Journal of Applied Informatics and Computing (JAIC)
Vol.7, No.1, Juli 2023, pp. 21~27
e-ISSN: 2548-6861

Analisis Sentimen Pencitraan Perguruan Tinggi di Yogyakarta Menggunakan Metode *Naïve Bayes Classifier*

Y.Yohakim Marwanta [1*], Badiyanto [2*]
* Informatika, Universitas Teknologi Digital Indonesia
yohakim@utdi.ac.id [1], badi@utdi,ac.id [2]

Article Info

Article history:

Received 2023-01-21
Revised 2023-03-08
Accepted 2023-07-19

Keyword:

Sentiment Analysis,
Twitter,
Classification,
Naïve Bayes.

ABSTRACT

This research utilizes data from Twitter to analyze sentiment in Yogyakarta's universities using the Naïve Bayes Classifier method. The Naive Bayes Classifier method is one of the text classification methods based on the probability of keywords in comparing training and testing documents. The data used consists of tweets in Indonesian language with keywords from the top 10 universities in Yogyakarta based on webometrics, as well as four other relevant keywords about Yogyakarta that are frequently searched through Google. From the conducted research, there are 1710 data collected from Twitter, which are used for classification and categorized into 3 labels: positive, negative, and neutral. The data is divided into 70% for training and 30% for testing randomly. The result of sentiment analysis classification from the test data shows that 82.1% of the data is categorized as neutral, 14.8% as positive, and 3.1% as negative, with an accuracy value of 73%.

I. PENDAHULUAN

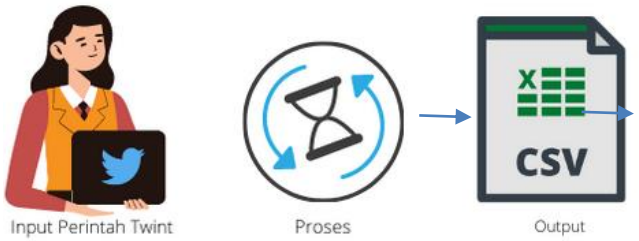
Pada era perkembangan big data yang terus meningkat, dimana terjadi pertumbuhan jumlah data secara eksponensial sebagai imbas dari perkembangan teknologi. Penggunaanya hampir menyeluruh dan menyentuh segala bidang mulai dari pendidikan, politik dan ekonomi. Fungsi utama yang ditawarkan big data adalah mengolah informasi dari data dan menyajikannya sesuai dengan kebutuhan. Penghasil data terbesar adalah pengguna Internet misalnya dari blog, situs organisasi dari pendidikan hingga pemerintahan, dan media social.

Daerah Istimewa Yogyakarta merupakan salah satu kota di Indonesia dengan beberapa perguruan tinggi baik negeri maupun swasta. Berdasarkan data LLDIKTI Wilayah V, Yogyakarta memiliki 104 perguruan tinggi swasta dengan 744 program studi. Dengan banyaknya perguruan tinggi, kota Yogyakarta memiliki magnet tersendiri bagi masyarakat Indonesia untuk menempuh pendidikan tinggi di Yogyakarta. Dengan banyaknya animo masyarakat untuk menempuh pendidikan tinggi di Yogyakarta membuat banyak orang dengan menggunakan media sosial seperti facebook, instagram, twitter dan lain-lain mencari informasi berita yang berhubungan dengan berita-berita atau sentimen baik sentiment negative maupun positif tentang Pendidikan tinggi di Yogyakrta. Pengguna Twitter, berdasarkan data PT Bakrie Telecom, memiliki 19,5 juta pengguna di Indonesia dari total 500 juta pengguna global. Twitter dipakai untuk mendapatkan berita atau bencana [1], mengikuti perkembangan tokoh terkenal [2], serta menjalin komunikasi dengan teman.

Beberapa penelitian sebelumnya algoritma Naïve Bayes digunakan untuk mengidentifikasi pola-pola dari ulasan-ulasan yang masuk dan memprediksi sentimen dari setiap ulasan oleh pengguna aplikasi e-commerce [3][4]. Metode Naïve Bayes Classifier digunakan untuk mengklasifikasikan sentimen dalam data tweet untuk memahami bagaimana persepsi dan pendapat masyarakat terhadap sistem zonasi sekolah [5]. Kemudian mengklasifikasikan sentimen setiap tweet menjadi kategori positif, negatif, atau netral menggunakan metode Naïve Bayes Classifier terkait dengan topik "Kurikulum 2013" [6]. Penelitian ini menggunakan opini publik khususnya pengguna twitter, melakukan analisis sentimen untuk mengetahui citra perguruan tinggi di Yogyakarta. Metode yang digunakan dalam penelitian ini adalah *Naïve Bayes Classifier* (NBC). Metode Naive Bayes Classifiers yaitu salah satu metode klasifikasi teks berdasarkan probabilitas kata kunci dalam membandingkan dokumen latih dan dokumen uji [7].

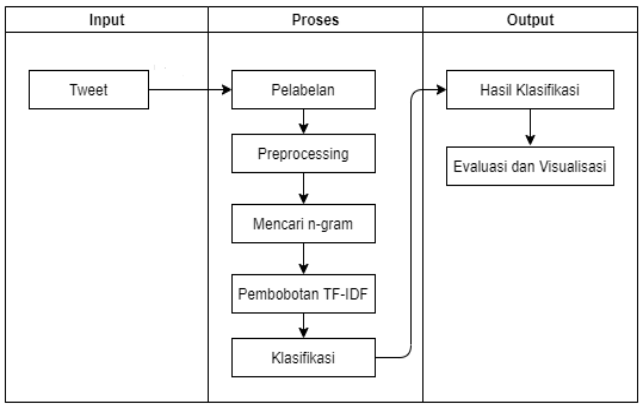
II. METODE PENELITIAN

Untuk dapat melakukan analisis sentimen diperlukan data. Data dapat diambil dari berbagai sumber dengan format data yang dihasilkan pun beragam disesuaikan dengan kebutuhan penelitian.



Gambar 1. Alur Pengambilan Data

Gambar 1 menunjukkan alur pengambilan data yang dimulai dengan memasukkan perintah untuk mengambil data dari twitter, proses pengumpulan dan hasil dari proses pengambilan data yaitu file csv berisi data dari twitter.



Gambar 2 Blok Diagram Sistem

Alur penelitian yang ditunjukkan pada Gambar 2 berupa blok diagram dengan 3 blok yang menunjukkan tahapan input terdiri dari *tweet* yang akan dilakukan analisis.
Pada tahap proses terdiri dari 5 sub tahapan meliputi :

1. Pelabelan dilakukan secara manual dibagi menjadi 3 label yang mewakili sentimen *tweet* yaitu positif, negatif, dan netral [9][10].
2. *Preprocessing* menjelaskan bagaimana proses awal terhahap teks untuk dipersiapkan menjadi data yang akan diolah lebih lanjut, terdiri dari beberapa proses sebagai berikut:
   a. *Case Folding* dan *Removing* merupakan proses untuk mengubah seluruh huruf di dalam dokumen menjadi huruf kecil dan menghapus karakter, tanda baca, angka, spasi dan username.
   b. *Tokenizing* merupakan proses untuk memisahkan setiap kata di dalam dokumen yang semula berupa kalimat-kalimat menjadi kata-kata.
   c. ***Filtering*** merupakan proses mengurangi jumlah kata-kata yang tidak penting.
   d. ***Stemming*** merupakan proses normalisasi yang digunakan untuk mencari kata dasar.
3. Mencari n-gram yaitu sub-urutan n karakter kata yang diberikan.
4. Pembobotan TF-IDF memiliki langkah awal menemukan nomor kata yang diketahui sebagai bobot atau *term frequency* di setiap dokumen setelah itu dilakukan pengalian *inverse document frequency*[11].
5. Klasifikasi terdiri dari 2 tahap yaitu pelatihan dan pengujian. Pelatihan dilakukan dengan data latih untuk membangun model klasifikasi dan pengujian dilakukan terhadap data uji untuk melihat hasil prediksi dari model yang sudah dibuat[13].

Dari proses klasifikasi tahap 5 akan didapatkan hasil prediksi dari data uji. Hasil tersebut yang menjadi Output penelitian ini. Dari setiap label yang berhasil diprediksi oleh model dihitung presentasenya untuk ditampilkan dalam bentuk Diagram. Kemudian hasil klasifikasi dilakukan evaluasi berupa perhitungan nilai akuransi.

III. HASIL DAN PEMBAHASAN

Data didapatkan dari twitter menggunakan Twint. Baris perintah untuk mengambil data :

```
twint  -s "perguruan tinggi yogyakarta" -o file.csv
--limit 999 --since YYYY-MM-DD --until YYYY-MM-DD -
-csv --lang id
```

Perintah twint -s digunakan untuk menerapkan *library* dengan mengumpulkan semua *tweet* berdasarkan kata kunci yang dituliskan dalam tanda petik dua (“ ”), kemudian output disimpan dalam file csv dengan perintah -o namafile.ekstensi --csv. Menggunakan --limit *tweet* yang diambil dibatasi sebanyak ± 150 untuk kata kunci universitas dan 175 kata kunci lainnya, selain itu *tweet* yang diambil dibatasi waktu menggunakan --since dan --until dengan format waktu YYYY-MM-DD yaitu sepanjang tahun 2022 sejak tanggal 1 Januari hingga 31 Desember dengan bahasa yang ditentukan menggunakan --lang adalah bahasa indonesia dengan kode bahasa id.

Data yang didapat terdiri dari 14 file csv terpisah sesuai dengan kata kunci yang telah ditentukan. Data file csv terdiri dari 34 kolom, kemudian diambil 1 kolom yaitu *tweet* dan disatukan menjadi 1 file dengan total 2304 baris data. Setelah itu, data diberikan label secara manual dan dilakukan penghapusan *tweet* ganda serta tidak memiliki makna sehingga tersisa 1710 baris data yang terdiri dari 1265 Netral, 143 Positif, 102 Negatif.

TABEL 1
DATA HASIL PELABELAN MANUAL

Tweet	Label
Universitas Sarjanawiyata Tamansiswa (UST) semakin dipercaya oleh masyarakat dan sampai hari ini jumlah calon mahasiswa baru yang mendaftar semakin meningkat...Biro Humas & KerjasamaUniversitas Sarjanawiyataâ€¦ https://www.instagram.com/p/Bzw5N8tHCsr/?igshid=1i6lxqvycybnrÂ â€¦	Positif
Jogja nih kota pelajar, banyak perguruan tinggi, banyak mahasiswa. Sistem transportasi massal ga mendukung. Motor-motor kelewat banyak dan semrawut. Mantap	Negatif
Liburan ke Jogja: Banyak alumni yang liburan ke Jogja menyempatkan mampir ke kampus. Seperti hari ini. Tak lupa mereka berfoto bersama di alun-alun APMD yang indah.	Netral
Tmn kampus masi ada yg di jogja emg?	Netral
Weits kita beda kampus beb :') w di jogja aku ga makek jakun tapi abu abu	Netral
Ist Akprind kampus kebanggaan para anak teknik Yogyakarta pic.twitter.com/xfmtFyX0Cg	Positif

Preprocessing merupakan tahap penting untuk dilakukan pada data sebelum dilakukan analisis sehingga memperoleh informasi yang dibutuhkan. Tahap ini terdiri dari mengubah huruf, menghapus tanda baca, menghapus *stopword* dan *stemming*.

a. *Case Folding* dan *Removing*

Proses *case folding* diperlukan untuk mengubah seluruh huruf dalam *tweet* menjadi huruf kecil, sehingga memiliki bentuk teks standar untuk menghindari perbedaan makna setiap kata degan bentuk yang berbeda. Selain *case folding*, pada tahap ini juga terdapat proses *removing* untuk menghilangkan karakter, tanda baca, angka, dan spasi berlebih.

```
def case_folding(tweet):
tweet = re.sub('(@[A-Za-z0-9]+)|([^A-Za-z
\t])|(\w+:\/\/\S+)|(pic\.[^\s]+)', '', tweet)
      return tweet.strip().lower()
```

Fungsi *case_folding* digunakan untuk proses *case folding* dan *removing*. Untuk menghapus *username*, karakter, huruf, dan url menggunakan fungsi regex(*regular expression*) yaitu barisan karakter yang mendefinisikan suatu pola pencarian yang terdapat pada *library* python yaitu re dengan fungsi sub() untuk mengganti karakter yang cocok pada pola pencarian dengan string yang ditentukan. Fungsi case_folding mengembalikan nilai variabel *tweet* dengan menerapkan fungsi strip() untuk menghapus spasi di awal dan akhir teks dan lower() untuk mengubah *tweet* menjadi huruf kecil.

TABEL 2
HASIL PROSES CASE FOLDING DAN REMOVING

Tweet	Case_Folding
Universitas Sarjanawiyata Tamansiswa (UST) semakin dipercaya oleh masyarakat dan sampai hari ini jumlah calon mahasiswa baru yang mendaftar semakin meningkat... Biro Humas & Kerjasama Universitas Sarjanawiyataâ€¦ https://www.instagram.com/p/Bzw5N8tHCsr/?igshid=1i6lxqvycybnrÂ â€¦	universitas sarjanawiyata tamansiswa ust semakin dipercaya oleh masyarakat dan sampai hari ini jumlah calon mahasiswa baru yang mendaftar semakin meningkatbiro humas kerjasamauniversitas sarjanawiyata

b. *Tokenizing*

Tokenizing adalah proses pemisahan teks menjadi potongan-potongan yang disebut token. *Tokenizing* disini dilakukan pada kata, yang artinya *tweet* dipisah menjadi himpunan kata.

```
def tokenizing(tweet):
      tokens = word_tokenize(tweet)
          return tokens
```

Fungsi *tokenizing* memisahkan setiap kata dalam *tweet* menggunakan fungsi word_tokenize() pada modul NLTK.

TABEL 3
HASIL PROSES TOKENIZING

Case_Folding	Tokenizing
universitas sarjanawiyata tamansiswa ust semakin dipercaya oleh masyarakat dan sampai hari ini jumlah calon mahasiswa baru yang mendaftar semakin meningkatbiro humas	['universitas', 'sarjanawiyata', 'tamansiswa', 'ust', 'semakin', 'dipercaya', 'oleh', 'masyarakat', 'dan', 'sampai', 'hari', 'ini', 'jumlah', 'calon', 'mahasiswa', 'baru', 'yang', 'mendaftar', 'semakin', 'meningkatbiro', 'humas',

kerjasamauniversitas sarjanawiyata	'kerjasamauniversitas', 'sarjanawiyata']

c. *Filtering* (*Stopword Removal*)

Filtering adalah proses mengambil kata-kata penting dari hasil token dengan menggunakan algoritma *stoplist*(membuang kata kurang penting) menggunakan *stopword*.

```
def filtering(tweet):
    factory = StopWordRemoverFactory()
    stopword_sastrawi =
factory.create_stop_word_remover()
    stopword_list =
[stopword_sastrawi.remove(word) for word  in
tweet]
     return stopword_list
```

Fungsi yang digunakan untuk proses *filtering*. Proses ini menggunakan *library* Sastrawi pada python dengan menggunakan StopWordRemoverFactory() untuk memberikan daftar kata tidak penting kemudian gunakan create_stop_word_remover() terhadap variabel factory dan hasil didapatkan dengan menghapus kata berdasarkan *stopword* sastrawi yang ada pada *tweet*.

TABEL 4
HASIL PROSES FILTERING

Tokenizing	Filtering
['universitas', 'sarjanawiyata', 'tamansiswa', 'ust', 'semakin', 'dipercaya', 'oleh', 'masyarakat', 'dan', 'sampai', 'hari', 'ini', 'jumlah', 'calon', 'mahasiswa', 'baru', 'yang', 'mendaftar', 'semakin', 'meningkatbiro', 'humas', 'kerjasamauniversitas', 'sarjanawiyata']	['universitas', 'sarjanawiyata', 'tamansiswa', 'ust', 'semakin', 'dipercaya', '', 'masyarakat', '', '', 'hari', '', 'jumlah', 'calon', 'mahasiswa', 'baru', '', 'mendaftar', 'semakin', 'meningkatbiro', 'humas', 'kerjasamauniversitas', 'sarjanawiyata']

d. *Stemming*

Stemming adalah proses yang diperlukan untuk memperkecil jumlah indeks yang bebeda dari suatu dokumen, juga untuk melakukan pengelompokan kata-kata lain yang memiliki kata dasar dan arti yang sama namun memiliki bentuk atau *form* yang berbeda karena mendapatkan imbuhan berbeda.

```
def stemming(tweet):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    stemmed_list = [stemmer.stem(word) for
word in tweet]
    return stemmed_list
```

Fungsi *stemming* menggunakan *library* Sastrawi di python dengan fungsi StemmerFactory() yang berisi kata dasar kemudian membuat fungsi stemmer dengan create_stemmer(). Hasil dari proses ini didapatkan dengan menerapkan fungsi stem pada kata yang ada di *tweet* berdasarkan *stemmer* yang ada.

TABEL 1
HASIL PROSES STEMMING

Filtering	Stemming
['universitas', 'sarjanawiyata', 'tamansiswa', 'ust', 'semakin', 'dipercaya', '', 'masyarakat', '', '', 'hari', '', 'jumlah', 'calon', 'mahasiswa', 'baru', '', 'mendaftar', 'semakin', 'meningkatbiro', 'humas', 'kerjasamauniversitas', 'sarjanawiyata']	['universitas', 'sarjanawiyata', 'tamansiswa', 'ust', 'makin', 'percaya', '', 'masyarakat', '', '', 'hari', '', 'jumlah', 'calon', 'mahasiswa', 'baru', '', 'daftar', 'makin', 'meningkatbiro', 'humas', 'kerjasamauniversitas', 'sarjanawiyata']

Pembobotan suatu kata sangat penting untuk merepresentasikan kata ke dalam bentuk numerik. Teknik yang digunakan dalam vektorisasi salah satunya adalah *Term Frequency - Inverse Document Frequency* (TF-IDF).

```
from sklearn.feature_extraction.text import
TfidfVectorizer

tfidf_vectorizer =
TfidfVectorizer(ngram_range=(1,2))
tfidf_text=
tfidf_vectorizer.fit_transform(df.tweet)
```

TF-IDF pada python dapat dilakukan dengan menggunakan kelas TfIdfVectorizer yang tergabung dalam kelas feature_extraction.text dari *library* sklearn. Pada variabel tfidf_vectorizer digunakan untuk mendefinisikan fungsi yang akan digunakan untuk pembobotan dengan mengubah nilai parameter ngram menjadi 1 dan 2 yang berarti menggunakan unigram dan bigram. Urutan n karakter yang dihasilkan terdiri dari minimal 1 kata dan maksimal 2 kata. Variabel tfidf_text digunakan untuk tokenisasi dan membangun kata dan mengkode ulang setiap dokumen menjadi vektor numerik, dimana fungsi fit() yang digunakan

untuk mempelajari kata dan transform() untuk mengkode ulang.

Berikut cara perhitungan TF-IDF berdasarkan program sklearn yang sudah dibuat.

- Tokenizing string menggunakan n-gram yaitu unigram dan bigram. Misalkan diberikan kalimat :
  1. tmn kampus mas yg jogja emg
  2. ist akprind kampus bangga anak teknik Yogyakarta

- Menghitung kemunculan token di setiap dokumen : $tf - idf(t,d) = tf(t,d) \times idf(t)$ dengan $idf(t) = \log\left(\frac{N+1}{df_i+1}\right) + 1$

  Dimana tf(t, d) = banyaknya kata-t dalam suatu dokumen/kalimat ke-d

  df(t) = banyaknya dokumen/kalimat yang mengandung kata ke-t

  N = total dokumen

  Misal menghitung bobot kata "Yogyakarta" pada kalimat kedua. Perhitungan log pada *library* TF-IDF di skelarn menggunakan logaritma natural, sehingga perhitungan menjadi:

  $N = 2$

  $tf_{yogyakartaa,d2} = 1$

  $df_{yogyakarta} = 1$

  $idf(t) = \ln\left(\frac{2+1}{1+1}\right) + 1 = 1{,}405465108$

  $tf - idf_{yogyakarta} = 1 \times 1{,}405465108$

  $= 1{,}405465108$

- Kemudian setiap baris di normalisasi dengan Euclidian(L2)

$$Vnorm = \frac{V}{||V||_2} = \frac{V}{\sqrt{V_1^2 + V_2^2 + \ldots + V_n^2}}$$

Data yang sudah diberikan label, melalui tahap *preprocessing* dan pembobotan TF-IDF kemudian dilakukan pengklasifikasian. Klasifikasi menggunakan *naïve bayes* terdiri dari pelatihan dan pengujian, data harus dibagi menjadi data latih dan data uji.

```
from sklearn.model_selection import
train_test_split
X_train, X_test, y_train, y_test =
train_test_split(tfidf_text.toarray(),
df['label'], test_size = 0.3)
```

Pembagian dataset menggunakan kelas train_test_split yang tergabung dalam kelas model_selection dari *library* sklearn. Data ditampung dalam 4 variabel, yaitu X untuk nilai tf-idf setiap data dan y berisi label setiap data yang akan dilatih dan diuji. Pembagian data terdiri dari 70% data latih dan 30% data uji. Kelas train_test_split sudah mampu menghasilkan pembagian data secara merata berdasarkan label, jadi dipastikan setiap label ada pada data uji dan data latih dengan ukuran yang telah ditentukan dan hasil sesuai dengan banyaknya data setiap label.

```
In [17]: y_train.value_counts()

Out[17]: Netral     863
         Positif    253
         Negatif     81
         Name: label, dtype: int64

In [18]: y_test.value_counts()

Out[18]: Netral     402
         Positif     90
         Negatif     21
         Name: label, dtype: int64
```

```
From sklearn.naive_bayes import GaussianNB
NBClassifier = GaussianNB().fit(X_train,
y_train)
predict_NBC= NBClassifier.predict(X_test)
```

Klasifikasi *Naïve Bayes* menggunakan kelas GuassianNB yang tergabung dalam kelas naïve_bayes dari *library* sklearn. Variabel NBClassifier digunakan untuk mendefiniskan fungsi GaussianNB() untuk menerapkan algoritma *Naïve Bayes* dan fungsi fit() digunakan untuk proses pembelajaran data latih oleh model atau membangun model. Variabel predict_NBC digunakan untuk memprediksi data uji berdasarkan model yang sudah dibuat.

Cara kerja algoritma *naïve bayes*, berikut adalah data yang akan digunakan.

Bobot	Label
0	Positif
0,168802	Positif
0	Negatif
0	Negatif
0,105912	Netral
0,190446	Netral
0,128315	?

Aturan Bayes :

$$P(v_j|x) = \frac{P(x|v_j)\,P(v_j)}{P(x)}$$

$$Posterior = \frac{Likelihood * Prior}{Evidence}$$

$$P(x) = \sum_{j=1}^{n} p(x|v_j)P(v_j)$$

Distribuasi Gaussian:

$$P(x|v_j) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Langkah penyelesaian :

1. Kelompokkan berdasarkan label, lalu hitung mean dan standar deviasi nya.

a. Netral

Bobot	Label
0,105912	Netral
0,190446	Netral

$$mean = \frac{0{,}105912 + 0{,}190446}{2} = 0{,}148179$$

$$S = \sqrt{\frac{(0{,}105912 - 0{,}148179)^2 + (0{,}190446 - 0{,}148179)^2}{2-1}} = 0{,}059775$$

b. Positif

Bobot	Label
0	Positif
0,168802	Positif

$$mean = \frac{0 + 0{,}168802}{2} = 0{,}084401$$

$$S = \sqrt{\frac{(0 - 0{,}084401)^2 + (0{,}168802 - 0{,}084401)^2}{2-1}} = 0{,}119361$$

c. Negatif

Bobot	Label
0	Negatif
0,154863	Negatif

$$mean = \frac{0 + 0{,}154863}{1} = 0{,}077432$$

$$S = \sqrt{\frac{(0 - 0{,}077432)^2 + (0{,}154863 - 0{,}077432)^2}{2-1}} = 0{,}109505$$

2. Hitung probabilitas *likelihood*

$$P(0{,}128315|Netral) = \frac{1}{\sqrt{2\pi(0{,}059775)^2}} e^{-\frac{(0{,}126315-0{,}148179)^2}{2\times(0{,}059775)^2}} = 0{,}007423$$

$$P(0{,}128315|Positif) = \frac{1}{\sqrt{2\pi(0{,}119361)^2}} e^{-\frac{(0{,}126315-0{,}084401)^2}{2\times(0{,}119361)^2}} = 0{,}029756$$

$$P(0{,}128315|Negatif) = \frac{1}{\sqrt{2\pi(0{,}109505)^2}} e^{-\frac{(0{,}126315-0{,}077432)^2}{2\times(0{,}109505)^2}} = 0{,}024111$$

3. Hitung probabilitas priori

$$P(Netral) = \frac{2}{6} = 0{,}33$$

$$P(Positif) = \frac{2}{6} = 0{,}33$$

$$P(Negatif) = \frac{2}{6} = 0{,}33$$

4. Hitung probabilitas posterior

$P(Netral \mid 0{,}128315) = 0{,}007423 \times 0{,}33 = 0{,}002449$

$P(Positif \mid 0{,}128315) = 0{,}029756 \times 0{,}33 = 0{,}009819$

$P(Negatif \mid 0{,}128315) = 0{,}024111 \times 0{,}33 = 0{,}007956$

Berdasarkan hasil perhitungan probabilitas posterior label positif lebih besar dari label negatif dan netral, maka label data adalah positif.

Hasil klasifikasi yang dilakukan terhadap 1710 data yang terdiri dari 70% data latih dengan 863 label Netral, 253 label Positif dan 81 label Negatif serta 30% data uji dengan jumlah 513 data berdasarkan hasil split data Gambar 4.8. Model dapat memprediksi label Netral sebanyak 343 dari 402 data, label Positif sebanyak 27 dari 90 data dan label Negatif sebanyak 3 dari 21 data.

Wordcloud merupakan salah satu visualisasi data teks, untum melihat frekuensi kemunculan kata dari seluruh data, dimana semakin besar ukuran kata pada gambar maka semakin sering kata tersebut muncul.

Gambar 3. Wordcloud tweet label Positif

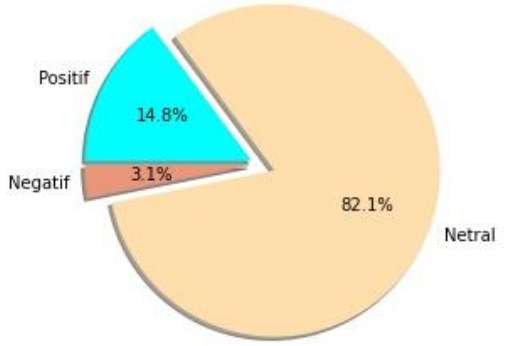
Berdasarkan Gambar 3 *wordcloud* untuk *tweet* dengan label Positif dapat diketahui kata-kata yang paling sering muncul adalah yang menulis tentang jogja, mahasiswa dan kampus.

Gambar 4. Wordcloud tweet label Negatif

Berdasarkan Gambar 4 *wordcloud* untuk tweet dengan label Negatif dapat diketahui kata-kata yang paling sering muncul adalah yang menulis tentang kata yg, kampus dan jogja.

Gambar 5. Wordcloud tweet label Netral

Berdasarkan Gambar 4.12 *wordcloud* untuk *tweet* dengan label Netral dapat diketahui kata-kata yang paling sering muncul adalah yang menulis tentang jogja, kuliah, dan beberapa nama kampus yang cukup banyak dituliskan pada *tweet*.



Gambar 6. Diagram Hasil Klasifikasi

Hasil analisis 30% data uji yang terdiri dari 513 data dari twitter berupa *tweet* mengenai perguruan tinggi di Yogyakarta yang diwakili melalui kata kunci 10 nama perguruan tinggi dengan rangking teratas dan kata kunci lainnya yang berkaitan sepanjang tahun 2022 menggunakan *Naïve Bayes* menghasilkan sentimen 82.1% netral, 14.8% positif, dan 3.1% negatif. Hal ini berarti bahwa hasil klasifikasi sentimen pengguna twitter paling banyak adalah netral, kemudian diikuti oleh positif dan paling sedikit adalah negatif. Pengguna twitter cenderung tidak memberikan pandangan positif atau negatif mengenai perguruan tinggi di Yogyakarta melalui twitter. Namun, pendapat positif masih lebih banyak daripada negatif sepanjang tahun 2022

IV. KESIMPULAN

Metode *Naïve Bayes* dapat mengklasifikasikan sentimen pengguna twitter terhadap perguruan tinggi di Yogyakarta kedalam sentimen positif, netral dan negatif. Penerapan unigram dan bigram pada pembobotan TF-IDF secara bersamaan dapat mengurangi ambiguitis pada setiap token yang dihasilkan. Hasil klasifikasi dari data uji didapat presentase sentimen netral 82.1%, positif 14.8% dan negatif 3.1%. Berdasarkan data uji model mampu memprediksi data dengan menghasilkan akurasi sebesar 73%.

DAFTAR PUSTAKA

[1] M. Kirana, N. Perkasa, M. Lubis, and M. Fani, "Visualisasi Kualitas Penyebaran Informasi Gempa Bumi di Indonesia Menggunakan Twitter", JAIC, vol. 3, no. 1, pp. 23-32, May 2019.

[2] D. Vonega, A. Fadila, and D. Kurniawan, "Analisis Sentimen Twitter Terhadap Opini Publik Atas Isu Pencalonan Puan Maharani dalam PILPRES 2024", JAIC, vol. 6, no. 2, pp. 129-135, Nov. 2022.

[3] B. Ramadhan, R. Adam, and I. Maulana, "Analisis Sentimen Ulasan pada Aplikasi E-Commerce dengan Menggunakan Algoritma Naïve Bayes", JAIC, vol. 6, no. 2, pp. 220-225, Dec. 2022.

[4] D. Kurniawan, A. Dzikri, and R. Permatasari, "E-Market Development for Fishermen and SMEs to Support Local Products in Hinterland Batam," presented at the Proceedings of the 2nd Multidisciplinary International Conference, MIC 2022, 12 November 2022, Semarang, Central Java, Indonesia, Feb. 2023. Accessed: Jul. 26, 2023. [Online]. Available: https://eudl.eu/doi/10.4108/eai.12-11-2022.2327385.

[5] R. Nooraeni, A. B. Safiruddin, A. F. Afifah, K. D. Agung, and N. N. Rosyad, "Analisis Sentimen Publik terhadap Sistem Zonasi Sekolah Menggunakan Data Twitter dengan Metode Naïve Bayes Classification," Fakt. Exacta, vol. 12, no. 4, Art. no. 4, Feb. 2020, doi: 10.30998/faktorexacta.v12i4.5205.

[6] Pamungkas, Dyarsa Singgih., Noor Ageng Setiyono dan Erlin Dolphina. 2015. Analisis Sentimen Pada Sosial Media Twitter Menggunakan Naïve Bayes Califier Terhadap Kata Kunci "Kurikulum 2013". Techno.COM, (hal. 299-314). Semarang: Universitas Dian Nuswantoro

[7] F. Pramono, Didi Rosiyadi, and Windu Gata, "Integrasi N-gram, Information Gain, Particle Swarm Optimation di Naïve Bayes untuk Optimasi Sentimen Google Classroom ", J. RESTI (Rekayasa Sist. Teknol. Inf.) , vol. 3, no. 3, pp. 383 - 388, Dec. 2019.

[8] R. S. Buana, W. Gata, A. Z. P. Widodo, H. . Setiawan, and K. Hilyati, "Analisis Sentimen pada Komen Twitter Pawang Hujan Mandalika dengan Support Vector Machine (SVM) dan Naïve Bayes", jtik, vol. 7, no. 2, pp. 194–200, Apr. 2023.

[9] Scikit-learn depelovers. 2020. Sklearn. feature_extraction. text.TfidfTransformer. Tersedia https://scikit-learn.org/stable/ modules/generated/sklearn.feature_extraction.text.TfidfTransformer. html#sklearn.feature_extraction.text.TfidfTransformer

[10] T. Amanattullah, H. Widyastuti, and F. Sari, "Identifikasi Fitur Laptop beserta Orientasinya dengan Metode Apriori dan Lexicon-Based", JAIC, vol. 1, no. 2, pp. 33-37, Dec. 2017.

[11] A. Sholihin, H. Haviluddin, N. Puspitasari, M. Wati, and I. Islamiyah, "Analisis Penyakit Difteri Berbasis Twitter Menggunakan Algoritma Naïve Bayes," Sains Apl. Komputasi Dan Teknol. Inf., vol. 1, no. 1, Art. no. 1, May 2019, doi: 10.30872/jsakti.v1i1.2215.

[12] T. E. Tarigan, R. C. Buwono, and S. Redjeki, "Extraction Opinion of Social Media in Higher Education Using Sentiment Analysis", bit-Tech, vol. 2, no. 1, pp. 11–19, Oct. 2019.

[13] A. P. Wijaya and H. A. Santoso, "Naive Bayes Classification pada Klasifikasi Dokumen Untuk Identifikasi Konten E-Government," J. Appl. Intell. Syst., vol. 1, no. 1, Art. no. 1, 2016, doi: 10.33633/jais.v1i1.1032.