# Perancangan *Dashboard* Sentimen Pariwisata Menggunakan Metode Complement Naïve Bayes pada Kabupaten Rembang

1st Matthew Ewaldo Amaniputra
*Fakultas Rekayasa Industri*
*Universitas Telkom*
Bandung, Indonesia
matthewewaldo@student.telkomuniversity.ac.id

2nd Rayinda Pramuditya Soesanto
*Fakultas Rekayasa Industri*
*Universitas Telkom*
Bandung, Indonesia
raysoesanto@telkomuniversity.ac.id

3rd Hilman Dwi Anggana
*Fakultas Rekayasa Industri*
*Universitas Telkom*
Bandung, Indonesia
hdadwi@telkomuniversity.ac.id

**Abstrak—Sumber devisa setiap negara yang saat ini menjadi sedang digalakan dimanapun adalah tempat pariwisata. Saat ini Indonesia mendapatkan sumber devisa yang tinggi salah satunya karena adanya pariwisata. Tempat wisata yang dijadikan tujuan destinasi para wisatawan sangat berpengaruh terhadap banyak tidaknya wisatawan yang datang ke tempat tersebut. Setiap tempat mempunyai daya tarik sendiri, sehingga sudah sewajarnya apabila pengelolaan terhadap daerah wisata tersebut harus menjadi fokus utama. Sebagai salah satu Kabupaten di Jawa Tengah, Kabupaten Rembang merupakan salah satu yang mempunyai daerah tempat tujuan wisata yang menjanjikan.**
**Output dari penelitian ini adalah performa Algoritma Naïve Bayes terkait komentar-komentar pengunjung terhadap beberapa pariwisata yang ada di Kabupaten Rembang yang dikemas dalam bentuk sebuah Dashboard yang menampilkan hasil dari analisis sentiment.**

**Kata Kunci—*dashboard, complementnaivebayes, google maps review***

## I. PENDAHULUAN

Sumber devisa setiap negara yang saat ini menjadi sedang digalakan dimanapun adalah tempat pariwisata. Saat ini Indonesia mendapatkan sumber devisa yang tinggi salah satunya karena adanya pariwisata. Tempat wisata yang dijadikan tujuan destinasi para wisatawan sangat berpengaruh terhadap banyak tidaknya wisatawan yang datang ke tempat tersebut. Setiap tempat mempunyai daya tarik sendiri, sehingga sudah sewajarnya apabila pengelolaan terhadap daerah wisata tersebut harus menjadi fokus utama. Sebagai salah satu Kabupaten di Jawa Tengah, Kabupaten Rembang merupakan salah satu yang mempunyai daerah tempat tujuan wisata yang menjanjikan. Capaian sasaran meningkatnya kontribusi pariwisata terhadap perekonomian daerah pada tahun 2021 sebesar 45,57%, jika disandingkan dengan capaian Tahun 2020 menurun 0,25% dari capaian kinerja Tahun 2020 sebesar 45,82% Nilai Realisasi Hasil Obyek Wisata merupakan indikator baru dalam perubahan RPJMD 2016-2021 yang berlaku untuk tahun 2020 hingga 2021. Adapun Realisasi Hasil Obyek Wisata selama kurun waktu dua tahun di Kabupaten Rembang adalah sebagai berikut:


GAMBAR I.1
(Nilai Obyek)

3.440 Milyar atau lebih rinci Rp 3.440.522.000, 45,57% dari Nilai Pencapaian Target Pariwisata 2021 dari tabel dan grafik Nilai Pencapaian Target Pariwisata 2021, yaitu Rp 7.550.000.000,- dengan 2020 Sebagai perbandingan nilai realisasi objek wisata mengalami peningkatan sebesar Rp.212.746.000,- (6,18%), dimana nilai realisasi pada tahun 2020 sebesar Rp.3.227.776.000,-. Sementara persentase pertumbuhannya turun 0,25% dari capaian kinerja 2020 sebesar 45,82%.

Tidak tercapainya target tersebut disebabkan dampak pandemi virus corona disease 2019 (Covid-19) yang berlangsung lebih lama dari perkiraan dan berdampak pada struktur perekonomian salah satunya pariwisata. Selama pandemi Covid-19, objek wisata ditutup sementara sehingga menyebabkan penurunan kunjungan wisatawan yang berdampak pada penurunan pendapatan objek wisata. Secara rinci, nilai realisasi hasil obyek wisata di Kabupaten Rembang pada tahun 2019-2021.

Adapun hambatan-hambatan yang ditemui dalam rangka pencapaian kinerja sasaran meningkatnya kontribusi pariwisata terhadap perekonomian daerah adalah sebagai berikut:

A. Adanya pandemi Corona Virus Disiase 2019 (Covid 19) yang sangat mempengaruhi pendapatan obyek wisata, pelaku usaha pariwisata dan ekonomi kreatif serta pelaku seni.

B. Aksesibiltas, amenitas, atraksi dan aktifitas pada destinasi pariwisata perlu ditingkatkan dan dikembangkan.
C. Kurang berkembangnya budaya lokal menjadi daya tarik wisata.
D. Pengembangan Daya Tarik Wisata perlu dioptimalkan.

Promosi potensi kebudayaan dan pariwisata perlu ditingkatkan.

Diketahui permasalahan dan turunannya yang membuat pariwisata di Kabupaten Rembang mengalami berkuangnya nilai pencapaian target pariwisata di Kabupaten Rembang. Tidak adanya analisis yang dilakukan mengenai komentar-komentar atau ulasan pengunjung sehingga terdapat ketidaktahuan *prespektif* pengunjung terhadap Pariwisata Rembang. Kurang berkembangnya budaya local yang menjadi daya tarik wisata dikarenakan kurang optimalnya kebijakan promosi yang dilakukan dan ada beberapa wisata local yang belum terdaftar. Adanya pandemic *Covid-19* juga menjadi masalah dikarenakan pada saat pandemic banyak destinasi wisata yang tutup akibatnya berkurangnya wisatawan yang berkunjung ke tempat-tempat wisata yang ada di Kabupaten Rembang.

Kemajuan teknologi pada kehidupan modern seperti saat ini menuntut masyarakat untuk selalu bergerak cepat dan *instant* dalam memenuhi segala aspek kebutuhan sehari-hari. Termasuk dalam hal penyampaian informasi, muncul berbagai macam *platform* yang digunakan untuk kebutuhan masyarakat tersebut. Dari *Google Maps* kami mendapatkan analisis komentar pengunjung. Pada ulasan pengunjung dapat dilihat positif atau negatif untuk tempat wisata. Untuk memeriksa ulasan positif atau negatif dapat dilakukan dengan menggunakan analisis sentimen. Analisis Sentimen adalah proses pemahaman mengekstrak dan memproses data teks secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam ulasan. Analisis sentiment dilakukan untuk melihat ulasan atau kecenderungan pendapat terhadap masalah atau item oleh seseorang yang memiliki pandangan atau pendapat negatif atau positif.

Adapun tujuan dari tugas akhir ini adalah menawarkan solusi dalam melakukan analisis sentiment terhadap ulasan pengunjung wisata di Kabupaten Rembang. Analisis sentiment ini dilakukan agar dapat mengukur atau mengetahui performa algoritma *Naïve Bayes Classifier* dalam melakukan klasifikasi berdasarkan ulasan pengunjung pariwisata di Kabupaten Rembang, sehingga dapat mempermudah Dinas Pariwisata dan Kebudayaan Kabupaten Rembang dalam memperhatikan perkembangan wisata dan memudahkan dalam pengambilan keputusan mengenai Langkah dalam mengembangkan pariwisata di Kabupaten Rembang

## II. KAJIAN TEORI

### *A. Sistem Informasi*

Sistem Informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan (Sutabri, 2016). Sistem informasi merupakan seperangkat komponen yang saling terkait yang mengumpulkan, memproses, menyimpan, dan menyebarkan data dan informasi. Sistem informasi menyediakan mekanisme umpan balik untuk memantau dan mengendalikan operasinya untuk memastikan terus memenuhi tujuan dan sasarannya. Mekanisme umpan balik sangat penting untuk membantu organisasi mencapai tujuannya, seperti meningkatkan keuntungan atau meningkatkan layanan pelanggan (Reynolds dan Stair, 2018).

### *B. Machine Learning*

Machine learning merupakan suatu ilmu yang membuat sistem dapat secara otomatis belajar sendiri tanpa harus berulang kali diprogram oleh manusia. Machine Learning sendiri merupakan salah satu disiplin ilmu dalam kecerdasan buatan atau yang sering biasa dikenal dengan *Artifical Intelligent* (AI). Machine Learning juga sering disebut dengan *Artifical Intelligent* (AI) konvensional karena merupakan kumpulan metode-metode yang digunakan dalam penerapannya. Machine Learning berfokus pada pengembangan program komputer yang dapat mengakses data dan menggunakannya untuk belajar sendiri. Sebelum Machine Learning bisa bekerja, maka ia membutuhkan data untuk training (latian) kemudian hasil dari training tersebut akan diuji atau di test dengan data yang sama atau bertolak belakang.

### *C. Pengolahan Bahasa Alami*

Pengolahan Bahasa Alami (PBA) atau *Natural Language Processing* (NLP) merupakan bagian penting dari text mining dan juga sub bidang dari kecerdasan buatan atau yang sering disebut dengan Artificial Intelligence atau AI. PBA mempelajari tentang bagaimana membuat komputer yang mampu mengerti dan memahami makna bahasa manusia, dengan cara mengubah bahasa manusia ke dalam dokumen atau teks dan menjadikannya lebih formal upaya lebih mudah untuk dimanipulasi oleh program komputer dan memberikan respon yang sesuai. Tujuan PBA adalah untuk melangkah melebihi manipulasi teks berbasis sintaks (yang sering kali disebut dengan ‘wordcounting’) ke pemahaman yang benar dan memproses bahasa alami dengan mempertimbangkan batasan semantik, gramatikal, dan konteks (Kumar, 2011). Komputer dapat memahami bahasa alami manusia dengan cara membuat gambaran Bahasa manusia dan mengubahnya menjadi angka-angka yang nantinya akan diproses dan dimasukkan ke dalam perhitungan dengan metode tertentu, yang kemudian akan menghasilkan respon kepada pengguna berupa tanggapan dari masukan yang telah diproses komputer sehingga membuat komputer terkesan dapat berinteraksi dengan pengguna menggunakan bahasa alami manusia.

PBA terdiri dari bagian utama yaitu:

1. *Parser*

*Parser* merupakan suatu sistem dimana dilakukan proses pengambilan kalimat input bahasa alami dan kemudian menguraikannya ke dalam beberapa bagian gramatikal (kata benda, kata kerja, kata sifat, dan lain-lain.

2. Sistem Representasi Pengetahuan

Sistem Representasi Pengetahuan merupakan sistem yang menganalisis output parser untuk menetukan maknanya yang nantinya akan menghasilkan outputsesuai dengan input yang diberikan *parser*.

3. *Output Translator*

*Output Translator* merupakan terjemahan yang mempresentasikan sistem pengetahuan dan melakukan langkah-langkah yang bisa berupa jawaban atas bahasa alami atau output khusus yang sesuai dengan program komputer.

Pada zaman modern seperti saat ini, PBA sudah banyak diimplementasikan ke dalam berbagai bidang. Beberapa diantara penerapannya seperti penjawab pesan otomatis atau sering disebut dengan chatbot, machine translation (aplikasi translator), spam filtering (pemfilteran pesan sampah), dan language identification (identifikasi bahasa), dan lain-lain. Ada beberapa tingkat pengelohan kata dalam PBA antara lain: *fonetik, morfologi, sintaksis, semantik*, *dicsource knowledge*, dan *pragmatik*.

*D. Naïve Bayes*

*Naïve Bayes Classification* merupakan teknikklasifikasi berdasarkan Teroema Bayes dengan asumsi indepedensi di antara para prediktor. Naïve Bayes Classifier memprediksi peluang di (Hmasa depan berdasarkan pengalaman di masa sebelumnya sehingga disebut Teorema Bayes. Dalam Bahasa sederhana, penggolongan Naïve Bayes menganggap bahwa kehadiran fitur tertentu di kelas tidak terkait dengan kehadiran fitur lainya (Hidayatullah, 2014). Naïve Bayes hanya membutuhkan jumlah data pelatihan (*training data)* yang kecil untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasian. Karena yang diasumsikan sebagai variabel independent, maka hanya varians dari suatu variabel dalam sebuah kelas yang dibutuhkan untuk menentukan klasifikasi, bukan keseluruhan dari matriks kovarians.

Rumus Bayes dalam (Muslehatin dkk, 2017) secara umum dapat diberikan sebagai berikut:

$$P(H|X) = \frac{P(H \mid X)\, P(H)}{P(X)}$$

Keterangan:

X = Data dengan class yang belum diketahui

H = Hipotesis data X merupakan suatu class spesifik

P(H|X) = Probabilitas hipotesis H berdasarkan kondisi x

P(H) = Probabilitas hipotesis H

P(X|H) = Probabilitas X berdasarkan kondisi tersebut

P(X) = Probabilitas dari X

I.1.1 *Text Mining*

Text mining adalah proses menambang data berupa informasi dan pengetahuan yang berguna dimana sumber data didapatkan dari dokumen atau teks, seperti dokumen Word, PDF, kutipan teks, atau sebagainya. Text mining sendiri memiliki tujuan untuk mencari kata-kata dan mendapatkan informasi yang berguna dimana informasi tersebut dapat mewakili isi dari dokumen yang berkaitan sehingga dapat dilakukan analisa yang berhubungan antar dokumen. Sumber data yang digunakan pada text mining adalah kumpulan teks yang memiliki format yang tidak terstruktur atau kurang terstruktur. Pada dasarnya, text mining bisa dipikir sebagai suatu proses (dengan dua langkah utama) yang mulai dengan memaksakan struktur ke berbagai sumber data berbasis teks yang diikuti dengan mengektrak informasi dan knowledge yang relevan dari data berbasis teks yang sudah terstruktur tersebut dengan menggunakan berbagai tools dan teknik data mining.

Text mining merupakan penerapan konsep dan teknik data mining untuk mencari pola dalam teks, yaitu proses penganalisisan teks guna mendapatkan informasi yang bermanfaat untuk tujuan tertentu. Berdasarkan ketidakteraturan struktur data teks, maka proses text mining memerlukan beberapa tahap awal yang pada intinya adalah mempersiapkan agar teks dapat diubah menjadi lebih terstruktur.

Pada saat ini, text mining sudah diterapkan di berbagai bidang, di antaranya:

1. *Information Extraction* (Ekstraksi Informasi)
2. *Topic Tracking* (Pelacakan Topik)
3. *Summarzation* (Peringkasan)
4. *Clustering* (Penggugusan)
5. *Concept Linking* (Penautan Konsep)
6. *Question Answering* (Penjawaban Otomatis)

*E. Analisis Sentimen*

Sentiment Analysis (SA) merupakan proses memahami dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat atau teks yang berupa opini. Tujuan dilakukan sentiment analysis untuk melihat pandangan atau pendapat teks yang berkaitan terhadap sebuah masalah atau objek, apakah cenderung berpandangan positif atau negatif. Sentiment analysis terdiri dari pemrosesan bahasa alami, analisis teks dan komputasi linguistik untuk mengidentifikasi sentimen dari suatu dokumen
(Vinodhini dan Chandrasekaran, 2015).
Sentiment Analysis dapat dibedakan berdasarkan sumber datanya, beberapa level yang sering digunakan dalam penelitian Sentiment Analysis adalah Sentiment Analysis pada level dokumen dan Sentiment Analysis pada level kalimat (Clayton, 2011). Berdasarkan level sumber datanya Sentiment Analysis terbagi menjadi 2 kelompok besar yaitu:

1. *Coarse-grained Sentiment Analysis*

Pada Sentiment Analysis jenis ini, Sentiment Analysis yang dilakukan adalah pada level dokumen. Secara garis besar fokus utama dari Sentiment Analysis jenis ini adalah menganggap seluruh isi dokumen sebagai sebuah sentiment positif atau sentiment negatif (Clayton, 2011).

2. *Fined-grained Sentiment Analysis*

Fined-grained Sentiment Analysis adalah Sentiment Analysis pada level kalimat. Fokus utama fined-greined Sentiment Analysis adalah menentukan sentiment pada setiap kalimat pada suatu dokumen, dimana kemungkinan yang terjadi adalah terdapat sentiment pada level kalimat yang berbeda pada suatu dokumen (Clayton, 2011).

*F. Classification*

Menurut Prasetyo (2012), klasifikasi merupakan suatu pekerjaan menilai objek data untuk memasukkannya ke

dalam kelas tertentu dari sejumlah kelas yang tersedia. Dalam klasifikasi ada dua pekerjaan utama yang dilakukan, yaitu pembangunan model sebagai *prototype* untuk disimpan sebagai memori dan penggunaan model tersebut untuk melakukan pengenalan/klasifikasi/prediksi pada suatu objek data lain agar diketahui di kelas mana objek data tersebut dalam model yang sudah disimpannya.

Model dalam klasifikasi mempunyai arti yang sama dengan kotak hitam, di mana ada suatu model yang menerima masukan, kemudian mampu melakukan pemikiran terhadap masukan tersebut, dan memberikan jawaban sebagai keluaran dari hasil pemikirannya.

*G. Split Validation*

Split validation merupakan teknik validasi yang membagi data menjadi dua bagian menjadi data training dan data testing. Metode yang dapat digunakan untuk memvalidasi silang model deret waktu adalah validasi silang secara bergulir, jadi dimulai dengan sebagian kecil data untuk pelatihan, dan dilanjutkan dengan data uji.

Karena data testing tidak digunakan untuk melatih model, maka model tidak mengetahui *outcome* dari data tersebut. Ini yang disebut dengan *out-of-sample testing*. Suatu model dikatakan bagus jika memiliki akurasi yang tinggi atau bagus untuk data *out-of-sample*, karena tujuan utama dibuatnya sebuah model tentunya adalah untuk memprediksi dengan benar data yang belum diketahui *outcome*-nya.

*H. Perfomance Evaluation Measure*

Performance Evaluation Measure (PEM) atau dalam Bahasa Indonesia bisa disebut pengukuran evaluasi performa adalah satu bundel tahapan yang digunakan untuk mengukur performa suatu sistem. PEM dalam banyak kasus digunakan dalam training data, tujuannya untuk mengevaluasi model yang sudah dibuat. Ada banyak perhitungan untuk mendapatkan nilai PEM, biasanya diterapkan sebagai kombinasi atau juga secara parsial. Beberapa perhitungan dalam PEM antara lain:

1. *Percision.*

   *Precision* adalah tingkat ketepatan antara request pengguna dengan jawaban sistem. Rumus *precision*(pre):

   $$\text{pre} = \frac{TP}{FP+TP}$$

2. *Accuration.*

   *Accuration* adalah pembandingan antara informasi yang dijawab oleh sistem dengan benar dengan keselurhan informasi. Rumus *accuration*(acc):

   $$\text{acc} = \frac{TN+TP}{FN+FP+TN+TP}$$

3. *Recall.*

   *Recall* adalah ukuran ketepatan antara informasi yang sama dengan informasi yang sudah pernah dipanggil sebelumnya. Rumus *recall*(rec)

   $$\text{rec} = \frac{TP}{FN+TP}$$

*I. Eksplorasi Data*

Analisis data eksploratif (*Exploratory Data Analysis – EDA*) merupakan metode eksplorasi data dengan menggunakan teknik aritmatika sederhana dan teknik grafis dalam meringkas data pengamatan. Eksplorasi data merupakan bagian yang integral dari persepsi kita. Apabila tujuan akhir dari penelitian bukan untuk menghasilkan inferensi kausal, analisis data selanjutnya sudah tidak diperlukan lagi. Namun apabila diperlukan, analisis data eksploratori sangat menunjang dalam menelaah dan menemukan tentang sifat-sifat data yang nantinya dapat berguna dalam menyeleksi model statistik yang tepat. Dengan demikian, pada analisis data eksploratif, sifat dari data pengamatanlah yang akan menentukan model analisis statistik yang sesuai (atau perbaikan dari analisis yang sudah direncanakan).

Langkah pertama dalam menganalisis data adalah mempelajari karakteristik dari data tersebut. Terdapat beberapa alasan penting yang perlu kita pertimbangkan secara cermat sebelum analisis data sebenarnya kita lakukan. Alasan pertama pemeriksaan data adalah untuk memeriksa kesalahan-kesalahan yang mungkin terjadi pada berbagai tahap, mulai dari pencatatan data di lapangan sampai pada entry data pada komputer. Alasan berikutnya adalah untuk tujuan eksplorasi data sehingga kita bisa menentukan model analisis yang tepat.

## III. METODE

*A. Pengumpulan Data*

Pengambilan data dilakukan dengan menggunakan bantuan dari Library Python *Selenium* dan diharuskan menginstall Psycharm dan Mozilla Firefox dengan versi terbaru.


GAMBAR III.1
(Lambang Mozilla Firefox)

Mozilla Firefox menjadi alat yang berjalan ototmatis dengan *script* yang sudah atau sedang dijalankan.


GAMBARIII.2
(Lambang PyCharm)

PyCharm sebagai software untuk menjalnkan file Pyhton secara otomatis atau menjalankan *script* untuk melakukan data *scraping* itu sendiri.

Langkah pertama yang harus dilakukan adalah pembuatan *Script* scraping data pada software Pycharm. Seperti mengimport beberapa library dan webdriver yang diperlukan untuk langkah awal *script* tersebut.

```
import asyncio
import time
from math import floor
import csv
from selenium import webdriver
from selenium.common.exceptions import NoSuchElementException
from selenium.webdriver.common.by import By
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.support.wait import WebDriverWait
from webdriver_manager.firefox import GeckoDriverManager
from selenium.webdriver.firefox.service import Service
from selenium.webdriver.firefox.options import Options
```

GAMBAR III.3
(Menginstall Library dan webdriver)

Selanjutnya adalah pembuatan argument yang berfungsi agar pada saat *script* di *run* proses *scraping* tetap berjalan otomatis tanpa adanya *visualisasi* proses *scraping* yang sedang berlangsung.

```
async def extract_google_reviews(query):
    options = Options()
    options.headless = True
    options.add_argument("--headless")
    options.add_argument("--no-sandbox")
    options.add_argument("--disable-dev-shm-usage")
    options.set_preference("dom.webdriver.enabled", False)
    options.set_preference("useAutomationExtension", False)
```

GAMBAR III.4
(Options add argument)

Langkah berikutnya adalah untuk mengambil data dari element yang ada di *web Google Reviews* tersebut. Pada kasus ini element yang diambil adalah *review* pengunjung pada Pariwisata Kabupaten Rembang.

```
driver = webdriver.Firefox(service=Service(GeckoDriverManager().install()),options=options)
driver.get("https://www.google.com/?hl=id")
driver.find_element(By.NAME, "q").send_keys(query)
WebDriverWait(driver, 5).until(
    EC.element_to_be_clickable((By.NAME, "btnK"))
).click()

reviews_header = driver.find_element(By.CSS_SELECTOR, "div.kp-header")
reviews_link = reviews_header.find_element(By.PARTIAL_LINK_TEXT, "ulasan Google")
number_of_reviews = int(reviews_link.text.replace(".", "").split()[0])
reviews_link.click()

all_reviews = WebDriverWait(driver, 3).until(
    EC.presence_of_all_elements_located(
        (By.CSS_SELECTOR, "div.gws-localreviews__google-review")
```

GAMBAR III.5
(Mengambil isi iframe)

Setelah itu kode program dilanjutkan seperti Gambar yang berfungsi untuk menjalankan jumlah *scroll page* atau *element* yang akan diambil datanya.

```
start = time.time()
last = 0
while len(all_reviews) < number_of_reviews:
    try:
        driver.execute_script("arguments[0].scrollIntoView(true);", all_reviews[-1])
        WebDriverWait(driver, 5, 0.25).until_not(
            EC.presence_of_element_located(
                (By.CSS_SELECTOR, "div[class$='activityIndicator']")
            )
        )
        all_reviews = driver.find_elements(
            By.CSS_SELECTOR, "div.gws-localreviews__google-review"
        )
        now = time.time()
        print(
            f"loading {len ( all_reviews )} reviews of {number_of_reviews} running for {floor ( now - start )} seconds."
        )
        print(len(all_reviews), " from ", number_of_reviews)
        if len(all_reviews) != last:
            writetocsv(all_reviews)
            last = len(all_reviews)
    except Exception as e:
        print(e)
        driver.implicitly_wait(1)
        pass
```

GAMBAR III.6
(Kode Program Jumlah Scroll Page)

Langkah terakhir yang dilakukan setelah semua *reviews* dari beberapa tempat wisata telah diperoleh adalah mengimport data yang sudah di dapatkan kedalam format *csv* adapun kode untuk program import data ke *csv* dapat dilihat di Gambar

```
def writetocsv(all_reviews):
    f = open("reviews.csv", "a", encoding="utf-8")
    wr = csv.writer(f)
    if len(all_reviews) <= 20:
        stop = 0
    else:
        stop = len(all_reviews) - 11
    for i in range(len(all_reviews) - 1, stop, -1):
        try:
            wr.writerow(
                [
                    all_reviews[i]
                    .find_element(By.CSS_SELECTOR, "span.review-full-text")
                    .get_attribute("textContent")
                ]
            )
        except NoSuchElementException:
            try:
                wr.writerow(
                    [
                        all_reviews[i]
                        .find_element(By.CSS_SELECTOR, "span[data-expandable-section]")
                        .get_attribute("textContent")
                    ]
                )
```

GAMBAR III.7.1
(Kode Program Import *csv)*

B. Eksplorasi Data

Sebelum *dataset* diolah lebih lanjut sebaiknya dilakukan eksplorasi data, Eksplorasi data merupakan langkah untuk memahami data sebelum dilakukan praproses. Pemahaman terhadap data yang akan di-mining dapat membantu dalam menentukan teknik-teknik pra-proses dan analisis data terhadap data sebelum dilakukan data mining.

Pada proses eskplorasi data penulis melakukan eksplorasi data dengan menggunakan bantuan *Microsoft excel* dengan membuat beberapa fitur penilaian yang biasa dijadikan acuan untuk menilai sebuah objek wisata, fitur yang digunakan adalah sebagai berikut:

TABEL III.1
(Fitur Penilaian Pariwisata)

| **Fitur** | **X** |
|---|---|
| Harga | 1 |
| Bagus | 2 |
| Bersih | 3 |
| Nyaman | 4 |

Pada Tabel nomor 1,2,3, dan 4 adalah pengganti dari kata fitur Harga, Bagus, Bersih, dan Nyaman secara berurutan, nomor tersebut digunakan untuk ekplorasi data karena eksplorasi data menggunakan rumus pada *Microsoft excel* sebagai berikut:

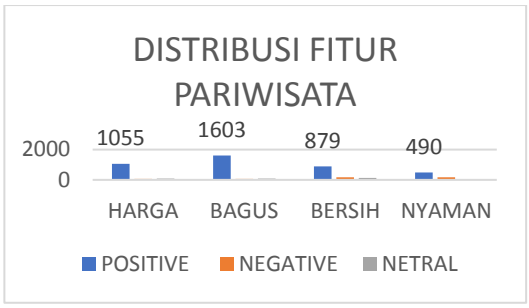
=MATCH(FALSE;ISERROR(SEARCH($I$2:$I$5;B2));0)

Rumus diatas digunakan untuk mencari kata-kata yang menjadi fitur penilaian pada *dataset* yang sudah di *labelling* sebelumnya, dan hasil dari esplorasi data dapat dilihat pada Tabel

TABEL III.2
(Hasil Eksplorasi Data)

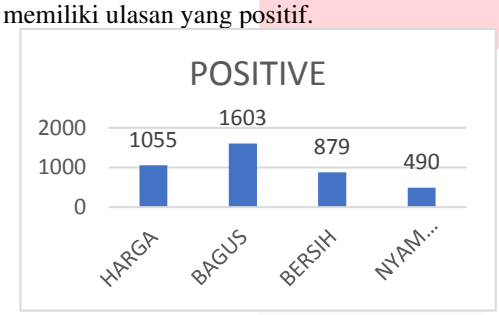
| | Positif | Negatif | Netral |
|---|---|---|---|
| Harga | 1055 | 58 | 86 |
| Bagus | 1603 | 74 | 90 |
| Bersih | 879 | 167 | 126 |
| Nyaman | 490 | 177 | 42 |

Setelah mendapatkan hasil dari Eksplorasi Data berdasarkan fitur penilaian pariwisata, data tersebut dibuat menjadi grafik sepert pada Gambar



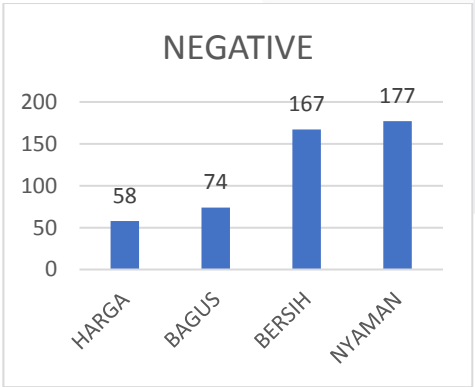

GAMBAR III.8
(Distribusi Fitur Pariwisata)

Setelah mendapatkan distribusi fitur pariwisata pada ulasan pengunjung yang sudah memiliki sentiment positif, negative, dan netral. Data tersebut kemudian di buat grafik berdasarkan fitur dan sentimen ulasan masing-masing supaya mempermudah dalam tahap analisisnya.
Pada Gambar IV.10 menunjukan fitur Bagus dan Harga memiliki jumlah ulasan terbanyak yang menunjukan pariwisata di Kab. Rembang pada fitur Bagus dan Harga sudah memiliki ulasan yang positif.



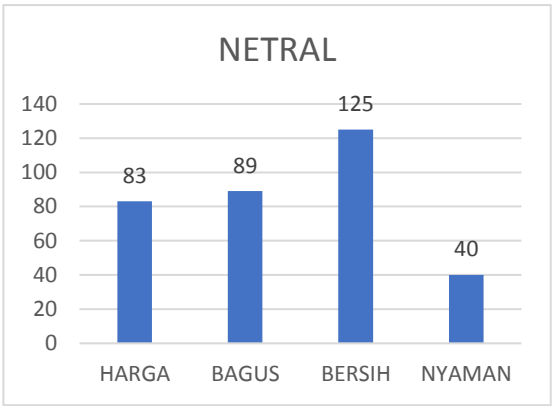

GAMBAR III.9
(Grafik Fitur Sentiemn Positif)

Pada Gambar menunjukan fitur Bersih dan Nyaman memiliki ulasan paling banyak pada sentiment negative yang menunjukan pariwisata di Kab. Rembang untuk fitur Bersih dan Nyaman masih memilik ulasan yang negative dan pada fitur tersebut Pemerintah Kabupaten Rembang dapat meningkatkan kebersihan dan kenyamanan pada tempat-tempat wisata yang ada.




GAMBAR III. 10
(Grafik Fitur Negatif)

Pada Gambar fitur bersih mendapatkan ulasan terbanyak pada proses eksplorasi data yang dilakukan peneliti.




GAMBAR III.11
(Grafik Fitur Netral)

*C. Pre-processing*

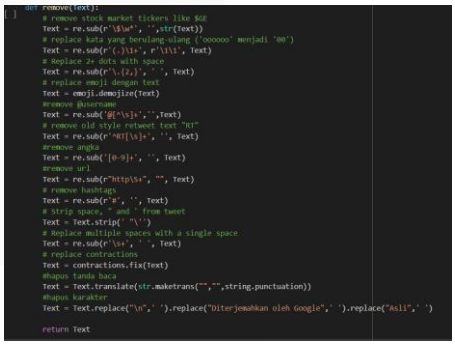
Tahapan ini terdiri dari beberapa proses karena data komentar tidak sepenuhnya menggunakan kata baku. Tahap *pre-processing* dilakukan dengan menggunakan bantuan library pada bahasa pemrograman Python3. Penerapan tahap *pre-processing* data pada penelitian ini dilakukan dengan melakukan 4 proses secara urut, di antaranya:

1. *Cleaning*

Proses cleaning pada tahap ini bertujuan untuk membersihkan data komentar dari hal yang tidak diperlukan seperti tanda baca, normalisasi unicode, dan sebagainya. Dalam melakukan proses cleaning tersebut dilakukan 4 tahapan untuk mendapatkan hasil yang maksimal, di antaranya ialah:

a. Menghapus tanda baca
b. Menghapus angka
c. Menyeragamkan huruf menjadi huruf kecil semua
d. Menghapus kelebihan spasi

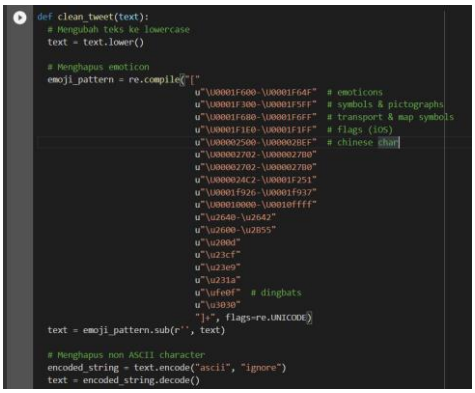
Adapun kode program yang memperlihatkan implementasi dari 4 tahap cleaning data ditunjukkan oleh Gambar




GAMBAR III. 12
(Kode Program tahap *cleaning)*

2. *Case Folding*

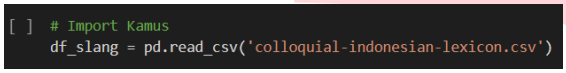
*Case Folding* adalah proses menyeragamkan bentuk kata-kata menjadi huruf kecil (*lowercase*) atau huruf kapital (*uppercase*). Pada penelitian yang penulis teliti ini proses case folding diseragamkan menjadi bentuk hufur kecil (*lowercase*).




GAMBAR III. 13
(Kode Program Tahap *Case Folding)*

3. *Slang Word*

Tahap *Slang Word* dilakukan dengan tujuan mengubah kata-kata tidak baku menjadi kata-kata yang baku, *Slang Word* dilakukan dengan bantuan kamus, berikut kode program untuk kamus yang digunakan.




GAMBAR III. 14
(Kode Program Import Kamus)

Adapun kode pemrograman yang menunjukan implementasi dari tahapan *Slang Word.*


GAMBAR III. 15
(Kode Program Tahap *Slang Word)*

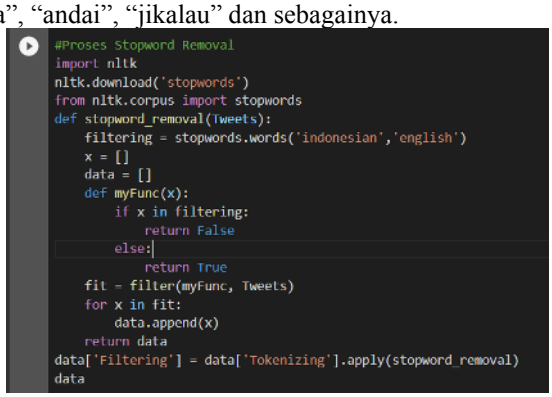
4. Tokenization

Tahap tokenization merupakan tahap dimana memisahkan kata, simbol, frase, dan entitas penting lainnya (yang disebut sebagai token) dari sebuah teks. Tahap tokenization dilakuan dengan menggunakan bantuan library pada Bahasa pemrograman Python3 yang bernama nltk.


GAMBAR III. 16
(Kode Program Tahap *Tokenization)*

5. *Filtering*

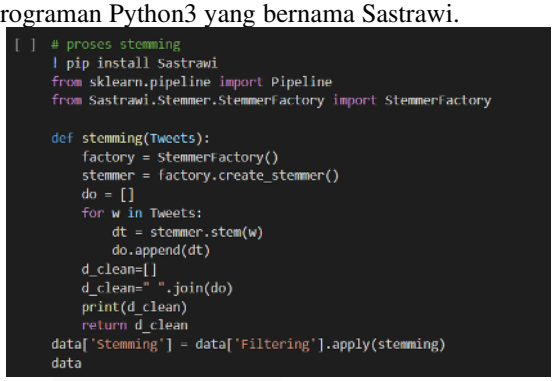
Tahap *Filtering* merupakan proses penghapusan kata-kata yang termasuk dalam stoplist. Stopword perlu dilakukan untuk efisiensi pemrosesan pada tahap berikutnya karena keberadaan stopword dianggap tidak mempengaruhi makna dari suatu kondokumen teks. Contoh dari kata-kata dalam stoplist bahasa Indonesia yaitu: “dan”, “seperti”, “meskipun”, “jika”, “andai”, “jikalau” dan sebagainya.




GAMBAR III. 17
(Kode Program Tahap *Filterring)*

6. *Stemming*

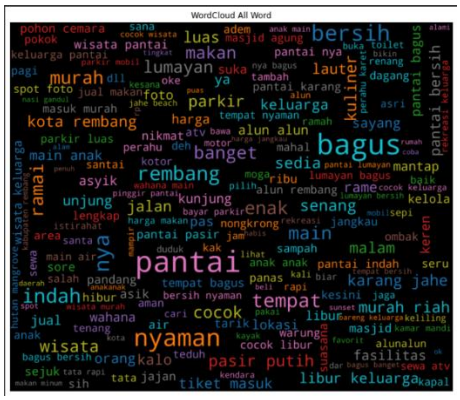
Tahap Stemming adalah tahap mengubah sebuah kata ke dalam bentuk kata dasarnya dengan menghapus kata imbuhan di depan maupun imbuhan di belakang kata. Tahap stemming dilakuan dengan menggunakan bantuan library pada bahasa pemrograman Python3 yang bernama Sastrawi.




GAMBAR III. 18
(Kode Program Tahap *Stemming)*

*D. Ekstrasi Fitur*

Setelah terbentuknya file yang akan dijadikan dataset, maka selanjutnya data tersebut akan dibentuk menjadi sebuah model klasifikasi. Namun sebelum membentuk model, ada beberapa tahapan yang harus dilakukan agar terbentuknya suatu model yang baik. Yang pertama dilakukan adalah membaca file xlsx dan kemudian dilakukan tokenisasi terhadap seluruh dokumen dalam file tersebut. Berdasarkan hasil tokenisasi yang dilakukan, maka penulis juga ingin mengetahui frekuensi kata yang banyak diperbincangkan oleh pengunjung wisata, untuk itu penulis memvisualisasikannya dalam bentuk *wordcloud.*




GAMBAR III. 19
(*WordCloud All Word)*

Berdasarkan hasil visualisasi kata dengan wordcloud, terlihat bahwa kata yang paling banyak dibicarakan dicetak lebih besar daripada kata yang lain. Dengan begitu kata pantai merupakan kata yang paling sering dibicarakan pada komentar yang diambil dari *Google Maps Review*. Berdasarkan hasil visualisasi tersebut dapat disimpulkan tempat yang paling sering dikunjungi ialah pantai. Selain kata pantai juga terdapat kata ‘pantai’, ‘bagus’, ‘nyaman’, dan ‘bersih’.

Pada proses ekstraksi fitur, tahap pertama yang dilakukan yaitu mengubah dataset ke dalam sebuah representasi vector dengan menggunakan library python yang bernama *CountVectorizer*.

Pada tahap selanjutnya tiap dokumen diwujudkan sebagai sebuah vektor dengan elemen, jika terdapat kata yang bersangkutan di dalam dokumen maka diberi nilai 1, jika tidak ada diberi nilai 0.

Pada penelitian ini proses pembuatan word vector dan pembobotan kata menggunakan bantuan library Python 3 yaitu *TfidfVectorizer.*

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| wiri | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| wis | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| wisat | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| wisata | 0.0 | 0.0 | 0.0 | 0.0 | 0.08285903613407235 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| wisatafasilitas | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| wisatajam | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| wisatarekomended | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| wisatasemua | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| wisatawan | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.2034044570054435 | 0.0 | 0.0 |
| wisel | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| wiskul | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| wisman | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| wistimir | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| with | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| wiuhh | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| wkk | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| wkwk | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| wkwkkoiya | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| wkwkkw | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| wkwkw | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| wkwkwk | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| wkwkwkk | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| wkwkwkkwk | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| wkwkwkrekomended | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| wkwkwm | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

GAMBAR III. 20
(Data *Word Vector* yang sudah Terbobot).

### *III.5 Implementasi Klasifikasi Naïve Bayes*

Dengan menggunakan Split Validation akan dilakukan percobaan training berdasarkan split ratio yang telah ditentukan sebelumnya, untuk kemudian sisa dari split ratio data training akan dianggap sebagai data testing. Data training adalah data yang akan dipakai dalam melakukan pembelajaran sedangkan data testing adalah data yang belum pernah dipakai sebagai pembelajaran dan akan berfungsi sebagai data pengujian kebenaran atau keakurasian hasil pembelajaran. Namun disini penulis menggunakan data training 0.9 (90%) dan data testing 0.1(10%).

```
# splitting data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(text_tf, data['label'], test_size=0.1, random_state=42)
print("X_train_shape : ",X_train.shape)
print("X_test_shape : ",X_test.shape)
print("y_train_shape : ",y_train.shape)
print("y_test_shape : ",y_test.shape)

X_train_shape :  (7183, 6923)
X_test_shape :  (799, 6923)
y_train_shape :  (7183,)
y_test_shape :  (799,)
```

Gambar III. 21 Kode Program *Split Validation*

X_train: Untuk menampung data source yang akan dilatih. X_test: Untuk menampung data target yang akan dilatih. y_train: Untuk menampung data source yang akan digunakan untuk testing. y_test: Untuk menampung data target yang akan digunakan untuk testing. Perlu diketahui bahwa metode ini akan membagi train set dan test set secara random atau acak. Jadi, jika kita mengulang proses running, maka tentunya hasil yang didapat akan berubah-ubah. Untuk mengatasinya, kita dapat menggunakan parameter random_state.

Algoritma Complement Naive Bayes merupakan modifikasi dari algoritma standar Multinomial Naive Bayes. Multinomial Naive Bayes tidak dapat bekerja dengan baik dengan data yang tidak stabil. Kumpulan data tidak seimbang adalah contoh di mana jumlah contoh milik kelas tertentu lebih besar dari jumlah contoh milik kelas yang berbeda. Ini menyiratkan penyebaran contoh tidak merata..

```
[12] from sklearn.naive_bayes import MultinomialNB  # Naive Bayes Classifier model Multinomial
     from sklearn.naive_bayes import ComplementNB # Naive Bayes Classifer Model Complement
     model_naive = ComplementNB().fit(X_train, y_train)
     predicted_naive = model_naive.predict(X_test)
```

Gambar III. 22 Kode Program Klasifikasi Naive Bayes

Pada proses klasifikasi yang dilakukan, penulis menggunakan data testing yang diambil sacara acak dari 10% data training.

```
[13] print(predicted_naive)

     'Positive' 'Positive' 'Positive' 'Negative' 'Neutral' 'Positive'
     'Positive' 'Negative' 'Positive' 'Negative' 'Positive' 'Positive'
     'Positive' 'Positive' 'Neutral' 'Positive' 'Positive' 'Positive'
     'Positive' 'Positive' 'Positive' 'Positive' 'Neutral' 'Positive'
     'Positive' 'Positive' 'Negative' 'Positive' 'Positive' 'Positive'
     'Positive' 'Positive' 'Positive' 'Positive' 'Positive' 'Positive'
     'Positive' 'Positive' 'Positive' 'Positive' 'Positive' 'Negative'
     'Neutral' 'Positive' 'Positive' 'Positive' 'Positive' 'Positive'
     'Positive' 'Positive' 'Positive' 'Positive' 'Positive' 'Positive'
     'Positive' 'Positive' 'Positive' 'Negative' 'Positive' 'Positive'
     'Neutral' 'Positive' 'Positive' 'Neutral' 'Positive' 'Positive' 'Neutral'
     'Positive' 'Positive' 'Positive' 'Negative' 'Positive' 'Positive'
     'Negative' 'Positive' 'Positive' 'Positive' 'Positive' 'Positive'
     'Positive' 'Positive' 'Positive' 'Positive' 'Positive' 'Neutral'
     'Positive' 'Positive' 'Negative' 'Positive' 'Positive' 'Positive'
     'Neutral' 'Positive' 'Negative' 'Positive' 'Positive' 'Positive'
     'Positive' 'Positive' 'Positive' 'Positive' 'Positive' 'Positive'
     'Neutral' 'Positive' 'Positive' 'Positive' 'Positive' 'Positive'
     'Positive' 'Negative' 'Positive' 'Positive' 'Positive' 'Positive'
     'Positive' 'Positive' 'Negative' 'Positive' 'Positive' 'Positive'
     'Positive' 'Positive' 'Positive' 'Negative' 'Positive' 'Positive'
     'Positive' 'Positive' 'Positive' 'Positive' 'Positive' 'Positive'
     'Positive' 'Positive' 'Positive' 'Positive' 'Positive' 'Positive'
```

Gambar III. 23 Hasil Prediksi Data *Testing*

## IV. HASIL DAN PEMBAHASAN

### *A. Uji Model*

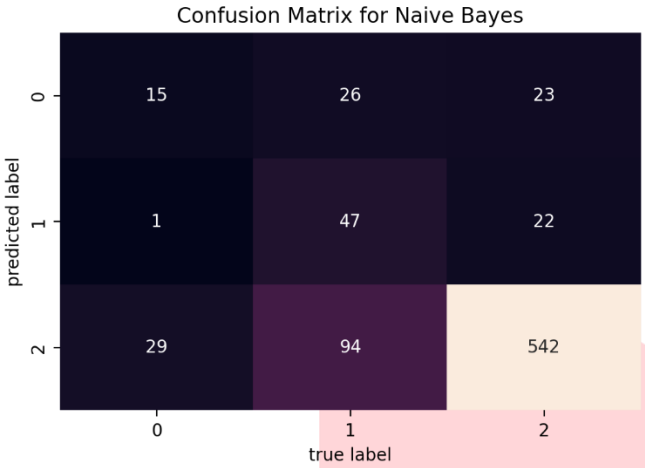
Untuk mengetahui performa dari Algoritma *Naive Bayes*, maka dilakukan pengujian terhadap model. Hasil klasifikasi akan ditampilkan dalam bentuk confusion matix. Tabel confusion matrix terdiri dari kelas predicted dan kelas actual. Model confusion matrix 3x3 ditunjukkan pada Tabel

TABEL IV.3
(*Confusion Matrix)*

| | | ***Predict Class*** | | |
|---|---|---|---|---|
| | | ***Class A*** | ***Class B*** | ***Class C*** |
| | ***Class A*** | … | … | … |

| | | | | |
|---|---|---|---|---|
| ***Actual Class*** | ***Class B*** | **…** | **…** | **…** |
| | ***Class C*** | **…** | **…** | **…** |

Pada proses uji model yang dilakukan menghasilkan nilai akurasi dan confusion matrix 3x3 yang dapat dilihat pada gambar.




GAMBAR IV.2
(*Confusion Matrix)*

Pada Gambar V.1 angka 2 melambangkan kelas positif, angka 1 melambangkan kelas netral, dan angka 0 melambangkan kelas negative.

*B. Evaluasi Model*

Evaluasi model dilakukan setelah proses uji model selesai dilakukan. Evaluasi model dilakukan untuk menghitung performa metode yang dipilih. Pada proses uji model yang dilakukan menghasilkan confusion matrix dengan ukuran 3x3 yang dapat dilihat pada Tabel

TABEL III.4
(Hasil *Confusion Matrix)*

| | | ***Predict Class*** | | |
|---|---|---|---|---|
| | | ***Positive*** | ***Neutral*** | ***Negative*** |
| ***Actual Class*** | ***Positive*** | **542** | **94** | **29** |
| | ***Neutral*** | **22** | **47** | **1** |
| | ***Negative*** | **23** | **26** | **15** |
| | **Jumlah** | **587** | **167** | **45** |

Seperti yang terlihat pada tabel V.2, confused matrix berupa matriks dengan ukuran 3x3, dimana setiap kolomnya mewakili setiap kelas yaitu kelas positif, kelas negatif, dan kelas netral.

Apabila ingin mendapatkan nilai true positif, true negatif, false positif dan false negatif dalam confused matrix dengan ukuran matriks 2x2 di setiap kelas, maka akan menjadi seperti berikut:

1. Kelas Positif

TABEL III.5
(*Confusion Matrix* Kelas Positif)

| **n=772** | **Aktual: *Positive*** | **Aktual: Bukan** |
|---|---|---|
| **Prediksi: *Positive*** | **542** | **123** |
| **Prediksi: Bukan** | **45** | **62** |
| **Jumlah** | **587** | **185** |

Pada Tabel menjelaskan tentang dari jumlah data testing sebanyak 772 komentar terbagi kedalam 542 TP, 123 FP, 45 FN, dan 62 TN pada kelas Positif. Maka dapat dihitung nilai *accuracy, precision, recall,* dan *F-1 score* sebagai berikut:

*Precision* = (TP) / (TP + FP)
= (543) / (543+123)
= 0.815315

*Recall* = TP / (TP + FN)
= 543 / (543+45)
= 0.923469

*F-1 score* = (2 x Recall x Precision) / (Recall + Precision)
= 0.866029

2. Kelas Negatif

TABEL III.6
(*Confsuin Matrix* Kelas Negatif)

| | **Aktual: *Negative*** | **Aktual: Bukan** |
|---|---|---|
| **Prediksi: *Negative*** | **15** | **49** |
| **Prediksi: Bukan** | **30** | **589** |
| **Jumlah** | **45** | **638** |

Pada Tabel menjelaskan tentang dari jumlah data testing sebanyak 772 komentar terbagi kedalam 15 TP, 49 FP, 30 FN, dan 589 TN pada kelas Positif. Maka dapat dihitung nilai *accuracy, precision, recall,* dan *F-1 score* sebagai berikut:

*Precision* = (TP) / (TP + FP)
= (15) / (15+49)
= 0.234375

*Recall* = TP / (TP + FN)
= 15 / (15+30)
= 0.333333

*F-1 score* = (2 x Recall x Precision) / (Recall + Precision)
= 0.275229

3. Kelas Netral

TABEL III.7
(*Confusion Matrix* Kelas Netral)

| | **Aktual: *Neutral*** | **Aktual: Bukan** |
|---|---|---|
| **Prediksi: *Neutral*** | **47** | **23** |
| **Prediksi: Bukan** | **120** | **557** |
| **Jumlah** | **167** | **580** |

Pada Tabel menjelaskan tentang dari jumlah data testing sebanyak 772 komentar terbagi kedalam 47 TP, 23 FP, 120 FN, dan 557 TN pada kelas Positif. Maka dapat dihitung nilai *accuracy, precision, recall,* dan *F-1 score* sebagai berikut:

*Precision* = (TP) / (TP + FP)
= (47) / (47+23)
= 0.671429

*Recall* = TP / (TP + FN)
= 47 / (47+120)
= 0.281437

*F-1 score* = (2 x Recall x Precision) / (Recall + Precision)
= 0.396624

Sehingga berdasarkan rumus nilai presisi pada keseluruhan sistem dapat dihitung dan sebesar 0.819 dan untuk nilai recall keseluruhan sistem sebesar 0.755. Sedangkan untuk nilai f-1 score untuk evaluasi dalam informasi temu kembali yang mengkombinasikan nilai presisi dan recall sebesar 0.777. Untuk menghitung nilai presisi, recall dan f-1 score pada sistem dapat menggunkan metode pada gambar

```
[15] from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
     from sklearn.metrics import classification_report


     print(classification_report(y_test, predicted_naive))
     print("Accuracy : ", accuracy_score(predicted_naive, y_test))
     print("Precision : ", precision_score(predicted_naive, y_test, average = 'weighted'))
     print("Recall : ", recall_score(predicted_naive, y_test, average = 'weighted'))
     print("F1_score : ", f1_score(predicted_naive, y_test, average = 'weighted'))
```

GAMBAR III.3
(Kode Program Perhitungan Nilai *Precision, Recall,* dan *F-1 Score)*

Dengan mengetahui besarnya nilai presisi, recall dan f-1 score pada kinerja keseluruhan sistem dapat dikatakan tingkat kemampuan sistem dalam mencari ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem dan tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi sebesar 75%.

Selanjutnya untuk performa metode klasifikasi dari setiap kelasnya dapat dilihat melalui nilai presisi, recall, dan f1-score pada setiap kelasnya. Hasil nilai presisi, recall, dan f1-score memiliki nilai sebesar 0-1. Semakin tinggi nilainya maka semakin baik. Hasil dari keseluruhan proses evaluasi model dapat dilihat pada Gambar.

```
              precision    recall  f1-score   support

    Negative       0.23      0.33      0.28        45
     Neutral       0.67      0.28      0.40       167
    Positive       0.82      0.92      0.87       587

    accuracy                           0.76       799
   macro avg       0.57      0.51      0.51       799
weighted avg       0.75      0.76      0.73       799

Accuracy :  0.7559449311639549
Precision :  0.8198427722977244
Recall :  0.7559449311639549
F1_score :  0.7774032104407894
```

GAMBAR III.4
(Hasil Pengukuran Evaluasi Performa)

Hasil presisi, recall, dan f-1 score di setiap kelasnya dapat dilihat pada Tabel

TABEL II.8
(Nilai *Precision, Recall,* dan *F-1 Score)*

| Jenis Klasifikasi | *Precision* | *Recall* | *F-1 Score* |
|---|---|---|---|
| ***Negative*** | **0,23** | **0,33** | **0,28** |
| ***Netral*** | **0,67** | **0,28** | **0,40** |
| **Positive** | **0,82** | **0,92** | **0,87** |

Hasil dari evaluasi model dapat dilihat nilai presisi dan recall di setiap kelasnya dapat dikatakan tingkat kemampuan sistem dalam mencari ketepatan antara informasi yang diminta oleh pengguna untuk kelas positif sebesar 82%, untuk kelas negatif sebesar 23%, kelas netral sebesar 67%. Sedangkan tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi untuk kelas positif sebesar 92%, untuk kelas negatif sebesar 33%, kelas netral sebesar 28%.

## V. KESIMPULAN

Berdasarkan hasil dari perancangan Tugas Akhir yang telah dilakukan, dapat disimpulkan bahwa perancangan Tugas Akhir ini menghasilkan Dashboard Visualisasi Hasil Analisis Sentimen yang didharapkan dapat membantu Pemerintah Kabupaten Rembang dalam melakukan analisis terhadap prespektif yang dimiliki pengunjung. Perancangan Tugas Akhir ini menggunakan Algoritma Naïve Bayes, dalam penelitian ini performa Algoritma Naïve Bayes terbukti akurat karena menghasilkan nilai akurasi sebesar 0,755 atau 76%. Pengujian Dashboard Visualisasi Hasil Analisis Sentimen ini menggunakan metode verifikasi greybox, yang dimana pengujian tersebut memiliki hasil yang memuaskan karena fitur dan proses yang ada pada sistem berhasil dalam menjalankan prosesnya.

## REFERENSI


[1] A. Rahman, E. Utami, and S. Sudarmawan. (2020) Sentimen Analisis Terhadap Aplikasi pada Google Playstore Menggunakan Algoritma Naïve Bayes dan Algoritma Genetika, J. Komtika Komputasi.

[2] Chandrasekaran., V. &. (2012). Sentiment Analysis and Opinion Mining: A Survey. International Journal of Advanced Research in Computer Science and Software Engineering.

[3] Clayton, F. (2011). Coarse-and Fine-Grained Sentiment Analysis of Social Media Text

[4] Hidayatullah, A. F. (2014). ANALISIS SENTIMEN DAN KLASIFIKASI KATEGORI TERHADAP TOKOH PUBLIK PADA TWITTER. Seminar Nasional Informatika 2014

[5] Kumar, E. (2011). NATURAL LANGUAGE PROCESSING. New Delhi: LK. Intrnational Publishing House Pvt, Ltd

[6] Muslehatin, W. I. (2017). Penerapan Naïve Bayes Classification untuk Klasifikasi Tingkat Kemungkinan Obesitas Mahasiswa Sistem Informasi UIN Suska Riau. Seminar Nasional Teknologi Informasi, Komunikasi dan Industri, 250-256..

[7] Prasetyo, E. (2012). Data Mining Konsep dan Aplikasi menggunakan MATLAB. Yogyakarta: Penerbit Andi.

[8] Reynolds, G. W. (2018). Principles of Information Systems. Boston: Cengage Learning

[9] Sutabri, T. (2016). Sistem Informasi Manajemen. Yogyakarta: Andi Offset.

[10] Wilianto, L., Rakhmat Umbara, F., & Hendro Pudjiantoro, T. (2017). ANALISIS SENTIMEN TERHADAP TEMPAT WISATA DARI KOMENTAR PENGUNJUNG DENGAN MENGGUNAKAN METODE NAÏVE BAYES CLASSIFIER STUDI KASUS JAWA BARAT. Prosiding SNATIF Ke -4.