LAPORAN FINAL PROJECT

“SISTEM IDENTIFIKASI SENTIMEN DATA TEKS MENGGUNAKAN METODE SUPPORT VECTOR MACHINE (SVM)”

PENGANTAR DAN PEMROSESAN DATA MULTIMEDIA

Disusun oleh Kelompok 5:

Saifulloh Rahman	(2108561028)
Nyoman Krisna Ari Sudarsana	(2108561072)
Ratri Desy Christirahma	(2108561003)
Yan Pieter Israel Rumere	(1808561082)

Dosen Pengampu:

Dr. Anak Agung Istri Ngurah Eka Karyawati, S.Si., M.Eng

PROGRAM STUDI INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS UDAYANA

2023

BAB I
PENDAHULUAN

1.1 Latar belakang

Dalam berkembangnya teknologi saat ini, data text menjadi salah satu kemajuan teknologi yang signifikan. Namun, memahami sentimen atau perasaan yang terkandung dalam data teks tersebut dapat menjadi tugas yang kompleks dan memakan waktu jika dilakukan secara manual. Oleh karena itu, pengembangan sistem yang dapat mengidentifikasi sentimen pada data teks menjadi sangat penting.

Salah satu metode yang telah terbukti efektif dalam mengidentifikasi sentimen pada data teks adalah metode Support Vector Machine (SVM). SVM adalah sebuah algoritma pembelajaran mesin yang digunakan untuk tugas klasifikasi. SVM bekerja dengan memisahkan dua kelas data dengan menggunakan hiperplane yang optimal. Dalam konteks identifikasi sentimen, SVM dapat mempelajari pola-pola penting yang terkait dengan sentimen positif dan negatif dalam data teks, sehingga dapat mengklasifikasikan teks-teks baru ke dalam salah satu kategori sentimen tersebut.

Dalam pengembangan sistem identifikasi sentimen data teks menggunakan metode SVM, diperlukan pemrosesan data teks yang cermat, termasuk tahapan seperti penghapusan tanda baca, tokenisasi, penghilangan kata-kata yang tidak penting, dan pemotongan kata ke bentuk dasarnya (stemming). Selain itu, pemilihan fitur yang tepat dan pemilihan parameter SVM yang optimal juga menjadi faktor penting untuk mendapatkan hasil yang baik.

Dengan adanya sistem identifikasi sentimen data teks menggunakan metode SVM, diharapkan pengolahan data teks dapat dilakukan secara efisien dan akurat. Sistem ini dapat membantu meningkatkan pemahaman terhadap sentimen yang terkandung dalam data teks, baik itu dalam konteks bisnis, penelitian, maupun pengambilan keputusan.

1.2 Problem Komputasi

Membangun sistem aplikasi untuk mengidentifikasi sentimen/emosi dari sebuah/beberapa ulasan. Terdapat dua sentimen yaitu sentimen positif (atau emosi *happy*) dan sentimen negatif (atau emosi *sad*). Dataset merupakan data Teks (ulasan. Semua data dibagi menjadi 2 bagian: 80% untuk dataset training dan 20% untuk dataset testing. Terdapat 3 tahapan utama proses komputasi untuk menghasilkan model klasifikasi yaitu: 1) Preprocessing; 2) Training; dan 3) Testing.

- Data text ulasan terbagi menjadi 2 label sentimen positif (1) dan negatif (0).

Tahapan preprocessing untuk data teks:

1) Feature Extraction untuk memperoleh nilai-nilai bobot TF atau TF IDF dari data ulasan. Tahapan feature extraction: tokenization, lower case converting, stop word removal, stemming, dan pembobotan (TF atau TFIDF). Untuk metode Multinomial Naive Bayes pembobotan kata menggunakan formula TF, sedangkan untuk metode yang lain menggunakan TF IDF. Total jumlah fitur dari setiap data text tergantung dari panjang/ jumlah vocabulary setelah preprocessing (jumlah term indeks) dari koleksi ulasan yang digunakan.
2) Feature Selection dengan menggunakan formula Chi-Square (lihat penjelasan di URL terkait), gunakan beberapa variasi jumlah fitur yang dipertahankan (10% atau 20% atau 30%, dsb).

Tahapan Training:

Training dilaksanakan untuk menghasilkan model klasifikasi yang terbaik.

Untuk metode SVM gunakan beberapa kombinasi hyper-parameter: nilai C= 0.1, atau 1, atau 10,atau 100, nilai gamma = 0.0001atau 0.001, atau 0.1, atau 1, dan fungsi kernel: rbf atau polynomial

Tahapan Testing:

Ukuran evaluasi yang digunakan: akurasi, precision, recall, an F1-Score.

Tahapan Deployment:

Model yang dihasilkan di deploy ke sistem aplikasi berbasis web dengan fitur utama adalah user menginput satu atau beberapa data dan outputnya adalah hasil sentimen atau identifikasi emosi.

1.3 Tujuan

Adapun tujuan dari laporan yang dibuat ini adalah

a. Mengevaluasi dan mengukur performa model Support Vector Machine dalam mengidentifikasi sentimen data teks.
b. Menyajikan interpretasi hasil dan temuan dari sistem identifikasi sentimen data teks menggunakan Support Vector Machine.

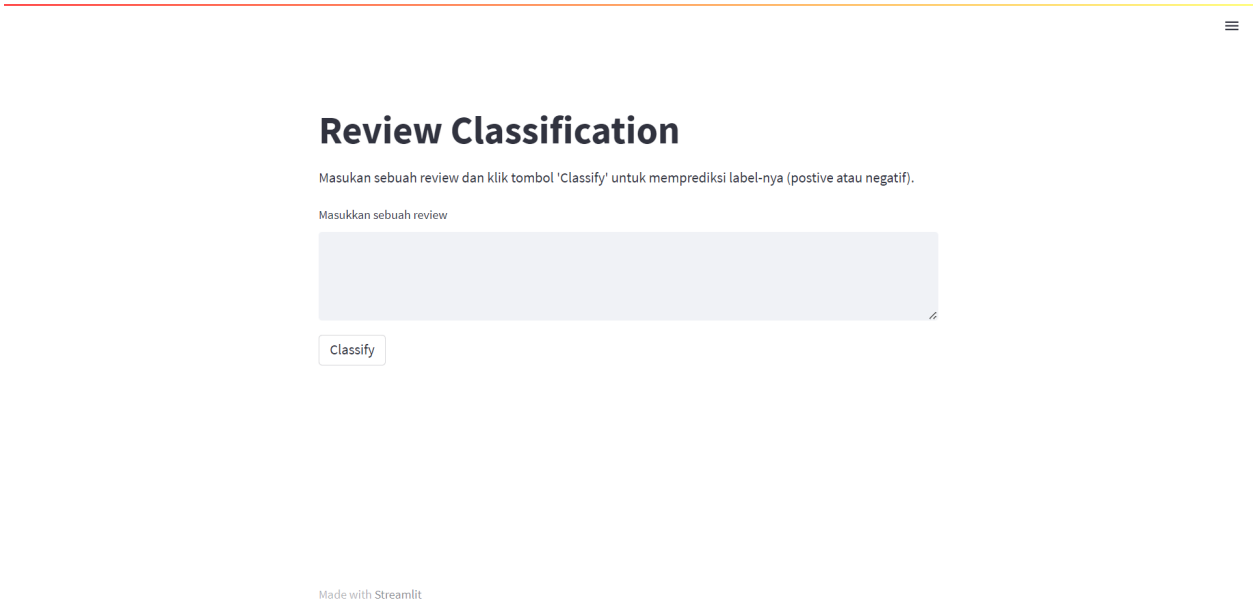
1.4 Manfaat

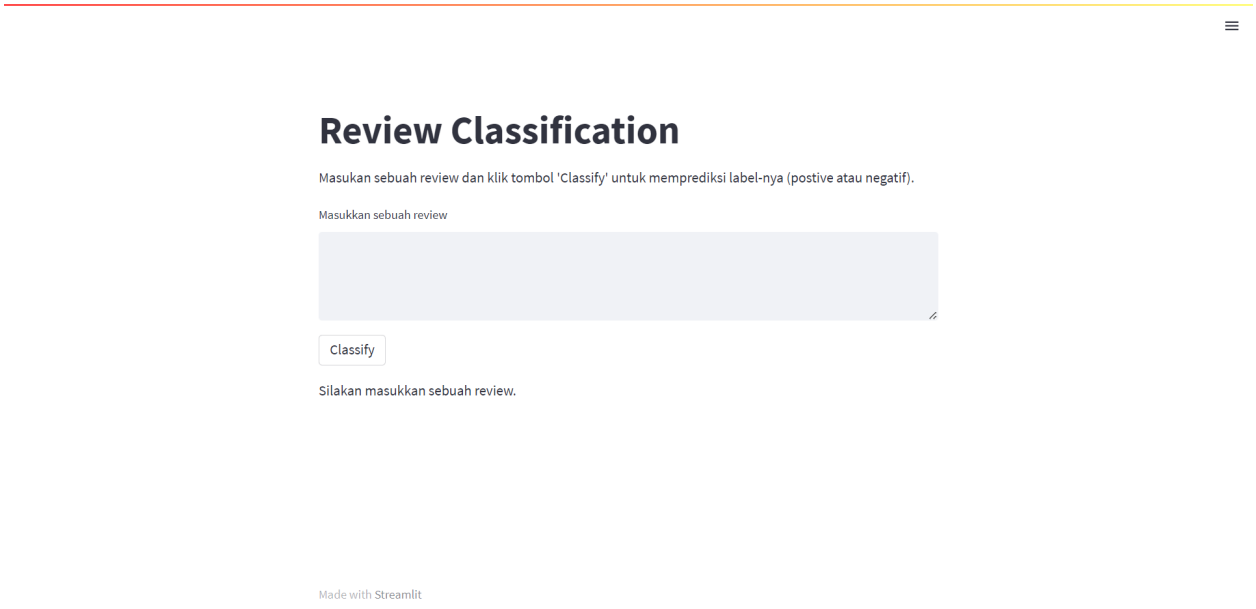
Manfaat dari laporan ini adalah dapat meningkatkan pemahaman tentang penggunaan Support Vector Machine dalam identifikasi sentimen data teks. Dengan mempelajari metode ini, pembaca akan mendapatkan wawasan tentang algoritma klasifikasi yang efektif untuk memproses data teks dan mengenali sentimen yang terkandung di dalamnya.

BAB II
ISI

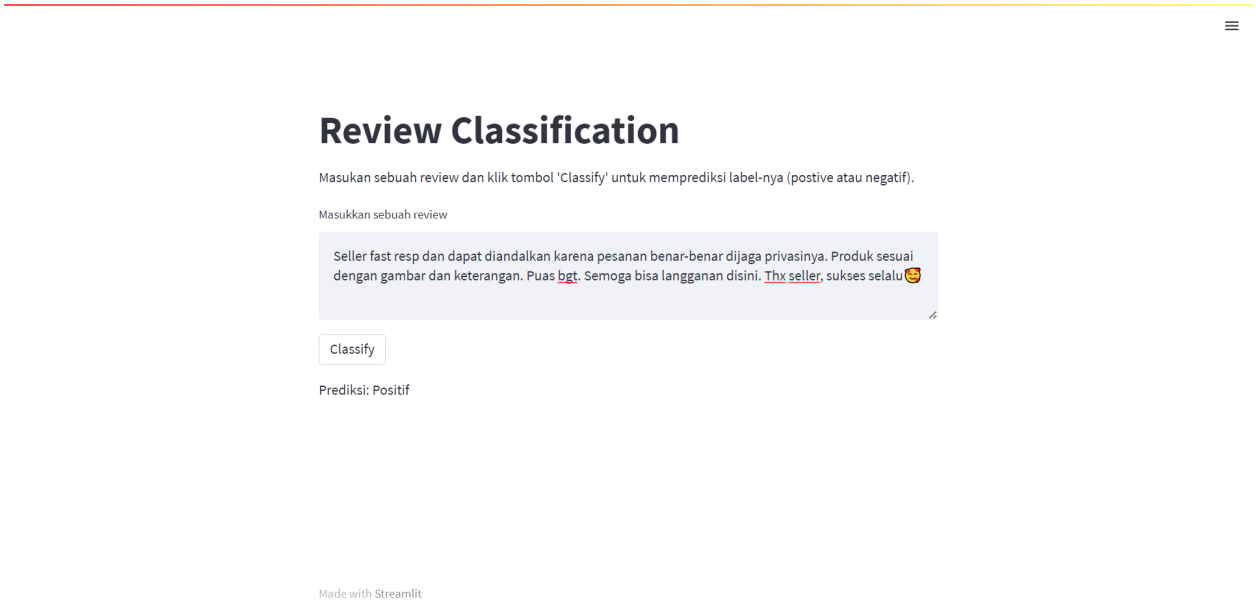
2.1 Manual Aplikasi

User bisa memasukkan/mengetikan/menuliskan kalimat review kemudian mengklik tombol Classify untuk melihat review tersebut termasuk positif atau negatif





Hasil prediksi klasifikasi atau sentimen review akan ditampilkan di bawah tombol Classify



2.2 Source Code Modul/ Fungsi Utama

Implementasi kode menggunakan bahasa pemrograman python. Terdapat dua file, yaitu model.ipynb dan app.py, yang kegunaan dan penjelasannya dijelaskan di bawah.

2.2.1 model.ipynb

File kode untuk membuat dan mencari model machine learning SVM terbaik

a. Mounting Google Drive

Mounting google drive agar bisa mengakses file yang ada di dalamnya dan juga bisa menyimpan ke dalam google drive

Data Google Drive

```
[34]  from google.colab import drive
      drive.mount('/content/drive')
      %cd /content/drive/MyDrive/Colab-Notebooks/final-project/
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
/content/drive/MyDrive/Colab-Notebooks/final-project
```

b. Library

Berikut beberapa library dan/atau modul yang digunakan untuk membuat model machine learning SVM

Library

```
[3]  from flask import Flask, render_template, url_for
     import numpy as np
     import pandas as pd
     import csv
     import matplotlib.pyplot as plt
     from sklearn import model_selection
     from sklearn.model_selection import train_test_split
     from sklearn.feature_extraction.text import TfidfVectorizer
     from sklearn import svm
     from sklearn.metrics import accuracy_score

     # Packages for visuals
     import matplotlib.pyplot as plt
     import seaborn as sns; sns.set(font_scale=1.2)

     import re
     !pip install Sastrawi
     from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
     from nltk.tokenize import word_tokenize
     from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
     from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
```

c. Import Data

Data berupa file csv disimpan ke dalam variabel “review” dan ditampilkan. Data berupa kumpulan ulasan yang berisi label 1=positif atau 0= negatif

Import Data

```
[4]  reviews = pd.read_csv("reviews.csv")

     reviews
```

reviews

	No	Reviews	Label
0	1	kemeja nya bagusss bgtttt😍😍😍aaaa mauuu nngisss...	1
1	2	Jahitannya sih rapi,cuman ada benang yang ikut...	0
2	3	Sesuai harga. Agak tipis tapi masih oke kok. W...	0
3	4	Wah gila sihhh sebagus itu, se worth it, se l...	1
4	5	Kain nya bagus halus \nTapi kok di bukak koto...	0
...	...	...	...
826	827	Terima kasih barang sudah sampai sesuai ukuran...	1
827	828	Mantapp realpicttt bangttt tapi pengemasan nya...	1
828	829	Suka bgt sama tasnya, ga kayak tas local. Kere...	1
829	830	kualitas produk sangat baik. produk original. ...	1
830	831	Barang udah sampai dg selamat, mantul banget d...	1

831 rows × 3 columns

d. Preprocessing

Preprocessing untuk menyiapkan data sebelum diolah, yang meliputi tahap tokenization, lower case converting, stop word removal, dan stemming. Lalu data hasil preprocessing ditampilkan sehingga terlihat perbedaannya dari sebelumnya.

Preprocessing

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

factory = StopWordRemoverFactory()
stopword = factory.create_stop_word_remover()

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

reviews['Reviews'] = reviews['Reviews'].apply(lambda x: x.lower())

reviews['Reviews'] = reviews['Reviews'].apply(lambda x: x.replace('\n', ''))

reviews['Reviews'] = reviews['Reviews'].apply(lambda x: re.compile('[^a-zA-Z]').sub(" ", x))

reviews['Reviews'] = reviews['Reviews'].apply(lambda x: x.split())

def removeAkhir(s):

  cadangan = s[-1]
  s = s.rstrip(s[-1])

  return s + cadangan

reviews['Reviews'] = reviews['Reviews'].apply(lambda x: [removeAkhir(i) for i in x])

reviews['Reviews'] = reviews['Reviews'].apply(lambda x: [i for i in x if len(i) > 2])

reviews['Reviews'] = reviews['Reviews'].apply(lambda x: " ".join(x))

reviews['Reviews'] = reviews['Reviews'].apply(lambda x: stopword.remove(x))

reviews['Reviews'] = reviews['Reviews'].apply(lambda x: stemmer.stem(x))
```

```
[24]  reviews
```

	No	Reviews	Label
0	1	kemeja nya bagus bgt mau nngis knpa dri dlu be...	1
1	2	jahit sih rapi cuman benang ikut jahit jadi jelek	0
2	3	sesuai harga tipis masih oke kok warna abu kal...	0
3	4	wah gila sih bagus worth lembut baju kirain ba...	1
4	5	kain nya bagus halus kok bukak kotor warna putih	0
...	...	...	...
826	827	terima kasih barang sampai sesuai ukur seesuai...	1
827	828	mantap realpict bangt emas nya cuman plastik a...	1
828	829	suka bgt sama tas kayak tas local keren parah ...	1
829	830	kualitas produk sangat baik produk original ha...	1
830	831	barang udah selamat mantul banget dah baju ses...	1

831 rows × 3 columns

```
[33]  # reviews.to_csv(r"/content/drive/MyDrive/Colab-Notebooks/final-project/preprocessed-data.csv")
      # reviews = pd.read_csv("preprocessed-data.csv")
```

e. Membagi Data Menjadi Train dan Test

Membagi data menjadi data test sebanyak 20% dan train menjadi 80%

```
Split the data into training and test sets

[6]  train_X, test_X, train_Y, test_Y = model_selection.train_test_split(reviews['Reviews'], reviews['Label'], test_size = 0.2, random_state = 0)

[7]  df_train80 = pd.DataFrame()
     df_train80['Reviews'] = train_X
     df_train80['Label'] = train_Y

     df_test20 = pd.DataFrame()
     df_test20['Reviews'] = test_X
     df_test20['Label'] = test_Y
```

f. Feature Extraction

Pembobotan TF-IDF

```
Feature Extraction

TF-IDF

[11]  from sklearn.feature_extraction.text import TfidfVectorizer

      tfidf_vect_8020 = TfidfVectorizer(max_features = 5000)
      tfidf_vect_8020.fit(reviews['Reviews'])
      train_X_tfidf_8020 = tfidf_vect_8020.transform(df_train80['Reviews'])
      test_X_tfidf_8020 = tfidf_vect_8020.transform(df_test20['Reviews'])
```

g. Feature Selection

Feature Selection dengan menggunakan formula Chi-Square dengan menggunakan beberapa variasi jumlah fitur yang dipertahankan (10% atau 20% atau 30%, dsb). Pada model ini Chi-Square yang digunakan adalah 10% atau 0.1

```
Feature Selection

Chi-Square = [10%, 20%, 30%]

     from sklearn.feature_selection import SelectKBest, chi2

     percent_kept = 0.1  # Ubah value ini sesuai keinginan
     num_features = int(percent_kept * train_X_tfidf_8020.shape[1])
     selector = SelectKBest(chi2, k=num_features)
     train_X_tfidf_8020_selected = selector.fit_transform(train_X_tfidf_8020, train_Y)
     test_X_tfidf_8020_selected = selector.transform(test_X_tfidf_8020)
```

h. Proses Pelatihan

Dilakukan proses pelatihan (training) dengan Chi Square = 0.1, parameter c = 10, kernel rbf, gamma = 0.1. Pada gambar di bawah ditampilkan beberapa kombinasi chi square dan parameter terbaik dari data yang ada.

i. Proses Pengujian

Setelah proses training dilakukan, selanjutnya proses testing terhadap model untuk melihat berapa akurasinya.

j. Accuracy, Precision, Recall, F-1 Score

Ditampilkan nilai accuracy, precision, recall, f-1 score dari model machine learning. Dari model dengan kombinasi chi-square = 0.1, parameter c = 10, kernel rbf, dan gamma = 0.1, didapatkan nilai accuracy, precision, recall, dan f-1 score di atas 90%.

k. Menyimpan Model dan Vectorizer yang sudah dilatih

Model yang sudah dibuat dan di-training kemudian disimpan dalam bentuk .pkl untuk digunakan dalam pembuatan web, setiap kali web digunakan tidak perlu melakukan tahap training. Ada dua file .pkl yang dihasilkan yaitu model.pkl dan vectorizer.pkl

Save the trained model and vectorizer

```
[61]  import joblib
      joblib.dump(model, "model.pkl")
      joblib.dump(tfidf_vect_8020, "vectorizer.pkl")

['vectorizer.pkl']
```

l. Mencari Kombinasi Parameter terbaik

Pada proses ini merupakan iterasi proses seleksi fitur, training, dan testing untuk mencari kombinasi chi square dan parameter (c, kernel, gamma) terbaik. Kemudian menghasilkan output kombinasi dan akurasinya. Adapun parameter yang digunakan adalah sebagai berikut: Chi-Square: 10%, 20%, 30%; C: 0.1, 1, 10, 100; kernel: rbf, poly; dan gamma: 0.0001, 0.001, 0.1, 1, scale, auto.

Mencari Parameter Terbaik

```
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.feature_selection import SelectKBest, chi2

def mulai(h, i, j, k):

  percent_kept = h  # Change this value to the desired percentage of features to keep
  num_features = int(percent_kept * train_X_tfidf_8020.shape[1])
  selector = SelectKBest(chi2, k=num_features)
  train_X_tfidf_8020_selected = selector.fit_transform(train_X_tfidf_8020, train_Y)
  test_X_tfidf_8020_selected = selector.transform(test_X_tfidf_8020)

  print(f"Chi Square = {h}\tParameter: c={i}, kernel={j}, gamma={k}")
  model = SVC(C=i, kernel=j, gamma=k )
  model.fit(train_X_tfidf_8020,train_Y)
```

```
  predictions_SVM_8020 = model.predict(test_X_tfidf_8020)
  test_prediction_8020 = pd.DataFrame()
  test_prediction_8020['Reviews'] = test_X
  test_prediction_8020['Label'] = predictions_SVM_8020
  SVM_accuracy_8020 = accuracy_score(predictions_SVM_8020, test_Y)*100
  SVM_accuracy_8020 = round(SVM_accuracy_8020,1)

  print(f"Akurasi: {SVM_accuracy_8020}")

  print ("Here is the classification report:")
  print (classification_report(test_Y, predictions_SVM_8020, zero_division=0.0), end="\n")
  print("============================================================")

chi = [0.1, 0.2, 0.3]
c = [0.1, 1, 10,  100]
kernel = ['rbf', 'poly']
gamma = [0.0001, 0.001, 0.1, 1, 'scale', 'auto' ]

for h in chi:
  for i in c:
    for j in kernel:
      for k in gamma:
        mulai(h, i, j, k)
```

Berikut adalah beberapa hasil evaluasi model dari beberapa kombinasi parameter

```
Chi Square = 0.1        Parameter: c=0.1, kernel=rbf, gamma=0.0001
Akurasi: 50.3
Here is the classification report:
              precision    recall  f1-score   support

           0       0.50      1.00      0.67        84
           1       0.00      0.00      0.00        83

    accuracy                           0.50       167
   macro avg       0.25      0.50      0.33       167
weighted avg       0.25      0.50      0.34       167

============================================================
```

```
==========================================================
Chi Square = 0.2        Parameter: c=1, kernel=rbf, gamma=scale
Akurasi: 92.8
Here is the classification report:
              precision    recall  f1-score   support

           0       0.88      0.99      0.93        84
           1       0.99      0.87      0.92        83

    accuracy                           0.93       167
   macro avg       0.93      0.93      0.93       167
weighted avg       0.93      0.93      0.93       167

==========================================================
```

All Combinations:

	Chi Square	C	Kernel	Gamma	Accuracy
0	0.2	0.1	rbf	0.0001	50.3
1	0.2	0.1	rbf	0.001	50.3
2	0.2	0.1	rbf	0.1	50.3
3	0.2	0.1	rbf	1	64.7
4	0.2	0.1	rbf	scale	57.5
5	0.2	0.1	rbf	auto	50.3
6	0.2	0.1	poly	0.0001	50.3
7	0.2	0.1	poly	0.001	50.3
8	0.2	0.1	poly	0.1	50.3
9	0.2	0.1	poly	1	50.3
10	0.2	0.1	poly	scale	51.5
11	0.2	0.1	poly	auto	50.3
12	0.2	1.0	rbf	0.0001	50.3
13	0.2	1.0	rbf	0.001	50.3
14	0.2	1.0	rbf	0.1	83.2
15	0.2	1.0	rbf	1	90.4
16	0.2	1.0	rbf	scale	92.8
17	0.2	1.0	rbf	auto	50.3
18	0.2	1.0	poly	0.0001	50.3
19	0.2	1.0	poly	0.001	50.3
20	0.2	1.0	poly	0.1	50.3
21	0.2	1.0	poly	1	51.5
22	0.2	1.0	poly	scale	62.3
23	0.2	1.0	poly	auto	50.3
24	0.2	10.0	rbf	0.0001	50.3
25	0.2	10.0	rbf	0.001	50.3
26	0.2	10.0	rbf	0.1	91.6
27	0.2	10.0	rbf	1	90.4
28	0.2	10.0	rbf	scale	91.6
29	0.2	10.0	rbf	auto	50.9
30	0.2	10.0	poly	0.0001	50.3
31	0.2	10.0	poly	0.001	50.3
32	0.2	10.0	poly	0.1	50.3
33	0.2	10.0	poly	1	64.7
34	0.2	10.0	poly	scale	74.9
35	0.2	10.0	poly	auto	50.3
36	0.2	100.0	rbf	0.0001	50.3
37	0.2	100.0	rbf	0.001	85.6
38	0.2	100.0	rbf	0.1	90.4
39	0.2	100.0	rbf	1	90.4
40	0.2	100.0	rbf	scale	90.4
41	0.2	100.0	rbf	auto	91.0
42	0.2	100.0	poly	0.0001	50.3
43	0.2	100.0	poly	0.001	50.3
44	0.2	100.0	poly	0.1	50.3
45	0.2	100.0	poly	1	75.4
46	0.2	100.0	poly	scale	79.6
47	0.2	100.0	poly	auto	50.3

```
Worst Combination:
    Chi Square      C Kernel Gamma  Accuracy
47         0.2  100.0   poly  auto      50.3

Moderate Combination:
Chi Square     0.2
C             10.0
Kernel        poly
Gamma          0.1
Accuracy      50.3
Name: 32, dtype: object

Best Combination:
    Chi Square    C Kernel  Gamma  Accuracy
16         0.2  1.0    rbf  scale      92.8
```

```
Accuracy Counts:
    Accuracy  Count
5       92.8      1
2       91.6      2
6       91.0      1
1       90.4      5
7       85.6      1
8       83.2      1
9       79.6      1
10      75.4      1
11      74.9      1
3       64.7      2
12      62.3      1
13      57.5      1
4       51.5      2
14      50.9      1
0       50.3     27
```

Berikut tabel akurasi dari beberapa kombinasi parameter:

1. Akurasi model SVM dengan **Chi-Square 10%**:

	Salah satu Kombinasi (C, kernel, gamma)	Akurasi	Jumlah kombinasi
Terjelek	C= 0.1, kernel= rbf, gamma= 0.0001 C= 0.1,kernel= poly, gamma= 0.0001	50.3%	27
Moderate	C= 0.1, kernel= rbf, gamma= 0.0001 C= 0.1,kernel= poly, gamma= 0.0001	50.3%	27
Terbaik	C=1.0, kernel= rbf, gamma= scale	90.4%	3

2. Akurasi model SVM dengan **Chi-Square 20%**:

	Salah satu Kombinasi (C, kernel, gamma)	Akurasi	Jumlah kombinasi
Terjelek	C= 0.1, kernel= rbf, gamma= 0.0001 C= 0.1,kernel= poly, gamma= 0.0001	50.3%	27
Moderate	C= 0.1, kernel= rbf, gamma= 0.0001 C= 0.1,kernel= poly, gamma= 0.0001	50.3%	27

Terbaik	C=1.0, kernel= rbf, gamma= scale	92.8%	1

3. Akurasi model SVM dengan **Chi-Square 30%**:

	Salah satu Kombinasi (C, kernel, gamma)	Akurasi	Jumlah kombinasi
Terjelek	C= 0.1, kernel= rbf, gamma= 0.0001 C= 0.1,kernel= poly, gamma= 0.0001	50.3%	28
Moderate	C= 0.1, kernel= rbf, gamma= 0.0001 C= 0.1,kernel= poly, gamma= 0.0001	50.3%	28
Terbaik	C=10, kernel= rbf, gamma= 1	91.6%	1

Berdasarkan tiga tebel di atas, dapat dilihat bahwa kombinasi *feature selection* dan parameter yang menghasilkan akurasi terbaik, yaitu 92,8%, adalah Chi-Square 20%, C = 1.0, kernel = rbf, dan gamma = scale.

2.2.2 app.py

File app.py adalah file web dari model yang dibuat. Library yang digunakan adalah streamlit

```
1  import streamlit as st
2  import joblib
3  from sklearn.feature_extraction.text import TfidfVectorizer
4  import nltk
5  from nltk.corpus import stopwords
6  from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
7  nltk.download('stopwords')
8
```

Ada tahap preprocessing untuk kalimat yang diinput user

```python
# Preprocessing
def preprocess_text(text):
    # Tokenisasi
    tokens = nltk.word_tokenize(text)

    # Konversi ke huruf kecil
    tokens = [token.lower() for token in tokens]

    # Penghapusan kata-kata stopword
    stop_words = set(stopwords.words('indonesian'))
    tokens = [token for token in tokens if token not in stop_words]

    # Stemming
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    tokens = [stemmer.stem(token) for token in tokens]

    return tokens

```

Model dan vectorizer yang telah dibuat pada file model.py diload ke dalam file app.py untuk kegunaan klasifikasi

```python
# Load the trained model and vectorizer
model = joblib.load("model.pkl")
tfidf_vect_8020 = joblib.load("vectorizer.pkl")

def classify_review(review):
    # Preprocess the input text
    review = " ".join(preprocess_text(review))
    review = tfidf_vect_8020.transform([review])
    # Make predictions
    prediction = model.predict(review)
    return prediction[0]

```

Pada bagian main() berisi kode tampilan dari web. User bisa menginputkan kalimat review dan melihat prediksi emosi atau label dari kalimat review tersebut

```python
# Create a Streamlit web app
def main():
    st.title("Review Classification")
    st.write("Masukan sebuah review dan klik tombol 'Classify' untuk memprediksi label-nya (postive atau negatif).")


    # Input text box
    review = st.text_area("Masukkan sebuah review", "")

    # Classify button
    if st.button("Classify"):
        if review:
            # Perform classification
            prediction = "Positif" if classify_review(review) == 1 else "Negatif"
            st.write("Prediksi:", prediction)
        else:
            st.write("Silakan masukkan sebuah review.")

if __name__ == "__main__":
    main()

```

BAB 3
PENUTUP

3.1 Kesimpulan

Metode *Support Vector Machine* (SVM) dapat digunakan untuk melakukan analisis sentimen data teks berdasarkan percobaan yang dilakukan. Model SVM yang dibuat menghasilkan akurasi yang baik pada salah satu kombinasi parameter dan seleksi fitur, yaitu kombinasi *feature selection* dan parameter yang menghasilkan akurasi terbaik, yaitu 92,8%, dengan Chi-Square 20%, C = 1.0, kernel = rbf, dan gamma = scale. Model terbaik menghasilkan rata rata nilai precision 93%, recall 93%, F1-Score 93%, dan akurasi 93%. Tahapan percobaan dimulai dari *preprocessing* data (*case folding*, tokenisasi, *stopword removal*, *stemming*), seleksi fitur, *training*, *testing*, evaluasi, dan *deploy* model ke web. Model yang dihasilkan dapat di-*deploy* ke sistem aplikasi berbasis web untuk identifikasi sentimen data teks, yaitu sentimen positif atau negatif.

DAFTAR PUSTAKA

[1] Andriyani, Nanang Arif. "OPTIMASI ALGORITMA K-NEAREST NEIGHBOR DALAM MENDETEKSI KOMENTAR SPAM BERBAHASA INDONESIA DI INSTAGRAM MENGGUNAKAN CONVERT NEGATIVE DAN TF-IDF (TERM FREQUENCY - INVERSE DOCUMENT FREQUENCY) PADA TAHAP PREPROCESSING." (2019).

[2] Rian Tineges. 2021. "Tahapan Text Preprocessing dalam Teknik Pengolahan Data". URL : https://dqlab.id/tahapan-text-preprocessing-dalam-teknik-pengolahan-data.

[3] Muhabatin, H., Prabowo, C., Ali, I., Rohmat, C. L., & Amalia, D. R. (2021). Klasifikasi Berita Hoax Menggunakan Algoritma Naïve Bayes Berbasis PSO. Informatics for Educators and Professionals, 5(2), 156-165. E-ISSN: 2548-3412.

[4] Hakim, B. (2021). ANALISA SENTIMEN DATA TEXT PREPROCESSING PADA DATA MINING DENGAN MENGGUNAKAN MACHINE LEARNING [Sentiment Analysis Data Text Preprocessing in Data Mining Using Machine Learning]. Journal of Base Research, 4(2). Versi Online: http://journal.ubm.ac.id/index.php/jbase. DOI: http://dx.doi.org/10.30813/jbase.v4i2.3000.

[5] Rofiqi, M. A., Fauzan, A. C., Agustin, A. P., Saputra, A. A., & Fahma, H. D. (2019). Implementasi Term-Frequency Inverse Document Frequency (TFIDF) Untuk Mencari Relevansi Dokumen Berdasarkan Query [Implementation of Term-Frequency Inverse Document Frequency (TFIDF) for Document Relevance Based on Query]. ILKOMNIKA: Journal of Computer Science and Applied Informatics, 1(2), 58-64. E-ISSN: 2715-2731. URL: http://journal.unublitar.ac.id/ilkomnika. DOI: https://doi.org/10.28926/ilkomnika.v1i2.18.

[6] Suaibi, Rifkillah. "Aplikasi Teks Mining untuk Automasi Pencarian Kalimat Inti dalam Dokumen Tunggal Berbahasa Indonesia dengan Metode Term Frecuency-Invers Document Frecuenfy (TF-IDF)." (2018).

[7] Widyasanti, Ni Komang et al. "Seleksi Fitur Bobot Kata dengan Metode TFIDF untuk Ringkasan Bahasa Indonesia." Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi Informasi) (2018): n.

[8] "Deteksi Suara Gitar Dengan Bahan Jenis Senar Berbeda Melalui Ciri Akustik Dengan Mel-Frequency Cepstral Coefficients (MFCC) Dan Support Vector Machine (SVM) Guitar String Detection Through Acoustic Characteristics Using Mel-Frequency Cepstral Coefficients (MFCC) And Support Vector Machine (SVM) Methods."

[9] Binus. 2022. "Support Vector Machine Algorithm". URL : https://sis.binus.ac.id/2022/02/14/support-vector-machine-algorithm/.