

# ANALISIS SENTIMEN ULASAN PENGGUNA APLIKASI DISCORD MENGGUNAKAN METODE K-NEAREST NEIGHBOR

Reva Qintara Rohmansa<sup>1)</sup>, Nunik Pratiwi\*<sup>2)</sup>, Muhammad Jeral Palepa<sup>3)</sup>

1. Universitas Muhammadiyah Prof. Dr. Hamka, Indonesia
2. Universitas Muhammadiyah Prof. Dr. Hamka, Indonesia
3. Universitas Muhammadiyah Prof. Dr. Hamka, Indonesia

## Article Info

**Kata Kunci:** Discord; Google Play Store; K-Nearest Neighbor; Sentimen

**Keywords:** *Discord; Google Play Store; K-Nearest Neighbor, Sentiment*

## Article history:

Received 4 December 2023

Revised 18 December 2023

Accepted 1 January 2024

Available online 1 March 2024

## DOI :

<https://doi.org/10.29100/jipi.v9i1.4943>

\* Corresponding author.

Nunik Pratiwi

E-mail address:

[npratiwi@uhamka.ac.id](mailto:npratiwi@uhamka.ac.id)

## ABSTRAK

Pada masa pandemi *COVID-19*, aplikasi *Discord* menjadi semakin populer karena dapat digunakan untuk bekerja dan berkomunikasi jarak jauh. Popularitas ini dibuktikan dengan meningkatnya jumlah unduhan aplikasi *Discord* di *Google Play Store*. Pada bulan Juni 2020, *Discord* mencatat 100 juta pengguna aktif per bulan, dan jumlah tersebut terus bertambah hingga saat ini. Selain itu, aplikasi *Discord* juga mendapatkan peringkat 4,1 di *Google Play Store*, yang menunjukkan bahwa aplikasi ini dinilai baik oleh pengguna. Penelitian ini dilakukan untuk menganalisis sentimen terhadap aplikasi *Discord*. Penelitian ini mengumpulkan 2.000 *dataset* ulasan pengguna, kemudian melakukan tahap *preprocessing* dan pelabelan, termasuk penghapusan data duplikat, sehingga tersisa 1.807 *dataset*. Setelah itu, peneliti mengimplementasikan algoritma *K-Nearest Neighbor* untuk melakukan uji evaluasi dan mengklasifikasikan sentimen positif dan negatif berdasarkan data latih dan data uji yang sudah ditentukan. Pengujian menggunakan teknik *k-5 fold*, dan menghasilkan akurasi rata-rata sebesar 82,76%.

## ABSTRACT

During the *COVID-19* pandemic, the *Discord* application became increasingly popular because it could be used for remote work and communication. This popularity is evidenced by the increasing number of *Discord* app downloads on the *Google Play Store*. In June 2020, *Discord* recorded 100 million active users per month, and that number continues to grow to this day. In addition, the *Discord* app also received a rating of 4.1 on the *Google Play Store*, which indicates that the app is well-rated by users. This study was conducted to analyze the sentiment towards the *Discord* application. The study collected 2,000 datasets of user reviews, then conducted preprocessing and labeling, including the removal of duplicate data, leaving 1,807 datasets. After that, the researcher implemented the *K-Nearest Neighbor* algorithm to perform the evaluation test and classify positive and negative sentiment based on the predetermined training and test data. The testing used the *k-5 fold* technique, and resulted in an average accuracy of 82.76%.

## I. PENDAHULUAN

TEKNOLOGI berkembang dengan cepat seiring dengan meningkatnya tuntutan hidup dan mobilitas kerja manusia yang terus meningkat. Kemajuan yang paling signifikan terjadi di bidang teknologi komunikasi [1]. Menerapkan teknologi berbasis *mobile* atau *remote* dalam operasional sehari-hari sangat penting untuk mencapai efisiensi dari waktu ke waktu tanpa mengorbankan komponen penting dari operasional tersebut [2]. Teknologi komunikasi secara signifikan mendukung perusahaan dalam menghadapi persaingan bisnis yang terus meningkat. Komunikasi melibatkan proses interaksi individu untuk mencapai tujuan yang disepakati. Sinergi antara teknologi dan komunikasi secara signifikan memengaruhi perkembangan sumber daya manusia kontemporer. Banyaknya layanan aplikasi komunikasi jarak jauh memberikan banyak pilihan bagi konsumen, memberikan manfaat dengan ketersediaan layanan yang beragam seperti *Discord*, *Google Meet*, *Skype*, *Zoom*, dan lainnya [3]. Awalnya *Discord* merupakan aplikasi yang sering digunakan oleh para *gamer* sebagai platform komunikasi, namun seiring berjalannya waktu, *Discord* telah memperluas basis penggunaannya untuk mencakup beragam komunitas besar [4]. Aplikasi *Discord* memiliki berbagai fasilitas komunikasi, salah satu fitur yang mencolok adalah fungsi komunikasi yang mirip dengan telepon, memungkinkan pengguna untuk menyampaikan

konten secara lancar kepada pengguna lainnya. Fitur lain yang menonjol adalah kemampuan *share screen* video, memungkinkan pengguna dan pengguna lainnya berkomunikasi melalui interaktif video. Selain itu, dalam fitur *share screen* ini, pengguna dapat membagikan layar mereka, mirip dengan fitur yang dimiliki oleh aplikasi konferensi video [5]. Fitur unggulan lain yang dimiliki oleh *Discord* adalah kemampuannya membuat *channel* atau ruang baru untuk membuat dan memulai diskusi sesuai topik dengan nama *channel* tersebut. Saat ini, *Discord* sangat populer dan berkembang dengan waktu yang relatif cepat, memiliki basis pengguna yang melebihi 100 juta. *Discord* dapat digunakan di berbagai platform, termasuk *Android*, *iOS*, *Linux*, *Mac*, dan *Windows* [6].

Analisis sentimen adalah metode untuk memahami emosi yang terkandung dalam teks. Metode ini dapat digunakan untuk secara otomatis mengidentifikasi sentimen positif, negatif, atau netral dari suatu opini [7]. Ada berbagai metode analisis sentimen yang tersedia, peneliti menggunakan metode *K-Nearest Neighbor*. *K-Nearest Neighbor* merupakan pendekatan *machine learning* yang terkenal dan sering digunakan dalam klasifikasi data [8]. Algoritma ini bekerja dengan menghitung jarak dari data baru ke data pelatihan, lalu mengklasifikasikannya ke kelas yang diwakili oleh mayoritas tetangga terdekatnya sejumlah  $k$  [9]. Dalam konteks analisis sentimen, objek yang serupa adalah objek yang memiliki sentimen yang sama. Oleh karena itu, *K-Nearest Neighbor* dapat digunakan untuk mengklasifikasikan sentimen dari sebuah teks dengan cara mencari  $k$  tetangga terdekat dari teks tersebut.  $K$  tetangga terdekat adalah objek-objek yang memiliki sentimen yang paling mirip dengan teks tersebut.

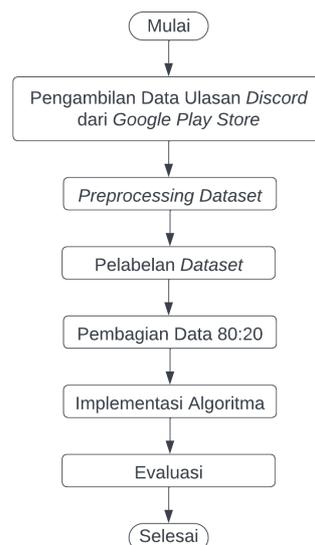
Penelitian yang dilakukan oleh Rizki dan Gilang terkait analisis sentimen terhadap aplikasi *Grab* dengan menggunakan metode *Support Vector Machine*, mereka menggunakan 1.000 ulasan dan menghasilkan akurasi sebesar 85,54% [10]. Penelitian lain yang dilakukan oleh Restu, dkk. terkait analisis sentimen terhadap aplikasi *Maxim* dengan menggunakan metode *K-Nearest Neighbor*, mereka menggunakan 10.000 ulasan dan menghasilkan akurasi sebesar 90,23% dengan menggunakan nilai  $k = 5$  [11].

Penelitian ini secara khusus berfokus pada analisis sentimen pengguna *Discord* dengan menggunakan data ulasan *Discord* untuk menganalisis sentimen penggunanya. *Discord* dapat menggunakan wawasan ini untuk meningkatkan platformnya dan memberikan pengalaman pengguna yang lebih baik. Penelitian ini mempertimbangkan aspek-aspek yang penting bagi pengguna, seperti kemudahan pengguna, dan kepuasan pengguna.

## II. METODE PENELITIAN

### A. Tahapan Penelitian

Peneliti melakukan serangkaian upaya untuk mencapai tujuan penelitian, dimulai dari mengumpulkan data, kemudian mengolah data dengan algoritma *K-Nearest Neighbor*, dan mengevaluasi hasil pengolahan data dengan *confusion matrix*. Gambar 1 memvisualisasikan alur tahapan penelitian.



Gambar. 1. Tahapan Penelitian

### B. Pengambilan Data

Data ulasan *Discord* diperoleh dengan menggunakan teknik *web scraping* dari *Google Play Store*. Data yang diambil adalah ulasan yang ditulis dalam bahasa Indonesia dan diunggah mulai dari September 2018 hingga Oktober 2023. Total data yang diambil adalah 2000 ulasan dan disimpan dalam format CSV. Kriteria pemilihan

data tersebut dipilih untuk memastikan bahwa data yang diambil dapat dianalisis dengan mudah dan memberikan informasi yang berharga. Gambar 2 merupakan ilustrasi diagram alir dari proses pengambilan data.



Gambar. 2. Ilustrasi Pengambilan Data

### C. Preprocessing Dataset

*Preprocessing dataset* bertujuan menghilangkan kata-kata yang tidak bermakna. Proses pembersihan membuat data lebih teragregasi, memudahkan penggunaannya dalam tahap berikutnya [12]. Setelah kumpulan data diperoleh, teks akan diurai menjadi bagian-bagian yang lebih kecil melalui proses tokenisasi. Tahap ini dilakukan untuk memastikan kelancaran proses saat melakukan analisis data [13]. Selama tahap ini berlangsung, berbagai langkah dilaksanakan dalam *preprocessing dataset*, yang mencakup :

- 1) *Case folding* dan *cleaning*, bertujuan untuk menyatukan karakter dan membersihkan teks dari karakter, simbol, angka, emotikon, dan tautan dalam kumpulan data. Selama proses ini berlangsung, melibatkan transformasi semua huruf menjadi huruf kecil, sehingga karakter 'A' hingga 'Z' dalam data diubah menjadi 'a' hingga 'z'.
- 2) *Tokenizing*, bertujuan untuk memisahkan sekelompok karakter dalam sebuah teks menjadi kata-kata individual.
- 3) *Filtering*, yaitu langkah yang dilakukan setelah *tokenizing*. Tujuan *filtering* adalah untuk mengekstrak kata-kata penting dari hasil proses sebelumnya. Selama tahap *filtering*, keputusan akan diambil mengenai apakah kata-kata lama akan dipertahankan atau dibuang.
- 4) *Stopwords removal*, yaitu menghilangkan kata-kata yang tidak bermakna atau tidak esensial dalam sebuah kalimat, dengan panduan daftar *stopword*.
- 5) *Stemming*, bertujuan mengubah *dataset* yang mengandung imbuhan menjadi kata dasar.

### D. Pelabelan Dataset

Pelabelan *dataset* diperlukan untuk memberikan label sentimen pada setiap *dataset* yang sudah diperoleh, dan label-label ini memiliki peran penting dalam menilai akurasi pada langkah-langkah selanjutnya. Proses pelabelan dilakukan dengan menggunakan modul *vader lexicon*. Modul ini memiliki daftar kata-kata yang memiliki skor sentimen positif dan negatif. Proses pelabelan dimulai dengan pembobotan setiap kata dalam ulasan. Pembobotan dilakukan berdasarkan skor sentimen dari kata tersebut dalam *vader lexicon*. Skor sentimen kata-kata dalam *vader lexicon* berkisar antara -1 hingga 1. Setelah bobot setiap kata diperoleh, maka dilakukan penjumlahan bobot dari setiap kata. Jumlah bobot ini kemudian digunakan untuk menentukan sentimen dari keseluruhan ulasan. Sentimen ulasan ditentukan berdasarkan jumlah bobot positif dan negatif. Ulasan dengan jumlah bobot positif lebih besar dari jumlah bobot negatif akan diberi label positif. Sedangkan ulasan dengan jumlah bobot negatif lebih besar dari jumlah bobot positif akan diberi label negatif.

### E. Implementasi Algoritma

Algoritma *K-Nearest Neighbor* merupakan metode *machine learning* yang dapat digunakan untuk mengklasifikasikan dan memprediksi data. Algoritma ini bekerja dengan cara mencari *k* tetangga terdekat dari objek yang akan diklasifikasikan atau diprediksi [14]. Dalam konteks penelitian ini, metode *machine learning K-Nearest Neighbor* digunakan untuk mengklasifikasikan sentimen terhadap ulasan *Discord*. Algoritma ini diimplementasikan dengan menggunakan bahasa *Python* dan *library scikit-learn*. Model algoritma *K-Nearest Neighbor* memiliki beberapa tahap kerja, yaitu:

- 1) Pertama, menentukan parameter *k* yang akan digunakan untuk membuat klasifikasi. Nilai *k* yang dipilih dapat mempengaruhi akurasi klasifikasi. Nilai *k* yang terlalu kecil dapat menyebabkan *overfitting*, sedangkan nilai *k* yang terlalu besar dapat menyebabkan *underfitting*.
- 2) Kedua, melakukan pendekatan dengan *euclidean distance* yang merupakan salah satu cara untuk mengukur seberapa dekat data baru dengan data yang sudah ada. *Euclidean distance* merupakan metode pengukuran jarak yang sederhana dan mudah diterapkan, serta memiliki makna yang jelas. Rumus untuk menghitung *euclidean distance* adalah sebagai berikut.

$$X = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_n - b_n)^2} \quad (1)$$

Keterangan:  
 a = Data latih  
 b = Data baru

3) Ketiga, menentukan  $k$  tetangga terdekat untuk digunakan dalam proses prediksi [15].

#### F. Evaluasi

Nilai akurasi dapat digunakan untuk menilai hasil penelitian. Evaluasi kinerja algoritma dapat dilakukan dengan menggunakan *confusion matrix*. *Confusion matrix* adalah metode evaluasi performa algoritma yang menghasilkan nilai akurasi. Rumus *confusion matrix* adalah sebagai berikut.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

### III. HASIL DAN PEMBAHASAN

#### A. Pengambilan Data

Sebelum memulai proses penelitian, peneliti mengimpor *library* yang diperlukan. Persiapan ini memastikan bahwa setelah data berhasil dikumpulkan, data tersebut dapat diatur menjadi sebuah *dataset*. Tujuan utamanya adalah untuk memudahkan pelaksanaan proses dan sesuai dengan algoritma. Proses pengambilan data dilakukan dalam bahasa *python* di *Google Colaboratory* dengan menggunakan teknik *web scraping* melalui domain aplikasi *Discord*, yaitu '*com.discord*'. Langkah selanjutnya, melakukan pengambilan data ulasan secara acak mulai dari bulan September 2018 hingga Oktober 2023, berdasarkan kueri tertentu yang sesuai dengan topik penelitian. Jumlah maksimum data yang akan dikumpulkan dalam proses ini diatur sebanyak 2000 data. Kemudian, data tersebut diubah menjadi sebuah *dataset* menggunakan *library pandas*.

```
[ ] #Library
from google_play_scraper import app
from google_play_scraper import Sort, reviews
import numpy as np
import pandas as pd
```

Gambar. 3. Import Library

#### B. Preprocessing Dataset

##### 1) Case Folding dan Cleaning

*Case folding* proses untuk menormalisasi huruf dalam teks, yaitu mengubah semua huruf menjadi huruf kecil. Proses *cleaning* juga dilakukan secara bersamaan untuk menghapus emotikon, simbol, dan karakter lainnya yang dianggap tidak penting. Proses *case folding* dan *cleaning* dapat ditemukan di TABEL I.

TABEL I  
 PROSES CASE FOLDING DAN CLEANING

No.	content	clean_content
1	Bagus bangt aku suka sama aplikasi ini kasih bintang 5 ya ☺	bagus bangt aku suka sama aplikasi ini kasih bintang ya
2	Apk ini bagus banget,cocok buat mabar bareng temen	apk ini bagus banget cocok buat mabar bareng temen
3	aplikasi yang bagus, memiliki fitur nya yang banyak, dan bisa juga sebagai media seperti komunitas dan game.	aplikasi yang bagus memiliki fitur nya yang banyak dan bisa juga sebagai media seperti komunitas dan game

## 2) Tokenization

Tokenisasi (*tokenization*) bertujuan untuk membagi sekelompok karakter dalam sebuah teks menjadi kata-kata yang berdiri sendiri. Proses tokenisasi yang dimaksud dapat ditemukan pada TABEL II.

TABEL II  
 PROSES *TOKENIZING*

No.	<i>clean_content</i>	<i>Tokenizing</i>
1	bagus banget aku suka sama aplikasi ini kasih bintang ya	['bagus', 'banget', 'aku', 'suka', 'sama', 'aplikasi', 'ini', 'kasih', 'bintang', 'ya']
2	apk ini bagus banget cocok buat mabar bareng temen	['apk', 'ini', 'bagus', 'banget', 'cocok', 'buat', 'mabar', 'bareng', 'temen']
3	aplikasi yang bagus memiliki fitur nya yang banyak dan bisa juga sebagai media seperti komunitas dan game	['aplikasi', 'yang', 'bagus', 'memiliki', 'fitur', 'nya', 'yang', 'banyak', 'dan', 'bisa', 'juga', 'sebagai', 'media', 'seper-ti', 'komunitas', 'dan', 'game']

## 3) Normalization

Normalisasi (*normalization*) bertujuan untuk mengekstrak kata-kata penting dari data yang telah diproses sebelumnya. Pada tahap ini, keputusan akan diambil untuk mempertahankan atau membuang kata-kata lama. Proses normalisasi yang dimaksud dapat ditemukan pada TABEL III.

TABEL III  
 PROSES *NORMALIZATION*

No.	<i>Tokenizing</i>	<i>Filtering</i>
1	['bagus', 'banget', 'aku', 'suka', 'sama', 'aplikasi', 'ini', 'kasih', 'bintang', 'ya']	['bagus', 'banget', 'aku', 'suka', 'sama', 'aplikasi', 'ini', 'kasih', 'bintang', 'ya']
2	['apk', 'ini', 'bagus', 'banget', 'cocok', 'buat', 'mabar', 'bareng', 'temen']	['apk', 'ini', 'bagus', 'banget', 'cocok', 'buat', 'mabar', 'bareng', 'teman']
3	['aplikasi', 'yang', 'bagus', 'memiliki', 'fitur', 'nya', 'yang', 'banyak', 'dan', 'bisa', 'juga', 'sebagai', 'media', 'seper-ti', 'komunitas', 'dan', 'game']	['aplikasi', 'yang', 'bagus', 'memiliki', 'fitur', 'nya', 'yang', 'banyak', 'dan', 'bisa', 'juga', 'sebagai', 'media', 'seper-ti', 'komunitas', 'dan', 'game']

## 4) Stopwords Removal

Proses penghapusan kata-kata yang tidak penting dalam kalimat disebut *stopwords removal*. Proses ini dilakukan dengan menggunakan daftar *stopwords*, yaitu daftar kata-kata yang dianggap tidak penting dalam sebuah kalimat. Selain menghapus kata-kata yang tidak penting, *stopwords removal* juga mengubah huruf kapital menjadi huruf kecil dalam teks. Proses *stopwords removal* yang dimaksud dapat ditemukan pada TABEL IV.

TABEL IV  
 PROSES STOPWORDS REMOVAL

No.	Filtering	Stopwords Removal
1	['bagus', 'banget', 'aku', 'suka', 'sama', 'aplikasi', 'ini', 'kasih', 'bintang', 'ya']	['bagus', 'banget', 'suka', 'aplikasi', 'kasih', 'bintang', 'ya']
2	['apk', 'ini', 'bagus', 'banget', 'cocok', 'buat', 'mabar', 'bareng', 'teman']	['apk', 'bagus', 'banget', 'cocok', 'mabar', 'bareng', 'teman']
3	['aplikasi', 'yang', 'bagus', 'memiliki', 'fitur', 'nya', 'yang', 'banyak', 'dan', 'bisa', 'juga', 'sebagai', 'media', 'seper-ti', 'komunitas', 'dan', 'game']	['aplikasi', 'bagus', 'memiliki', 'fitur', 'nya', 'media', 'komunitas', 'game']

### 5) Stemming

*Stemming* bertujuan untuk menormalkan representasi kata dalam *dataset*, sehingga mempermudah algoritma untuk mengidentifikasi pola dan hubungan antar kata. Proses *stemming* yang dimaksud dapat ditemukan pada TABEL V.

TABEL V  
 PROSES STEMMING

No.	Stopwords Removal	Stemming
1	['bagus', 'banget', 'suka', 'aplikasi', 'kasih', 'bintang', 'ya']	['bagus', 'banget', 'suka', 'aplikasi', 'kasih', 'bintang', 'ya']
2	['apk', 'bagus', 'banget', 'cocok', 'mabar', 'bareng', 'teman']	['apk', 'bagus', 'banget', 'cocok', 'mabar', 'bareng', 'teman']
3	['aplikasi', 'bagus', 'memiliki', 'fitur', 'nya', 'media', 'komunitas', 'game']	['aplikasi', 'bagus', 'milik', 'fitur', 'nya', 'media', 'komunitas', 'game']

### C. Pelabelan Dataset

#### 1) Translate

Proses *translate* menggunakan *library googletrans* untuk menerjemahkan teks dari bahasa Indonesia ke bahasa Inggris, karena *library TextBlob* hanya dapat mengenali Bahasa Inggris.

#### 2) Klasifikasi Sentimen

Setelah mendapatkan hasil terjemahan dalam Bahasa Inggris, langkah berikutnya adalah menentukan klasifikasi untuk mendapatkan skor polaritas.

```
def convert (polarity) :
    if polarity == 'Positif' :
        return 1
    else :
        return -1

df ['Polarity'] = df ['Sentiment'].apply (convert)
df.head()
```

Gambar. 4. Proses Klasifikasi Untuk Mencari Polaritas

TABEL VI  
 ULASAN POSITIF DAN NEGATIF

No.	Ulasan Positif	Ulasan Negatif
1	Saya suka dengan aplikasi ini	Ga bisa verifikasi email notif ya ga muncul dan dari dulu ga bisa masuk discord delvover tolong atasi bug ya secepatnya karena saya ingin sekali masuk komunitas discord tapi terhalang oleh bugg !!!!!
2	Aplikasi keren mantap dah pokok nya,, semoga saya bisa win di event game mlbb terus 🤔👍	Udah bagus, tapi akhir2 ini banyak banget orang yang kena hack akunnnya padahal udah pake 2FA, tolong tingkatkan lagi sistem keamanannya, kasian kan orang yg jualan, komunikasi dan lain sebagainya kena hack. Minta tolong di verifikasi ulang kalo misal login ke perangkat lain / perangkat yang berbeda 🤔
3	Discord Sangatlah berguna karena discord bisa membuat kita menemukan orang orang menarik	Gak bisa login lagi padahal semua dah bener email ama password nya semua dah bener tapi di bilang salah dah gandi sandi malah sama aja hasilnya ini gimana si dari pihak discord tolong ini kenapa jangan cuman di baca aja ya langsung carik kesalahan sistem nya 🤔👍

Berdasarkan klasifikasi yang sudah dilakukan, peneliti mendapatkan total 1519 sentimen positif dan 288 sentimen negatif.

### 3) Implementasi Algoritma

Pemisahan data untuk klasifikasi dilakukan dengan membagi 1807 data menjadi 80% sebagai data latih (1445 data) dan 20% sebagai data uji (362 data). Pengujian dilakukan sebanyak 5 *fold*. Nilai *k* adalah parameter dalam algoritma *K-Nearest Neighbor* yang menentukan jumlah tetangga terdekat yang akan dipertimbangkan. Nilai *k* yang dipilih dapat mempengaruhi akurasi klasifikasi. Nilai *k* yang terlalu kecil dapat menyebabkan *overfitting*, yaitu model terlalu pas dengan data pelatihan sehingga tidak dapat generalisasi dengan baik ke data uji. Sedangkan nilai *k* yang terlalu besar dapat menyebabkan *underfitting*, yaitu model tidak cukup pas dengan data pelatihan sehingga tidak dapat menangkap pola data dengan baik. Dalam penelitian ini, nilai *k* yang dipilih adalah 5, karena nilai *k*=5 memiliki akurasi yang baik [16]. Tahap ini dimulai dengan persiapan data, yaitu pemilihan data yang sesuai dan validasi variabel yang tersedia, dilanjutkan dengan persiapan perubahan data [17].

```
x_train, x_test, y_train, y_test = train_test_split (X, y, test_size = 0.2)
```

Gambar. 5. Proses Pembagian Antara Data Latih dan Data Uji di 80:20

```
n_neighbors = 5
knn_tfidf = KNeighborsClassifier (n_neighbors, weights = 'distance', metric= 'euclidean')
knn_tfidf.fit (X_train_vect, y_train)
y_predict = knn_tfidf.predict(X_test_vect)

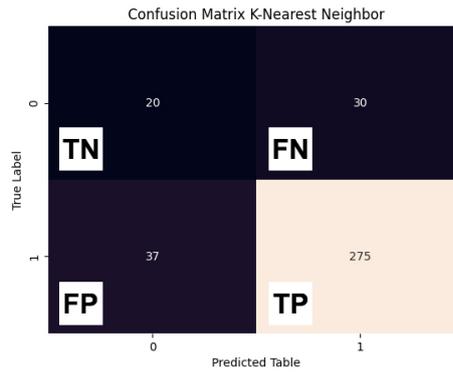
print (classification_report (y_test, y_predict))
print ('Confusion Matrix:', confusion_matrix (y_test, y_predict))
```

Gambar. 6. Proses Pengujian Menggunakan Algoritma *K-Nearest Neighbor*

### D. Evaluasi

#### 1) Pengujian K-1

Pengujian *k*-1 terhadap 362 data uji menghasilkan akurasi 81,49%, diilustrasikan pada Gambar 7.



Gambar. 7. Visualisasi *Confusion Matrix K-1*

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

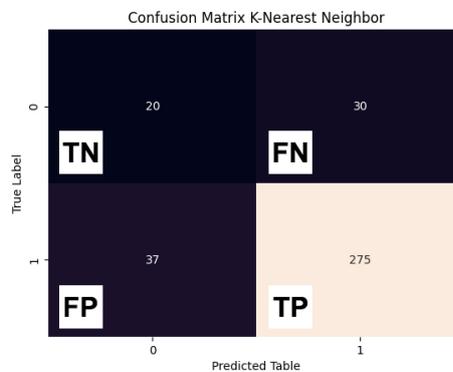
$$Accuracy = \frac{275 + 20}{275 + 20 + 37 + 30} = 0,8149 = 81,49\% \quad (4)$$

Keterangan :

- TN : Dari 362 data, 20 data diprediksi negatif dan label sebenarnya negatif.
- TP : Dari 362 data, 275 data diprediksi positif dan label sebenarnya positif.
- FN : Dari 362 data, 30 data diprediksi negatif, tetapi label sebenarnya positif.
- FP : Dari 362 data, 37 data diprediksi positif, tetapi label sebenarnya negatif.

2) Pengujian K-2

Pengujian *k-2* terhadap 362 data uji menghasilkan akurasi 81,49%, diilustrasikan pada Gambar 8.



Gambar. 8. Visualisasi *Confusion Matrix K-2*

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

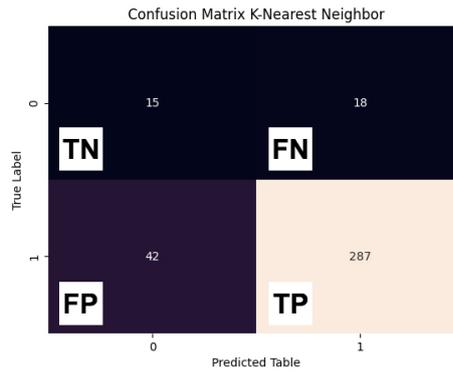
$$Accuracy = \frac{275 + 20}{275 + 20 + 37 + 30} = 0,8149 = 81,49\% \quad (6)$$

Keterangan :

- TN : Dari 362 data, 20 data diprediksi negatif dan label sebenarnya negatif.
- TP : Dari 362 data, 275 data diprediksi positif dan label sebenarnya positif.
- FN : Dari 362 data, 30 data diprediksi negatif, tetapi label sebenarnya positif.
- FP : Dari 362 data, 37 data diprediksi positif, tetapi label sebenarnya negatif.

3) Pengujian K-3

Pengujian *k-3* terhadap 362 data uji menghasilkan akurasi 83,42%, diilustrasikan pada Gambar 9.



Gambar. 9. Visualisasi *Confusion Matrix K-3*

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

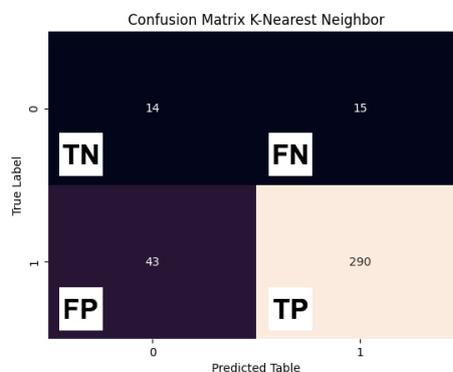
$$Accuracy = \frac{287 + 15}{287 + 15 + 42 + 18} = 0,8342 = 83,42\% \quad (8)$$

Keterangan :

- TN : Dari 362 data, 15 data diprediksi negatif dan label sebenarnya negatif.
- TP : Dari 362 data, 287 data diprediksi positif dan label sebenarnya positif.
- FN : Dari 362 data, 18 data diprediksi negatif, tetapi label sebenarnya positif.
- FP : Dari 362 data, 42 data diprediksi positif, tetapi label sebenarnya negatif.

#### 4) Pengujian K-4

Pengujian *k-4* terhadap 362 data uji menghasilkan akurasi 83,97%, diilustrasikan pada Gambar 10.



Gambar. 10. Visualisasi *Confusion Matrix K-4*

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

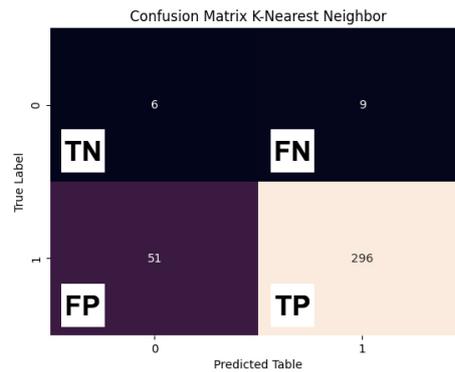
$$Accuracy = \frac{290 + 14}{290 + 14 + 43 + 15} = 0,8397 = 83,97\% \quad (10)$$

Keterangan :

- TN : Dari 362 data, 14 data diprediksi negatif dan label sebenarnya negatif.
- TP : Dari 362 data, 290 data diprediksi positif dan label sebenarnya positif.
- FN : Dari 362 data, 15 data diprediksi negatif, tetapi label sebenarnya positif.
- FP : Dari 362 data, 43 data diprediksi positif, tetapi label sebenarnya negatif.

#### 5) Pengujian K-5

Pengujian  $k$ -5 terhadap 362 data uji menghasilkan akurasi 83,42%, diilustrasikan pada Gambar 11.



Gambar. 11. Visualisasi *Confusion Matrix K-5*

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (11)$$

$$Accuracy = \frac{296 + 6}{296 + 6 + 51 + 9} = 0,8342 = 83,42\% \quad (12)$$

Keterangan :

- TN : Dari 362 data, 6 data diprediksi negatif dan label sebenarnya negatif.
- TP : Dari 362 data, 296 data diprediksi positif dan label sebenarnya positif.
- FN : Dari 362 data, 9 data diprediksi negatif, tetapi label sebenarnya positif.
- FP : Dari 362 data, 51 data diprediksi positif, tetapi label sebenarnya negatif.

Hasil akhir dari algoritma *K-Nearest Neighbor* melibatkan penggunaan tabel *confusion matrix*. Terdapat penelitian terdahulu tentang analisis sentimen terhadap aplikasi *Discord* menggunakan metode *Naïve Bayes Classifier* yang dilakukan oleh Syafrida, Betha dan Rini menunjukkan akurasi yang baik dari algoritma *Naïve Bayes* dengan angka tertinggi mencapai 84% [18].

Berdasarkan pengujian yang sudah dilakukan, peneliti melakukan perbandingan pengujian yang dilakukan oleh Syafrida, dkk. yang menggunakan *Naïve Bayes Classifier* dan pengujian peneliti yang menggunakan metode *K-Nearest Neighbor*. Pengujian yang dihasilkan oleh peneliti menghasilkan akurasi rata-rata mencapai 82,76%, *precision*: 85,30%, *recall*: 97.04%, dan *f1-score*: 90.79% dan nilai tertinggi ada pada  $k$ -4 yang akurasinya mencapai 83,97%, menunjukkan bahwa algoritma *K-Nearest Neighbor* telah mencapai akurasi yang baik. Oleh karena itu, algoritma *K-Nearest Neighbor* menjadi salah satu pilihan (alternatif) yang baik untuk mengklasifikasikan sentimen ulasan *Discord*. Namun, perlu diketahui juga untuk mendapatkan hasil pengujian yang optimal, algoritma *K-Nearest Neighbor* membutuhkan jumlah tetangga atau parameter yang tepat.

	precision	recall	f1-score	support
-1	0.40	0.11	0.17	57
1	0.85	0.97	0.91	305
accuracy			0.83	362
macro avg	0.63	0.54	0.54	362
weighted avg	0.78	0.83	0.79	362

Confusion Matrix: [[ 6 51]  
[ 9 296]]

Gambar. 12. Hasil Pengujian Evaluasi

#### IV. KESIMPULAN

Berdasarkan dari proses dan pengujian yang telah dilakukan, terlihat bahwa terdapat 1519 sentimen positif dan 288 sentimen negatif dari total 1807 ulasan, setelah menghapus data duplikat dari kumpulan 2000 ulasan dengan menghasilkan akurasi rata-rata sebesar 82,76%, *precision*: 85,30%, *recall*: 97.04%, dan *f1-score*: 90.79%. Akurasi dari setiap uji algoritma *K-Nearest Neighbor* adalah sebagai berikut; pengujian  $k$ -1: 81,49%, pengujian  $k$ -2: 81,49%, pengujian  $k$ -3: 83,42%, pengujian  $k$ -4: 83,97%, dan pengujian  $k$ -5: 83,42%. Oleh karena itu, ulasan aplikasi *Discord* sudah sangat baik, dengan adanya pernyataan tersebut calon pengguna aplikasi jarak jauh dapat mempertimbangkan untuk memilih *Discord* sebagai medianya, karena dengan hasil yang begitu baik, pengguna

tidak perlu khawatir akan apa yang ditawarkan *Discord*.

#### DAFTAR PUSTAKA

- [1] I. Artikel, "Sistem Informasi Akademik Berbasis Web Menggunakan Node JS , Express JS dan MongoDB," vol. 23, pp. 51–56, 2023.
- [2] M. Raffi, A. Suharso, and I. Maulana, "Analisis Sentimen Ulasan Aplikasi Binar Pada Google Play Store Menggunakan Algoritma Naïve Bayes," *INTECOMS J. Inf. Technol. Comput. Sci.*, vol. 6, no. 1, pp. 450–462, 2023, doi: 10.31539/intecom.v6i1.6117.
- [3] C. Kesuma *et al.*, "Rancang Bangun Sistem Informasi Penyewaan Mobil Berbasis Web," vol. 23, pp. 124–128, 2023.
- [4] E. Efriani, J. A. Dewantara, and A. Afandi, "Pemanfaatan Aplikasi Discord Sebagai Media Pembelajaran Online," *J. Teknol. Inf. dan Pendidik.*, vol. 13, no. 1, pp. 61–65, 2020, doi: 10.24036/tip.v13i1.283.
- [5] A. Rakhmawan *et al.*, "Analisis Pemanfaatan Aplikasi Discord Dalam Pembelajaran Daring Di Era Pandemi Covid-19," *Pros. Semin. Nas. Pendidik. FKIP*, vol. 3, no. 1, pp. 55–59, 2020.
- [6] M. R. Ridho, M. Muhaimin, and H. S. Harjono, "Pengaruh Aplikasi Discord Dalam Pembelajaran Daring Terhadap Hasil Belajar Pada Matakuliah Komputer," *J. Ilm. Bina Edukasi*, vol. 14, no. 1, pp. 22–35, 2021, doi: 10.33557/jedukasi.v14i1.1367.
- [7] F. V. Sari and A. Wibowo, "Analisis Sentimen Pelanggan Toko Online Jd.Id Menggunakan Metode Naïve Bayes Classifier Berbasis Konversi Ikon Emosi," *J. SIMETRIS*, vol. 10, no. 2, pp. 681–686, 2019.
- [8] Maruli Tua Silaen, "Klasifikasi Karakteristik Kepribadian Siswa Berdasarkan the Big Five Personality Dengan Menggunakan Metode K- Nearest Neighbor (Knn)," *J. Inform. dan Rekayasa Elektron.*, vol. 6, no. 1, pp. 121–129, 2023, doi: 10.36595/jire.v6i1.860.
- [9] Muflih Ihza Rifatama, Mohammad Reza Faisal, Rudy Herteno, Irwan Budiman, and Muhammad Itqan Mazdadi, "Optimasi Algoritma K-Nearest Neighbor Dengan Seleksi Fitur Menggunakan Xgboost," *J. Inform. dan Rekayasa Elektron.*, vol. 6, no. 1, pp. 64–72, 2023, doi: 10.36595/jire.v6i1.723.
- [10] R. Wahyudi and G. Kusumawardana, "Analisis Sentimen pada Aplikasi Grab di Google Play Store Menggunakan Support Vector Machine," *J. Inform.*, vol. 8, no. 2, pp. 200–207, 2021, doi: 10.31294/ji.v8i2.9681.
- [11] R. Ramadhan, M. Afdal, I. Permana, and M. Jazman, "Analisis Sentimen pada Ulasan Aplikasi Maxim di Google Play Store dengan K-Nearest Neighbor," *JURIKOM (Jurnal Ris. Komputer)*, vol. 10, no. 3, pp. 715–724, 2023, doi: 10.30865/jurikom.v10i3.6396.
- [12] T. A. Sari, E. Sinduningrum, and F. Noor Hasan, "KLIK: Kajian Ilmiah Informatika dan Komputer Analisis Sentimen Ulasan Pelanggan Pada Aplikasi Fore Coffee Menggunakan Metode Naïve Bayes," *Media Online*, vol. 3, no. 6, pp. 773–779, 2023, doi: 10.30865/klik.v3i6.884.
- [13] A. Wibowo, Firman Noor Hasan, Rika Nurhayati, and Arief Wibowo, "Analisis Sentimen Opini Masyarakat Terhadap Keefektifan Pembelajaran Daring Selama Pandemi COVID-19 Menggunakan Naïve Bayes Classifier," *J. Asimetrik J. Ilm. Rekayasa Inov.*, vol. 4, pp. 239–248, 2022, doi: 10.35814/asiimetrik.v4i1.3577.
- [14] J. Homepage, S. R. Cholil, T. Handayani, R. Prathivi, and T. Ardianita, "IJCIT (Indonesian Journal on Computer and Information Technology) Implementasi Algoritma Klasifikasi K-Nearest Neighbor (KNN) Untuk Klasifikasi Seleksi Penerima Beasiswa," *IJCIT (Indonesian J. Comput. Inf. Technol.)*, vol. 6, no. 2, pp. 118–127, 2021.
- [15] H. Said, N. H. Matondang, and H. N. Irmada, "Penerapan Algoritma K-Nearest Neighbor Untuk Memprediksi Kualitas Air Yang Dapat Dikonsumsi," *Techno.Com*, vol. 21, no. 2, pp. 256–267, 2022, doi: 10.33633/tc.v21i2.5901.
- [16] N. Soleha, D. Syaury, and E. Setiawan, "Sistem Deteksi dan Klasifikasi Pergerakan Kepala Menggunakan K-Nearest Neighbor," vol. 3, no. 10, pp. 9929–9934, 2019, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [17] L. A. Rahman Malik, M. Kamayani, and F. N. Hasan, "Faktor-Faktor Yang Mempengaruhi Minat Calon Mahasiswa Baru Mendaftar Pada Ftii Uhamka Menggunakan Algoritma K-Nearest Neighbor (K-Nn)," *Infotech J. Technol. Inf.*, vol. 9, no. 1, pp. 85–94, 2023, doi: 10.37365/jti.v9i1.163.
- [18] S. N. Salsabila, B. N. Sari, and R. Mayasari, "Klasifikasi Ulasan Pengguna Aplikasi Discord Menggunakan Metode Information Gain Dan Naïve Bayes Classifier," *INFOTECH J.*, vol. 9, no. 2, pp. 383–392, 2023, doi: 10.31949/infotech.v9i2.6277.