JDMIS: Journal of Data Mining and Information Systems
Vol. 2, No. 1, 2024, p. 26~36
ISSN: 2986-3473 | p-ISSN 2986-5271
DOI: 10.54259/jdmis.v2i1.2418

Sentimen Komentar Universitas Pelita Harapan Pada TikTok Menggunakan Metode *K-Nearest Neighbor*

Robert Wijaya[1], Albert Suwandhi[2]
[1]Informatika, Universitas Pelita Harapan, Indonesia
[2] Teknologi Informasi, Universitas IBBI, Indonesia
Email: [1]03082200011@student.uph.edu, [2]albertsuwandhi@ibbi.ac.id

ABSTRAK

Di era digital yang sedang berkembang, media sosial, khususnya TikTok, telah menjadi platform penting untuk berbagi informasi dan komunikasi, termasuk oleh institusi pendidikan seperti Universitas Pelita Harapan (UPH). Penggunaan TikTok di UPH menghasilkan beragam komentar yang memerlukan pengelolaan efektif, mendorong penelitian ini untuk mengembangkan sentimen menggunakan algoritma K-Nearest Neighbor (KNN). Penelitian ini bertujuan untuk mengatasi dua masalah utama: menganalisa tingkat akurasi algoritma *K-Nearest Neighbor* dalam sentimen terhadap kalimat komentar dan mengukur kinerja algoritma *K-Nearest Neighbor* dalam menghitung hasil analisis pada kalimat komentar. Penelitian ini menggunakan metode KNN dengan *dataset* berjumlah 1213 data dari tahun 2021 sampai 2023 yang memiliki kata kunci terkait UPH dari konten *platform* TikTok. Penelitian ini dikelola dan dilakukan di *Google Colab* dengan bahasa pemrograman *python*. Berdasarkan hasil data *training* dan data *testing* didapatkan *accuracy* sebesar 91% dengan *precision* sebesar 93%, *recall* sebesar 91% dan *f-1 score* sebesar 92%. Dari hasil performa algoritma KNN, dapat disimpulkan metode KNN dapat mengklasifikasikan sentiment komentar.

Kata Kunci: : KNN, TikTok, Analisis Sentimen

ABSTRACT

In the evolving digital era, social media, particularly TikTok, has become a pivotal platform for information sharing and communication, including by educational institutions such as Universitas Pelita Harapan (UPH). The use of TikTok at UPH has generated diverse comments that require effective management, prompting this research to develop sentiment by using the K-Nearest Neighbor (KNN) algorithm. This study aims to address two main issues: analyzing the accuracy of the K-Nearest Neighbor algorithm in sentiment of comment sentences and measuring the performance of the K-Nearest Neighbor algorithm in calculating analysis results on comment sentences. This research employs the KNN method with a dataset of 1213 entries from 2021 to 2023 containing keywords related to UPH from TikTok platform content. The study is managed and conducted on Google Colab using the Python programming language. Based on the results of training and testing data, an accuracy of 91% is obtained, with precision at 93%, recall at 91%, and an f-1 score of 92%. From the performance of the KNN algorithm, it can be concluded that the KNN method can classify sentiment in comments.

Keywords: *KNN, TikTok, Sentiment Analysis*

Penulis Korespondensi:

Robert Wijaya
03082200011@student.uph.edu

Article Info

Diterima: 29 Januari 2024
Direvisi: 30 Januari 2024
Disetujui: 31 Januari 2024

1. PENDAHULUAN

Seiring berjalannya waktu, teknologi informasi berkembang sangat pesat, kegiatan yang pada umumnya banyak menggunakan peranan teknologi informasi di dalamnya. Hal tersebut disebabkan oleh kebutuhan teknologi yang semakin

meningkat, salah satunya teknologi yang memberikan banyak kemampuan untuk digunakan sebagai media komunikasi yang dapat mempercepat kerja manusia, salah satunya untuk media promosi.

Menurut laporan *We Are Social*, pada bulan Januari 2023, terdapat 167 juta orang yang aktif menggunakan media sosial di Indonesia, yang setara dengan 60,4% dari total populasi dalam negeri. Jumlah pengguna aktif ini mengalami penurunan sebesar 12,57% dibandingkan dengan tahun sebelumnya, di mana jumlahnya mencapai 191 juta orang. Penurunan ini merupakan yang pertama kalinya terjadi dalam satu dekade terakhir [1].

Terdapat beragam komentar yang muncul di media sosial, mulai dari yang informatif dan mendukung hingga yang negatif atau tidak relevan. Penggunaan teknologi analisis sentimen dapat membantu dalam mengelola komentar di media sosial. Beberapa penelitian menyoroti isu-isu terkait komentar di media sosial, seperti perlindungan hukum terkait penyebaran informasi negatif di TikTok [2], penggunaan bahasa sarkasme [3] dan disfemisme oleh netizen [4], serta model pembelajaran daring menggunakan *platform* media sosial untuk mendukung keterampilan peserta didik [5]. Dengan demikian, penggunaan teknologi analisis sentimen dapat menjadi salah satu langkah yang efektif dalam mengelola komentar di media sosial untuk menjaga reputasi dan menyajikan diskusi yang konstruktif.

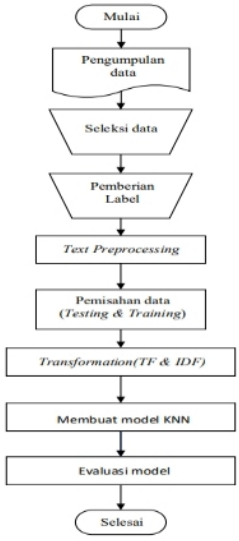
Berdasarkan penelitian terdahulu yang menggunakan algoritma KNN berdasarkan studi kasus yang berbeda-beda, maka muncullah penelitian yang dilakukan oleh Florensius Sianipar, dkk [6] tentang "Analisis Sentimen Publik Terhadap Pengambilalihan Jalan Rusak" menyatakan bahwa algoritma KNN memberikan hasil akurasi sebesar 82,69%, Ada juga penelitian lainnya yang dilakukan oleh Baita dan Cahyono [7] tentang "Analisis sentimen mengenai vaksin sinovac" menyatakan bahwa algoritma KNN memberikan hasil akurasi sebesar 60%. Penelitian selanjutnya yang dilakukan oleh Puspitasari [8] tentang "Analisis sentimen terhadap inflasi pasca COVID-19", KNN menunjukkan hasil akurasi sebesar 54%. Selain itu ada penelitian dilakukan oleh Ginantra, dkk [9] menyatakan bahwa algoritma KNN memberikan hasil akurasi yang sangat tinggi, yakni 91,26%. Terakhir penelitian yang dilakukan oleh Angkasa dan Pangaribuan [10] yang membandingkan KNN dan Random Forest dalam mendiagnosis penyakit kanker payudara, dimana KNN mempunyai akurasi yang lebih akurat yakni 99,59% dibandingkan Random Forest 99,51%. Dapat disimpulkan bahwa terdapat perbedaan dalam tingkat akurasi algoritma *K-Nearest Neighbor* (KNN) sehingga menciptakan ketidakpastian mengenai kecocokan algoritma KNN untuk analisis sentimen. Penelitian ini diperlukan untuk membuktikan apakah algoritma KNN bisa dipakai untuk mengklasifikasi komentar.

Berdasakan latar belakang yang telah diuraikan, maka perlu dilakukan penelitian mengenai pengembangan sentimen komentar UPH pada TikTok dengan menggunakan metode KNN.

2. METODE PENELITIAN

2.1. Tahapan Penelitian

Tahapan penelitian merupakan susunan kerja dari penelitian ini secara sistematis agar mecapai tujuan yang diinginkan. Adapun tahapan penelitian ini meliputi pengumpulan data, seleksi data, pemberian label, *text preprocessing*, permisahan data(Testing dan Training), transformation(TF dan IDF), membuat model KNN, evaluasi model. Tahapan penelitian ini dapat dilihat pada Gambar 1.



Gambar 1.Tahapan Penelitian

1. Pada tahap *crawling data* dilakukan penarikan *data text* TikTok dengan *rapidminer* pada *browser* untuk menarik data
2. Pada tahap seleksi data dilakukan dengan penghapusan atribut yang tidak relevan untuk tujuan analisis sentimen.

3. Pada tahap Pemberian label dilakukan pemberian label positif, negatif pada *text* dokumen.
4. Pada tahap *Text Preprocessing* dilakukan *Case Normalization,* Pembersihan, *Tokenization*, *Stopword Removal*, *Lemmatization*, *Stemming.*
5. Pada tahap TF-IDF dilakukan pembobotan pada kata dalam dokumen untuk mengetahui bobot dari kata tersebut.
6. Pada tahap pemisahan data, penulis membagi *data set* menjadi *data training* dan *data testing*
7. Pada tahap membuat model KNN, kumpulan data terstruktur diproses menggunakan metode KNN dan menentukan nilai k.
8. Pada tahap evaluasi model, penulis mengevaluasi kinerja algoritma KNN yang digunakan dalam penelitian ini untuk mengklasifikasikan data berdasarkan perhitungan *precision*, *recall*, dan *f1- score.*

2.2. Prosedur Pengumpulan Data

Dataset yang digunakan untuk penelitian ini terdiri dari komentar positif dan negatif yang diperoleh dari *platform* TikTok. Data yang dikumpulkan dalam penelitian ini mencakup topik UPH dan rentang waktu dari tahun 2021 hingga 2023. Proses pengumpulan data dilakukan dengan menggunakan ekstensi Data *Miner* pada *browser*. Data yang dikumpulkan terdiri dari : "*Date*", "*Commentar Name*", dan *"Tags"*, dan "*Commentar*".

2.3. Seleksi Data

Dataset ini terdiri dari empat atribut (*Date, Tags, Commentar Name dan Commentar*). Untuk seleksi data, penulis akan menghapus atribut "*Date*", "*Commentar Name*", dan *"Tags"* karena tidak relevan untuk tujuan analisis sentimen

2.4. Pemberian Label

Berita yang dianggap *positive* diberi label 1 dan berita yang dianggap *negative* diberi label 0.

2.5. Text Preprocessing

Text preprocessing yaitu pengolahan teks yang akan digunakan sebagai data dan menghindari masalah pada proses selanjutnya. Langkah pertama dalam *text preprocessing* adalah *case Normalization.*

a. *Case Normalization*

Normalization adalah salah satu tahap penting dalam *text preprocessing* yang bertujuan untuk mengubah teks menjadi bentuk yang lebih standar atau normal. Ini membantu mengatasi variasi dalam teks, seperti perbedaan antara singkatan dan bentuk lengkapnya, atau angka dalam berbagai format. Hal ini dapat dilihat pada tabel di bawah ini:

Tabel 1. Sebelum dan sesudah *normalization*

Sebelum *normalization*	**Sesudah *normalization***
kayak di drama korea drama korea ya, pas liat di drama korea kek apasih masa sekolah ada kek gini, ternyata emang ada, dan ya dunia kita berbeda.	seperti di drama korea, ya. pas lihat di drama korea, seperti apa sih masa sekolah ada seperti ini. ternyata memang ada, dan ya, dunia kita berbeda.

b. *Pembersihan*

Pembersihan adalah proses mengidentifikasi dan menangani data yang tidak *valid*, tidak benar, tidak lengkap, atau tidak konsisten dalam kumpulan data. Dalam kasus teks, pembersihan data melibatkan penghapusan karakter dan simbol yang tidak relevan serta mengatasi masalah seperti duplikat, salah eja, dan *format* yang tidak konsisten. Hal ini dapat dilihat pada tabel di bawah ini:

Tabel 2. Sebelum dan sesudah pembersihan

Sebelum pembersihan	**Sesudah pembersihan**
seperti di drama korea, ya. pas lihat di drama korea, seperti apa sih masa sekolah ada seperti ini. ternyata memang ada, dan ya, dunia kita berbeda.	seperti di drama korea ya pas lihat di drama korea seperti apa sih masa sekolah ada seperti ini ternyata memang ada dan ya dunia kita berbeda

c. *Tokenization*

Sebelum memproses data/teks lebih lanjut, data/teks tersebut perlu disegmentasi menjadi kata-kata, sebuah proses yang disebut *Tokenization*, memisahkan kalimat menjadi kata-kata individual. Hal ini dapat dilihat pada tabel di bawah ini:

Tabel 3. Sebelum dan sesudah *tokenization*

Sebelum *tokenization*	Sesudah *tokenization*
kayak di drama korea drama korea ya pas liat di drama korea kek apasih masa sekolah ada kek gini ternyata emang ada dan ya dunia kita berbeda	seperti, di, drama, korea, ya, pas, lihat, di, drama, korea, seperti, apa, sih, masa, sekolah, ada, seperti, ini, ternyata, memang, ada, dan, ya, dunia, kita, berbeda

d. *Stopword Removal*

Stopwords removal adalah tahap penting dalam *text preprocessing* yang bertujuan untuk menghilangkan kata-kata yang sering muncul dan umumnya tidak memberikan makna yang signifikan dalam analisis teks. Ini dilakukan untuk memfokuskan analisis pada kata-kata kunci atau entitas yang lebih informatif. Berikut adalah penjelasan lebih *detail* tentang *stopword removal* pada tabel di bawah ini:

Tabel 4. Sebelum dan sesudah *stopword removal*

Sebelum *stopword removal*	Sesudah *stopword removal*
seperti, di, drama, korea, ya, pas, lihat, di, drama, korea, seperti, apa, sih, masa, sekolah, ada, seperti, ini, ternyata, memang, ada, dan, ya, dunia, kita, berbeda	drama, korea, lihat, drama, korea, masa, sekolah, ternyata, dunia, berbeda

e. *Lemmatization*

Lemmatization adalah proses mengurangi sebuah kata menjadi bentuk dasar (bentuk kata dalam kamus). Hal ini dapat dilihat pada tabel di bawah ini:

Tabel 5. Sebelum dan sesudah *lemmatization*

Sebelum *stemming* atau *lemmatization*	Sesudah *stemming* atau *lemmatization*
drama, korea, lihat, drama, korea, masa, sekolah, ternyata, dunia, berbeda	drama, korea, lihat, drama, korea, masa, sekolah, ternyata, dunia, beda

f. *Stemming*

Stemming adalah proses menghilangkan akhiran kata (*suffix*) dari kata-kata dalam teks untuk menghasilkan bentuk dasarnya. Ini dilakukan untuk mengidentifikasi kata-kata yang memiliki akar yang sama [8]. Misalnya, kata "berlari" akan diubah menjadi "lari," dan kata "menyanyikan" akan diubah menjadi "nyanyi.". Hal ini dapat dilihat pada tabel di bawah ini:

Tabel 6. Sebelum dan sesudah *stemming*

Sebelum *stemming*	Sesudah *stemming*
drama, korea, lihat, drama, korea, masa, sekolah, ternyata, dunia, beda	drama, korea, lihat, drama, korea, masa, sekolah, ternyata, dunia, beda

2.6. Permisahan Data *(Testing & Training)*

Pemisahan antara *dataset data training* dan *data testing* dilakukan untuk memungkinkan model *machine learning* yang telah dibangun untuk di *testing* pada data yang belum pernah tersedia sebelumnya. Penulis memisahkan *dataset* menjadi 80% untuk data *training* dan 20% untuk data *testing.*

2.7. *Transformation (TF-IDF)*

TF-IDF (Term Frequency-Inverse Document Frequency) adalah metode yang pemrosesan bahasa alami untuk mengekstrak fitur-fitur penting dan merepresentasikan dokumen secara numerik. *TF-IDF* diperoleh dengan mengalikan dua nilai, yaitu *TF (Term Frequency)* dan *IDF* (*TF* adalah ukuran seberapa sering sebuah kata muncul dalam sebuah dokumen, sedangkan *IDF* adalah ukuran pentingnya sebuah kata dalam kumpulan dokumen, memberikan bobot yang lebih tinggi pada kata yang lebih jarang muncul dalam sebuah dokumen [11].

Rumus pada TF pada nomor 1 [11].

$$TF = \frac{jumlah\ kata\ dalam\ 1\ dokumen}{\text{total seluruh kata dalam dokumen}} \quad \textbf{(1)}$$

Rumus pada IDF pada nomor 2 [11].

$$IDF = Log\ (\frac{total\ dari\ semua\ dokumen}{\text{jumlah kata di dalam dokumen}}) \quad \textbf{(2)}$$

Rumus pada TF – IDF pada nomor 3 [11].

$$TF - IDF = TF * IDF \quad \textbf{(3)}$$

2.7. Membuat Model KNN

Penelitian ini menggunakan metode SEMMA *(Sample, Explore, Modify, Model, Assess)* untuk analisis. Pada tahap *sample*, data dikumpulkan dari penelitian melalui komentar TikTok. Pada tahap *explore*, atribut yang tidak penting akan dihapus. Tahap *modify,* melibatkan pengolahan data tak terstruktur, mencakup *case normalization*, *cleaning*, *tokenization*, *stopword removal*, dan *stemming* atau *lemmatization* untuk mengubahnya menjadi data terstruktur. Pada tahap *model*, kumpulan data terstruktur diproses menggunakan metode KNN dan menentukan nilai k. Tahap *assess* mengevaluasi model berdasarkan metrik seperti *accuracy*, *precision*, dan *recall.*

2.8. Mengevaluasi Model KNN

Dalam tahap evaluasi model, penulis mengevaluasi kinerja algoritma KNN yang digunakan dalam penelitian ini untuk mengklasifikasikan data berdasarkan perhitungan *precision*, *recall*, dan *f1- score*. Hasil dari *True Positive, True Negative, False Positive*, dan *False Negative* ditampilkan menggunakan *confusion matrix*, sedangkan accuracy, *precision*, *recall*, dan *f1-score* ditampilkan menggunakan *classification report* yang disediakan oleh *library Python "sklearn"*.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Pengumpulan Data

Setelah melakukan pengambilan komentar dari *platform* media sosial TikTok dengan menggunakan *tools*, berhasil terkumpul sebanyak 1213 komentar. Komentar-komentar ini merupakan respons terhadap konten UPH yang diunggah oleh seseorang. Data yang berhasil terkumpul meliputi rentang waktu dari tahun 2021 hingga 2023. *Dataset* ini tersedia dalam format CSV (*Comma Separated Values*). Gambar 2 memberikan informasi mengenai *dataset* komentar UPH yang diperoleh melalui fungsi *data.info()*.

```
data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1213 entries, 0 to 1212
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Date            1213 non-null   object
 1   Commenter Name  1213 non-null   object
 2   Comment         1213 non-null   object
 3   label           1213 non-null   object
 4   Comment_Length  1213 non-null   int64 
dtypes: int64(1), object(4)
memory usage: 47.5+ KB
```

Gambar 2. Hasil Pengumpulan Data

Dataset ini terdiri dari empat atribut dan bisa terlihat dari Gambar 2. Untuk seleksi data, penulis hanya perlu menghapus atribut "*Date*", "*Commentar Name*", dan "*Comment_length*".

3.2 Hasil Data Label

Sebelum *training* model *KNN*, *dataset* dilabeli secara manual terlebih dahulu, supaya *dataset* bisa melakukan *training* dan *testing*. Gambar di bawah ini menunjukkan total *data set* yang sudah dilabel.

```
jumlah_positif = (data['label'] == 'POSITIVE').sum()
jumlah_negatif = (data['label'] == 'NEGATIVE').sum()

print(f'Jumlah data positif: {jumlah_positif}')
print(f'Jumlah data negatif: {jumlah_negatif}')

Jumlah data positif: 667
Jumlah data negatif: 546
```

Gambar 3. Total Jumlah *Positive* dan *Negative*

3.3 Hasil Pengumpulan Data

Setelah proses pelabelan data, data akan memasuki tahap *text preprocessing* agar bisa dibaca oleh *machine learning*. Hasil dari *text preprocessing* bisa dilihat pada gambar di bawah ini:

a. *Case Normalization*

Tahap pertama pada *text preprocessing*, penulis akan melakukan *case normalization* pada *dataset,* seluruh *dataset* akan diubah menjadi huruf kecil untuk membuat teks menjadi lebih konsisten dan memudahkan analisis teks.

```
0       kayak di drama korea drama korea ya pas liat d...
1                                          uang papa habis
2                   sekolah kami sekolah jaman majapahit
3                            malam tapi jam  siang keren
4                                  sekolah orang kayaa
                              ...
1208                                pasti itu mahal ya ica
1209    sekolah saya panggungnya dari kayu biasa duduk...
1210                                 uph  uang papa habis
1211                         sekolah kita beda galaksi kak
1212                           sekolah kita beda kasta kak
```

Gambar 4. Hasil *Case Normalization*

b. Pembersihan

Penulis akan melakukan pembersihan pada karakter khusus, tanda baca, dan angka.

```
Pembersihan :
0       kayak di drama korea drama korea ya pas liat d...
1                                          uang papa habis
2                   sekolah kami sekolah jaman majapahit
3                            malam tapi jam  siang keren
4                                  Sekolah orang kayaa
                              ...
1208                                pasti itu mahal ya ica
1209    sekolah saya panggungnya dari kayu biasa duduk...
1210                                 uph  uang papa habis
1211                         Sekolah kita beda galaksi kak
1212                           sekolah kita beda kasta kak
```

Gambar 5. Hasil Pembersihan

c. *Tokenization*

Penulis akan memecah seluruh kumpulan data menjadi bagian-bagian kecil. Seperti pada Gambar 6.

```
Tokenization :
0       [kayak, di, drama, korea, drama, korea, ya, pa...
1                                     [uang, papa, habis]
2           [sekolah, kami, sekolah, jaman, majapahit]
3                    [malam, tapi, jam, siang, keren]
4                             [sekolah, orang, kayaa]
                                ...
1208                        [pasti, itu, mahal, ya, ica]
1209    [sekolah, saya, panggungnya, dari, kayu, biasa...
1210                           [uph, uang, papa, habis]
1211                [sekolah, kita, beda, galaksi, kak]
1212                  [sekolah, kita, beda, kasta, kak]
```

Gambar 6. Hasil *Tokenization*

d. *Stopword Removal*

Selanjutnya, penulis akan melakukan *stopword removal* yaitu menghapus kata-kata pada *dataset* yang tidak memiliki nilai informasi yang penting.

```
Stopwords :
0       [kayak, drama, korea, drama, korea, ya, pas, l...
1                                     [uang, papa, habis]
2                   [sekolah, sekolah, jaman, majapahit]
3                            [malam, jam, siang, keren]
4                               [sekolah, orang, kayaa]
                                ...
1208                                    [mahal, ya, ica]
1209    [sekolah, panggungnya, kayu, duduk, langsung, ...
1210                             [uph, uang, papa, habis]
1211                          [sekolah, beda, galaksi, kak]
1212                            [sekolah, beda, kasta, kak]
```

Gambar 7. Hasil *Stopword Removal*

e. *Lemmatization*

Setelah melewati tahap *stopword removal,* penulis akan melakukan *lemmatization*. *Lemmatization* adalah proses yang mengurangi akhiran kata, seperti pada Gambar 8.

```
Lemmatization :
0       [kayak, drama, korea, drama, korea, ya, pa, li...
1                                    [uang, papa, habis]
2                   [sekolah, sekolah, jaman, majapahit]
3                             [malam, jam, siang, keren]
4                                [sekolah, orang, kayaa]
                                ...
1208                                      [mahal, ya, ica]
1209    [sekolah, panggungnya, kayu, duduk, langsung, ...
1210                               [uph, uang, papa, habis]
1211                          [sekolah, beda, galaksi, kak]
1212                           [sekolah, beda, kasta, kak]
```

Gambar 8. Hasil *Lemmatization*

f. *Stemming*

Tahap terakhir, penulis akan melakukan *stemming*. *Stemming* adalah mengubah kata-kata ke dalam bentuk dasar.

```
Stemming :
0       [kayak, drama, korea, drama, korea, ya, pa, li...
1                                    [uang, papa, habis]
2                   [sekolah, sekolah, jaman, majapahit]
3                             [malam, jam, siang, keren]
4                                [sekolah, orang, kayaa]
                                ...
1208                                      [mahal, ya, ica]
1209    [sekolah, panggung, kayu, duduk, langsung, patah]
1210                               [uph, uang, papa, habis]
1211                          [sekolah, beda, galaksi, kak]
1212                           [sekolah, beda, kasta, kak]
```

Gambar 9. Hasil *Stemming*

3.4 Permisahan Data (*Testing & Training*)

Setelah melakukan proses *text preprocessing*. Selanjutnya penulis melakukan permisahan data(*testing & training*). Berikut di bawah ini merupakan hasil dari permisahan data.

```
Data Training:
Labels:
971     NEGATIVE
352     POSITIVE
676     NEGATIVE
671     POSITIVE
982     POSITIVE
         ...
1044    POSITIVE
1095    POSITIVE
1130    POSITIVE
860     POSITIVE
1126    POSITIVE
Name: label, Length: 970, dtype: object

Data Testing:
Labels:
382    POSITIVE
787    POSITIVE
43     NEGATIVE
155    POSITIVE
493    NEGATIVE
        ...
59     POSITIVE
837    POSITIVE
63     POSITIVE
722    POSITIVE
644    NEGATIVE
Name: label, Length: 243, dtype: object
```

Gambar 10. Hail Permisahan Data *(Testing & Training)*

3.5 *Transformation*

Setelah melakukan proses permisahan data. Selanjutnya penulis melakukan TF –IDF untuk mengekstrak fitur-fitur penting dan merepresentasikan dokumen secara numerik. Berikut di bawah ini merupakan hasil dari TF – IDF.

```
-3.4170e-01  3.6592e-01  7.1084e-01 -1.1752e+00
-9.1876e-02 -1.1798e-02 -2.3171e-01  3.3501e-01
 4.1060e-01 -4.6063e-02 -3.2066e-01 -1.3558e-01
 2.3512e-01 -1.1266e-01 -6.3805e-01  3.0731e-02
-7.1437e-01 -9.2071e-01  9.2736e-02  9.9276e-02
-7.1901e-02 -9.0255e-01 -6.1942e-01 -9.8787e-01
-7.0859e-02 -6.2859e-02  3.6730e-01  3.6571e-01
-5.4735e-01 -2.2454e-01 -4.1741e-01 -5.1566e-05
-4.4910e-01  1.1944e-01 -6.6511e-01  3.1428e-02
 9.9204e-02 -7.3010e-01 -4.9230e-01 -2.8550e-01
-7.5013e-01 -3.0712e-01 -1.4554e-01  4.9327e-01
-4.6061e-01  1.0729e+00  3.2161e-01 -3.6459e-01
-3.8077e-01 -5.6719e-01  3.9581e-01  2.3752e-01
 1.1016e+00  7.4036e-01 -2.8486e-01 -1.2475e+00
 2.7672e-01 -2.0743e-01  4.0379e-01 -1.0002e+00
 2.0305e-01 -8.5739e-01  7.6791e-01]
.37051    0.67706    0.48728    0.34663    0.154
3.59518    0.2452    -0.30288   -0.43952    0.34
3.22144   -0.023228  -0.13451    0.082757   0.25
3.18507    0.044146   0.56987   -0.26005   -0.38
3.23008   -0.048999  -0.0053947  0.53103   -0.11
3.12819    0.1278     0.49488    0.16101    0.33
3.90574    0.65931   -0.062931   0.20454    0.31
3.27613   -0.27964   -0.027583   0.33466    0.52
3.079882  -0.43247   -0.13602    0.24493    0.46
3.40932    0.3817    -0.33738    0.11546   -0.25
3.25754   -0.17855   -0.90601    0.101      0.26
3.75323   -0.22981   -1.0301     0.0034974 -0.03
3.3006    -0.26094    0.49965    0.17487   -0.30
3.56809   -0.82957    0.55824    0.087272   0.77
3.53515    0.7754     0.45141   -0.73914   -0.21
3.59694   -0.14201    0.20747   -0.31695   -0.55
3.73554   -0.43075    0.76173  ]
3881   -0.70512    0.45632   -0.66897    0.30411
3.50283   -0.41815    0.31758   -0.31735   -0.86
3.74826    0.35545   -0.59321   -0.34876   -0.92
3.8072    -0.1712    -0.505     -0.86671    0.25
3.11075    0.48538    0.49432    0.43909   -0.00
3.93604    0.11284    0.42269   -0.50771    1.08
3.70915    0.36104    0.41545   -0.31608   -0.11
3.19856    0.27928   -0.27468   -0.046731   1.08
3.3355    -0.60383    0.27725    0.41607    0.55
3.59629   -0.62935   -0.38876    0.18153   -0.27
3.32394    0.019943  -0.99724   -0.79633   -0.25
3.16058   -0.79156   -0.47805   -0.31824   -1.17
3.1694    -0.67325    0.10182    0.46863   -0.29
```

Gambar 11. Hasil *Transformation*

3.6 Membuat Model KNN

Setelah melakukan proses TF-IDF. Selanjutnya penulis akan membuat model KNN untuk klasifikasi sentimen pada data komentar. Pertama – tama model KNN diinisialisasi dengan menggunakan knn = *KNeighborsClassifier()*. Berikut di bawah ini merupakan cara membuat model KNN.

```
# Inisialisasi model KNN
knn = KNeighborsClassifier()
```

Gambar 12. Inisialisasi Model KNN

Selanjutnya, pada Gambar 13. Pencarian *grid* dilakukan dengan menggunakan *GridSearchCV* untuk menemukan *hyperparameter* yang optimal dengan menggunakan *Stratified K-Fold Cross-Validation*. Setelah itu, melakukan pencarian grid dengan menggunakan data training menggunakan *gridSearch.fit(xTrain, yTrain)*.

```
gridSearch = GridSearchCV(knn, paramGrid, cv=StratifiedKFold(n_splits=5, shuffle=True, random_state=42))
gridSearch.fit(xTrain, yTrain)
```

Gambar 13. Pencarian *Grid*

Pada Gambar 14. Penulis mengambil nilai terbaik untuk *hyperparameter* dalam model KNN ke dalam suatu variabel.

```
# Mendapatkan hyperparameter terbaik
bestK = gridSearch.best_params_['n_neighbors']
bestWeights = gridSearch.best_params_['weights']
bestMetric = gridSearch.best_params_['metric']
```

Gambar 14. Mendapatkan *Hyperparameter*

Setelah mendapatkan *hyperparameter* yang optimal dari hasil *grid search*, model KNN diinisialisasi ulang dengan *hyperparameter* yang optimal dan dilatih pada semua data training. Tujuannya memastikan bahwa model KNN yang akan

digunakan untuk prediksi selanjutnya telah dikonfigurasi dengan kombinasi *hyperparameter* terbaik yang ditemukan selama proses *search grid*.

```
knn = KNeighborsClassifier(n_neighbors=bestK, weights=bestWeights, metric=bestMetric)
knn.fit(xTrain, yTrain)
```

Gambar 15. Melatih Model KNN

Pada Gambar 16. Model digunakan untuk melakukan prediksi pada *data testing*.

```
# Prediksi sentimen pada data testing
yPredTest = knn.predict(xTest)
```

Gambar 16. Prediksi Sentimen

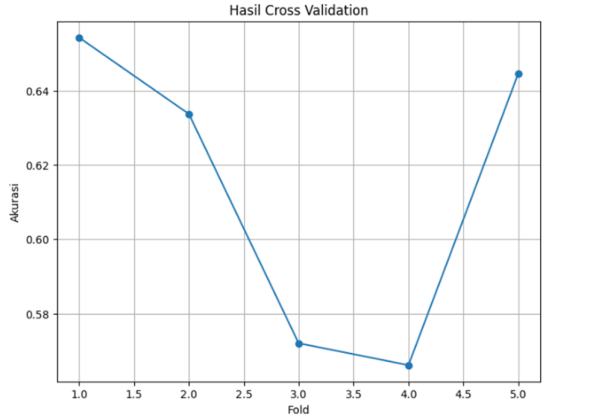
Pada Gambar 17. Penulis melakukan laporan klasifikasi sejumlah metrik evaluasi seperti *precision, recall, f1-score.*

```
# Klasifikasi pada data analisis
classificationReport = classification_report(y, yPredTest)
```

Gambar 17. Klasifikasi *Report*

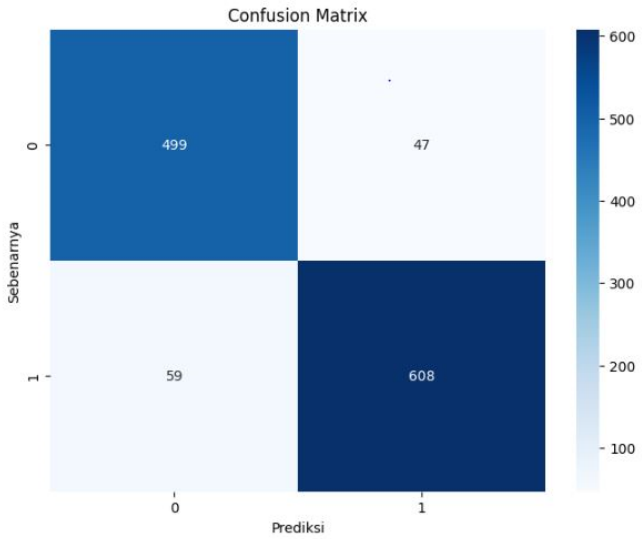
3.7 Hasil Penelitian

Pada Gambar 18 menunjukkan grafik *cross validation* dari model KNN. Dimana k = 1 dan k = 5 memiliki *accuracy* 0.64 atau 64%. *accuracy* tertinggi pada hasil *cross validation* adalah k = 1 dengan *accuracy* 68%. Sedangkan *accuracy* terendah berada di k = 4 dengan *accuracy* di bawah 0.58 atau 58%.Berikut tampilan dari hasil *cross validation* dan *confusion matrix*:



Gambar 18. Hasil *Cross Validation*

Gambar 19 menunjukkan *confusion matrix* dari model KNN. Dimana angka 499 adalah *True Negative* sedangkan angka 608 adalah *True Positive.* Dan juga angka ke 59 adalah *False Negative* sedangkan angka 47 adalah *False Positive*.



Gambar 19. *Confusion Matrix*

3.8 Pembahasan

Dari Gambar 19 kita bisa menghitung *precision*, *recall*, *f-1 score* dan nilai *accuracy*. Dibawah ini adalah hasil perhitungannya:

a. *Positive :*

Precision : $\frac{TP}{TP+FP} = \frac{608}{608+47} = \frac{608}{655} = 0{,}928\ atau\ 0{,}93$

Recall : $\frac{TP}{TP+FN} = \frac{608}{608+59} = \frac{608}{667} = 0{,}911$

F-1 score : $\frac{2TP}{2TP+FP+FN} = \frac{2(608)}{2(608)+47+59} = \frac{1216}{1216+47+59} = \frac{1216}{1322} = 0{,}918$

b. *Negative :*

Precision : $\frac{TP}{TP+FP} = \frac{499}{499+59} = \frac{499}{558} = 0{,}89\ atau\ 0{,}90$

Recall : $\frac{TP}{TP+FN} = \frac{499}{499+47} = \frac{499}{546} = 0{,}913$

F-1 score : $\frac{2TP}{2TP+FP+FN} = \frac{2(499)}{2(499)+59+47} = \frac{998}{998+59+47} = \frac{998}{1104} = 0{,}903$

c. *Accuracy :*

$\frac{TP+FN}{TP+TN+FP+FN} = \frac{608+499}{608+499+47+59} = \frac{1107}{1213} = 0{,}912$

Pada Gambar 20 penulis dapat membuat kesimpulan sebagai berikut :

1. *Accuracy* 91%: Model klasifikasi memiliki tingkat *accuracy* yang tinggi, sekitar 91%. Ini berarti sebagian besar data telah diklasifikasikan dengan benar oleh *model*.
2. *Precision positive* 93%: *Precision positive* mengukur seberapa akurat *model* mengidentifikasi kelas *positive*.
3. *Precision negative* 89%: *Precision negative* mengukur seberapa akurat *model* mengidentifikasi kelas *negative*.
4. *Recall positive* 91%: Dengan *positive recall* 91%, *model* hampir dapat mendeteksi semua contoh aktual dari kelas *positive*.
5. *Recall negative* 91%: Dengan *negative recall* sebesar 91%, model hampir dapat mendeteksi semua contoh aktual dari kelas *negative*.

```
Akurasi : 0.91
Parameter Terbaik untuk Model KNN:
n_neighbors: 3
weights: distance
metric: manhattan
Grid Search Best Score: 0.61
              precision    recall  f1-score   support

    NEGATIVE       0.89      0.92      0.90       546
    POSITIVE       0.93      0.91      0.92       667

    accuracy                           0.91      1213
   macro avg       0.91      0.91      0.91      1213
weighted avg       0.91      0.91      0.91      1213
```

Gambar 1. Hasil perhitungan *Confusion Matrix*

Pada Gambar 21 penulis membuat analisis sentimen untuk mengklasifikasi komentar-komentar menentukan sentimen positif atau negatif. Data analisis digunakan untuk mengukur kinerja model yang dihasilkan oleh algoritma. Penulis mengambil 10 contoh komentar sebagai representasi dari hasil validasi. Dari 10 komentar tersebut ada 2 komentar yang terdapat kesalahan dalam mengklasifikasikan sentimen. Secara keseluruhan, model klasifikasi ini berkinerja baik dengan *accuracy* yang tinggi, *precision positive* dan *negative* yang baik, dan kemampuan yang kuat untuk mendeteksi contoh dari kelas *positive* dan *negative.*

```
Data analisis :
Komentar: kayak drama korea drama korea ya pa liat drama korea kek apasih sekolah kek gini emang ya dunia berbeda
Label Asli: POSITIVE
hasil analisis Sentimen: POSITIVE

Komentar: uang papa habis
Label Asli: NEGATIVE
hasil analisis Sentimen: NEGATIVE

Komentar: sekolah sekolah jaman majapahit
Label Asli: NEGATIVE
hasil analisis Sentimen: NEGATIVE

Komentar: malam jam siang keren
Label Asli: POSITIVE
hasil analisis Sentimen: POSITIVE

Komentar: sekolah orang kayaa
Label Asli: POSITIVE
hasil analisis Sentimen: POSITIVE

Komentar: seru banget sekolahnya
Label Asli: POSITIVE
hasil analisis Sentimen: NEGATIVE

Komentar: cakep banget
Label Asli: POSITIVE
hasil analisis Sentimen: POSITIVE

Komentar: keren banget sekolahnyaa
Label Asli: POSITIVE
hasil analisis Sentimen: POSITIVE

Komentar: vibe wp banget sekolahnya
Label Asli: POSITIVE
hasil analisis Sentimen: NEGATIVE

Komentar: kak sekolahku aja gedungnya minjem
Label Asli: NEGATIVE
hasil analisis Sentimen: NEGATIVE
```

Gambar 2. Data analisis

4. KESIMPULAN

Dari hasil penelitian diatas mengenai analisis sentimen mengenai Universitas Pelita Harapan pada platform TikTok yang berjumlah 1213 data dengan menggunakan metode KNN dan sudah melewati tahap *preprocessing text, transformation* dan klasifikasi menggunakan algoritma KNN serta evaluasi data dengan confusion matrix. Dengan persentase 91% pada akurasi sehingga dapat disimpulkan bahwa KNN bisa dipakai untuk analisis sentimen.

REFERENSI

[1] S. Widi, "Pengguna Media Sosial di Indonesia Sebanyak 167 Juta pada 2023." [Online]. Available: https://dataindonesia.id/internet/detail/pengguna-media-sosial-di-indonesia-sebanyak-167-juta-pada-2023

[2] V. S. Virginia, "Perlindungan Hukum Korban Yang Dirugikan Akibat Pencemaran Nama Baik di Media Sosial Tiktok," *Supremasi Jurnal Hukum* , vol. 5, no. 02, pp. 134–143, 2021.

[3] Andi Saadillah, Andi Haryudi, Muhammad Reskiawan, and Alam Ikhsanul Amanah, "Penggunaan Bahasa Sarkasme Netizen di Media Sosial," *Jurnal Onoma: Pendidikan, Bahasa, dan Sastra*, vol. 9, no. 2, pp. 1437–1447, 2023, doi: 10.30605/onoma.v9i2.2367.

[4] R. Selgianita and M. N. Antono, "Disfemisme Warganet dalam Kolom Komentar Media Sosial Instagram @Kpipusat (Kajian Semantik)," *Journal of Educational Language and Literature*, vol. 1, no. 1, pp. 9–19, 2023, doi: 10.21107/jell.v1i1.19386.

[5] D. Menur, "Model Online Learning dalam Mendukung Keterampilan Menulis Descriptive Text Peserta Didik pada Sosial Media," Kurikula. Accessed: Sep. 15, 2020. [Online]. Available: https://www.neliti.com/publications/406832/model-online-learning-dalam-mendukung-keterampilan-menulis-descriptive-text-pese

[6] J. Florensius Sianipar, Y. R. Ramadhan, and I. Jaelani, "Analisis Sentimen Pembangunan Kereta Cepat Jakarta-Bandung di Media Sosial Twitter Menggunakan Metode Naive Bayes," *Media Online)*, vol. 4, no. 1, pp. 360–367, 2023, doi: 10.30865/klik.v4i1.1033.

[7] A. Baita and N. Cahyono, "Analisis Sentimen Mengenai Vaksin Sinovac Menggunakan Algoritma Support Vector Machine (Svm) Dan K-Nearest Neighbor (Knn)," *Infos*, vol. 4, no. 2, pp. 42–42, 2021.

[8] R. Puspitasari, Y. Findawati, M. A. Rosid, P. S. Informatika, and U. M. Sidoarjo, "Sentiment Analysis of Post-Covid-19 Inflation Based on Twitter Using the K-Nearest Neighbor and Support Vector Machine Analisis Sentimen Terhadap Inflasi Pasca Covid-19 Berdasarkan Twitter Dengan Metode Klasifikasi K-Nearest Neighbor Dan," vol. 4, no. 4, pp. 1–11, 2023.

[9] N. L. W. S. R. Ginantra, C. P. Yanti, G. D. Prasetya, I. B. G. Sarasvananda, and I. K. A. G. Wiguna, "Analisis Sentimen Ulasan Villa di Ubud Menggunakan Metode Naive Bayes, Decision Tree, dan K-NN," *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 11, no. 3, pp. 205–215, 2022, doi: 10.23887/janapati.v11i3.49450.

[10] V. Angkasa and J. Junifer Pangaribuan, "Komparasi Tingkat Akurasi Random Forest dan KNN Untuk Mendiagnosis Penyakit Kanker Payudara," *Journal Information System Development (ISD)*, 2022.

[11] S. Qaiser and R. Ali, "Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents," *Int J Comput Appl*, vol. 181, no. 1, pp. 25–29, 2018, doi: 10.5120/ijca2018917395.