LAPORAN PROJECT

ANALISIS SENTIMEN PRODUK SKINCARE

PEMROSESAN TEKS

OLEH :

Kelompok 8

- Susy Susanty (22031554012)
- Abdini Qolbi Sahlina (22031554015)
- Michael Luwi Pallea' (22031554055)

PROGRAM STUDI SAINS DATA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS NEGERI SURABAYA

2023

BAB I
PENDAHULUAN

Industri perawatan kulit adalah bidang yang berkembang pesat yang didorong oleh meningkatnya kesadaran masyarakat terhadap kesehatan dan penampilan. Di era digital ini, konsumen semakin bersedia berbagi pengalaman dan pendapatnya mengenai produk perawatan kulit melalui berbagai platform online. Melimpahnya informasi produk kecantikan tidak lepas dari layanan mereka yang berbagi pengalaman setelah menggunakan produk kecantikan melalui media sosial, blog pribadi, atau website yang khusus memiliki fitur untuk mengulas produk kecantikan [1].

Dengan menganalisis data dari berbagai sumber online, project ini bertujuan untuk mengungkap secara detail bagaimana reaksi konsumen terhadap produk perawatan kulit dari berbagai merek. Klasifikasi sentimen bertujuan untuk mengatasi masalah ini dengan secara otomatis mengelompokkan review pengguna menjadi opini positif atau negatif [2]. Analisis sentimen menentukan apakah terdapat reaksi positif atau negatif secara keseluruhan terhadap produk tertentu dan faktor mana yang paling mempengaruhi persepsi konsumen. Oleh karena itu, project ini tidak hanya memberikan gambaran tren pasar, namun juga memberikan wawasan mendetail mengenai karakteristik produk yang paling dihargai konsumen.

Selain itu, project ini juga akan berkontribusi dalam pengembangan strategi pemasaran, peningkatan kualitas produk, dan pemahaman kekuatan dan kelemahan berbagai merek di pasar perawatan kulit. Review juga dapat untuk mengetahui umpan balik dari masyarakat terhadap brand kosmetik bagi perusahaan industri kosmetik [3]. Dengan menggabungkan analisis sentimen dengan data demografi dan tren konsumen, proyek ini diharapkan dapat memberikan gambaran komprehensif dan berkelanjutan mengenai tren industri perawatan kulit. Pendekatan ini diharapkan dapat memberikan nilai tambah bagi perusahaan di sektor ini dengan merespons kebutuhan dan harapan konsumen secara efektif.

BAB II
METODE

1. **Ekstraksi Fitur dengan TF-IDF:**

- Term Frequency-Inverse Document Frequency (TF-IDF) adalah teknik yang mengukur seberapa penting suatu kata dalam sebuah dokumen dalam sebuah koleksi dokumen.
- TF-IDF memberikan bobot pada kata-kata dalam teks berdasarkan seberapa sering kata tersebut muncul dalam dokumen tertentu (TF) dan seberapa umum kata tersebut di seluruh koleksi dokumen (IDF).
- Misalnya, kata-kata yang sering muncul dalam satu ulasan tetapi jarang muncul dalam ulasan lainnya cenderung memiliki bobot yang tinggi, menunjukkan kemungkinan pentingnya kata tersebut dalam merangkum sentimen ulasan.

2. **Word2Vec:**

- Word2Vec adalah teknik pemodelan bahasa yang mengubah kata-kata menjadi representasi vektor dalam ruang dimensi tertentu.
- Ini memungkinkan kata-kata dengan makna serupa untuk memiliki representasi vektor yang lebih dekat satu sama lain dalam ruang vektor.
- Dengan menggunakan Word2Vec, kita dapat memperoleh representasi vektor kata-kata dari ulasan produk yang kemudian dapat digunakan sebagai fitur dalam analisis sentimen.

3. **Algoritma Klasifikasi Naive Bayes:**

- Naive Bayes adalah algoritma klasifikasi yang berdasarkan pada teorema probabilitas Bayes dengan asumsi independensi antara fitur-fitur yang digunakan dalam klasifikasi.
- Algoritma ini cocok untuk klasifikasi teks dan sering digunakan dalam analisis sentimen karena kinerjanya yang baik bahkan dengan dataset yang relatif kecil.
- Dalam konteks analisis sentimen produk, Naive Bayes dapat digunakan untuk mengklasifikasikan ulasan menjadi kategori sentimen positif, negatif, atau netral berdasarkan fitur-fitur yang diekstraksi dari teks menggunakan TF-IDF atau representasi vektor Word2Vec.

Alasan kami memilih ekstraksi fitur TF-IDF dan Word2vec dalam project kami yaitu karena dapat memberikan keuntungan yang lebih besar dengan memanfaatkan kekuatan keduanya. Dalam Word2vec kami dapat menggunakan vektor untuk menangkap semantik dan konteks, sementara TF-IDF digunakan untuk menyoroti kata-kata yang mungkin penting secara spesifik dalam data serta lebih efisien dalam menghitung matriks. Sedangkan untuk Naive Bayes, algoritma ini termasuk kedalam algoritma klasifikasi yang sederhana dan efisien. Cara kerjanya yang mampu menangani jumlah data yang besar tanpa memerlukan waktu komputasi yang signifikan. Jika ulasan produk cenderung pendek, seperti ada beberapa platform e-commerce, Naive Bayes dapat memberikan hasil yang memuaskan. Algoritma ini juga dapat memanfaatkan kata-kata yang muncul dalam teks pendek untuk menghasilkan prediksi sentimen.

BAB III
HASIL DAN ANALISIS

- Pencarian data melalui github
  Data yang diperoleh adalah tentang review produk skincare, dimana terdapat 6 fitur yaitu ID Data, Nama Produk, Jenis Kulit, Review Produk, Rating, dan Label.

dataset.csv ×

1 to 10 of 500 entries Filter

ID Data	Nama Produk	Jenis Kulit	Review Produk	Rating	Label
A001	Cetaphil: Gentle Skin Cleanser	Oily	Aku suka banget sama ini. Cuma ini cleanser yang cocok buat aku. Aku udah sering banget gonta-ganti cleanser karena wajahku yang penuh jerawat. Aku udah coba yang low-end dan high-end sekalipun tapi tetep gak ada yang sebagus ini. Karena produknya gentle jadi gak bikin iritasi diwajah, kebanyakan produk cleanser lain yang harsh untuk wajah justru dapat menyebabkan iritasi sehingga munculah jerawat. Very recommended!	5	Positive
A002	Cetaphil: Gentle Skin Cleanser	Oily	Walaupun produknya mengklaim dapat digunakan sebagai makeup removal, aku tetap gak menggunakannya sebagai makeup removal. Tapi sebagai pencuci muka, iya. Karena sangat lembut. Awal pemakaian agak kaget, karena tidak berbusa sama sekali. Gak bikin breakout what soever. Kalau dibandingin sama produk pencuci mukaku sebelumnya, yang meninggalkan kesan kesat di wajah, maka produk ini membuat kulit terasa bersih tapi tetap lembut dan kenyal. Harga 150.000 untuk kemasan 16oz di century, ya. Tempat lain mungkin berbeda harganya, bisa lebih mahal atau lebih murah. Kesimpulan: puas :)	5	Positive
A003	Cetaphil: Gentle Skin Cleanser	Dry	Thanks God for this product. Produk ini udah paling cocok buat aku. Bisa dibuat cuci muka, pembersih make up dan sabun mandi. Kulitku sangat kering dan alergi. Produk ini direkomendasi sama dokter alergiku. Dan memang, udah paling best banget buat aku. Aku sih sekarang lebih buat pake pembersih make up dan sabun muka, untuk sabun beralih ke sabun baby aja, secara harganya lumayan kalo dipake buat badan yang paling gede pun cepet aja abisnya :D	5	Positive
A004	Cetaphil: Gentle Skin Cleanser	Oily	Another favorite product from Cetaphil. Seneng banget dengan produk ini karena aman untuk kulitku yang oily-acne prone yang mudah sensitive ini. Setelah 2 bulanan pakai cleanser ini kulit terasa lebih "kalem", walau sedang PMS pun, jerawat yang tadinya banyak berdatangan jadi jarang, paling hanya 1-3 saja which is a very good sign for me. Selain itu, setiap habis cuci muka, kulit terasa lebih kenyal dan lembut.	5	Positive
A005	Cetaphil: Gentle Skin Cleanser	Oily	awalnya kaget karena cleanser ini enggak berbusa sama sekali, tapi lama-lama jadi kebiasaan. saya suka cleanser ini karena sangat gentle di kulit saya yang sensitif dan acne prone. kebnyakan cleanser bakal bikin kulit saya terasa tertarik, tapi produk ini tidak.	5	Positive
A006	Cetaphil: Gentle Skin Cleanser	Combination	So far aku cocok sama cetaphil ini. Dan sudah purchase yang ke dua kalinya dalam botol besar buat dipakai bareng sama mami dan adik aku yang kulitnya cenderung normal-dry dan cocok juga buat mereka. Kalau mereka pakai purely buat cuci muka tapi aku pakai cetaphil ini pake bersihin make up. Karena cetaphil ini memang bisa dipake tanpa dan atau dengan air. Aku coba buat bersihin make up eh enak dan bersih juga dengan dua kali apply, dan sampai sekarang stuck pakai ini karena after feelingnya yang fresh dan nggak lengket. Teksturenya gel gitu nggak ada wanginya, sangat gentle di kulit. Definitely gonna do a repurchase!	5	Positive
A007	Cetaphil: Gentle	Combination	My HG cleanser! Dengan kulit yg oily tapi kering mengarah ke dehidrasi, pembersih ini cocok banget. Ga bikin tambah kering, bersihin dan bikin lembab.	5	Positive

Activate Windows
Go to Settings to activate Windows.

- *Pre-processing* data
  Preprocessing dilakukan untuk menghindari dataset yang kurang sempurna, terdapat noise pada dataset, data-data yang tidak konsisten dan mempercepat pemrosesan terhadap dokumen [4]. Pre-processing yang dilakukan yaitu lower casing, normalisasi label, koreksi duplikasi, menghapus spesial karakter dan punctuation, normalisasi whitespace, menghapus stopword, lemmatization, dan stemming. Sehingga, munculah data yang telah dibersihkan dengan jumlah fitur menjadi 2, yaitu Review Produk dan Label.

data_clean.csv ×

1 to 10 of 500 entries Filter

ID Produk	Produk	Review Produk	Label
A001	Cetaphil: Gentle Skin Cleanser	aku suka banget sama cuma cleanser cocok buat aku aku udah sering banget gonta ganti cleanser wajah penuh jerawat aku udah coba low end high end sekalipun tetep ada bagus karena produk gentle jadi bikin iritasi wajah banyak produk cleanser yang harsh wajah justru sebab iritasi sehinga munculah jerawat very recomended	positive
A002	Cetaphil: Gentle Skin Cleanser	walaupun produk klaim guna makeup removal aku tetap mengunakanya makeup removal bagai cuci muka iya sangat lembut awal pakai kaget busa sama sekali tidak bikin breakout what soever kalau dibandingin sama produk cuci muka belum meningalkan kesan kesat wajah produk buat kulit asa bersih tetap lembut kenyal harga kemas oz century tempat mungkin beda harga lebih mahal lebih murah simpul puas	positive
A003	Cetaphil: Gentle Skin Cleanser	thanks good for this product produk udah paling cocok buat aku buat cuci muka bersih make up sabun mandi kulit sangat kering alergi produk rekomendasi sama dokter alergi memang udah paling best banget buat aku aku sih sekarang lebih buat pake bersih make up sabun muka sabun alih sabun baby aja harga lumayan kalo dipake buat badan paling gede cepet aja abis d	positive
A004	Cetaphil: Gentle Skin Cleanser	another favorite product from cetaphil neng banget produk karena aman kulit oily acne prone mudah sensitive telah bulan pakai cleanser kulit asa lebih kalem sedang pm jerawat tadi banyak datang jadi jarang paling saja which is a very good sign for me itu habis cuci muka kulit asa lebih kenyal lembut	positive
A005	Cetaphil: Gentle Skin Cleanser	awal kaget cleanser engak busa sama sekali lama lama jadi bias suka cleanser karena sangat gentle kulit yang sensitif acne prone kebnyakan cleanser bakal bikin kulit asa tarik produk tidak	positive
A006	Cetaphil: Gentle Skin Cleanser	so far aku cocok sama cetaphil purchase ke kali botol besar buat pakai bareng sama mami adik aku kulit cenderung normal dry cocok buat kalau pakai purely buat cuci muka aku pakai cetaphil pake bersihin make up cetaphil memang dipake dengan air aku coba buat bersihin make up eh enak bersih dengan kali aply dan sampai sekarang stuck pakai karena after felingnya fresh dan ngak lengket teksturenya gel gitu ngak wangi sangat gentle kulit definitely gona do a repurchase	positive
A007	Cetaphil: Gentle Skin Cleanser	my hg cleanser kulit yg oily kering arah dehidrasi bersih cocok banget ga bikin tambah kering bersihin bikin lembab awal pake aneh sensasi sat cuci wajah cetaphil ga asa pake sabun khawatir ga bersih kan telah bilas rasa bersih enak banget udah repurchase botol maybe	positive
A008	Cetaphil: Gentle Skin	sekarang tetap setia mengunakan cethapil clenaser wajah sejak mengunakan produk hampir bagi masalah wajah seperti beruntus timbul mengunakanya adan wajah kering masage sedikit kemudian angkat tisue kemudian ulang sampai ada bekas make up kotor tisue kemudian biar efek bersih lebih nyata tahap akhir bilas air di lap kering tisue	positive

Activate Windows
Go to Settings to activate Windows.

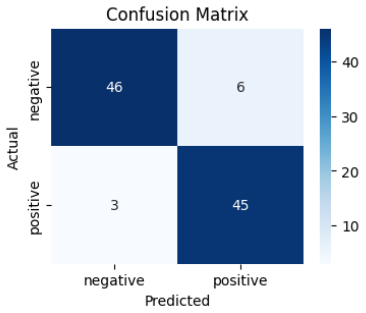
- Ekstraksi Fitur
  - ➔ TF-IDF
    TF-IDF dapat dirumuskan sebagai berikut [5] :

$$TF\ IDF(\ tkdj) = TF(\ tkdj) \times IDF(\ tk)$$

    Akurasi yang diperoleh untuk pengimplementasian TF-IDF dengan distribusi multinomial adalah 0,91 sedangkan dengan distribusi gaussian adalah 0,76.

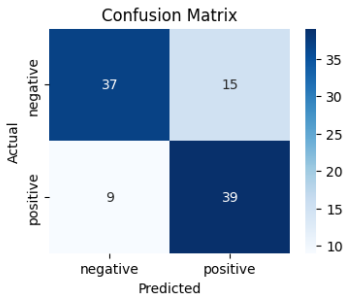
Classifier Multinomial Naive Bayes

```
Accuracy: 0.91
              precision    recall  f1-score   support

    negative       0.94      0.88      0.91        52
    positive       0.88      0.94      0.91        48

    accuracy                           0.91       100
   macro avg       0.91      0.91      0.91       100
weighted avg       0.91      0.91      0.91       100
```



Classifier Gaussian Naive Bayes

```
Accuracy: 0.76
              precision    recall  f1-score   support

    negative       0.80      0.71      0.76        52
    positive       0.72      0.81      0.76        48

    accuracy                           0.76       100
   macro avg       0.76      0.76      0.76       100
weighted avg       0.76      0.76      0.76       100
```



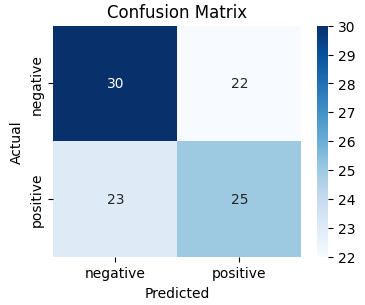
➔ Word2vec

Akurasi yang diperoleh untuk pengimplementasian word2vec dengan distribusi gaussian adalah 0,53.

Dalam ekstraksi fitur tersebut, sudah terdapat gabungan dengan pengimplementasian algoritma Naive Bayes.

Classifier Gaussian Naive Bayes

```
Accuracy: 0.55
              precision    recall  f1-score   support

    negative       0.57      0.58      0.57        52
    positive       0.53      0.52      0.53        48

    accuracy                           0.55       100
   macro avg       0.55      0.55      0.55       100
weighted avg       0.55      0.55      0.55       100
```



- Hasil Sentimen
  Setelah melakukan semua rangkaian diatas, kita dapat memunculkan hasil dari kalimat review produk yang diinput yaitu antara positif atau negatif, sesuai dengan analisis yang sudah dilakukan oleh tahap-tahap sebelumnya.

```
from sklearn.feature_extraction.text import TfidfVectorizer
import joblib
import pandas as pd

tfidf_vectorizer = TfidfVectorizer()
tfidf_train = tfidf_vectorizer.fit_transform(X_train)
joblib.dump(tfidf_vectorizer, 'tfidf_vectorizer.pkl')
joblib.dump(classifier, 'classifier_model.pkl')



tfidf_vectorizer = joblib.load('tfidf_vectorizer.pkl')
classifier = joblib.load('classifier_model.pkl')
```

```
data_baru = ["Pencuci muka ini memberikan hasil bersih yang memuaskan, teksturnya ringan dan mudah diratakan di wajah. Aromanya yang lembut memberikan sensasi k
tfidf_data_baru = tfidf_vectorizer.transform(data_baru)

hasil_prediksi = classifier.predict(tfidf_data_baru)
print(hasil_prediksi)
```

```
['positive']
```

BAB IV
KESIMPULAN

Project Analisis Sentimen Produk Skincare memberikan wawasan mendalam tentang persepsi konsumen terhadap berbagai produk perawatan kulit. Dengan menganalisis data dari berbagai sumber seperti ulasan online, media sosial, dan platform e-commerce, proyek ini dapat menyimpulkan bahwa secara umum terdapat sentimen positif atau negatif terhadap produk tertentu. Selain itu, proyek ini mengidentifikasi faktor-faktor berikut yang penting dalam membentuk opini konsumen, seperti kualitas bahan, efektivitas produk, harga, dan layanan pelanggan.

Menganalisis data dari waktu ke waktu dapat mengidentifikasi tren perilaku konsumen, memberikan wawasan berharga bagi bisnis untuk merespons perubahan pasar. Penyelesaian project ini akan memberikan landasan bagi perusahaan perawatan kulit untuk merekomendasikan perbaikan dan pengembangan produk, meningkatkan strategi merek, dan merancang strategi pemasaran yang lebih efektif. Selain itu, project ini memberikan wawasan mengenai citra merek dan reputasi perusahaan dimata konsumen, membantu mengembangkan strategi keterlibatan konsumen yang lebih efektif. Oleh karena itu, analisis sentimen terhadap produk skincare menjadi alat penting untuk mengambil keputusan yang lebih baik dalam industri perawatan kulit.

REFERENSI

[1] N. B. N-gram, I. Pujadayanti, M. A. Fauzi, and Y. A. Sari, “Prediksi Rating Otomatis pada Ulasan Produk Kecantikan dengan Metode Prediksi Rating Otomatis pada Ulasan Produk Kecantikan dengan Metode Naïve Bayes dan N-gram,” no. April, 2018.

[2] E. Indrayuni, “Klasifikasi Text Mining Review Produk Kosmetik Untuk Teks Bahasa Indonesia Menggunakan Algoritma Naive Bayes,” J. Khatulistiwa Inform., vol. 7, no. 1, pp. 29–36, 2019, doi: 10.31294/jki.v7i1.1.

[3] B. Gunawan, H. S. Pratiwi, and E. E. Pratama, “Sistem Analisis Sentimen pada Ulasan Produk Menggunakan Metode Naive Bayes,” J. Edukasi dan Penelit. Inform., vol. 4, no. 2, p. 113, 2018, doi: 10.26418/jp.v4i2.27526.

[4] Ratino, N. Hafidz, S. Anggraeni, and W. Gata, “Sentimen Analisis Informasi Covid-19 menggunakan Support Vector Machine dan Naïve Bayes,” J. JUPITER, vol. 12, no. 2, pp. 1–11, 2020.

[5] C. H. Yutika, A. Adiwijaya, and S. Al Faraby, “Analisis Sentimen Berbasis Aspek pada Review Female Daily Menggunakan TF-IDF dan Naïve Bayes,” J. Media Inform. Budidarma, vol. 5, no. 2, p. 422, 2021, doi: 10.30865/mib.v5i2.2845.

LAMPIRAN

Code Preposessing

```
import pandas as pd
import numpy as np
import pandas as pd
import re
import nltk
import spacy
import string
import csv
pd.options.display.max_columns=None
pd.options.display.max_rows=None
pd.options.display.max_colwidth=None
```

```
import pandas as pd
url = 'https://raw.githubusercontent.com/chlaudiah/Sentiment-
Classification-FD-Reviews/master/dataset.csv'
data = pd.read_csv(url)
data
```

```
def clean_lower(kecil):
    kecil = kecil.lower()
    return kecil
data['Lower_case'] = data['Review Produk'].apply(clean_lower)
huruf_kecil=pd.DataFrame(data['Lower_case'])
huruf_kecil.head(2)
```

```
def normalize_label(kecil):
    kecil = kecil.lower()
    return kecil
data['normalize_label'] = data['Label'].apply(normalize_label)
huruf_kecil=pd.DataFrame(data['normalize_label'])
```

```
def koreksi_duplikasi(text):
  return re.sub(r'(.)\1+', r'\1', text)

data['koreksi_duplikasi'] = data['Lower_case'].apply(koreksi_duplikasi)
kor_dup = pd.DataFrame(data ['koreksi_duplikasi'])
kor_dup
```

```
url = 'https://raw.githubusercontent.com/nadyndyaa/Kamus-
Alay/main/Kamus%20Alay.csv'
formal = pd.read_csv(url,sep = ',')
```

```
singkatan_dict = formal.set_index('Alay')['Baik'].to_dict()
def ganti_singkatan(teks):
    singkatan = re.findall(r'\b\w+\b', teks)
    for kata in singkatan:
        if kata in singkatan_dict:
            teks = re.sub(r'\b{}\b'.format(kata), singkatan_dict[kata],
teks, flags=re.IGNORECASE)
    return teks
data['no_singkatan'] = data['koreksi_duplikasi'].apply(ganti_singkatan)
tanpa_singkatan = pd.DataFrame(data['no_singkatan'])
tanpa_singkatan
```

```
formal = pd.read_csv('/content/KamusAlay+.csv',sep = ';')
singkatan_dict = formal.set_index('NonFormal')['Formal'].to_dict()

def ganti_singkatan(teks):
    singkatan = re.findall(r'\b\w+\b', teks)
    for kata in singkatan:
        if kata in singkatan_dict:
            teks = re.sub(r'\b{}\b'.format(kata), singkatan_dict[kata],
teks, flags=re.IGNORECASE)
    return teks

data['no_singkatan+'] = data['no_singkatan'].apply(ganti_singkatan)
tanpa_singkatan = pd.DataFrame(data['no_singkatan+'])
tanpa_singkatan
```

```
def hapus_angka(teks):
  return re.sub(r"\d+", " ", teks)
data['tanpa_angka'] = data['no_singkatan+'].apply(hapus_angka)
tanpa_angka=pd.DataFrame(data['tanpa_angka'])
tanpa_angka.head(2)
```

```
clean_spcl = re.compile('[/(){}\[\]\|@,;]')
clean_symbol = re.compile('[^0-9a-z]')
def clean_punct(text):
    text = clean_spcl.sub('', text)
    text = clean_symbol.sub(' ', text)
    return text
data['clean_punct'] = data['tanpa_angka'].apply(clean_punct)
data_bersih = pd.DataFrame(data['clean_punct'])
```

```
def _normalize_whitespace(text):
    corrected = str(text)
    corrected = re.sub(r"//t",r"\t", corrected)
    corrected = re.sub(r"( )\1+",r"\1", corrected)
    corrected = re.sub(r"(\n)\1+",r"\1", corrected)
```

```
    corrected = re.sub(r"(\r)\1+",r"\1", corrected)
    corrected = re.sub(r"(\t)\1+",r"\1", corrected)
    return corrected.strip(" ")
data['normal_space'] = data['clean_punct'].apply(_normalize_whitespace)
normal_space = pd.DataFrame(data ['normal_space'])
```

```
from nltk.tokenize import word_tokenize
from Sastrawi.StopWordRemover.StopWordRemoverFactory import
StopWordRemoverFactory
factory = StopWordRemoverFactory()
stopwords = factory.get_stop_words()


factory = StopWordRemoverFactory()
stopword = factory.create_stop_word_remover()
def remove_stopwords(text):
    return stopword.remove(text)

data['hasil_stopword'] = data['normal_space'].apply(remove_stopwords)
hasilstop_words = pd.DataFrame(data['hasil_stopword'])
hasilstop_words
```

```
import nltk
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize
nltk.download('wordnet')

def lemmatize_text(text):
    lemmatizer = WordNetLemmatizer()
    tokens = nltk.word_tokenize(text)
    lemmatized_tokens = [lemmatizer.lemmatize(token) for token in
tokens]
    lemmatized_text = ' '.join(lemmatized_tokens)
    return lemmatized_text

data['hasil_lemma'] = data['hasil_stopword'].apply(lemmatize_text)
hasilstop_words = pd.DataFrame(data['hasil_lemma'])
hasilstop_words
```

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
factory = StemmerFactory()
stemmer = factory.create_stemmer()

def steming(text):
  return stemmer.stem(text)
```

```
data['hasil_stem'] = data['hasil_lemma'].apply(steming)
hasil_stem = pd.DataFrame(data['hasil_stem'])
hasil_stem
```

```
import os
nama_folder = 'Folder_baru'

if not os.path.exists(nama_folder):
    os.makedirs(nama_folder)
data_clean = {'ID Produk': (data['ID Data']),'Produk': (data['Nama
Produk']),'Review Produk':
(data['hasil_stem']),'Label':(data['normalize_label'])}
df = pd.DataFrame(data_clean)
nama_file_csv = 'data_clean.csv'
df.to_csv(os.path.join(nama_folder, nama_file_csv), index=False)
df
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(data_clean['Review
Produk'], data_clean['Label'],
                                                    test_size = 0.20,
                                                    random_state = 42)
```

```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf_vectorizer = TfidfVectorizer()
tfidf_train = tfidf_vectorizer.fit_transform(X_train)
tfidf_test = tfidf_vectorizer.transform(X_test)

X_train_vec = pd.DataFrame(tfidf_train.toarray())
X_test_vec = pd.DataFrame(tfidf_test.toarray())

X_train_vec.head()
```

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report
from sklearn.naive_bayes import GaussianNB

from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
import seaborn as sns
```

```
classifier = MultinomialNB()
classifier.fit(tfidf_train, y_train)

prediksi = classifier.predict(X_test_vec)
confus_mat = confusion_matrix(y_test, prediksi)
akurasi = accuracy_score(y_test, prediksi)
print(f'Accuracy: {akurasi}')
print(classification_report(y_test, prediksi))

plt.figure(figsize=(4, 3))
sns.heatmap(confus_mat, annot=True, fmt='d', cmap='Blues',
xticklabels=classifier.classes_, yticklabels=classifier.classes_)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from gensim.models import Word2Vec
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
import nltk

def word2vec_embedding(text, model):
    tokens = word_tokenize(text)
    word_vectors = [model.wv[word] for word in tokens if word in
model.wv]
    text_vector = np.mean(word_vectors, axis=0) if word_vectors else
np.zeros(model.vector_size)
    return text_vector

sentences = [word_tokenize(text) for text in X_train]
word2vec_model = Word2Vec(sentences, vector_size=100, window=5,
min_count=1, workers=4)

train_vectors = np.array([word2vec_embedding(text, word2vec_model) for
text in X_train])
test_vectors = np.array([word2vec_embedding(text, word2vec_model) for
text in X_test])
```

```
nb_classifier = GaussianNB()
nb_classifier.fit(train_vectors, y_train)

predictions = nb_classifier.predict(test_vectors)
accuracy = accuracy_score(y_test, predictions)
confusion = confusion_matrix(y_test, predictions)

print(f'Accuracy: {accuracy}')
print(classification_report(y_test, predictions))
plt.figure(figsize=(4, 3))
sns.heatmap(confusion, annot=True, fmt='d', cmap='Blues',
xticklabels=classifier.classes_, yticklabels=classifier.classes_)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```

```
nb_classifier = GaussianNB()
nb_classifier.fit(tfidf_train.toarray(), y_train)

predictions = nb_classifier.predict(tfidf_test.toarray())
accuracy = accuracy_score(y_test, predictions)
confusion = confusion_matrix(y_test, predictions)

print(f'Accuracy: {accuracy}')
print(classification_report(y_test, predictions))
plt.figure(figsize=(4, 3))
sns.heatmap(confusion, annot=True, fmt='d', cmap='Blues',
xticklabels=classifier.classes_, yticklabels=classifier.classes_)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
import joblib
import pandas as pd

tfidf_vectorizer = TfidfVectorizer()
tfidf_train = tfidf_vectorizer.fit_transform(X_train)
joblib.dump(tfidf_vectorizer, 'tfidf_vectorizer.pkl')
joblib.dump(classifier, 'classifier_model.pkl')
```

```
tfidf_vectorizer = joblib.load('tfidf_vectorizer.pkl')
classifier = joblib.load('classifier_model.pkl')
```

```
data_baru = ["Pencuci muka ini memberikan hasil bersih yang memuaskan, teksturnya ringan dan mudah diratakan di wajah. Aromanya yang lembut memberikan sensasi kesegaran yang menyenangkan, dan kulit terasa lembut setelah penggunaan. Selain itu, formulanya cocok untuk berbagai jenis kulit, membuatnya menjadi pilihan yang baik untuk perawatan kulit sehari-hari."]
tfidf_data_baru = tfidf_vectorizer.transform(data_baru)


hasil_prediksi = classifier.predict(tfidf_data_baru)
print(hasil_prediksi)
```

Code GUI

```
from customtkinter import *
import joblib
from PIL import Image, ImageTk
import re
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import pandas as pd
from nltk.tokenize import word_tokenize
from Sastrawi.StopWordRemover.StopWordRemoverFactory import
StopWordRemoverFactory
import nltk
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize
nltk.download('wordnet')
nltk.download('punkt')
factory = StemmerFactory()
stemmer = factory.create_stemmer()


app = CTk()
app.geometry("900x600")


background_image = Image.open("bckgrnd.png")
background_photo = ImageTk.PhotoImage(background_image)

background_label = CTkLabel(app, image=background_photo)
background_label.place(relwidth=1, relheight=1)


tfidf_vectorizer = joblib.load('tfidf_vectorizer.pkl')
classifier = joblib.load('classifier_model.pkl')


main_frame = CTkFrame(master=background_label, fg_color='#4EAC7D')
```

```
main_frame.place(relx=0.5, rely=0.5, anchor="center")

frame_3 = CTkFrame(master=main_frame, fg_color="#4EAC7D")
frame_3.grid(row=1, column=1, padx=50, pady=50)


label = CTkLabel(master=frame_3, text="Analisis Review",
                 font=("Arial Bold", 30), justify="left")
label.pack(expand=True, pady=(30, 15))


entry = CTkEntry(
    master=frame_3, placeholder_text="Masukkan Review Disini",
width=400)
entry.pack(expand=True, pady=15, padx=20)

hasil_entry = CTkTextbox(master=frame_3, width=400)
hasil_entry.pack(expand=True, pady=10, padx=20)


factory = StemmerFactory()
stemmer = factory.create_stemmer()

factory = StopWordRemoverFactory()
stopwords = factory.get_stop_words()


url = 'https://raw.githubusercontent.com/nadyndyaa/Kamus-
Alay/main/Kamus%20Alay.csv'
formal = pd.read_csv(url, sep=',')
singkatan_dict = formal.set_index('Alay')['Baik'].to_dict()

formal = pd.read_csv('KamusAlay+.csv', sep=';')
singkatan_dict = formal.set_index('NonFormal')['Formal'].to_dict()


def ganti_singkatan(teks):
    singkatan = re.findall(r'\b\w+\b', teks)
    for kata in singkatan:
        if kata in singkatan_dict:
            teks = re.sub(r'\b{}\b'.format(kata),
                          singkatan_dict[kata], teks,
flags=re.IGNORECASE)
    return teks


def ganti_singkatan_(teks):
    singkatan = re.findall(r'\b\w+\b', teks)
    for kata in singkatan:
```

```
        if kata in singkatan_dict:
            teks = re.sub(r'\b{}\b'.format(kata),
                          singkatan_dict[kata], teks,
flags=re.IGNORECASE)
    return teks


def _normalize_whitespace(text):
    corrected = str(text)
    corrected = re.sub(r"//t", r"\t", corrected)
    corrected = re.sub(r"( )\1+", r"\1", corrected)
    corrected = re.sub(r"(\n)\1+", r"\1", corrected)
    corrected = re.sub(r"(\r)\1+", r"\1", corrected)
    corrected = re.sub(r"(\t)\1+", r"\1", corrected)
    return corrected.strip(" ")


stopword = factory.create_stop_word_remover()


def lemmatize_text(text):
    lemmatizer = WordNetLemmatizer()
    tokens = nltk.word_tokenize(text)
    lemmatized_tokens = [lemmatizer.lemmatize(token) for token in
tokens]
    lemmatized_text = ' '.join(lemmatized_tokens)
    return lemmatized_text


def click_button():
    entered_text = entry.get()
    teks_bersih = entered_text.lower()
    # teks_bersih = re.sub(r'(.)\1+', r'\1', teks_bersih)
    teks_bersih = ganti_singkatan(teks_bersih)
    teks_bersih = ganti_singkatan_(teks_bersih)
    teks_bersih = re.sub(r"\d+", " ", teks_bersih)
    teks_bersih = re.compile('[/(){}\[\]\|@,;]').sub('', teks_bersih)
    teks_bersih = re.compile('[^0-9a-z]').sub(' ', teks_bersih)
    teks_bersih = _normalize_whitespace(teks_bersih)
    teks_bersih = stopword.remove(teks_bersih)
    teks_bersih = lemmatize_text(teks_bersih)
    teks_bersih_ = stemmer.stem(teks_bersih)

    hasil_entry.delete("1.0", "end")
    hasil_entry.insert("1.0", entered_text)
    tfidf_data_baru = tfidf_vectorizer.transform([teks_bersih_])
    hasil_prediksi = classifier.predict(tfidf_data_baru)
    hasil_entry.insert(
```

```
        "end", f"\n\nSetelah Pembersihan:{teks_bersih_}\n\n Hasil
Prediksi: {hasil_prediksi[0]}")
    print(hasil_prediksi)


button = CTkButton(master=frame_3, text="Prediksi",
command=click_button)
button.pack(expand=True, fill="both", pady=(30, 15), padx=30)


app.mainloop()
```

Tampilan GUI

Kontribusi Anggota Kelompok

- Michael Luwi Pallea' : pre-processing data, ekstraksi fitur TF-IDF, implementasi algoritma dalam GUI.
- Susy Susanty dan Abdini Qolbi Sahlina : ekstraksi fitur Word2vec dan design GUI.

Penambahan setelah presentasi project :

- Mencantumkan alasan mengapa menggunakan metode TF-IDF dan Word2vec dalam project ini.
  Keterangan : sudah tercatat pada Bab II Metode.
- Menambahkan fitur upload file pada GUI
  Code GUI:

```
from customtkinter import *
import joblib
```

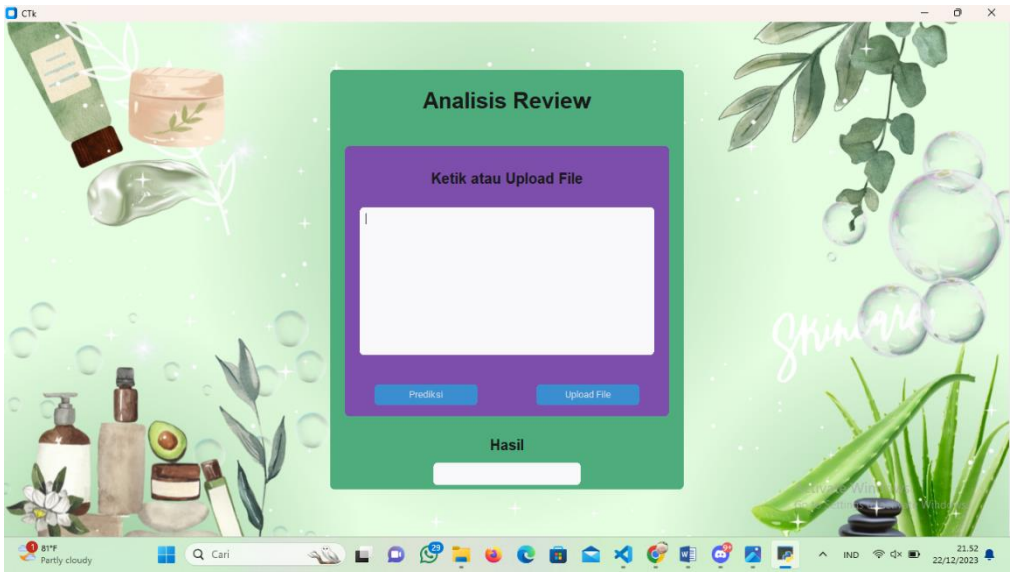
```
from PIL import Image, ImageTk
import re
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import pandas as pd
from nltk.tokenize import word_tokenize
from Sastrawi.StopWordRemover.StopWordRemoverFactory import
StopWordRemoverFactory
import nltk
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize
from docx import Document
# from tkinter import filedialog
# nltk.download('wordnet')
# nltk.download('punkt')

factory = StemmerFactory()
stemmer = factory.create_stemmer()


app = CTk()
app.geometry("900x6000")


background_image = Image.open("background.png")
background_photo = ImageTk.PhotoImage(background_image)
''
background_label = CTkLabel(master=app, image=background_photo)
background_label.place(relwidth=1, relheight=1)


tfidf_vectorizer = joblib.load('tfidf_vectorizer.pkl')
classifier = joblib.load('classifier_model.pkl')


main_frame = CTkFrame(master=app, fg_color='#4EAC7D')
main_frame.place(relx=0.5, rely=0.5, anchor='center')


frame_3 = CTkFrame(master=main_frame, fg_color="#7D4EAC")
frame_3.grid(row=1, column=1, padx=20, pady=20)

label = CTkLabel(master=main_frame, text="Analisis Review",
                 font=("Arial Bold", 30), justify="left")
label.grid(row=0, column=1, padx=50, pady=24)

label = CTkLabel(master=frame_3, text="Ketik atau Upload File",
                 font=("Arial Bold", 20), justify="left")
label.pack(expand=True, pady=(30, 15))
```

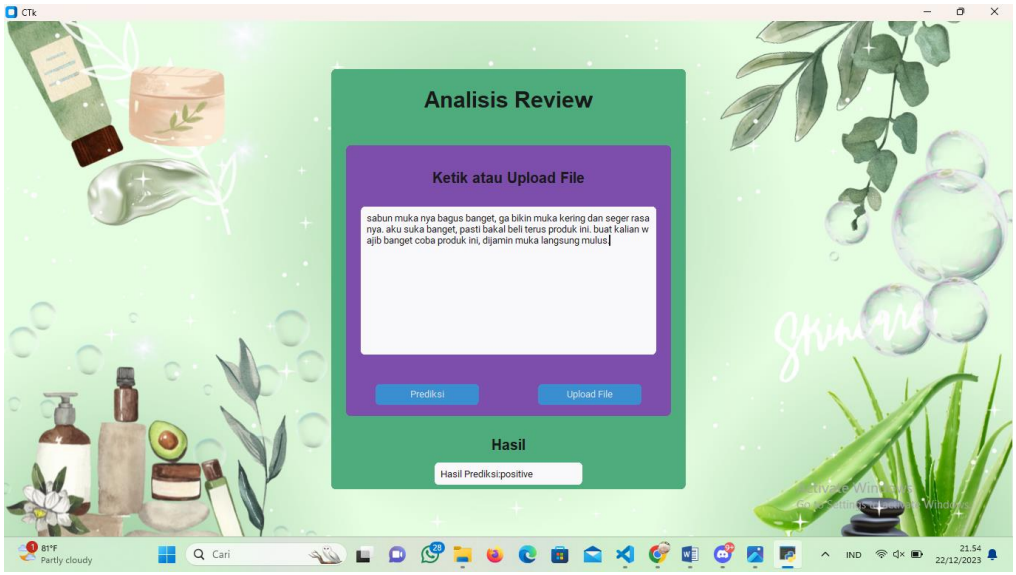
```
hasil_entry = CTkTextbox(master=frame_3, width=400)
hasil_entry.pack(expand=True, pady=10, padx=20)


factory = StemmerFactory()
stemmer = factory.create_stemmer()

factory = StopWordRemoverFactory()
stopwords = factory.get_stop_words()


url = 'https://raw.githubusercontent.com/nadyndyaa/Kamus-
Alay/main/Kamus%20Alay.csv'
formal = pd.read_csv(url, sep=',')
singkatan_dict = formal.set_index('Alay')['Baik'].to_dict()

formal = pd.read_csv('KamusAlay+.csv', sep=';')
singkatan_dict = formal.set_index('NonFormal')['Formal'].to_dict()


def ganti_singkatan(teks):
    singkatan = re.findall(r'\b\w+\b', teks)
    for kata in singkatan:
        if kata in singkatan_dict:
            teks = re.sub(r'\b{}\b'.format(kata),
                          singkatan_dict[kata], teks,
flags=re.IGNORECASE)
    return teks


def ganti_singkatan_(teks):
    singkatan = re.findall(r'\b\w+\b', teks)
    for kata in singkatan:
        if kata in singkatan_dict:
            teks = re.sub(r'\b{}\b'.format(kata),
                          singkatan_dict[kata], teks,
flags=re.IGNORECASE)
    return teks


def _normalize_whitespace(text):
    corrected = str(text)
    corrected = re.sub(r"//t", r"\t", corrected)
    corrected = re.sub(r"( )\1+", r"\1", corrected)
    corrected = re.sub(r"(\n)\1+", r"\1", corrected)
    corrected = re.sub(r"(\r)\1+", r"\1", corrected)
    corrected = re.sub(r"(\t)\1+", r"\1", corrected)
    return corrected.strip(" ")
```

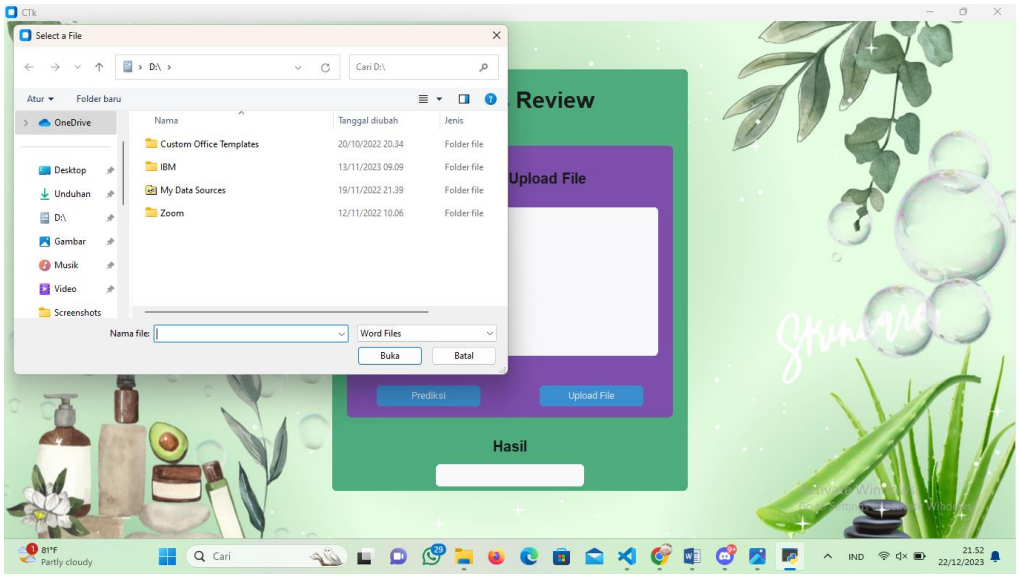
```
stopword = factory.create_stop_word_remover()


def lemmatize_text(text):
    lemmatizer = WordNetLemmatizer()
    tokens = nltk.word_tokenize(text)
    lemmatized_tokens = [lemmatizer.lemmatize(token) for token in
tokens]
    lemmatized_text = ' '.join(lemmatized_tokens)
    return lemmatized_text


def click_button():
    entered_text = hasil_entry.get("1.0", "end-1c")
    if not entered_text:
        hasil_entry.delete("1.0", "end")
        hasil_entry.insert("1.0", "Error: Masukkan kalimat terlebih
dahulu.")
        return
    teks_bersih = entered_text.lower()
    teks_bersih = re.sub(r'(.)\1+', r'\1', teks_bersih)
    teks_bersih = ganti_singkatan(teks_bersih)
    teks_bersih = ganti_singkatan_(teks_bersih)
    teks_bersih = re.sub(r"\d+", " ", teks_bersih)
    teks_bersih = re.compile('[/(){}\[\]\|@,;]').sub('', teks_bersih)
    teks_bersih = re.compile('[^0-9a-z]').sub(' ', teks_bersih)
    teks_bersih = _normalize_whitespace(teks_bersih)
    teks_bersih = stopword.remove(teks_bersih)
    teks_bersih = lemmatize_text(teks_bersih)
    teks_bersih_ = stemmer.stem(teks_bersih)

    hasil_entry.delete("1.0", "end")
    hasil_entry.insert("1.0", entered_text)
    tfidf_data_baru = tfidf_vectorizer.transform([teks_bersih_])
    hasil_prediksi = classifier.predict(tfidf_data_baru)
    hasil.delete("1.0", "end")
    hasil.insert(
        "end", f"Hasil Prediksi:{hasil_prediksi[0]}")


def open_file():
    file_path = filedialog.askopenfilename(
        title="Select a File",
        filetypes=[("Word Files", "*.docx"), ("All Files", "*.*")]
    )

    if file_path:
        doc = Document(file_path)
```

```
        content = "\n".join([paragraph.text for paragraph in
doc.paragraphs])

        hasil_entry.delete("1.0", "end")
        hasil_entry.insert("1.0", content)


button = CTkButton(master=frame_3, text="Prediksi",
command=click_button)
button.pack(side='left', expand=True, pady=(30, 15), padx=10)


button2 = CTkButton(master=frame_3, text="Upload File",
                    command=open_file)
button2.pack(side="left", expand=True, pady=(30, 15), padx=10)

hasil = CTkTextbox(master=main_frame, height=20)

hasil.grid(row=3, column=1, padx=5, pady=5)

hasil_label = CTkLabel(master=main_frame, text='Hasil',
                       font=("Arial Bold", 20), justify="left")

hasil_label.grid(row=2, column=1, padx=5, pady=5)


app.mainloop()
```

Tampilan GUI:




CTk
Analisis Review
Ketik atau Upload File
sabun muka nya bagus banget, ga bikin muka kering dan seger rasa nya. aku suka banget, pasti bakal beli terus produk ini. buat kalian w ajib banget coba produk ini, dijamin muka langsung mulus.
Prediksi
Upload File
Hasil
Hasil Prediksi:positive
Cari
IND
21.54
22/12/2023


CTk
Select a File
D:\
Cari D:\
Atur
Folder baru
OneDrive
Desktop
Unduhan
D:\
Gambar
Musik
Video
Screenshots
Nama
Tanggal diubah
Jenis
Custom Office Templates
20/10/2022 20.34
Folder file
IBM
13/11/2023 09.09
Folder file
My Data Sources
19/11/2022 21.39
Folder file
Zoom
12/11/2022 10.06
Folder file
Nama file:
Word Files
Buka
Batal
Review
Upload File
Prediksi
Upload File
Hasil
Cari
IND
21.52
22/12/2023