Laporan Projek Tugas Akhir Mata Kuliah

Informatika Pariwisata Kelas A

Analisis Sentiment Review Tempat Pariwisata dengan Automated Lexicon Word2Vec dan Naive Bayes

(Studi Kasus: Pantai Toroan, Kabupaten Sampang)

**200411100082 R. Bella Aprillia Damayanti**

**200411100186 Triasmi Dwi Farawati**

# Abstrak

Sistem analisis sentimen merupakan sistem yang digunakan untuk melakukan proses analisis otomatis pada Pantai Toroan Kabupaten Sampang, Madura. Dari proses yang di Scraping melalui situs Google Maps dengan menggunakan tools Instant Data Scraper total data yang diperoleh sejumlah 1075 data dimana terdapat ulasan atau data yang NaN (data kosong) sejumlah 198 data. Setelah melakukan drop data maka jumlah total data yang akan di proses sebanyak 877 data. Data tersebut diklasifikasikan menggunakan Naive Bayes. Sistem analisis sentimen dibagi menjadi 5 (lima) tahap, yaitu crawling, pre-processing, pembobotan kata, pembentukan model dan klasifikasi sentimen. Pada pembobotan kata digunakan metode Metode Word2Vec merupakan hasil pembelajaran dari algoritme word embedding yang bertujuan untuk melakukan pemetaan dari kata menjadi vektor. Metode ini bertujuan untuk membuat leksikon otomatis berupa kata-kata positif dan negatif. Data tersebut kemudian akan dievaluasi menggunakan pengujian confusion matrix dengan parameter akurasi, recall, dan precision. Hasil pengujian menunjukkan pada pengujian 2 kelas (negatif dan positif) dan telah diuji dengan 80% data latih dan 20% data uji dengan nilai akurasi 57%, recall 0,87% dan precision 0,54. Hasil prediksi kelas data uji yang relevan dan kelas sentimen yang dihasilkan oleh sistem analisis sentimen walaupun belum sepenuhnya akurat

Kata kunci - Naive bayes, Klasifikasi sentimen, Word2vec, Confusion matrix, Akurasi

# Pendahuluan

Objek wisata merupakan salah satu sektor penunjang sumber perekonomian masyarakat dan juga sebagai salah satu sumber pendapatan penghasilan pemerintah daerah. Kabupaten Sampang merupakan daerah yang memiliki keunggualan dalam segi letak gografis, kekayaan alam, dan budaya yang tinggi sebagai modal daya tarik untuk pengembangan tempat objek wisata sehingga dapat mendatangkan banyak parawisatawan[3].

Sistem analisis sentimen yang dibangun menggunakan algoritma klasifikasi Naive Bayes. Ciri utama dari algoritma Naive Bayes adalah asumsi yang sangat kuat (naif) akan independensi dari masing-masing kondisi atau kejadian[7].

Analisis sentimen berfungsi untuk menemukan pola pada kalimat lalu mengelompokkan ke dalam sentimen positif atau negatif. Adapun metode yang digunakan pada penelitian ini adalah Word2Vec dan Naive-Bayes. Metode Word2Vec merupakan hasil pembelajaran dari algoritme word embedding yang bertujuan untuk melakukan pemetaan dari kata menjadi vektor. Metode ini bertujuan untuk membuat leksikon otomatis berupa kata-kata positif dan negatif. Selanjutnya, klasifikasi data tersebut ke dalam sentimen positif atau negatif dengan menggunakan metode Naive-Bayes. Naive-Bayes berfungsi untuk mencari nilai probabilitas tertinggi pada data. [1],[2].

Algoritma Naïve Bayes merupakan salah satu algoritma yang termasuk kedalam klasifikasi. Metode ini merupakan metode terbaik dalam pengklasifikasian yang dikemukakan oleh Thomas Bayes dengan memprediksi peluang di masa depan berdasarkan data di masa sebelumnya. Naive Bayes menerapkan fungsi statistik sederhana berdasarkan teorema bayes dengan asumsi keberadaan dari suatu fitur tertentu terhadap suatu kelas yang tidak berhubungan dengan fitur lainnya. Naïve Bayes merupakan suatu metode klasifikasi yang menggunakan perhitungan probabilitas [4],[5].

Tugas dasar sentiment analysis adalah mengklasifikasikan beberapa teks dari dokumen, kalimat atau fitur, kalimat dari fitur tersebut bisa bersifat positif, negative dan netral. Pada tahapan sentiment analysis terdapat extraksi fitur yang beguna mengubah teks menjadi data yang dapat diklasifikasi[6],[7].

# Metode Usulan

Dalam penelitian ini menggunakan metode Word2Vec untuk proses word embeddingnya dan metode Naive Bayes untuk klasifikasinya.

Naïve Bayes merupakan suatu metode klasifikasi yang menggunakan perhitungan probabilitas[4].

Persamaan dari probabilitas prior:

P(H) = Nj

——

N

Nj : Jumlah data pada suatu class

N : Jumlah total data

Persamaan dari teorema bayes:

P(H|X) = P(X\H).P(H)

——————

P(X)

X : Data class belum diketahui

H : Hipotesis data class spesifik

P(H|X) : Probabilitas Hipotesis H terhadap kondisi X

P(H) : Probabilitas Hipotesis H

P(X|H) : Probabilitas X terhadap kondisi hipotesis H

P(X) : Probabilitas X

Text Preprocessing merupakan tahapan dari proses awal terhadap teks untuk mempersiapkan teks menjadi data yang akan diolah lebih lanjut. Data yang didapat dari hasil crawling belum bisa langsung diklasifikasikan karena data tersebut masih terdapat banyak simbol dan kata-kata yang tidak diperlukan, karena itu kita memerlukan pre-processing data agar data lebih terstruktur dan bersih sehingga bisa diklasifikasikan. Text pre-processing adalah bagian dimana data yang sudah didapat selanjutnya diolah dengan tahapan-tahapan case folding, tokenizing, filtering dan stemming [8],[9],[10], serta tokenisasi untuk memisahkan dokumen menurut tokennya. Filtering yaitu membuang kata-kata yang tidak berguna dalam proses klasifikasi. Stemming yaitu merubah setiap kata ke dalam kata dasarnya, pada penelitian ini proses stemming menggunakan library Sastrawi[12] .

Pada proses pembobotan kata ini menggunakan metode proses pembobotan kata TF-IDF (Term Frequency-Inverse Document Frequency). Frequency Invers Document Frequency

(TF-IDF) merupakan metode yang digunakan menentukan seberapa jauh keterhubungan kata (term) Terhadap dokumen dengan memberikan bobot setiap kata [3].

Untuk meningkatkan akurasi, beberapa penelitian analisis sentimen menggunakan word2vec sebagai model untuk membentuk word vector dari setiap kata [11].

Crawling merupakan pengunduhan atau pengambilan data dengan tujuan untuk mengumpulkan data dari suatu database, misalnya data yang diunduh dari server Twitter berupa user dan tweet termasuk atributnya menggunakan fasilitas add-one Twitter archiver yang disediakan oleh Google pada Spreadsheet [13].

Salah satu algoritma yang digunakan untuk klasifikasi teks adalah Naive Bayes. Algoritma ini menggunakan perhitungan probabilitas dan statistik dalam pengoperasiannya. Metode klasifikasi Naive Bayes digunakan untuk mengambil keputusan dengan melakukan prediksi suatu kasus berdasarkan hasil dari klasifikasi yang telah diperoleh. Pada penelitian ini, metode Naive Bayes digunakan untuk menentukan sentimen dari dokumen tweet. Proses ini berdasarkan metode probabilitas dan statistik [14],[15].

# Dataset

Jenis data yang digunakan dalam penelitian ini yaitu:

Dataset yang digunakan merupakan data opini wisatawan Pantai Toroan Kabupaten Sampang yang berasal dari ulasan komentar Google Maps. Dataset ini berupa opini masyarakat baik opini positif, negatif, yang ditulis masyarakat pada ulasan Google Maps. Proses pengumpulan data ini menggunakan jenis data sekunder yang diperoleh dari literatur ataupun browsing pada internet yang dimana data yang digunakan adalah data yang diambil dari ulasan komentar melalui situs Google Maps pengunjung yang berwisata di Pantai Toroan Kabupaten Sampang, Madura. Dari proses yang di Scraping melalui situs Google Maps dengan menggunakan tools Instant Data Scraper total data yang diperoleh sejumlah 1075 data dimana terdapat ulasan atau data yang NaN (data kosong) sejumlah 198 data. Setelah melakukan drop data maka jumlah total data yang akan di proses sebanyak 877 data.

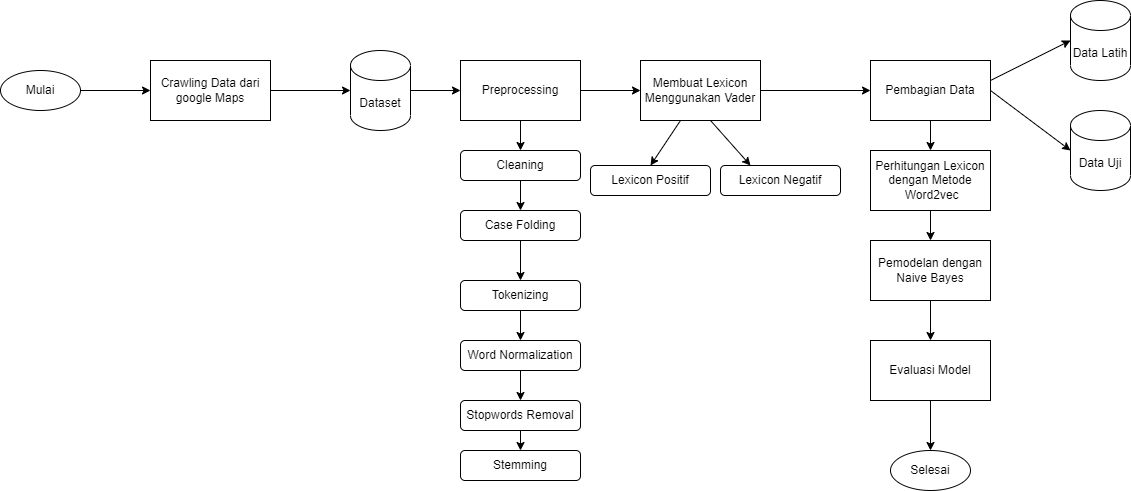
Pengambilan data dilakukan secara menyeluruh tanpa periode waktu tertentu, dikarenakan jumlah data ulasan terhadap tempat pariwisata tersebut pada kurun waktu tertentu terlalu sedikit. Namun data yang paling lama yang yakni data ulasan pada tahun 2017 dan data terbaru yakni data ulasan tahun 2023 ini.

**Contoh Data Yang Diperoleh**

|  |  |  |
| --- | --- | --- |
| Nama | Ulasan | Label |
| Antonina Suryantari | Tempatnya bagus tetapi agak kotor. Ada limbah tebu yang sepertinya dibuang di sekitar area sehingga ada bau tidak sedap yang tercium. Kalau areanya bersih, akan lebih menyenangkan menghabiskan waktu di sana. | Negatif |
| izdiharuddin  Muhammad | Pemandangan yg indah, adem pula suasanyanya meskipun pinggir laut. Cuman air terjunya hanya ada ketika musim hujan | Positif |
| Hendra Susanto | Mantap viewnya bagus banget, terlihat masih asri dan air terjunnya langsung ke pantai | Positif |
| Gabriel Pramana | Pemandangannya bagus tempatnya nyaman. Tetapi sangat di sayangkan dialiran air terjun masih banyak sampah sampah plastik yang berserakan, alangkah lebih baik pengelola dapat membersihkan sampah sampah tersebut agar pemandangan jauh lebih baik lagi | Negatif |
| muhammad ansani | Tempatnya nyaman lesehan, makanannya lumayan banyak menu yg disajikan tentunya pinggir laut jadi dapat pemandangan sekaligus bisa menyantap hidangan | Positif |
| Zain Mughozin | Karena habis hujan jadi airnya keruh | Negatif |
| Andika Bayangkara | Satu satunya air terjun di madura yang airnya terjun dari sungai langsung ke laut...bagus banget...kulinernya mantap...yg recomended ikan bakar kembung maknyus | Positif |
| Nugraha Fahlevi | PENGELOLA TEMPAT WISATA NYA RAMAH. TEMPATNYA SEJUK. | Positif |
| Luki Hidayah | Patut dicoba, masih sangat alami, sebelum masuk infokan ke petugas airnya ada, krn kalau musin kemarau suka kecil debit air terjunnya | Positif |
| REANITA PUTRI ARDANA | Kondisi setelah hujan, airnya jadi kecoklatan | Negatif |

# Arsitektur Sistem

Penelitian ini memiliki perencanaan sistematis yang diilustrasikan pada Gambar 1 yang menunjukkan proses penelitian secara umum. Dalam penelitian ini menggunakan metode Word2Vec untuk proses word embeddingnya dan metode Naive Bayes untuk klasifikasinya. Dalam analisis sentimen Penelitian ini termasuk menggunakan Naive Bayes algoritma klasifikasi untuk memeriksa sentimen pengunjung wisata dan kinerja dari model klasifikasi yang digunakan. Data yang digunakan dalam hal ini penelitian diambil dari Google Maps menggunakan crawling yang kemudian mengolah data dengan proses preprocessing dan kemudian menerapkan klasifikasi ke data dan mendapat visualisasi sentimen oleh frekuensi pengolahan kata pada pengunjung wisata data sentimen. Data yang ada akan diklasifikasikan ke dalam 2 (dua) kelas, yaitu negatif, positif.



Gambar 1. Diagram Alir Proses Klasifikasi untuk Analisis Sentimen

1. Crawling data

Pengumpulan data yang digunakan pada penelitian menggunakan data yang diambil dari Wisata Pantai Toroan di daerah kec. Sampang , Kab. Sampang , Jawa Timur 69216. dengan menggunakan ekstensi “Instant Data Scraper” pada Google Chrome. Data yang digunakan berupa ulasan atau komentar pengunjung wisata Pantai Toroan dengan sejumlah kurang lebih 877 data yang digunakan dalam penelitian.

1. Preprocessing

Tahapan ini merupakan proses dalam pengolahan data dimana mengubah data yang belum dikelola menjadi data yang siap untuk digunakan. Dalam proses ini terdiri dari beberapa proses yang dilakukan, diantaranya:

* Cleaning data

Merupakan proses untuk menghilangkan tanda baca, simbol yang tidak diperlukan, dan spasi yang lebih dari satu pada suatu kalimat

* Case Folding

Proses pengolahan data yang berfungsi untuk mengubah teks dari huruf kapital menjadi huruf kecil.

* Tokenizing

Proses ini dilakukan dengan cara memisahkan kata pada teks dengan tujuan untuk dijadikan serangkaian kelompok kata dengan bentuk token.

* Word Normalization

Proses normalisasi teks yaitu mengubah kata tidak baku di dalam dataset menjadi kata baku

* Stop Removal

Tahapan dalam pemrosesan kata pada dokumen dengan menghapus kata yang tidak mempunyai arti yang selaras pada teks.

* Stemming

Proses ini dilakukan dengan mengubah kata yang memiliki imbuhan pada kalimat dalam teks ke dalam bentuk kata dasar dari

1. Pembuatan Lexicon

VADER (Valence Aware Dictionary and Sentiment Reasoner) merupakan metode analisis lexicon based. Vader akan menganalisis text berdasarkan lexicon (a library) yang menghasilkan class sentiment berupa positif, negatif dan netral dengan tambahan skor total atau compound score. Vader Sentiment Lexicon merupakan salah satu kamus lexicon, kamus ini mempunyai 7.500 tokens yang mengandung kata berbahasa inggris emoticon serta sentimen yang terkait dengan akronim dan inisial.

1. Pembagian Data

Dataset yang kita gunakan dalam sistem ini berjumlah 663 setelah dilakukan preprocessing. Data ulasan positif berjumlah 601 data dan data ulasan negatif berjumlah 62 data. Dikarenakan adanya imbalance data lalu dilakukan proses balancing data dengan mengambil sample data positif sebanyak 1000 data dan data negatif sebanyak 1000 data. Pembagian data training dan data testing yaiut 80% data training yang berjumlah 800 data pada masing-masing kelas dan 20% data testing yang berjumlah 200 data pada masing-masing kelas.

1. Perhitungan Word2vec

Word2vec adalah salah satu metode embedding word yang berguna untuk merepresentasikan kata menjadi sebuah vektor dengan panjang N. Misalnya sebuah kata “Indonesia” di representasikan menjadi sebuah vektor dengan panjang 5 yaitu : [0.2, 0.4, -0.8, 0.9, -0.5]. Vektor tersebut tidak hanya merepresentasikan kata secara sintaktik tapi juga secara semantik atau secara makna. Terdapat 2 jenis arsitektur neural network dari Word2Vec yaitu “Skip-gram” dan “Continous Bag of Word” (CBOW). Pada sistem ini menggunakan arsitektur Skip-gram. Tujuan dari arsitektur skip-gram adalah untuk memprediksi konteks (output) di sekitar current word (input).

1. Pembentukan Model

Pada tahap ini dilakukan pembentukan model dengan algoritma klasifikasi Naive Bayes. Pembentukan model ini menggunakan library yang telah disediakan oleh Python. Untuk mengetahui model tersebut dapat melakukan klasifikasi sentimen maka model diimplementasikan menggunakan beberapa data uji nantinya. Data uji yang digunakan adalah data yang berupa review atau komentar yang terdapat dalam Google Maps.

1. Klasifikasi Sentimen

Proses ini bertujuan untuk mencari tahu polaritas dari sentimen positif, negatif. Klasifikasi yang dilakukan dalam penelitian ini adalah data yang bersumber dari google maps dengan objek wisata pantai toroan. Data yang didapat akan diolah menggunakan model yang digunakan dan menggunakan metode Naive Bayes untuk melihat sentimen yang dihasilkan.

# Preprocessing / Proses Bisnis

Text Preprocessing merupakan tahapan dari proses awal terhadap teks untuk mempersiapkan teks menjadi data yang akan diolah lebih lanjut. Suatu teks tidak dapat diproses langsung oleh algoritma pencarian, oleh karena itu dibutuhkan preprocessing text untuk mengubah teks menjadi data numerik. Dalam tahap pre-processing data pada dataset akan diolah agar data dapat menjadi lebih baik atau maksimal. Terdapat empat tahap dalam tahap pre-processing, yaitu tokenizing, cleaning, case folding,stemming, dan filtering.

Pada bagian ini dijelaskan mengenai tahapan pre processing yang digunakan, hasil dari masing-masing tahapan dan library yang digunakan (jika menggunakan library). Jika tidak ada proses preprocessing, maka bagian ini dapat dihapus.

Dalam tahapan preprocessing ini bertujuan untuk mengolah data mentah menjadi data yang siap diolah. Pada proses ini dilakukan beberapa tahapan yaitu :

1. Cleaning Data

Dalam tahapan preprocessing ini, Cleaning Data bertujuan untuk menghapus karakter, menghapus tanda baca, angka, teks berupa url, dan tanda hubung.

contoh ulasan : Dewasa 10.000/org, anak 7.500/anak.\nParkir mobil

output: Dewasa ./org, anak ./anak.\nParkir mobil

1. Case Folding

Case folding adalah tahapan dalam text preprocessing yang bertujuan untuk mengubah semua huruf yang ada dalam dokumen menjadi huruf kecil, hanya huruf abjad a sampai z yang akan di proses, sementara karakter lain akan dihilangkan. Dalam hal ini kita bisa memanfaatkan modul yang tersedia di python.

contoh ulasan: Tempatnya bagus. Air terjunnya beneran bagus.

output: tempatnya bagus air terjunnya beneran bagus

1. Tokenizing

Tokenizing adalah tahapan dari text preprocessing yang merupakan proses pemisahan teks menjadi beberapa potongan yang disebut token dan di analisa. Pada tahap ini kita bisa menggunakan modul NLTK *(Natural Language Toolkit)*.

contoh ulasan: "Tempatnya bagus tetapi agak kotor. Ada limbah dipinggir pantai"

output:

"Tempatnya", "bagus", "tetapi", "agak", "kotor", "Ada", "limbah", "dipinggir", "pantai".

1. Word Normalization

Proses normalisasi teks yaitu mengubah kata tidak baku di dalam dataset menjadi kata baku dengan menambahkan list kata tidak baku dari sebuah dokumen tambahan yang tidak terdapat dalam library sastrawi.

1. Stopword Removal (Filtering)

Stopword removal adalah proses filtering, memasukkan kata kata penting dari hasil token dan biasanya memakai algoritma *stoplist* yaitu membuang kata kurang penting dan *wordlist* yaitu menyimpan kata penting. Filtering ini bisa dilakukan menggunakan NLTK *(Natural Language Toolkit)* dan sastrawi.

output:

"Tempatnya bagus tetapi agak kotor. Ada limbah dipinggir pantai"

1. Stemming

Stemming adalah salah satu teknik dalam *text preprocessing* yang digunakan untuk mengurangi kata infleksi ke kata dasar, hal ini seringkali berguna untuk menangani kelangkaan atau standarisasi kosa kata. Stemming juga memungkinkan NLP untuk mempelajari hubungan antara infleksi dan kata induknya, dan akhirnya membantu model memahami penggunaannya. Ada 2 cara dalam stemming, stemming dengan NLTK (bahasa inggris), dan stemming bahasa indonesia menggunakan python sastrawi.

output:

tempatnya : tempat

diatasnya : atas

kecoklatan : coklat

petugas : tugas

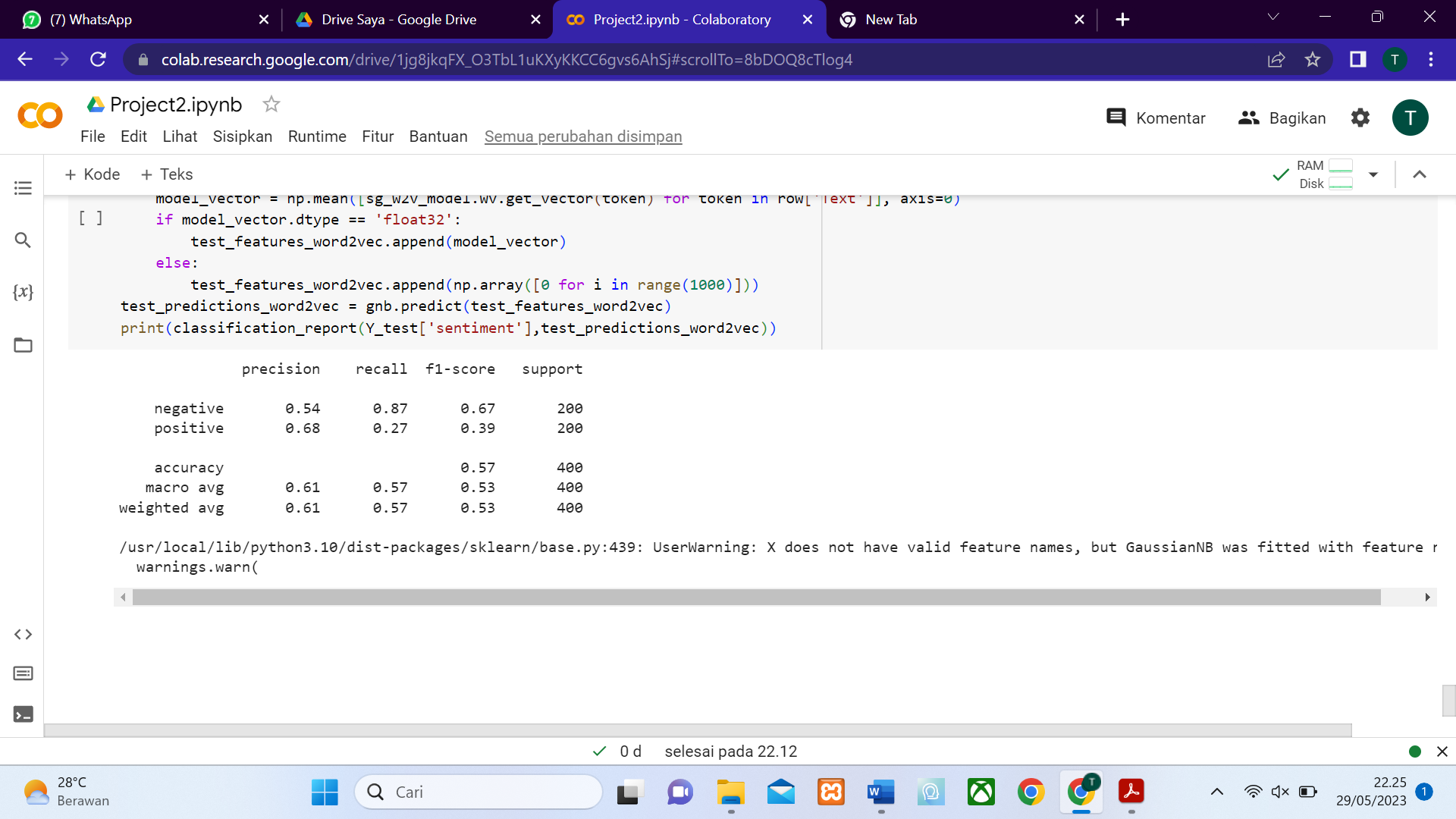
pemandangannya : pemandang

makanannya : makan

# Hasil dan pembahasan

Sistem ini menggunakan dataset yang jumlahnya 601 ulasan positif dan 62 ulasan negatif. Untuk menyeimbangkan data, maka melakukan balancing data dengan mengambil sample sebanyak 1000 data positif dan 1000 data negatif. Selanjutnya membuat lexicon positif dan negatif menggunakan library vader. Dengan tujuan memperoleh kelas untuk setiap ulasan berdasarkan skor compound yang dimilikinya. Lalu membagi data atau split data yaitu sebanyak 800 data training dan 200 data testing pada masing-masing kelas sentiment. Metode word embedding yang diterapkan menggunakan word2vec untuk merepresentasikan setiap kata membentuk sebuah vektor. Metode yang digunakan untuk mentraining data latih yaitu menggunakan Gaussian Naive Bayes yang ada pada library sklearn.

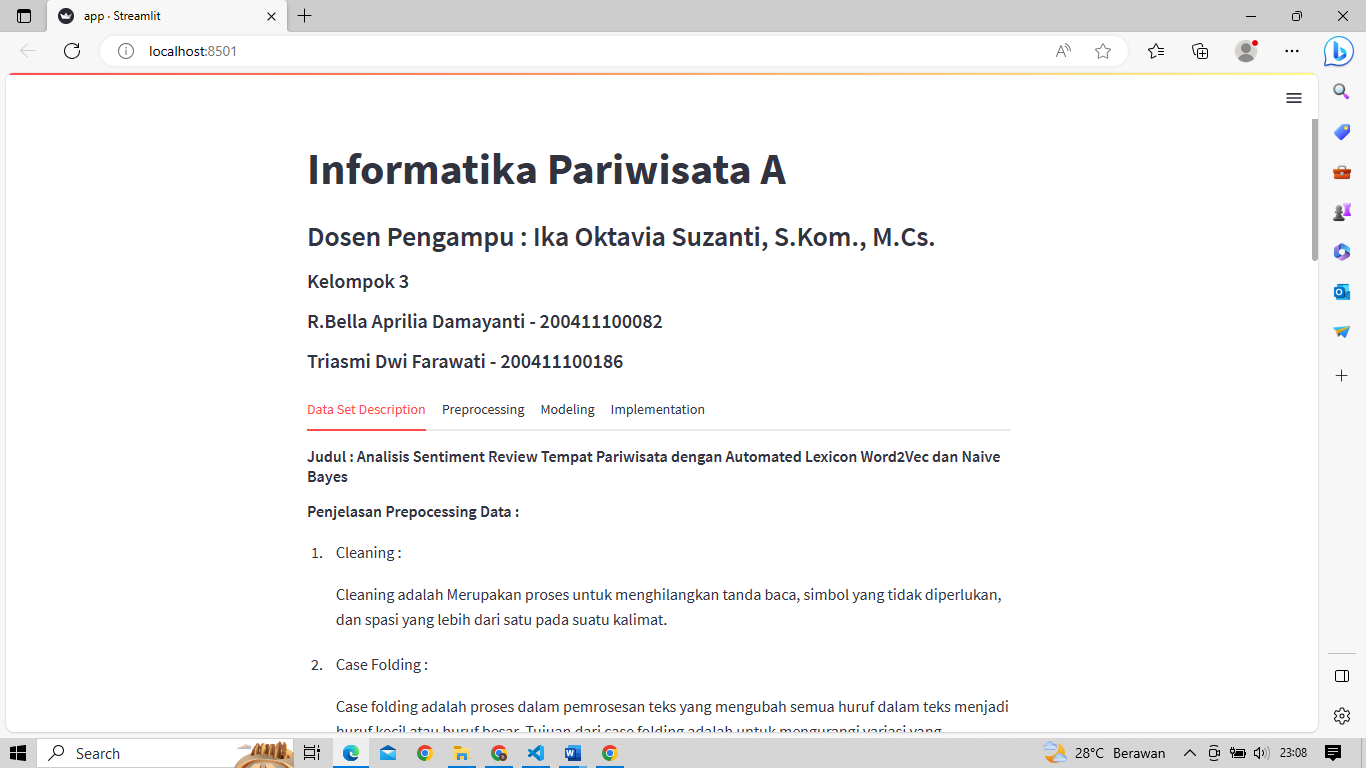
Kemudian untuk mengetahui nilai performa model klasifikasi pada setiap kelasnya. Setelah meimplementasi kode program, maka hasil nilai performa model klasifikasi precision score, recall score, dan f1-score terdapat pada tabel sebagai berikut.

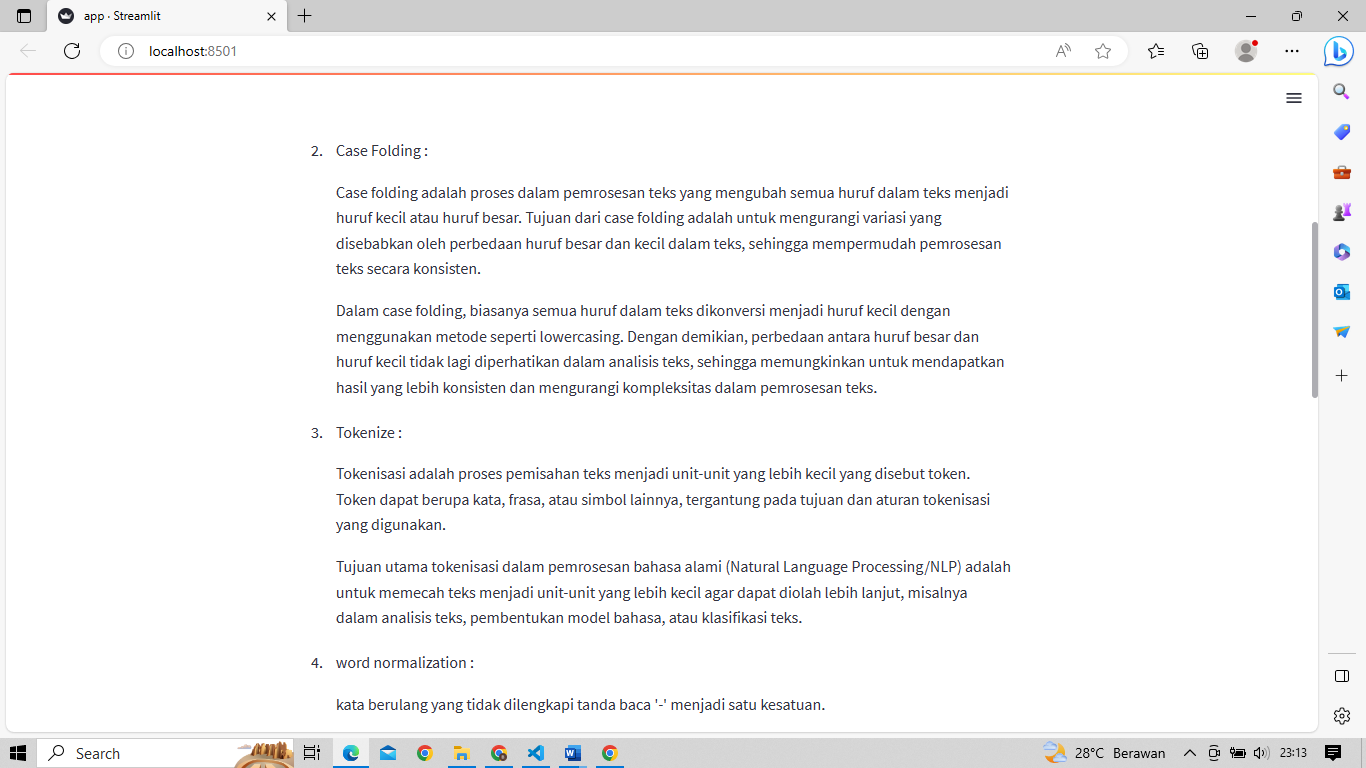


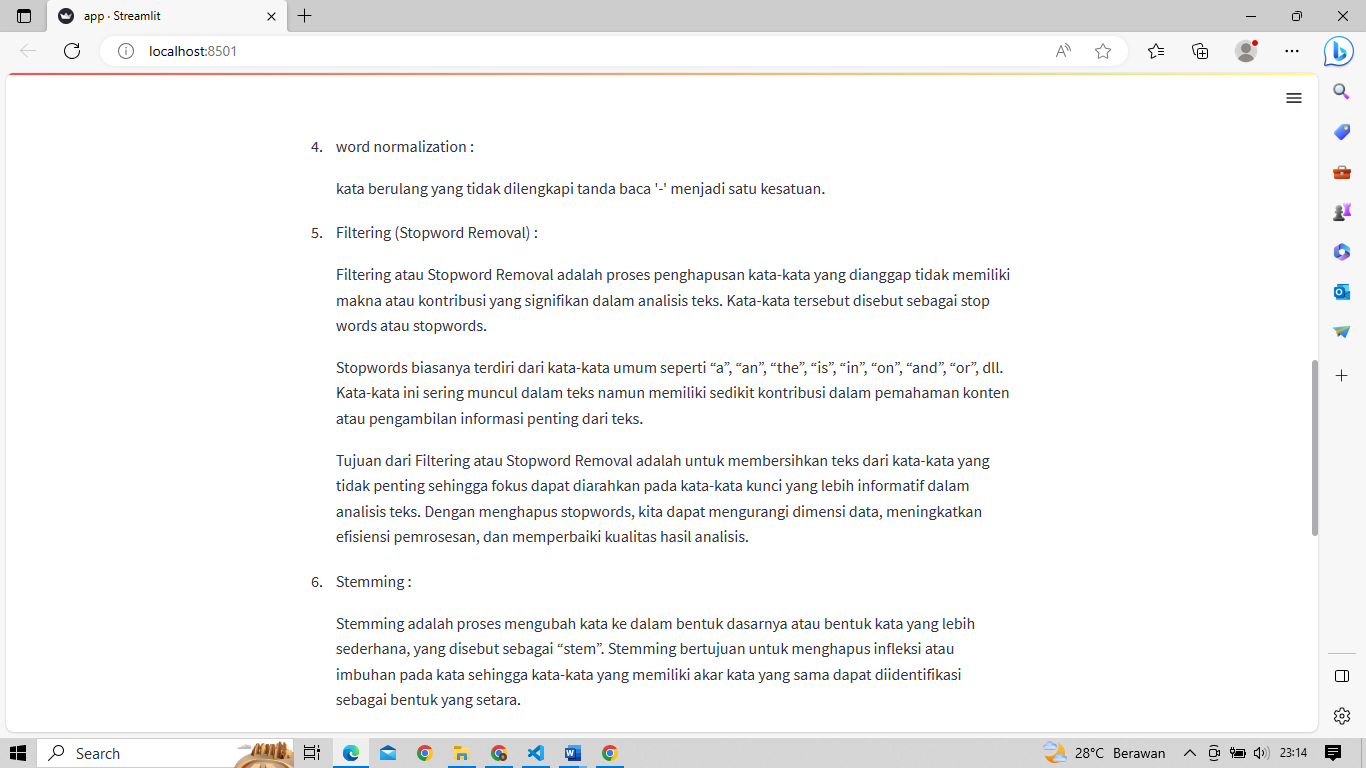
Dengan nilai performa model pada setiap kelasya diatas, maka hasil tahap evaluasi model pada sistem dalam mencari informasi yang diberikan oleh pengguna terdapat presentase 54% pada kelas negatif, 68% pada kelas positif. Lalu kemampuan sistem dalam menemukan kembali informasi pada pengguna terdapat presentase 87% dan pada kelas negatif, 27% pada kelas positif. Nilai f1-score yang dihasilkan pada sistem yaitu 67% pada kelas negatif dan 39% pada kelas positif.

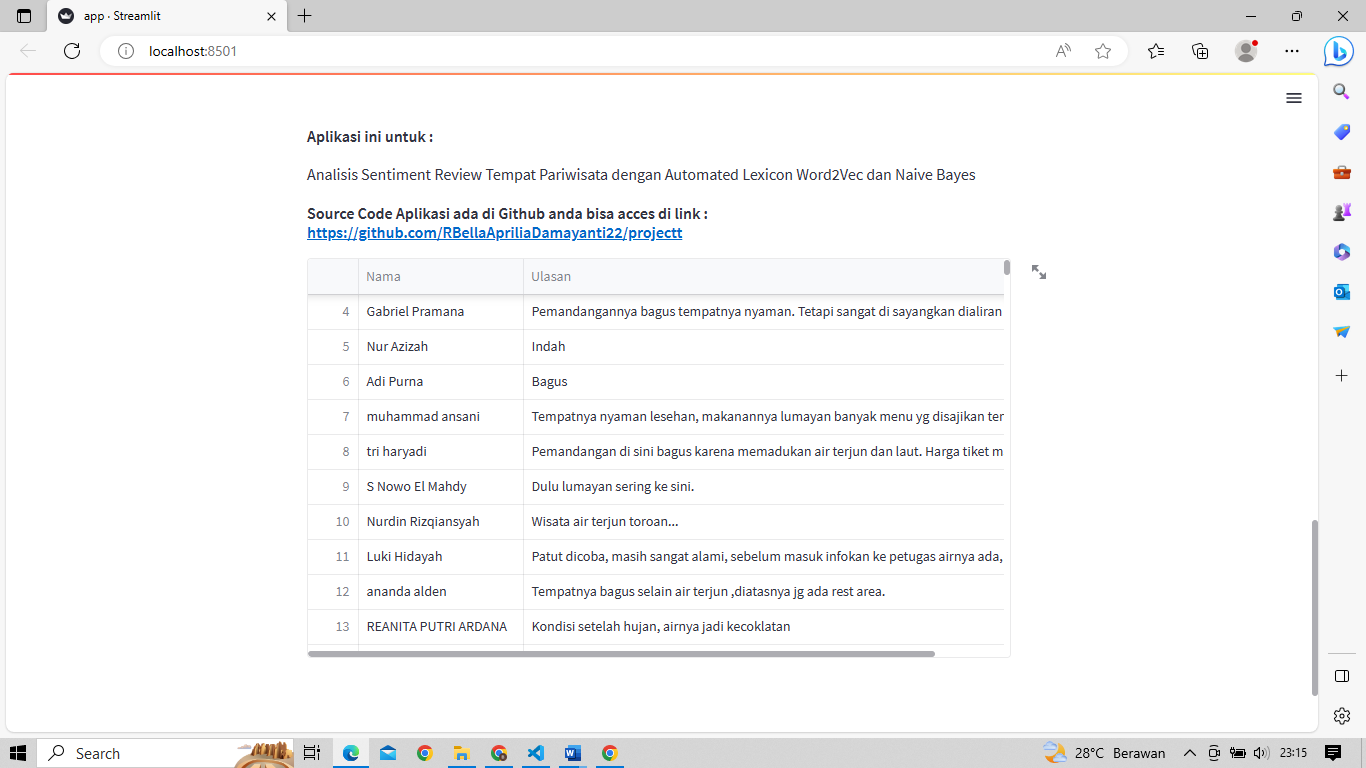
Berikut merupakan hasil tampilan user interface menggunakan streamlit:

* Tampilan Home page







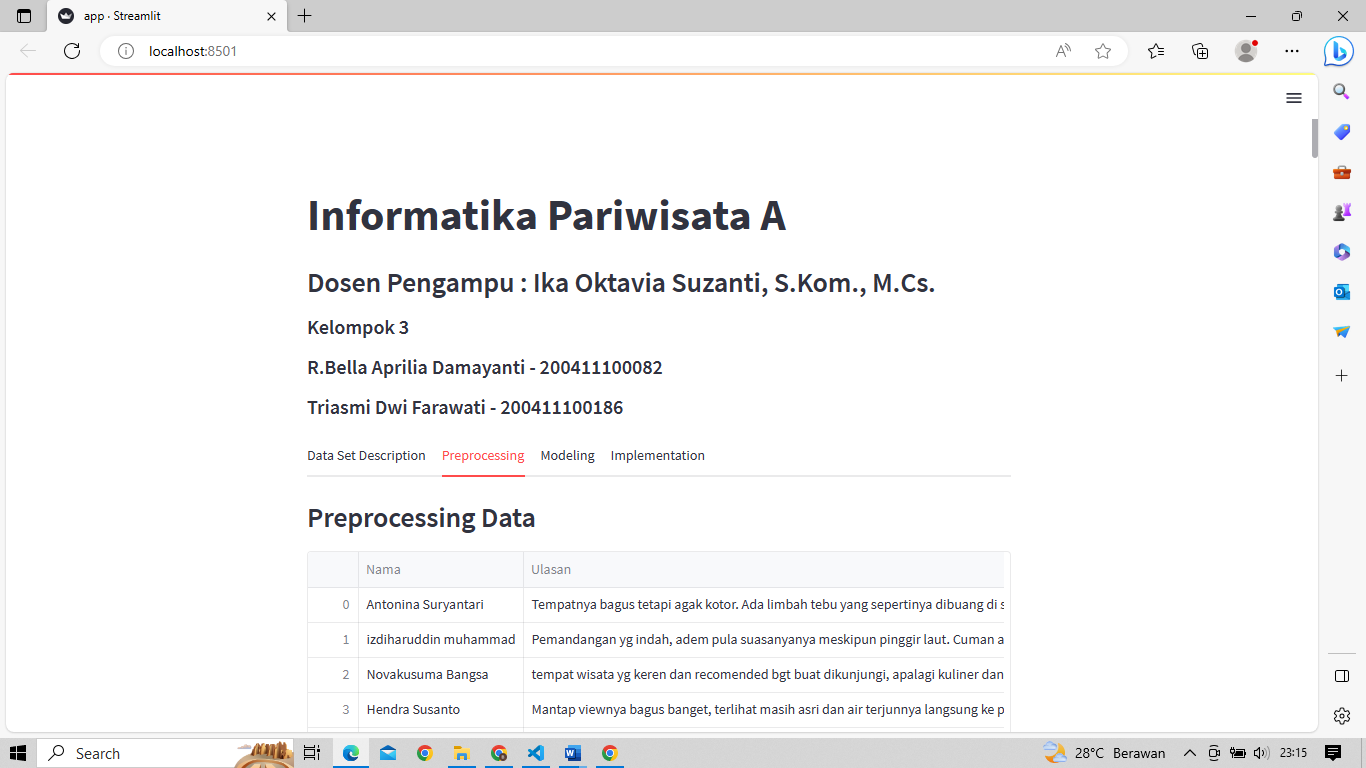


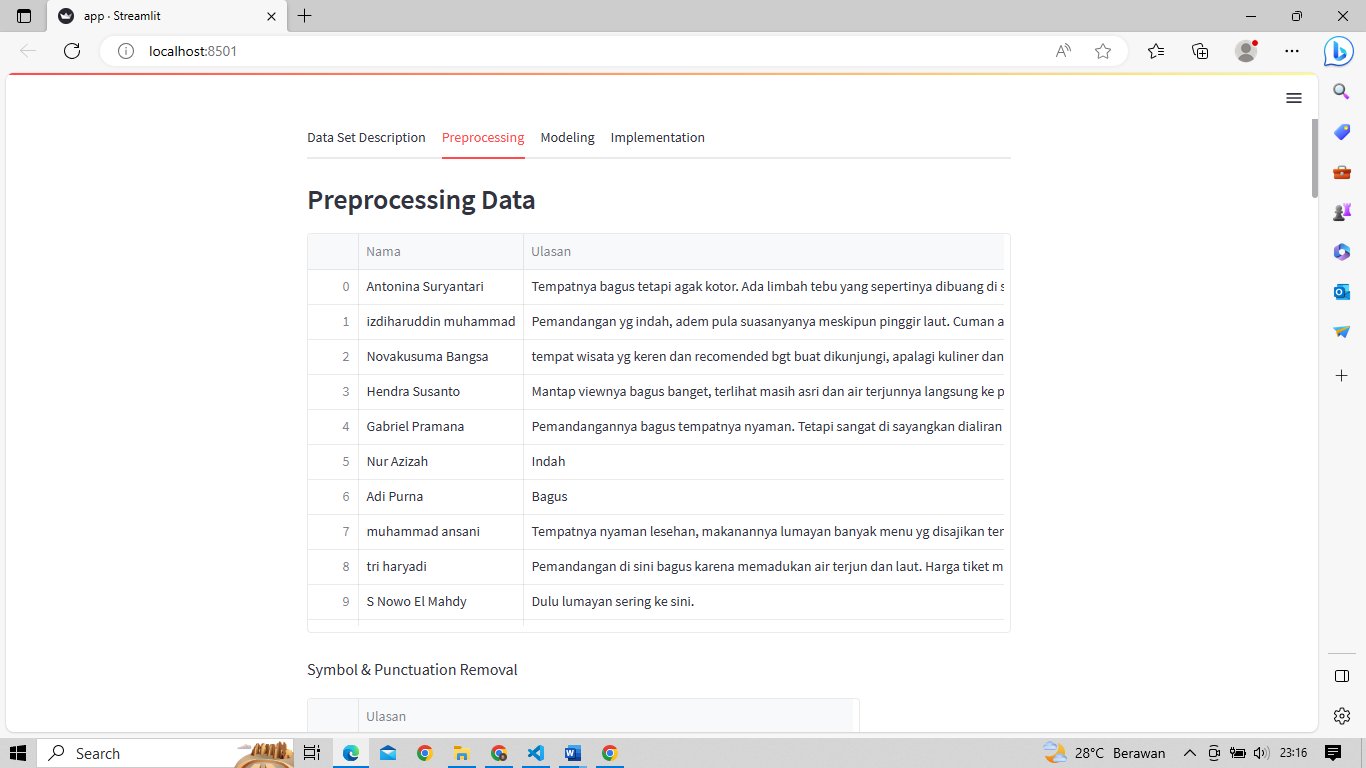
* Code :

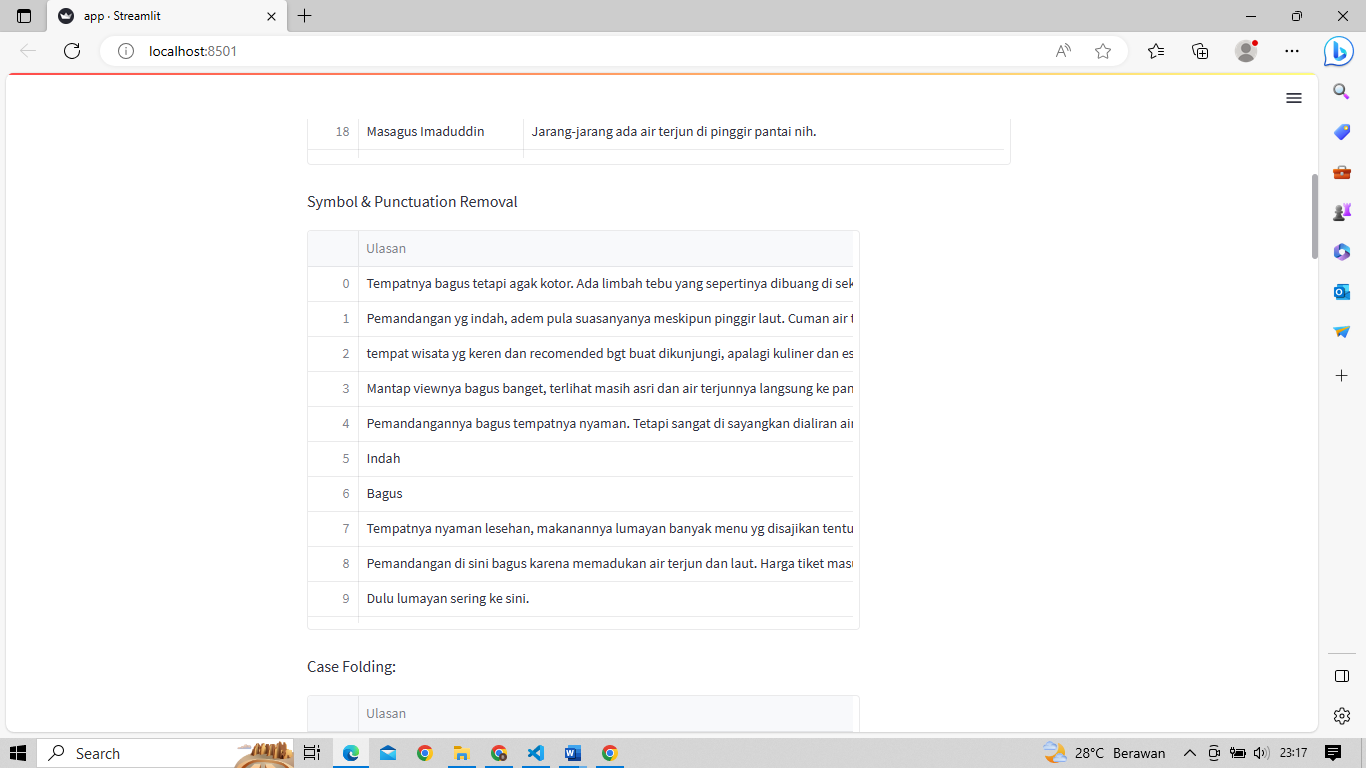
|  |
| --- |
| with data\_set\_description:  st.write("###### Judul : Analisis Sentiment Review Tempat Pariwisata dengan Automated Lexicon Word2Vec dan Naive Bayes ")  st.write("""###### Penjelasan Prepocessing Data : """)  st.write("""1. Cleaning :    Cleaning adalah Merupakan proses untuk menghilangkan tanda baca, simbol yang tidak diperlukan, dan spasi yang lebih dari satu pada suatu kalimat.  """)  st.write("""2. Case Folding :    Case folding adalah proses dalam pemrosesan teks yang mengubah semua huruf dalam teks menjadi huruf kecil atau huruf besar. Tujuan dari case folding adalah untuk mengurangi variasi yang disebabkan oleh perbedaan huruf besar dan kecil dalam teks, sehingga mempermudah pemrosesan teks secara konsisten.    Dalam case folding, biasanya semua huruf dalam teks dikonversi menjadi huruf kecil dengan menggunakan metode seperti lowercasing. Dengan demikian, perbedaan antara huruf besar dan huruf kecil tidak lagi diperhatikan dalam analisis teks, sehingga memungkinkan untuk mendapatkan hasil yang lebih konsisten dan mengurangi kompleksitas dalam pemrosesan teks.  """)  st.write("""3. Tokenize :  Tokenisasi adalah proses pemisahan teks menjadi unit-unit yang lebih kecil yang disebut token. Token dapat berupa kata, frasa, atau simbol lainnya, tergantung pada tujuan dan aturan tokenisasi yang digunakan.  Tujuan utama tokenisasi dalam pemrosesan bahasa alami (Natural Language Processing/NLP) adalah untuk memecah teks menjadi unit-unit yang lebih kecil agar dapat diolah lebih lanjut, misalnya dalam analisis teks, pembentukan model bahasa, atau klasifikasi teks.  """)  st.write("""4. word normalization :  kata berulang yang tidak dilengkapi tanda baca '-' menjadi satu kesatuan.  """)  st.write("""5. Filtering (Stopword Removal) :  Filtering atau Stopword Removal adalah proses penghapusan kata-kata yang dianggap tidak memiliki makna atau kontribusi yang signifikan dalam analisis teks. Kata-kata tersebut disebut sebagai stop words atau stopwords.  Stopwords biasanya terdiri dari kata-kata umum seperti “a”, “an”, “the”, “is”, “in”, “on”, “and”, “or”, dll. Kata-kata ini sering muncul dalam teks namun memiliki sedikit kontribusi dalam pemahaman konten atau pengambilan informasi penting dari teks.  Tujuan dari Filtering atau Stopword Removal adalah untuk membersihkan teks dari kata-kata yang tidak penting sehingga fokus dapat diarahkan pada kata-kata kunci yang lebih informatif dalam analisis teks. Dengan menghapus stopwords, kita dapat mengurangi dimensi data, meningkatkan efisiensi pemrosesan, dan memperbaiki kualitas hasil analisis.  """)  st.write("""6. Stemming :  Stemming adalah proses mengubah kata ke dalam bentuk dasarnya atau bentuk kata yang lebih sederhana, yang disebut sebagai “stem”. Stemming bertujuan untuk menghapus infleksi atau imbuhan pada kata sehingga kata-kata yang memiliki akar kata yang sama dapat diidentifikasi sebagai bentuk yang setara.  """)      st.write("###### Aplikasi ini untuk : ")  st.write("""Analisis Sentiment Review Tempat Pariwisata dengan Automated Lexicon Word2Vec dan Naive Bayes """)  st.write("###### Source Code Aplikasi ada di Github anda bisa acces di link : https://github.com/RBellaApriliaDamayanti22/projectt")  st.write(dataset) |

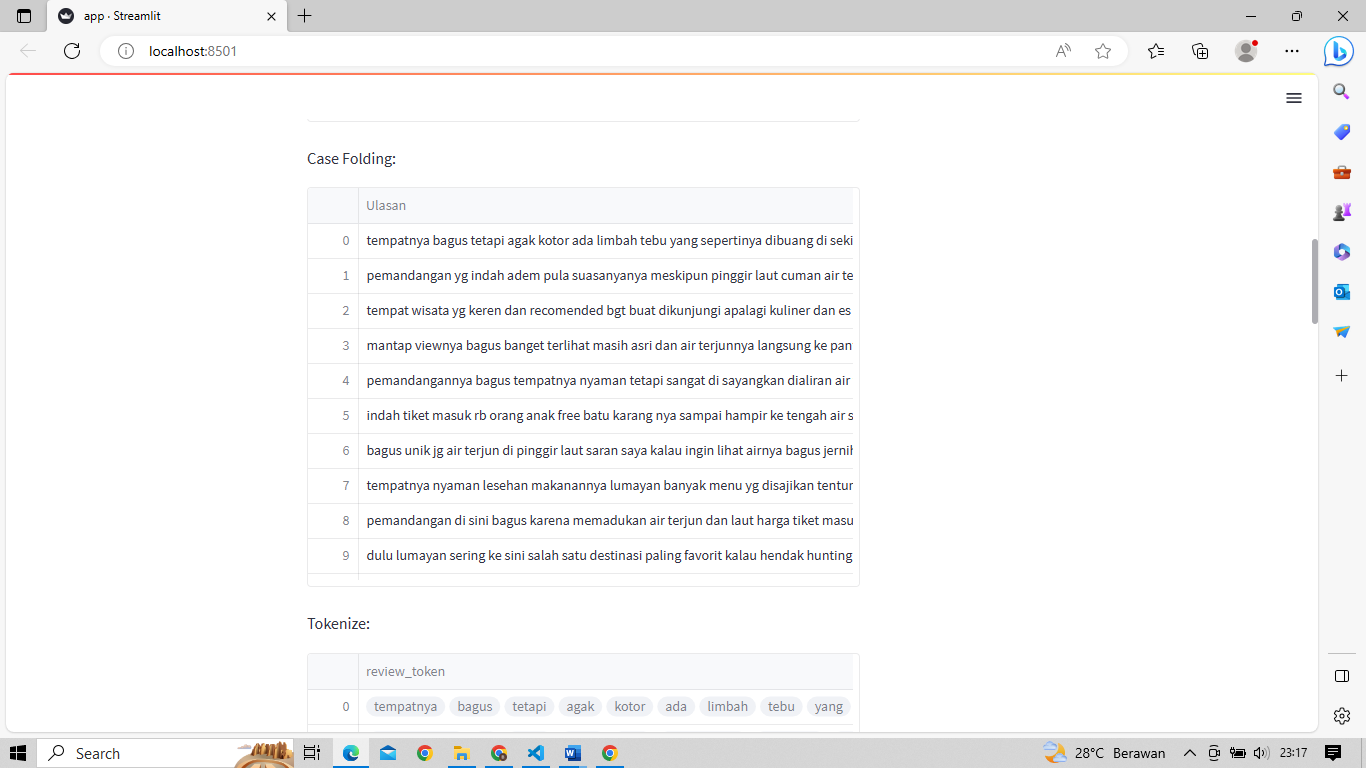
Pada tampilan *Home Page* dari system analisis Pantai Toroan Menggunakan Metode Naïve Bayes. Terdapat informasi mengenai pantai toroan dan dataset yang akan digunakan untuk penelitian ini yang ada pada sistem.

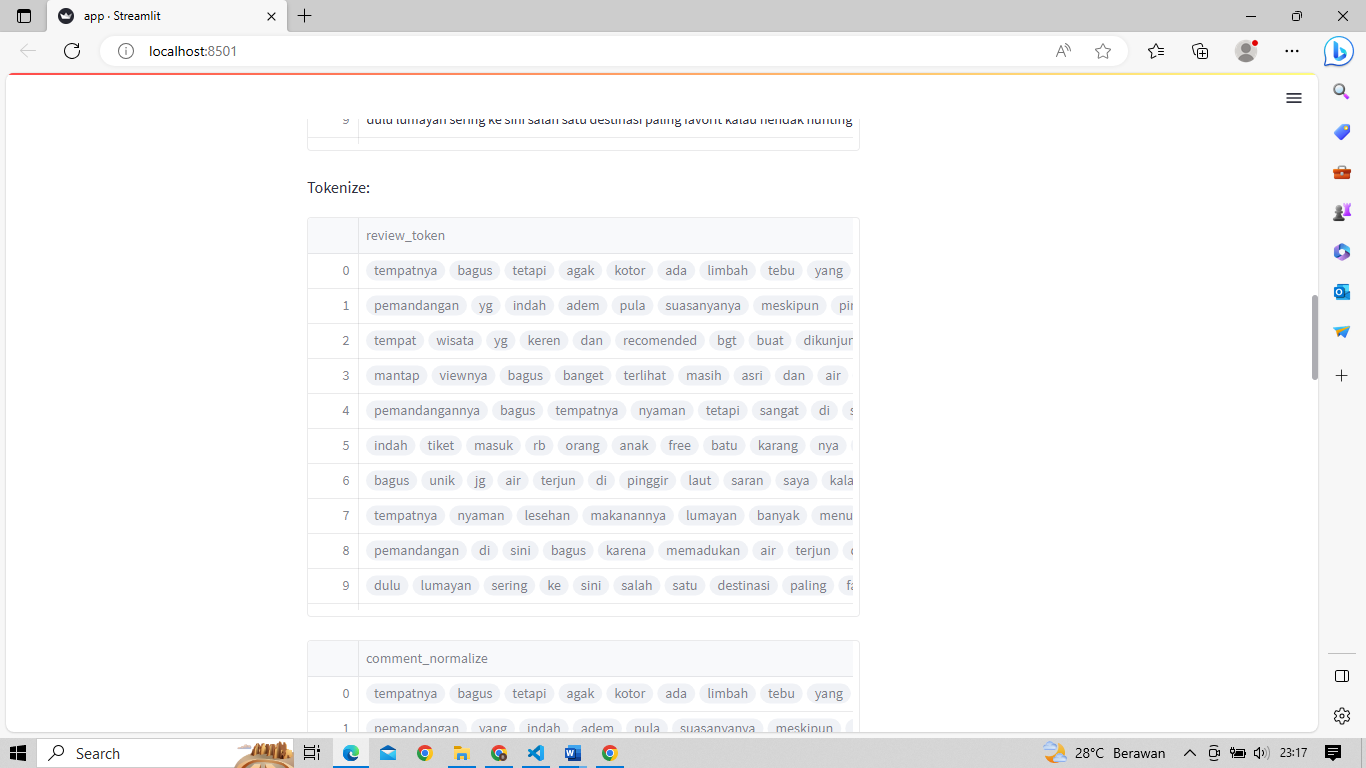
* Tampilan Menu Prepocessing

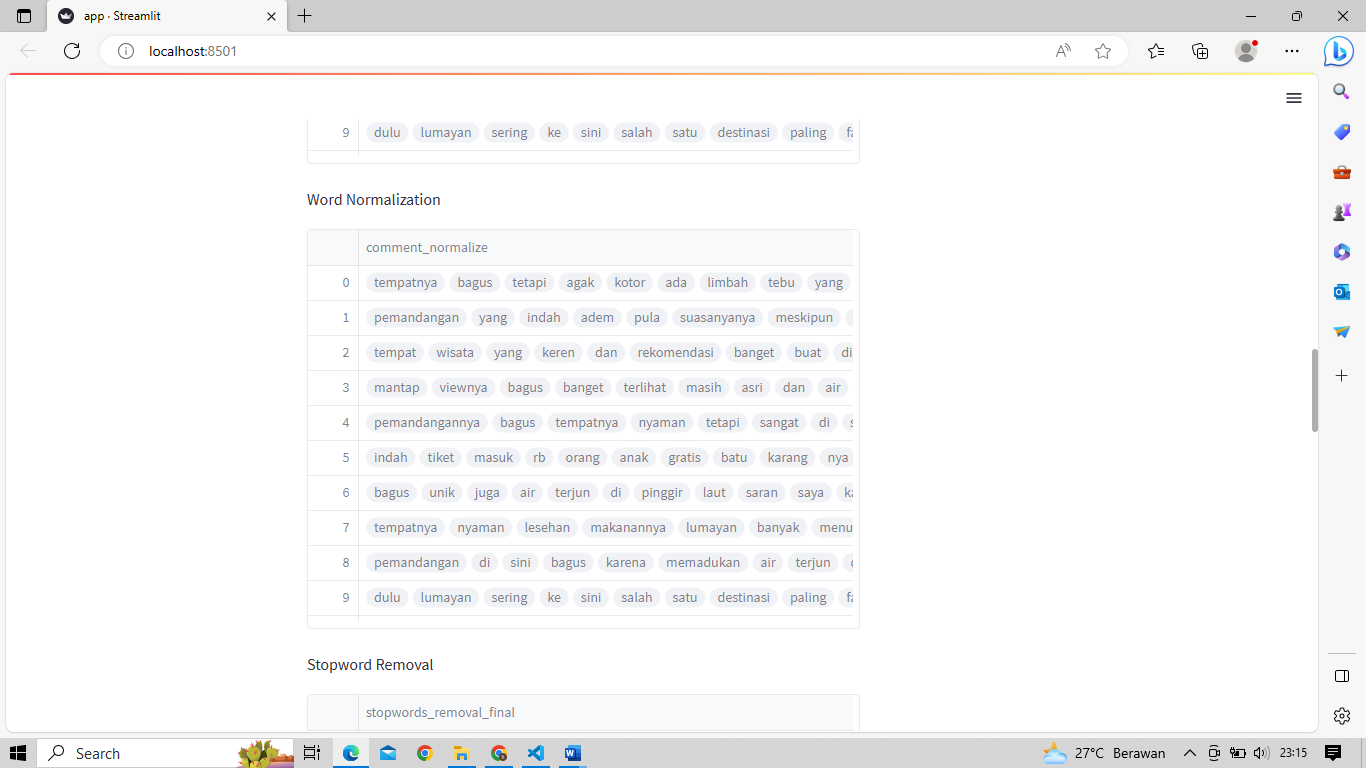


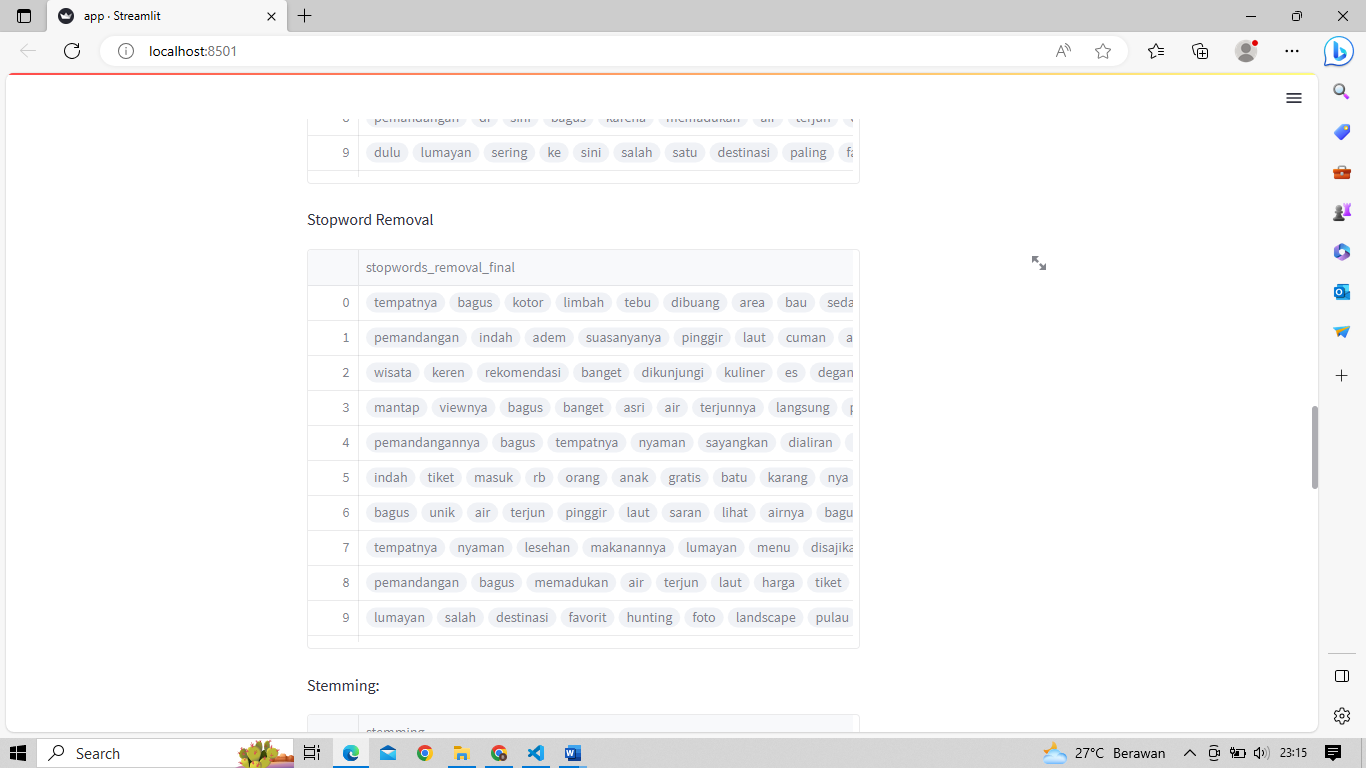


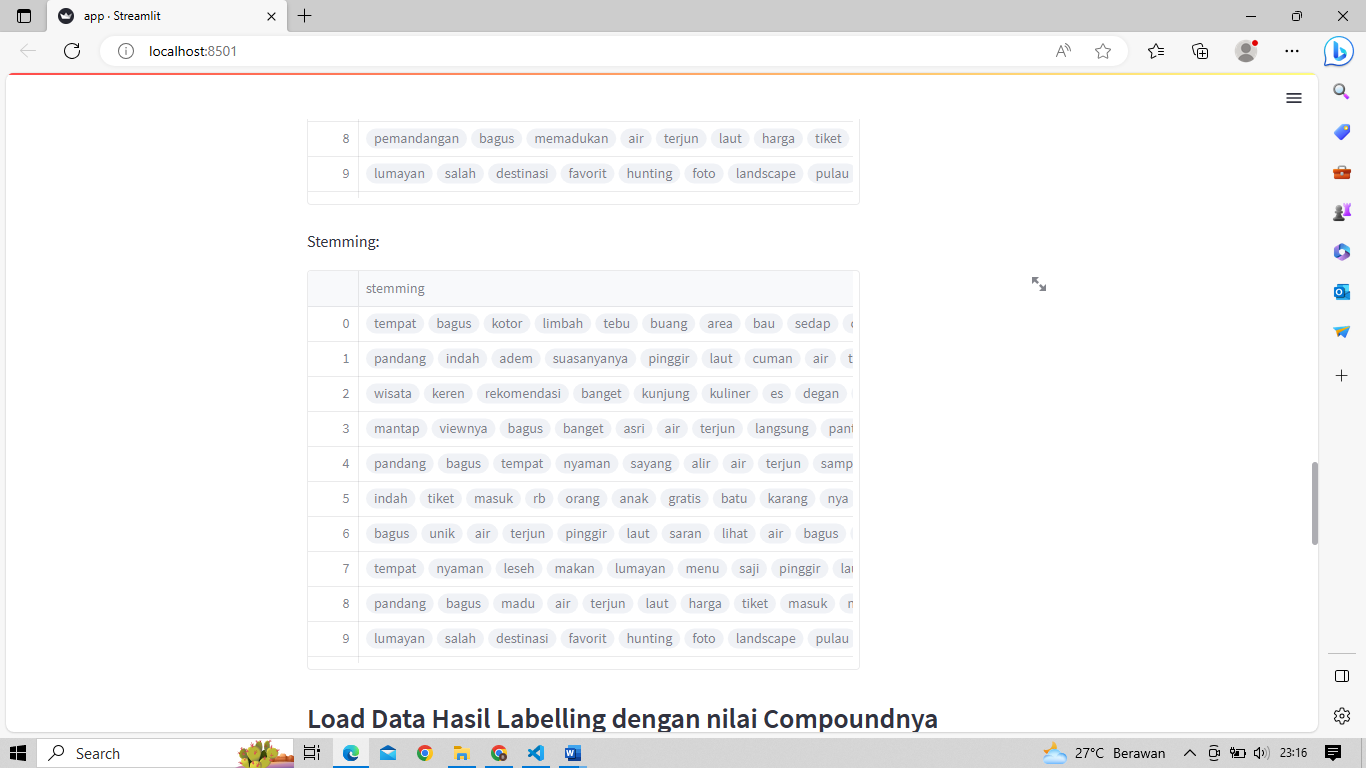


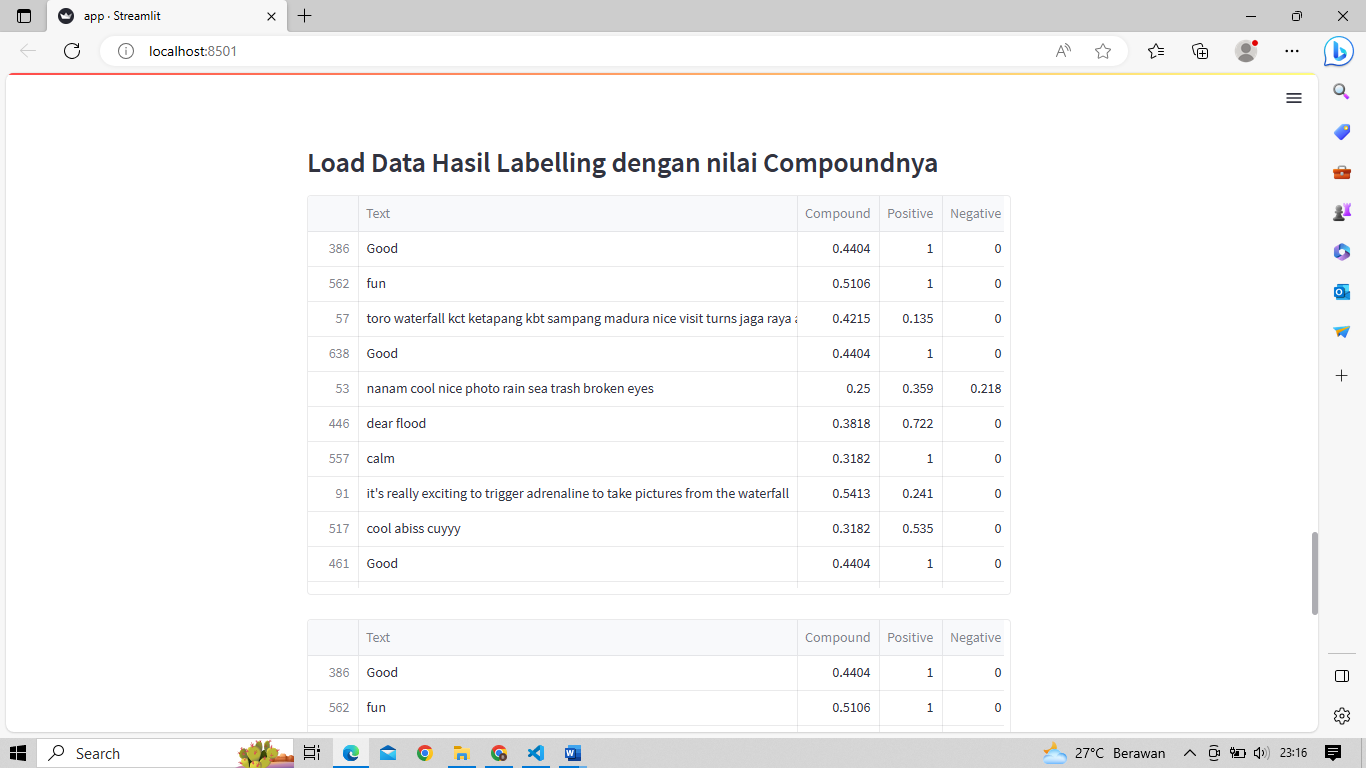












* Code :

|  |
| --- |
| * Code Preprocessing :   import pandas as pd  import numpy as np  from string import punctuation  import re  import nltk  from Sastrawi.Stemmer.StemmerFactory import StemmerFactory  import string  import os  import pickle  nltk.download('stopwords')  nltk.download('punkt')  from nltk.corpus import stopwords  def remove\_text\_special (text:str)->str:  text = text.replace('\\t',"").replace('\\n',"").replace('\\u',"").replace('\\',"")  text = text.encode('ascii', 'replace').decode('ascii')  return text.replace("http://"," ").replace("https://", " ")  def remove\_tanda\_baca(text:str)->str:  text = re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)"," ",text)  return text    def remove\_numbers (text:str)->str:  return re.sub(r"\d+", "", text)  """CASE FOLDING """  #proses casefolding  def casefolding(Comment:str)->str:  Comment = Comment.lower()  return Comment  """TOKENIZING"""  def tokenizing(sentence:str)->list:  return nltk.word\_tokenize(sentence)  """NORMALIZE"""  print(os.getcwd())  normalize = pd.read\_excel("./data/Normalization Data.xlsx")  normalize\_word\_dict = {}  for row in normalize.iterrows():  if row[0] not in normalize\_word\_dict:  normalize\_word\_dict[row[0]] = row[1]  def normalized\_term(comment:list)->list:  return [normalize\_word\_dict[term] if term in normalize\_word\_dict else term for term in comment]  """STOPWARD REMOVAL"""  txt\_stopwords = stopwords.words('indonesian')  def stopwords\_removal(filtering:list)->list:  filtering = [word for word in filtering if word not in txt\_stopwords]  return filtering  #stopword removal 2  data\_stopwords = pd.read\_excel("./data/list\_stopwords.xlsx")  def stopwords\_removal2(filter:list)->list :  filter = [word for word in filter if word not in data\_stopwords]  return filter  """STEMMING"""  factory = StemmerFactory()  stemmer = factory.create\_stemmer()  def stemming (document:list)->list:  result = []  for term in document:  result.append(stemmer.stem(term))  return result  def preprocessing(text:str)->list:  text = remove\_text\_special(text)  text = remove\_tanda\_baca(text)  text = remove\_numbers(text)  text = casefolding(text)  tokenize = tokenizing(text)  tokenize = normalized\_term(tokenize)  tokenize = stopwords\_removal(tokenize)  tokenize = stopwords\_removal2(tokenize)  tokenize = stemming(tokenize)  return tokenize  with open("./model/tfidf.pickle", 'rb') as file:  vectorizer = pickle.load(file) |

Pada tampilan menu preprocessing ada beberapa tahapan. Dalam tahapan preprocessing ini bertujuan untuk mengolah data mentah menjadi data yang siap diolah. Pada proses ini dilakukan beberapa tahapan yaitu :

1. Cleaning Data

Dalam tahapan preprocessing ini, Cleaning Data bertujuan untuk menghapus karakter, menghapus tanda baca, angka, teks berupa url, dan tanda hubung.

contoh ulasan : Dewasa 10.000/orang, anak 7.500/anak.\Parkir mobil

output: Dewasa : Dewasa orang anak parkir mobil

1. Case Folding

Case folding adalah tahapan dalam text preprocessing yang bertujuan untuk mengubah semua huruf yang ada dalam dokumen menjadi huruf kecil, hanya huruf abjad a sampai z yang akan di proses, sementara karakter lain akan dihilangkan. Dalam hal ini kita bisa memanfaatkan modul yang tersedia di python.

contoh ulasan: Tempatnya bagus. Air terjunnya beneran bagus.

output: tempatnya bagus air terjunnya beneran bagus

1. Tokenizing

Tokenizing adalah tahapan dari text preprocessing yang merupakan proses pemisahan teks menjadi beberapa potongan yang disebut token dan di analisa. Pada tahap ini kita bisa menggunakan modul NLTK *(Natural Language Toolkit)*.

contoh ulasan: "Tempatnya bagus tetapi agak kotor. Ada limbah dipinggir pantai"

output: "Tempatnya", "bagus", "tetapi", "agak", "kotor", "Ada", "limbah", "dipinggir", "pantai".

1. Word Normalization

Proses normalisasi teks yaitu mengubah kata tidak baku di dalam dataset menjadi kata baku dengan menambahkan list kata tidak baku dari sebuah dokumen tambahan yang tidak terdapat dalam library sastrawi.

1. Stopword Removal (Filtering)

Stopword removal adalah proses filtering, memasukkan kata kata penting dari hasil token dan biasanya memakai algoritma *stoplist* yaitu membuang kata kurang penting dan *wordlist* yaitu menyimpan kata penting. Filtering ini bisa dilakukan menggunakan NLTK *(Natural Language Toolkit)* dan sastrawi.

output: "Tempatnya bagus tetapi agak kotor. Ada limbah dipinggir pantai"

1. Stemming

Stemming adalah salah satu teknik dalam *text preprocessing* yang digunakan untuk mengurangi kata infleksi ke kata dasar, hal ini seringkali berguna untuk menangani kelangkaan atau standarisasi kosa kata. Stemming juga memungkinkan NLP untuk mempelajari hubungan antara infleksi dan kata induknya, dan akhirnya membantu model memahami penggunaannya. Ada 2 cara dalam stemming, stemming dengan NLTK (bahasa inggris), dan stemming bahasa indonesia menggunakan python sastrawi.

output:

tempatnya : tempat

diatasnya : atas

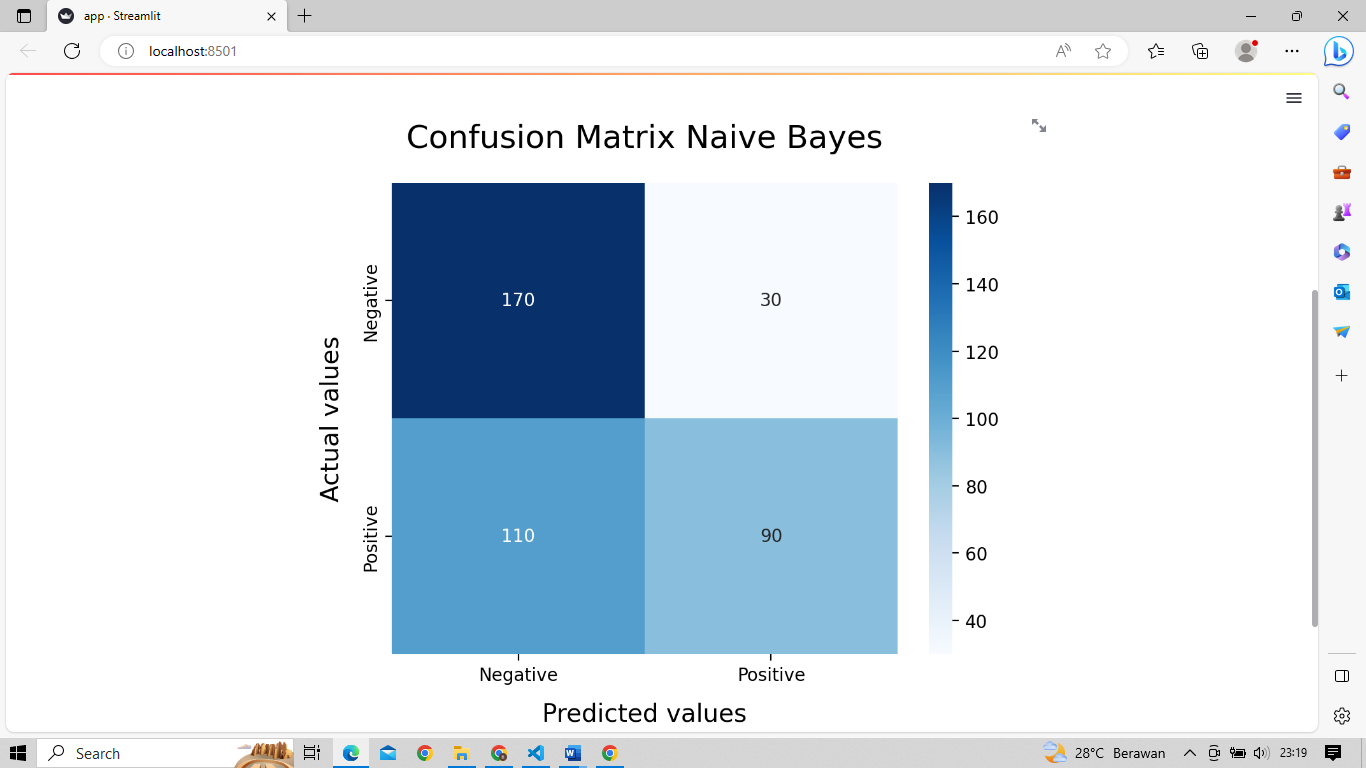
kecoklatan : coklat

petugas : tugas

pemandangannya : pemandang

makanannya : makan

* Tampilan Modelling

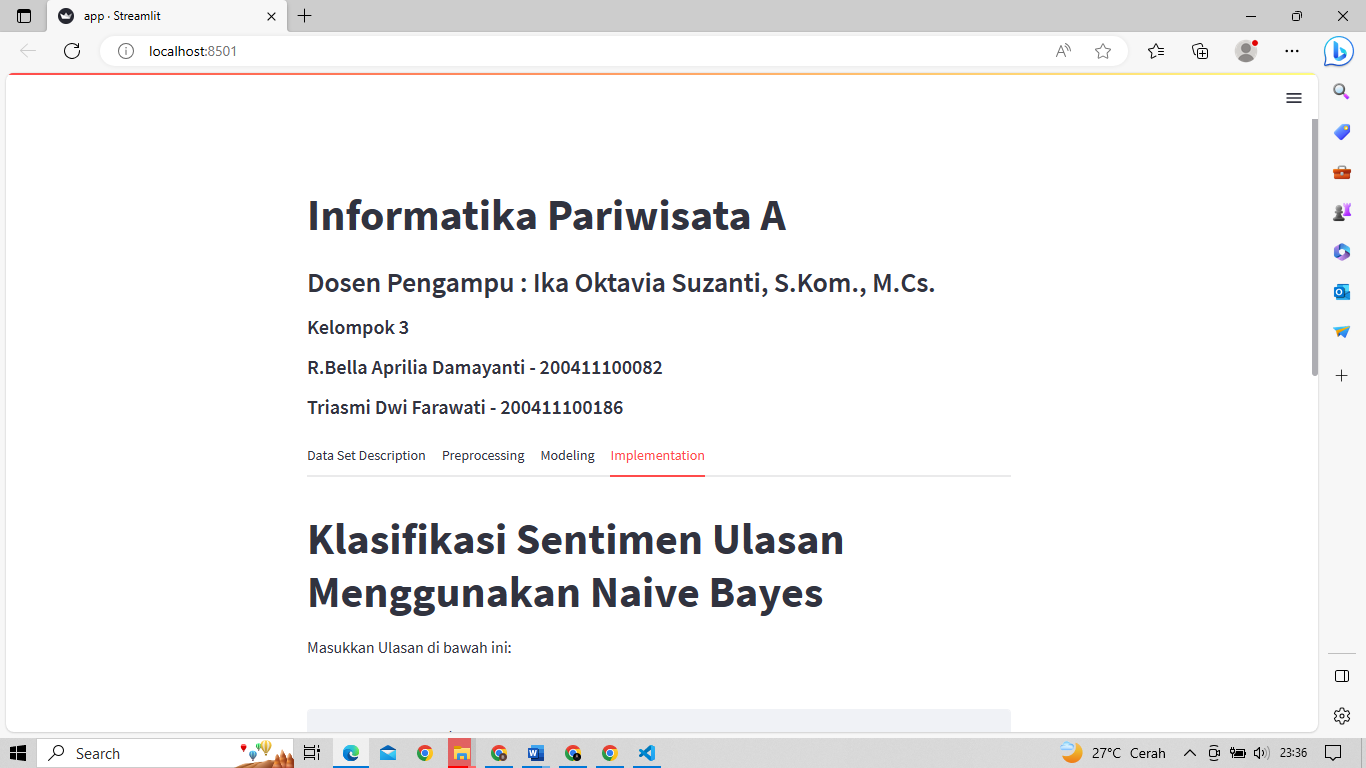


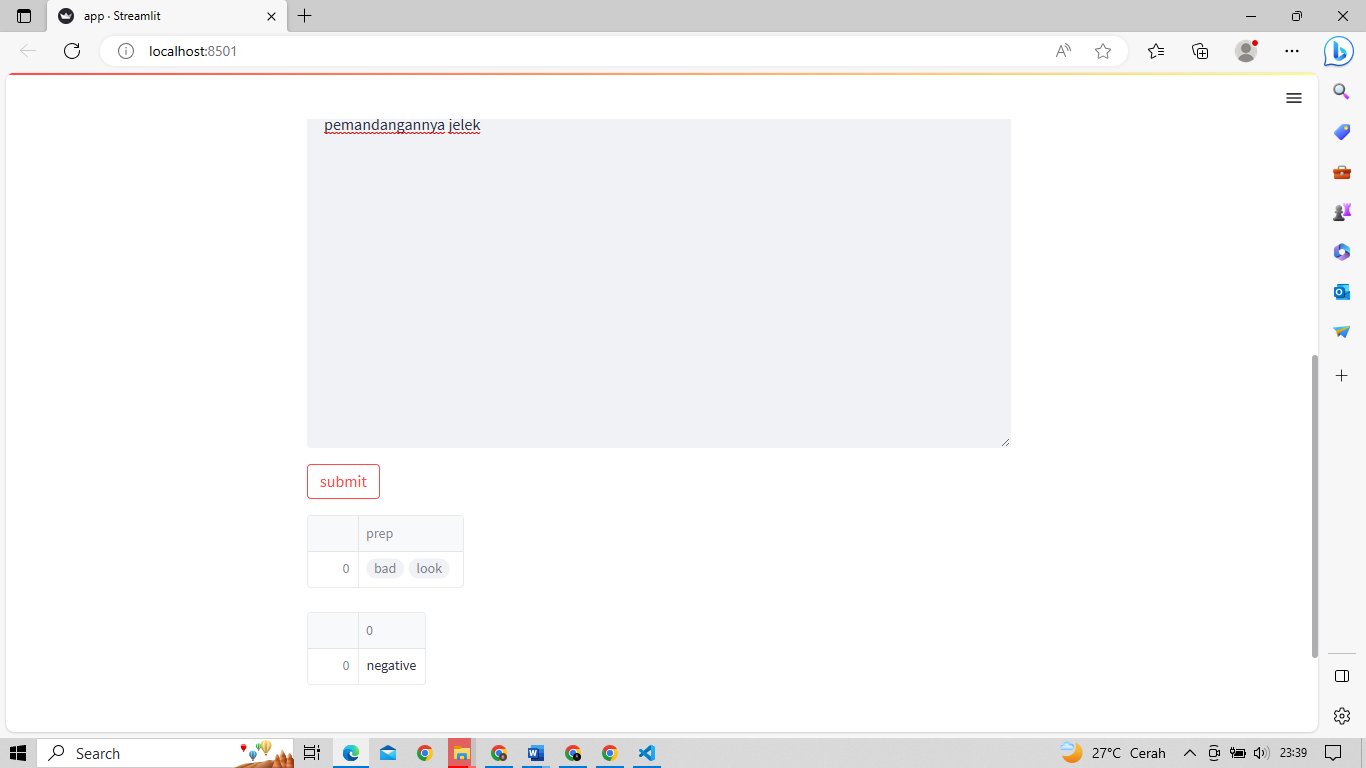
* Code

|  |
| --- |
| import time  from sklearn.naive\_bayes import GaussianNB  # Load from the filename  word2vec\_df = pd.read\_csv(word2vec\_filename)  gnb = GaussianNB(priors=None, var\_smoothing= 2.848035868435799e-06)  start\_time = time.time()  # Fit the model  gnb.fit(word2vec\_df, Y\_train['sentiment'])  print("Time taken to fit the model with word2vec vectors: " + str(time.time() - start\_time))  from sklearn.metrics import classification\_report  test\_features\_word2vec = []  for index, row in X\_test.iterrows():  model\_vector = np.mean([sg\_w2v\_model.wv.get\_vector(token) for token in row['Text']], axis=0)  if model\_vector.dtype == 'float32':  test\_features\_word2vec.append(model\_vector)  else:  test\_features\_word2vec.append(np.array([0 for i in range(1000)]))  test\_predictions\_word2vec = gnb.predict(test\_features\_word2vec)  print(classification\_report(Y\_test['sentiment'],test\_predictions\_word2vec))  from sklearn.metrics import classification\_report, confusion\_matrix  #Prediciting the test results  from sklearn.metrics import classification\_report, confusion\_matrix  #Prediciting the test results  y\_predict\_test = gnb.predict(test\_features\_word2vec)  cm = confusion\_matrix(Y\_test['sentiment'], y\_predict\_test)  categories = ['Negative','Positive']  plt.figure()  st.write("Confusion Matrix")  sns.heatmap(cm, annot=True, fmt='', cmap='Blues', xticklabels = categories, yticklabels = categories)  plt.xlabel("Predicted values", fontdict = {'size':14}, labelpad = 10)  plt.ylabel("Actual values" , fontdict = {'size':14}, labelpad = 10)  plt.title ("Confusion Matrix Naive Bayes", fontdict = {'size':18}, pad = 20)  st.set\_option('deprecation.showPyplotGlobalUse', False)  st.pyplot() |

Pada tampilan modelling yang berisi informasi tentang confusion matrix naïve bayes.

* Tampilan Hasil klasifikasi Metode Naïve Bayes





* Code :

|  |
| --- |
| with implementation:  st.title("Klasifikasi Sentimen Ulasan Menggunakan Naive Bayes")  st.write("Masukkan Ulasan di bawah ini:")  from preprocessor import preprocessing, vectorizer, model  word2vec = Word2Vec.load("word2vec\_model\_file")  # Fungsi untuk mendapatkan vektor kata dari model  def get\_word\_vector(word):  try:  return word2vec.wv[word]  except KeyError:  return None  inp = st.text\_area(label="inp", label\_visibility="hidden", placeholder="Please input text...", height=350)  btn\_submit = st.button("submit")  if btn\_submit:  if inp == "":  st.write("error")  else:  prep = preprocessing(inp)  prep = " ".join([str(elem) for elem in prep])  def translate\_text(text, src\_lang, dest\_lang):  translator = Translator()  translated = translator.translate(text, src=src\_lang, dest=dest\_lang)  return translated.text  # Contoh penggunaan  translated\_text = translate\_text(prep, "id", "en")  clean = re.sub("[^a-zA-Zï ]+"," ", translated\_text)  tekons = nltk.word\_tokenize(clean.lower())  data = pd.DataFrame([[tekons]],columns=["prep"])  st.write(data)  test\_features\_word2vec = []  for index, row in data.iterrows():  model\_vector = np.mean([word2vec.wv.get\_vector(token) for token in row["prep"]], axis=0)  if model\_vector.dtype == 'float32':  test\_features\_word2vec.append(model\_vector)  else:  test\_features\_word2vec.append(np.array([0 for i in range(1000)]))  label = model.predict(test\_features\_word2vec)  st.write(label) |

Pada tampilan hasil klasifikasi metode naïve bayes ini ketika memasukkan kalimat “pemandangannya jelek” maka setelah di submit menampilkan kalimat yang telah di preprocessing “bad look“ dengan label “negative”

# Kesimpulan

Berdasarkan hasil analisis dan pengujian terhadap sistem analisis sentimen pada komentar pantai toroan kabupaten sampang menggunakan metode Naive Bayes, maka dapat ditarik kesimpulan sebagai berikut.

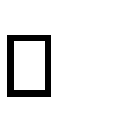
1. Metode Naive Bayes dapat memprediksi kelas sentimen pada ulasan komentar pantai toroan kabupaten sampang sesuai dengan sistem yang disiapkan.
2. Sistem analisis yang dibentuk belum sepenuhnya relevan dalam memprediksi kelas sentimen yang sesuai terhadap pemberian kelas dari supervisor.
3. Sistem analisis sentimen pada komentar pantai toroan kabupaten sampang menggunakan metode Naive Bayes menghasilkan nilai akurasi terendah pada pengujian 2 kelas menggunakan dataset 80% latih dan 20% data uji sebesar 57%.
4. Jumlah data latih dalam sistem analisis sentimen memiliki pengaruh terhadap prediksi sistem. Selain jumlah, kualitas data latih juga berperan karena semakin tinggi kualitas data maka sistem akan mendapatkan vocabulary yang semakin besar sehingga akan lebih tepat dalam memprediksi kelas sentimen.
5. Review yang diluar batasan cenderung akan memiliki kelas sentimen yang salah dikarenakan sistem tidak dilatih diluar batasan yang telah ditentukan. Seperti review dalam bahasa Inggris atau kata-kata gaul baru yang tidak terdapat dalam kamus

# Referensi

1. A. Fitri Niasita, P. P. Adikara, and S. Adinugroho, “Analisis Sentimen Pembangunan Infrastruktur di Indonesia dengan Automated Lexicon Word2Vec dan Naive-Bayes,” 2019. [Online].

Available: http://j-ptiik.ub.ac.id

1. R. Fajar, S. Program, P. Rekayasa, N. Lunak, and R. Bengkalis, “Implementasi Algoritma Naive Bayes Terhadap Analisis Sentimen Opini Film Pada Twitter,” vol. 3, no. 1.
2. B. R. Atmadja, “Analisis Sentimen Bahasa Indonesia Pada Tempat Wisata di Kabupaten Sukabumi Dengan Naïve Bayes,” vol. 15, no. 2, pp. 371–382, 2022, [Online]. Available:

http://journal.stekom.ac.id/index.php/elkom page371

1. R. Sari, R. Yulia Hayuningtyas, and S. Nusa Mandiri, “Penerapan Algoritma Naive Bayes Untuk Analisis Sentimen Pada Wisata TMII Berbasis Website,” *IJSE-Indonesian Journal on Software Engineering*, vol. 5, no. 2, pp. 51–60, 2019.
2. D. Darwis, N. Siskawati, and Z. Abidin, “Penerapan Algoritma Naive Bayes untuk Analisis Sentimen Review Data Twitter BMKG Nasional,” vol. 15, no. 1.
3. M. Rusli, M. R. Faisal, and I. Budiman, “EKSTRAKSI FITUR MENGGUNAKAN MODEL WORD2VEC

UNTUK ANALISIS SENTIMEN PADA KOMENTAR FACEBOOK,” *Seminar Nasional Ilmu Komputer (SOLITER)*, vol. 2, 2019.

1. B. Gunawan, H. Sasty, P. #2, E. Esyudha, and P. #3, “JEPIN (Jurnal Edukasi dan Penelitian Informatika) Sistem Analisis Sentimen pada Ulasan Produk Menggunakan Metode Naive Bayes,” vol. 4, no. 2, pp. 17–29, 2018, [Online]. Available: www.femaledaily.com
2. T. Krisdiyanto, E. Maricha, and O. Nurharyanto, “Analisis Sentimen Opini Masyarakat Indonesia Terhadap Kebijakan PPKM pada Media Sosial Twitter Menggunakan Naïve Bayes Clasifiers,” *Jurnal CoreIT*, vol. 7, no. 1, 2021.
3. B. Mas Pintoko and K. Muslim, “Analisis Sentimen Jasa Transportasi Online pada Twitter Menggunakan Metode Naïve Bayes Classifier.”
4. F. V. Sari and A. Wibowo, “ANALISIS SENTIMEN PELANGGAN TOKO ONLINE JD.ID

MENGGUNAKAN METODE NAÏVE BAYES CLASSIFIER BERBASIS KONVERSI IKON EMOSI,” *Jurnal SIMETRIS*, vol. 10, no. 2, 2019.

1. W. Widayat, “Analisis Sentimen Movie Review menggunakan Word2Vec dan metode LSTM Deep Learning,” *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 5, no. 3, p. 1018, Jul. 2021, doi: 10.30865/mib.v5i3.3111.
2. S. Fanissa, M. A. Fauzi, and S. Adinugroho, “Analisis Sentimen Pariwisata di Kota Malang Menggunakan Metode Naive Bayes dan Seleksi Fitur Query Expansion Ranking,” 2018. [Online]. Available: http://j-ptiik.ub.ac.id
3. D. Normawati and S. A. Prayogi, “Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter,” 2021.
4. N. M. A. J. Astari, Dewa Gede Hendra Divayana, and Gede Indrawan, “Analisis Sentimen

Dokumen Twitter Mengenai Dampak Virus Corona Menggunakan Metode Naive Bayes

Classifier,” *Jurnal Sistem dan Informatika (JSI)*, vol. 15, no. 1, pp. 27–29, Nov. 2020, doi:

10.30864/jsi.v15i1.332.

1. H. Utama and A. Masruro, “Analisis Sentimen pada Twitter menggunakan Word Embedding dengan Pendekatan Word2Vec,” 2022.