrix = tfidf_vectorizer.fit_transform(df['Review'])

# Convert the resulting sparse matrix to a dense matrix
tfidf_dense = tfidf_matrix.todense()

# Convert the dense matrix to a DataFrame and assign column names
tfidf_df = pd.DataFrame(tfidf_dense, columns=tfidf_vectorizer.get_feature_names())

# Concatenate the original DataFrame 'df' with the new TF-IDF DataFrame 'tfidf_df'
df_tfidf = pd.concat([df, tfidf_df], axis=1)
df_tfidf.head(20)
```

C:\Users\HP\anaconda3\lib\site-packages\sklearn\utils\deprecation.py:87: FutureWarning: Function get_feature_names is deprecated; get_feature_names is deprecated in 1.0 and will be removed in 1.2. Please use get_feature_names_out instead. warnings.warn(msg, category=FutureWarning)

Out[10]:

	Rating	Review	label sentimen	Review_tokenized	aa	aaa	aaaa	aaaaaaa	aaaaaaaa	aaaaaaaaa	...	yoyok
0	1	masukp	negatif	[masukp]	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
1	5	suka	positif	[suka]	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
2	1	biasany cepetinikoklambat	negatif	[biasany, cepetinikoklambat]	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
3	1	sulit update	negatif	[sulit, update]	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
4	4	zozot gtwhiogreawawztzdxuhrzgugtwzfhxtzdw	positif	[zozot, gtwhiogreawawztzdxuhrzgugtwzfhxtzdw]	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
5	5	nice	positif	[nice]	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
6	1	nih	negatif	[nih]	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
7	5	ok	positif	[ok]	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
8	1	buka	negatif	[buka]	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
9	5	bagus	positif	[bagus]	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0

Gambar 11. Perhitungan TF-IDF

Gambar 11 Setelah melakukan preprocessing dilakukan perhitungan TF-IDF dengan menggunakan librari Tfidfvectorizer.

- Penerapan Metode *Logistic Regression*

```
In [17]: import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report

# Define the target variable
df['sentiment'] = df['Rating'].apply(lambda x: 'positive' if x > 3 else 'negative' if x < 3 else 'neutral')

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(df['Review'], df['sentiment'], test_size=0.2, random_state=42)

# Create a TfidfVectorizer object
tfidf = TfidfVectorizer()

# Fit and transform the training data
X_train_tfidf = tfidf.fit_transform(X_train)

# Transform the testing data
X_test_tfidf = tfidf.transform(X_test)

# Create a Logistic Regression model and fit the training data
lr = LogisticRegression()
lr.fit(X_train_tfidf, y_train)

# Predict the sentiment of the testing data
y_pred = lr.predict(X_test_tfidf)

# Print the classification report
print(classification_report(y_test, y_pred, digits=4, target_names=['negative', 'neutral', 'positive']))
```

	precision	recall	f1-score	support
negative	1.0000	0.1667	0.2857	24
neutral	0.0000	0.0000	0.0000	10
positive	0.7902	1.0000	0.8828	113
accuracy			0.7959	147
macro avg	0.5967	0.3889	0.3895	147
weighted avg	0.7707	0.7959	0.7253	147

Gambar 12. Penerapan metode *logistic regression*

Setelah itu dilakukan penerapan *Metode Logistic regression* dengan kesimpulan :

- ✓ *'negative'* memiliki precision 100%, recall 16.67%, dan f1-score 28.57%. Hal ini menunjukkan bahwa model memiliki tingkat akurasi yang tinggi dalam memprediksi data yang termasuk dalam kelas *'negative'*, namun recall-nya rendah. Artinya, model cenderung mengabaikan beberapa data yang seharusnya termasuk dalam kelas *'negative'*.
- ✓ *'neutral'* memiliki precision 0%, recall 0%, dan f1-score 0%. Hal ini menunjukkan bahwa model tidak dapat memprediksi data yang termasuk dalam kelas *'neutral'*.
- ✓ *'positive'* memiliki precision 79.02%, recall 100%, dan f1-score 88.28%. Hal ini menunjukkan bahwa model memiliki tingkat akurasi yang cukup baik dalam memprediksi data yang termasuk dalam kelas *'positive'*, dengan recall 100% artinya model dapat mengklasifikasikan semua data yang seharusnya termasuk dalam kelas *'positive'*.
- ✓ Dari nilai *accuracy* sebesar 79.59%, model dapat memprediksi dengan akurasi sekitar 79.59%.
- ✓ Nilai *macro avg* menunjukkan rata-rata precision, recall, dan f1-score dari semua label kelas yang ada di dataset. Rata-rata precision sebesar 59.67%, recall sebesar 38.89%, dan f1-score sebesar 38.95%.
- ✓ Nilai *weighted avg* juga menunjukkan rata-rata precision, recall, dan f1-score, namun mempertimbangkan bobot dari masing-masing kelas yang ada di dataset. Rata-rata precision sebesar 77.07%, recall sebesar 79.59%, dan f1-score sebesar 72.53%.

telah memberikan wadah dalam penelitian ini. Medika Risnasari, S.ST., M.T selaku Koordinator Prodi Pendidikan Informatika yang telah bekerja sama dalam proses publikasi penelitian ini. Laili Cahyani, S.Kom., M.Kom selaku dosen pengampu mata kuliah text mining yang telah memberikan arahan, saran dan masukan dalam penyempurnaan penelitian ini.

DAFTAR PUSTAKA

- Agustina, L., Fayardi, A. O., & Irwansyah, I. (2018). *Online Review: Indikator Penilaian Kredibilitas Online dalam Platform E-commerce. Jurnal ILMU KOMUNIKASI*, 15(2), 141–154. <https://doi.org/10.24002/jik.v15i2.1320>
- Hermanto, H., & Noviriandini, A. (2021). Analisa Sentimen Terhadap Belajar Online Pada Masa Covid-19 Menggunakan Algoritma Support Vector Machine Berbasis Particle Sarm Optimization. *Jurnal Informatika Kaputama (JIK)*, 5(1), 129–136. <https://jurnal.kaputama.ac.id/index.php/JIK/article/view/451>
- Ramli, Yuniarti, D., & Goejantoro, R. (2013). Perbandingan Metode Klasifikasi Regresi Logistik Dengan Jaringan Saraf Tiruan (Studi Kasus : Pemilihan Jurusan Bahasa dan IPS pada SMAN 2 Samarinda Tahun Ajaran Comparison of Classification Methods Between Logistic Regression and Artificial Neural Networ. *Jurnal Eksponensial*, 4(1), 17–24.
- Septian, J. A., Fachrudin, T. M., & Nugroho, A. (2019). Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TF-IDF dan K-Nearest Neighbor. *Journal of Intelligent System and Computation*, 1(1), 43–49. <https://doi.org/10.52985/insyst.v1i1.36>
- Sodik, F., & Kharisudin, I. (2021). Analisis Sentimen dengan SVM , NAIVE BAYES dan KNN untuk Studi Tanggapan Masyarakat Indonesia Terhadap Pandemi Covid-19 pada Media Sosial Twitter. *Prisma*, 4, 628–634.
- Widyasanti, N. K., Darma Putra, I. K. G., & Dwi Rusjyanthi, N. K. (2018). Seleksi Fitur Bobot Kata dengan Metode TFIDF untuk Ringkasan Bahasa Indonesia. *Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi Informasi)*, 6(2), 119. <https://doi.org/10.24843/jim.2018.v06.i02.p06>