

**OPTIMASI PARAMETER SVM MENGGUNAKAN ALGORITMA
GENETIKA UNTUK ANALISIS SENTIMEN TERHADAP BAKAL
CALON PRESIDEN 2024 PADA TWITTER**



Disusun Oleh :

Valentino Prasetya Putra (22031554002)

Fadhilah Nuria Shinta (22031554003)

Daffa Fazly Rashidan (22031554006)

**UNIVERSITAS NEGERI SURABAYA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
PROGRAM STUDI SAINS DATA 2023**

DAFTAR ISI

Isi

DAFTAR ISI.....	i
BAB I PENDAHULUAN.....	1
1.1 LATAR BELAKANG.....	1
1.2 TUJUAN DAN MANFAAT	1
BAB II METODE	2
2.1 Dataset	2
2.2 Pre-processing	3
2.3 Ekstraksi Fitur	3
BAB III HASIL	3
BAB IV USER MANUAL GUIDE	6

BAB I PENDAHULUAN

1.1 LATAR BELAKANG

Pemilihan umum, terutama dalam konteks pemilihan presiden, merupakan momen penting dalam kehidupan demokrasi sebuah negara. Dalam era digital ini, media sosial, khususnya Twitter, menjadi salah satu wadah utama bagi masyarakat untuk menyampaikan pendapat, ekspresi, dan sentimen terkait bakal calon presiden. Analisis sentimen pada data Twitter menjadi semakin relevan untuk memahami pandangan publik terhadap calon presiden pada Pemilihan Presiden 2024.

Pada penelitian ini, fokus diberikan pada penggunaan Support Vector Machine (SVM) sebagai model klasifikasi, yang diperkuat dengan algoritma genetika untuk mengoptimalkan parameter SVM. SVM dipilih karena kemampuannya dalam menangani masalah klasifikasi pada data berdimensi tinggi, seperti data teks dari Twitter. Algoritma genetika digunakan untuk mencari kombinasi parameter SVM yang optimal, meningkatkan kinerja model dalam mengklasifikasikan sentimen dari tweet.

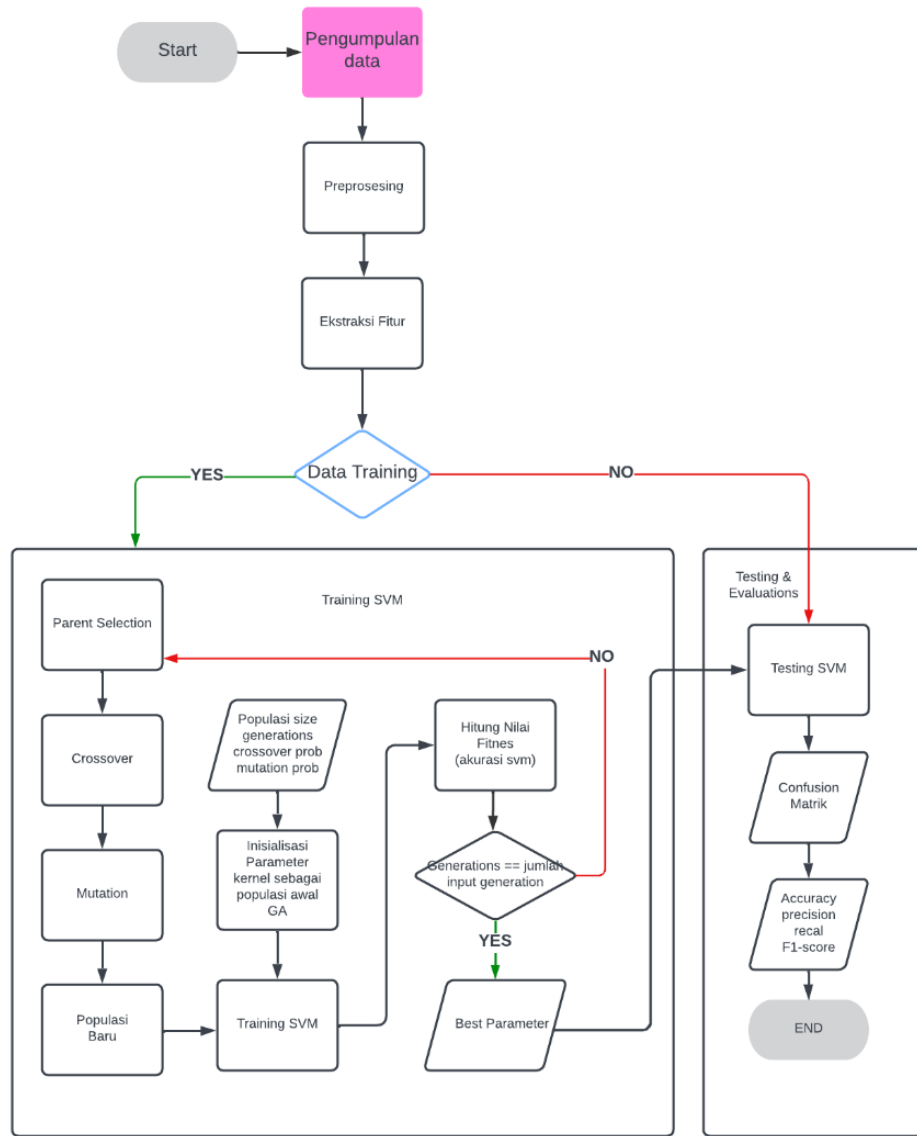
Optimasi parameter SVM menjadi langkah kritis dalam menjamin keakuratan dan kehandalan model analisis sentimen. Kombinasi parameter yang tepat akan memungkinkan SVM untuk mengatasi permasalahan kompleksitas dan variasi dalam data sentimen yang bervariasi di Twitter. Dengan melakukan analisis sentimen terhadap tweet yang berkaitan dengan bakal calon presiden pada Pemilihan Presiden 2024, diharapkan dapat memberikan wawasan yang mendalam tentang dukungan atau ketidaksetujuan masyarakat terhadap para calon presiden.

1.2 TUJUAN DAN MANFAAT

Tujuan dan manfaat dari dibuatnya aplikasi analisis sentimen terhadap bakal calon presiden 2024 adalah

1. Memberikan akses yang lebih mudah dan ramah pengguna kepada berbagai kalangan masyarakat untuk mengakses dan memahami analisis sentimen terkait bakal calon presiden melalui antarmuka grafis pengguna (GUI).
2. Menyajikan hasil analisis sentimen dengan cara yang jelas dan mudah dimengerti melalui GUI, sehingga pengguna yang tidak memiliki latar belakang teknis dapat dengan mudah menginterpretasikan data sentimen.
3. Memanfaatkan GUI untuk menyederhanakan proses analisis sentimen, sehingga pengguna dapat dengan mudah menjalankan analisis tanpa memerlukan pengetahuan teknis yang mendalam.

BAB II METODE



2.1 Dataset

Dalam penelitian ini, media platform X dijadikan tempat pengumpulan data untuk dijadikan data uji pada model. Proses pengumpulan data dilakukan menggunakan teknik *scrapping* yang mengambil tweet yang terkait dengan calon presiden 2024. Seleksi atribut dalam proses scrapping dilakukan dengan memanfaatkan kata kunci dan didapatkan hasil scrapping sebanyak 1288 baris. Setelah tahap *scrapping* diselesaikan, data disimpan dalam format CSV untuk keperluan analisis lebih lanjut. Dataset kedua yakni diambil dari data pilkada DKI Jakarta 2017 yang terdiri dari 900 baris.

2.2 Pre-processing

Langkah-langkah yang dilakukan seperti mengonversi seluruh huruf menjadi huruf kecil, menerjemahkan emoji menjadi teks, menghapus hal yang tidak penting pada dataset, mengubah kata menjadi bentuk dasarnya (*Stemming*), menghapus kata yang sering muncul tapi dianggap tidak memiliki makna, dan menghapus simbol-simbol yang tidak bermakna, serta menyimpan hasil preprocessing dalam format CSV.

2.3 Ekstraksi Fitur

Rekayasa fitur atau *Feature Engineering* merupakan tahap mengubah data teks menjadi representasi angka. Dalam penelitian ini, digunakan metode TF-IDF sebagai teknik ekstraksi fitur. Metode TF-IDF, atau *Term Frequency-Inverse Document Frequency*, dalam ekstraksi fitur teks menawarkan beberapa kelebihan. Pertama, bobot yang diberikan oleh TF-IDF lebih tinggi pada kata-kata yang sering muncul dalam satu dokumen tetapi jarang muncul dalam dokumen-dokumen lain, hal ini akan memungkinkan identifikasi kata-kata khas untuk setiap dokumen. Kedua, dengan memfilter kata-kata umum yang sering muncul di banyak dokumen, TF-IDF membantu mengurangi dampak kata-kata yang tidak informatif, dengan memberikan bobot IDF rendah pada kata-kata tersebut. Ketiga, representasi vektor yang dihasilkan oleh TF-IDF bersifat jarang (*sparse*), fokus pada kata-kata tertentu dengan bobot signifikan, yang mempermudah dalam pemrosesan dan penyimpanan data. Terakhir, fleksibilitas TF-IDF memungkinkan penyesuaian dengan memodifikasi parameter seperti bobot TF atau IDF, memberikan kemampuan untuk menyesuaikan model dengan karakteristik khusus dari data yang dihadapi. Sehingga, penggunaan TF-IDF dalam ekstraksi fitur akan memberikan pendekatan yang efektif dalam mewakili teks secara numerik.

BAB III HASIL

1. Hasil preprocessing

1 to 25 of 900 entries Filter ?			
index	Text Tweet	preprocessing	preprobaru
0	Banyak akun kloning seolah2 pendukung #agussilvy mulai menyerang paslon #aniesandi dengan opini dan argumen pmbenaran. jangan terkecoh	akun kloning dukung agussilvy serang pasangan calon aniesandi opini argumen pmbenaran kecoh	akun kloning dukung agussilvy serang pasang calon aniesandi opini argumen pmbenaran kecoh
1	#agussilvy bicara apa kasihan yaa... lap itu air matanya wkwkwkwk	agussilvy bicara kasihan ya lap air mata wkwk	agussilvy bicara kasihan ya lap air mata wkwk
2	Kalau aku sih gak nunggu hasil akhir QC tp lagi nunggu motif cutan pak @SBYudhoyono kayak apa... pasca #AgusSily Nyungsep..	sih tidak menunggu hasil qc tetapi menunggu motif cut susilo bambang yudhoyono kayak pasca agussilvy nyungsep	sih tunggu hasil qc tunggu motif cut susilo bambang yudhoyono kayak pasca agussilvy nyungsep
3	Kasian oh kasian dengan peluru 1milyar untuk tiap RW #agussilvy tidak mempan menangi pilkada #QuickCount #PilkadaSerentak2017	kasih oh kasih peluru milyar rukun warga agussilvy mempan memenangkan pilihan kepala daerah quickcount pilkadaserentak	kasih oh kasih peluru milyar rukun warga agussilvy mempan menang pilih kepala daerah quickcount pilkadaserentak
4	Maaf ya pendukung #AgusSily. hayo dukung #AniesSandi diputar 2 @ronavioleta @NetizenTofa	maaf ya dukung agussilvy hayo dukung aniesandi putar ronavioleta netizentofa	maaf ya dukung agussilvy hayo dukung aniesandi putar ronavioleta netizentofa
5	aneh deh lebay masa ini di sangkut pautkan sama kandidat Calgub, anda lebay seperti yg anda dukung #agussilvy	aneh deh lebay sangkut paut kandidat calgub lebay sepe i yang dukung agussilvy	aneh deh lebay sangkut paut kandidat calgub lebay sepe i dukung agussilvy
6	Kepada Allah SWT kami ucapkan rasa syukur dan kepada pak @SBYudhoyono @presidenSBY #AHY kami sampaikan terimakasih. Makna 17% suara anda	allah subhanahu wa taala ucap syukur susilo bambang yudhoyono presidensby agus harimurti yudhoyono terima kasih makna suara	allah subhanahu wa taala syukur susilo bambang yudhoyono presidensby agus hari i yudhoyono terima kasih makna suara
7	Terima Kasih teruntuk pendukung #AHY #SYLVI kalianlah pembeda kali ini	terima kasih untuk dukung agus harimurti yudhoyono sylvi kalian beda kali	terima kasih dukung agus hari i yudhoyono sylvi beda kali
8	Trima Kasih atas Keistiqomahan Relawan #AHY @AgusYudhoyono lah yg tih mengantarkan #AniesSandiK11 #JakartaGubernurBaru dg cukup signifikan	terima kasih keistiqomahan rawan agus harimurti yudhoyono agusyudhoyono yang telah antar aniesandidki jika agubernurburu dengan signifikan	terima kasih keistiqomahan rawan agus hari i yudhoyono agusyudhoyono aniesandidki jika agubernurburu signifikan
9	Mengenang pidato kekalahan #ahy	kenang pidato kalah agus harimurti yudhoyono	kenang pidato kalah agus hari i yudhoyono
10	damnnn,!! politik cantik #SBY ngorbanin #AHY	damnnn politik cantik susilo bambang yudhoyono mengorbankan agus harimurti yudhoyono	damnnn politik cantik susilo bambang yudhoyono korban agus hari i yudhoyono

2. Hasil Akurasi

```

Generation 1 - Best Accuracy: 0.8444, Best Parameters: {'C': 5.252192049238519, 'gamma': 0.2799137219241583, 'kernel': 'rbf'}
Generation 2 - Best Accuracy: 0.8333, Best Parameters: {'C': 1.5970103001658664, 'gamma': 0.164675341309767, 'kernel': 'linear'}
Generation 3 - Best Accuracy: 0.8444, Best Parameters: {'C': 34.86305326839542, 'gamma': 0.9738182681403921, 'kernel': 'rbf'}
Generation 4 - Best Accuracy: 0.8667, Best Parameters: {'C': 3.8068647391343458, 'gamma': 0.5083072393477999, 'kernel': 'rbf'}
Generation 5 - Best Accuracy: 0.8556, Best Parameters: {'C': 1.0903346316878493, 'gamma': 0.938615999311941, 'kernel': 'linear'}
Generation 6 - Best Accuracy: 0.8444, Best Parameters: {'C': 67.34913310165878, 'gamma': 0.9738182681403921, 'kernel': 'rbf'}
Generation 7 - Best Accuracy: 0.8778, Best Parameters: {'C': 1.7200902055543636, 'gamma': 0.32593613994532966, 'kernel': 'rbf'}
Generation 8 - Best Accuracy: 0.8444, Best Parameters: {'C': 11.36045136197521, 'gamma': 0.9244409870270014, 'kernel': 'rbf'}
Generation 9 - Best Accuracy: 0.8444, Best Parameters: {'C': 67.34913310165878, 'gamma': 0.8934118838182689, 'kernel': 'rbf'}
Generation 10 - Best Accuracy: 0.8444, Best Parameters: {'C': 83.62669868589104, 'gamma': 0.8934118838182689, 'kernel': 'rbf'}
Generation 11 - Best Accuracy: 0.8444, Best Parameters: {'C': 27.7919926375733, 'gamma': 0.8913570442756177, 'kernel': 'rbf'}
Generation 12 - Best Accuracy: 0.8333, Best Parameters: {'C': 61.73307913312223, 'gamma': 0.6698273775643114, 'kernel': 'rbf'}
Generation 13 - Best Accuracy: 0.8333, Best Parameters: {'C': 4.933914451289067, 'gamma': 0.20692966594282988, 'kernel': 'rbf'}
Generation 14 - Best Accuracy: 0.8444, Best Parameters: {'C': 49.22648286269753, 'gamma': 0.9885968921190732, 'kernel': 'rbf'}
Generation 15 - Best Accuracy: 0.8111, Best Parameters: {'C': 61.73307913312223, 'gamma': 0.3217439216135575, 'kernel': 'rbf'}
Generation 16 - Best Accuracy: 0.8444, Best Parameters: {'C': 3.429960086835625, 'gamma': 0.3179960735085249, 'kernel': 'rbf'}
Generation 17 - Best Accuracy: 0.8556, Best Parameters: {'C': 3.429960086835625, 'gamma': 0.6774765502769096, 'kernel': 'rbf'}
Generation 18 - Best Accuracy: 0.8556, Best Parameters: {'C': 81.05048461447133, 'gamma': 0.006929036042977661, 'kernel': 'rbf'}
Generation 19 - Best Accuracy: 0.8556, Best Parameters: {'C': 3.429960086835625, 'gamma': 0.6657862546436784, 'kernel': 'rbf'}
Generation 20 - Best Accuracy: 0.8556, Best Parameters: {'C': 94.97232353451949, 'gamma': 0.006929036042977661, 'kernel': 'rbf'}

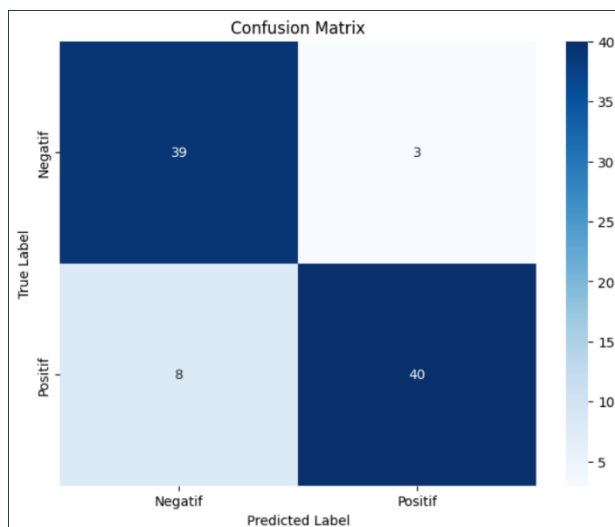
```

	precision	recall	f1-score	support
negative	0.83	0.93	0.88	42
positive	0.93	0.83	0.88	48
accuracy			0.88	90
macro avg	0.88	0.88	0.88	90
weighted avg	0.88	0.88	0.88	90

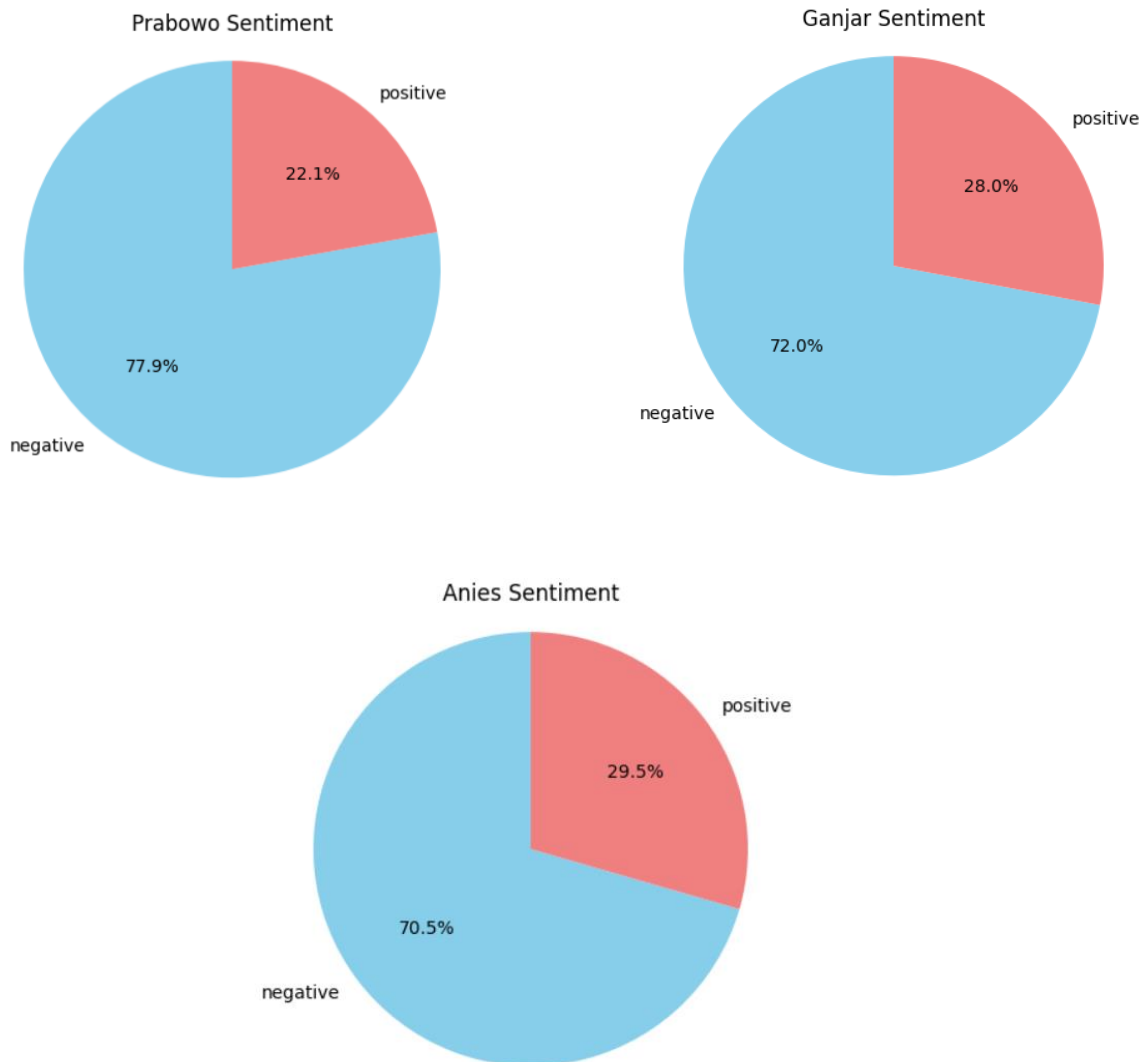
3. Tabel Perbandingan dan Confusion Matrix

Test size = 0.1 dan random state = 123

Population Size	Generation	Best Accuracy	Best Parameter 'C'	Gamma	Kernel
10 (10)	10	0.8444	34.23078597599125	0.8708579057265304	'rbf'
20 (3)	10	0.8667	2.8703359235402788	0.16080613102423055	'rbf'
30 (6)	20	0.8778	1.8682881378593799	0.2730602756870769	'rbf'
50 (30)	50	0.8778	1.4946746626325293	0.40992255660509846	'rbf'



4. Hasil prediksi yang sudah kami latih dari data bakal calon presiden 2024

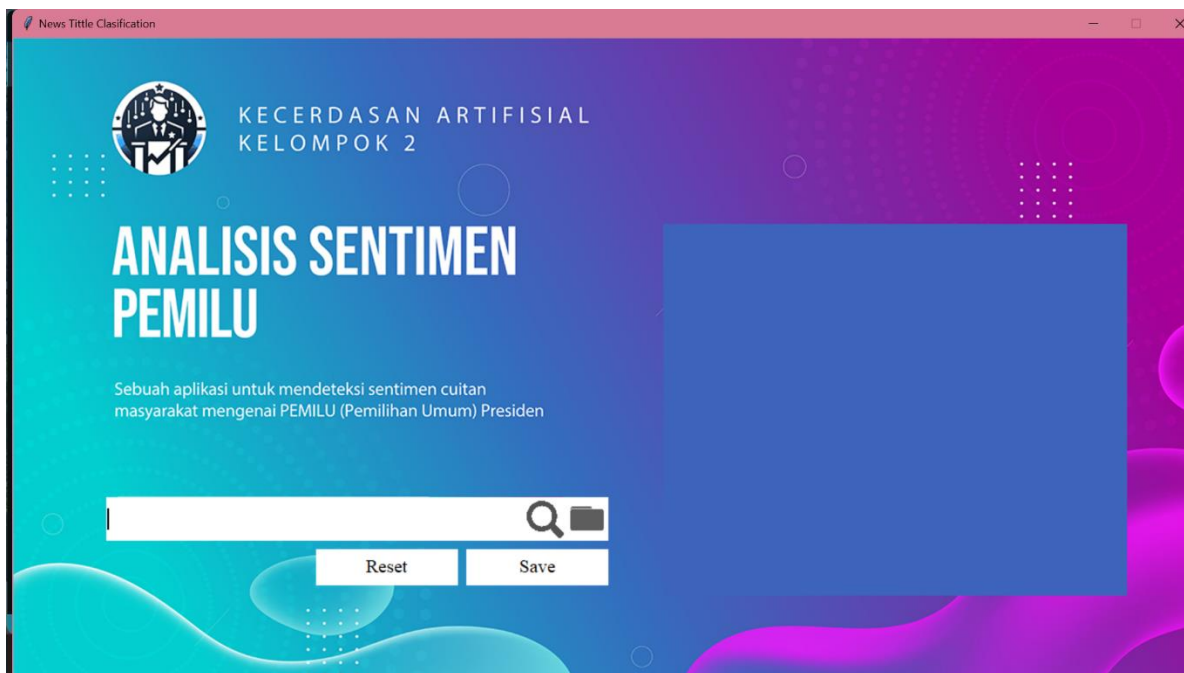


5. Representasi parameter C, kernel dan gamma

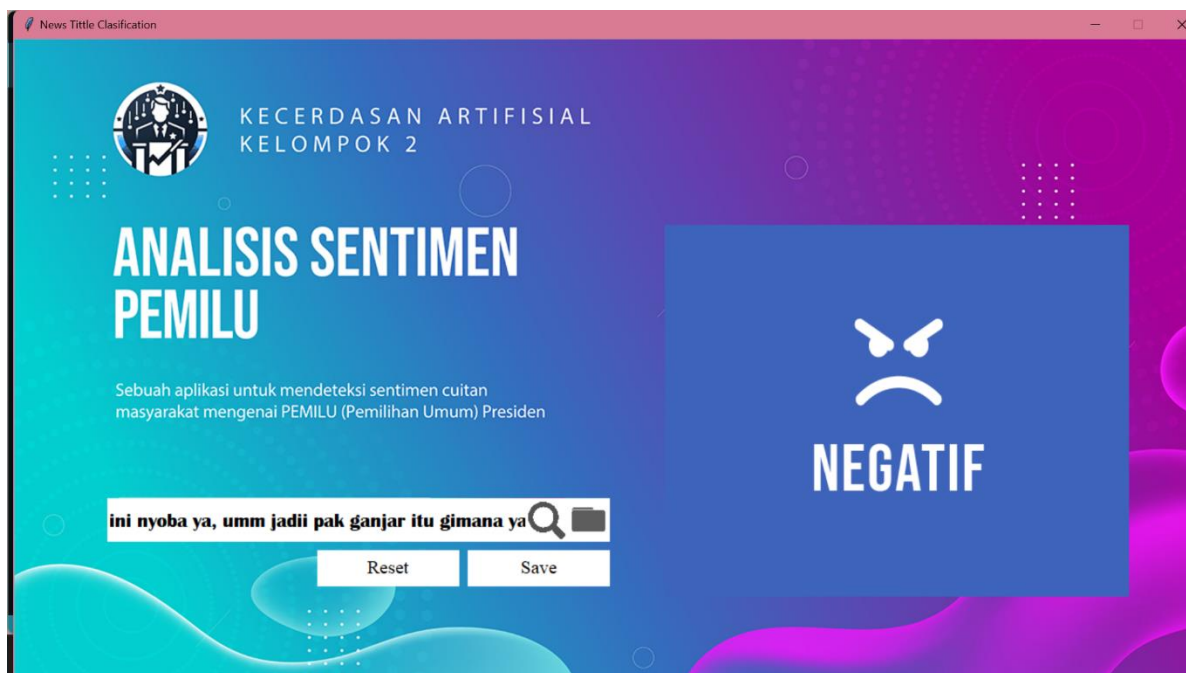
Parameter C dalam model SVM (*Support Vector Machine*) mengontrol trade-off antara kesalahan klasifikasi pada data pelatihan dan kompleksitas model. Semakin besar nilai C, model akan cenderung berusaha untuk meminimalkan kesalahan pada data pelatihan, namun dapat menghasilkan batas keputusan yang kompleks. Sebaliknya, nilai `gamma` mengontrol bentuk dari fungsi kernel yang digunakan (linear, rbf, poly), dan semakin besar nilainya dapat mengakibatkan penyesuaian yang lebih rumit.

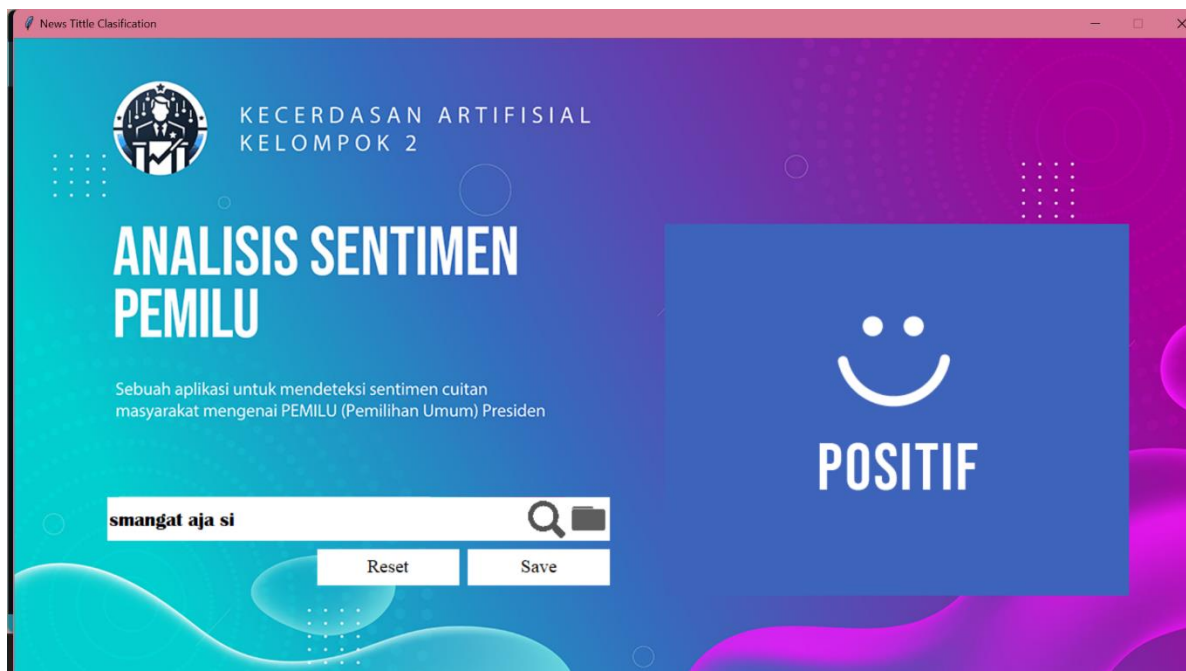
Dalam kode ini, kita menggunakan algoritma genetika untuk mencari kombinasi terbaik dari parameter `C`, `gamma`, dan `kernel` untuk meningkatkan akurasi model SVM pada data uji. Jadi, tujuannya adalah menemukan setting parameter yang optimal untuk SVM agar dapat menghasilkan hasil yang lebih baik pada data yang diberikan.

BAB IV USER MANUAL GUIDE

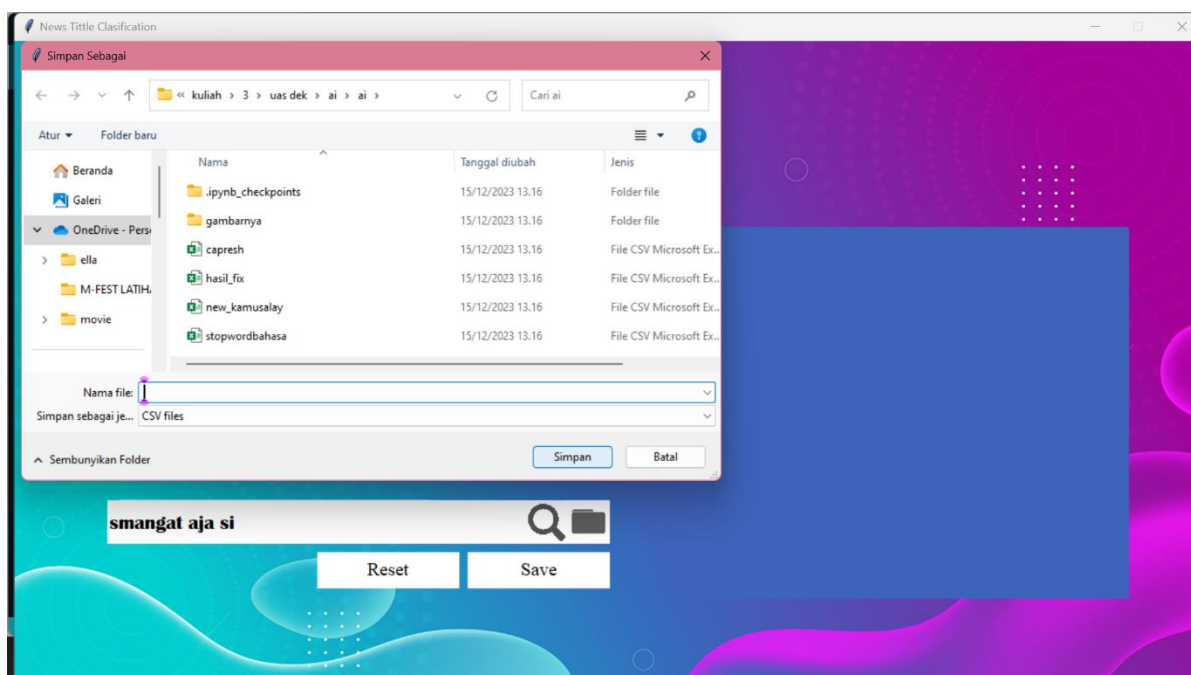


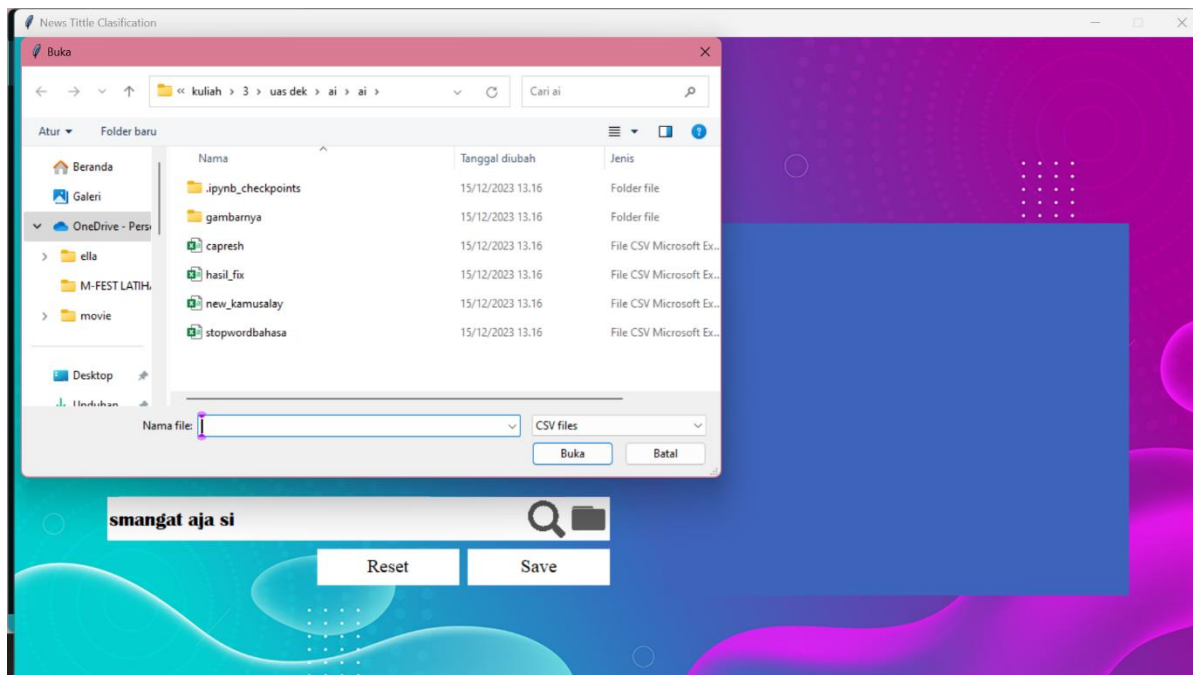
Pada halaman awal ini terdapat tampilan di mana di sebelah kiri, *user* dapat menginput teks yang akan dianalisis sentimen. Lalu terdapat juga button reset untuk mereset ulang hasil prediksi. Tampilan hasil analisis sentimen akan muncul seperti gambar di bawah ini





Adapun button ‘Save’ untuk menyimpan hasil dalam bentuk csv, dan button bergambar folder untuk membuka file csv yang berisi banyak teks untuk di analisis sentimen. Hasil analisis sentimen itu sendiri nantinya dikelompokkan menjadi 2 yakni kategori positif/negatif





Pembagian Tugas (semua mencoba code SVM+GA)

1. Valentino Prasetyo Putra 22031554002
Code, GUI
2. Fadhilah Nuria Shinta 22031554003
Code Scrapping, ppt, laporan akhir
3. Daffa Fazly Rashidan 22031554007
Code Pre processing, GUI

Listing Code SVM+GA

```

import numpy as np
import pandas as pd
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score

def initialize_population(pop_size):
    population = []
    for _ in range(pop_size):
        parameters = {
            'C': np.random.uniform(0.1, 100),
            'gamma': np.random.uniform(0.001, 1),
            'kernel': np.random.choice(['linear', 'rbf', 'poly'])
        }
        population.append(parameters)
    return population

def calculate_fitness(parameters, X_train, X_test, y_train, y_test):
    svm_model = SVC(C=parameters['C'], gamma=parameters['gamma'],
kernel=parameters['kernel'])
    svm_model.fit(X_train, y_train)
    predictions = svm_model.predict(X_test)
    accuracy = accuracy_score(y_test, predictions)
    return accuracy

def select_parents(population, fitness_scores):
    fitness_scores = np.array(fitness_scores)
    selected_indices = np.random.choice(len(population), size=2,
p=fitness_scores/fitness_scores.sum(), replace=False)
    return [population[i] for i in selected_indices]

def crossover(parent1, parent2):
    if isinstance(parent1, dict) and isinstance(parent2, dict):
        child = {
            'C': np.random.choice([parent1['C'], parent2['C']]),
            'gamma': np.random.choice([parent1['gamma'], parent2['gamma']]),
            'kernel': np.random.choice([parent1['kernel'], parent2['kernel']])
        }
    elif isinstance(parent1, list) and isinstance(parent2, list):
        child = {
            'C': np.random.choice([parent1[0]['C'], parent2[0]['C']]),

```

```

        'gamma': np.random.choice([parent1[0]['gamma'], parent2[0]['gamma']]),
        'kernel': np.random.choice([parent1[0]['kernel'], parent2[0]['kernel']])
    }
    else:
        raise ValueError("Invalid data types for parents.")
    return child

def mutate(child):
    mutation_prob = 0.2
    if np.random.rand() < mutation_prob:
        child['C'] = np.random.uniform(0.1, 100)
    if np.random.rand() < mutation_prob:
        child['gamma'] = np.random.uniform(0.001, 1)
    if np.random.rand() < mutation_prob:
        child['kernel'] = np.random.choice(['linear', 'rbf', 'poly'])
    return child

def genetic_algorithm(X_train, X_test, y_train, y_test, pop_size=10, generations=5):
    population = initialize_population(pop_size)

    for generation in range(generations):
        fitness_scores = [calculate_fitness(params, X_train, X_test, y_train, y_test) for params
in population]

        best_idx = np.argmax(fitness_scores)
        best_params = population[best_idx]
        best_accuracy = fitness_scores[best_idx]

        print(f"Generation {generation+1} - Best Accuracy: {best_accuracy:.4f}, Best
Parameters: {best_params}")
        new_population = []

        for _ in range(pop_size // 2):
            parent1 = select_parents(population, fitness_scores)
            parent2 = select_parents(population, fitness_scores)

            child1 = crossover(parent1, parent2)
            child1 = mutate(child1)

            child2 = crossover(parent1, parent2)
            child2 = mutate(child2)

        new_population.extend([child1, child2])

```

```

    population = new_population
    return best_params
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=123)
best_parameters = genetic_algorithm(X_train, X_test, y_train, y_test, pop_size=43,
generations=20)

```

Listing Code GUI

```

#import library
from tkinter import *      # Untuk membuat GUI
from PIL import ImageTk, Image # Untuk membaca Gambar
from tkinter import ttk     # Untuk mengatur style widget
import pandas as pd         # untuk membaca dan menyimpan csv
from tkinter import messagebox # untuk menampilkan showinfo
from docx import Document   # untuk menyimpan entryan ke document
import datetime as dt       # untuk menampilkan tanggal bulan dan tahun pada dataset
import customtkinter        # untuk memperbagus button, combobox
import os
os.system('cls')
import shutil
import joblib
import pickle
from tkinter import filedialog
import csv
import re
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from sklearn.feature_extraction.text import TfidfVectorizer

class News_title:
    def __init__(self, master):
        self.window = master

        # Atur ukuran jendela menjadi 1280x720 piksel
        self.window.geometry('1280x720')

        # Nonaktifkan kemampuan pengubahan ukuran jendela (0, 0 berarti tidak dapat diubah
        # ukurannya)
        self.window.resizable(0, 0)

        # Memberikan judul pada GUI
        self.window.title('News Tittle Clasification')

        # Background

```

```

photo_bgDasar = ImageTk.PhotoImage(Image.open('gambarnya/bg_GUI.jpg'))
self.bgAwal = Label(self.window, image=photo_bgDasar, width=1280, height=730)
self.bgAwal.image = photo_bgDasar
self.bgAwal.place(x=0, y=0)

self.timpa = Frame(self.window, bg="#3e63bb", width=500, height=400)
self.timpa.place(x=700, y=200)

# Login to your account
self.entry = Entry(self.window, background="white", fg="black",
                    font=("britannic bold", 12), relief=FLAT)
self.entry.place(x=100, y=494, width=495, height=47)

alay_dict = pd.read_csv('new_kamusalay.csv', header=None, encoding='latin-1')
# Mengganti nama kolom
alay_dict = alay_dict.rename(columns={0: 'original',
                                     1: 'replacement'})
alay_dict.head()

id_stopword_dict = pd.read_csv('stopwordbahasa.csv', header=None, encoding='latin-1')
id_stopword_dict = id_stopword_dict.rename(columns={0: 'stopword'})

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

def lowercase(text):
    return text.lower()

def remove_unnecessary_char(text):
    text = re.sub('\n', ' ', text) # Remove every '\n'
    text = re.sub('rt', ' ', text) # Remove every retweet symbol
    text = re.sub('user', ' ', text) # Remove every username
    text = re.sub('((www\.[^\s]+)|(https?://[^\s]+))', ' ', text) # Remove every url
    text = re.sub(' +', ' ', text) # Remove extra spaces
    return text

def remove_symbolnumeric(text):
    # Menghapus simbol selain huruf dan spasi, dan menggantinya dengan spasi
    text_without_symbols = re.sub(r'[^\a-zA-Z\s]', ' ', text)

    # Mengganti multiple spasi dengan satu spasi
    text_without_symbols = re.sub(r'\s+', ' ', text_without_symbols)

```

```

    return text_without_symbols.strip()

alay_dict_map = dict(zip(alay_dict['original'], alay_dict['replacement']))

def normalize_alay(text):
    return ' '.join([alay_dict_map[word] if word in alay_dict_map else word for word in
text.split(' ')])

def remove_stopword(text):
    text = ' '.join([ ' ' if word in id_stopword_dict.stopword.values else word for word in
text.split(' ')])
    text = re.sub(' +', ' ', text) # Remove extra spaces
    text = text.strip() # Remove space in begining and end of sentences
    return text

# Mengubah kata menjadi kata dasar nya. Ex: Programmer, programming = program
def stemming (text):
    return stemmer.stem(text)

def remove_hashtags_user(text):
    tweet_without_hashtags = ' '.join(word for word in text.split() if not
word.startswith('#'))

    tweet_without_mentions = re.sub(r'@\w+', '', tweet_without_hashtags)
    return tweet_without_mentions

def preprocess(text):
    text = lowercase(text)
    text = remove_unnecessary_char(text)
    text = remove_hashtags_user(text)
    text = remove_symbolnumeric(text)
    text = normalize_alay(text)
    text = remove_stopword(text)
    text = stemming(text)
    return text

def prediksi():
    if self.entry.get():
        try:
            self.pred.destroy()

        except:
            pass

```



```

# Memuat vokabuler yang telah disimpan
with open('vokabulerbaru.pkl', 'rb') as file:
    vokabuler_loaded = pickle.load(file)
vectorizer_baru = TfidfVectorizer(vocabulary=vokabuler_loaded)
teks_baru = self.entry.get()
teks_baru = preprocess(teks_baru)
teks_baru_transformed = vectorizer_baru.fit_transform([teks_baru])

svm_tfidf_model = joblib.load('svm88.sav')
prediksi = svm_tfidf_model.predict(teks_baru_transformed)

if prediksi[0] == 'positive':
    photo_pred = ImageTk.PhotoImage(Image.open('gambarnya/positif-01.png'))
    self.pred = Label(self.timpa, image=photo_pred, bg='#3e63bb', width=1280,
height=730)
    self.pred.image = photo_pred
    self.pred.place(x=-390, y=-170)

else:
    photo_pred = ImageTk.PhotoImage(Image.open('gambarnya/negatif-01.png'))
    self.pred = Label(self.timpa, image=photo_pred, bg='#3e63bb', width=1280,
height=730)
    self.pred.image = photo_pred
    self.pred.place(x=-390, y=-170)

else:
    messagebox.showinfo("Peringatan", "Entry belum di isi")

def predik_csv():
    def process_csv(file):
        global df
        df = pd.read_csv(file_path)
        df = df.head(20)
        X = df["preprocessing"].apply(preprocess)

    # Memuat vokabuler yang telah disimpan
    with open('vokabulerbaru.pkl', 'rb') as file:
        vokabuler_loaded = pickle.load(file)

    vectorizer_baru = TfidfVectorizer(vocabulary=vokabuler_loaded)

```

```

teks_baru_transformed = vectorizer_baru.fit_transform(X)

svm_tfidf_model = joblib.load('svm88.sav')
prediksi = svm_tfidf_model.predict(teks_baru_transformed)

df["Sentiment"] = prediksi
def display_dataframe():
    for i, row in df.iterrows():
        tree.insert("", 'end', values=(row['preprocessing'], row['Sentiment']))

# Buat TreeView
columns = ('preprocessing', 'Sentiment')
tree = ttk.Treeview(self.timpa, columns=columns, show='headings')
tree.tag_configure('mytag', font=('Gungsuh', 13))

# Atur lebar kolom
for col in columns:
    if col == "preprocessing":
        tree.heading(col, text=col)
        tree.column(col, width=400)
        continue

    tree.heading(col, text=col)
    tree.column(col, width=100) # Ganti lebar sesuai kebutuhan

tree.place(x=0,y=0, height=400)

# Tambahkan data DataFrame ke dalam TreeView
display_dataframe()

for item in tree.get_children():
    tree.item(item, tags=('mytag',))

# Fungsi untuk membuka jendela dialog pemilihan file CSV
global file_path
file_path = filedialog.askopenfilename(filetypes=[("CSV files", "*.csv")])
if file_path:
    process_csv(file_path)

def export_to_csv():
    file_path_export = filedialog.asksaveasfilename(defaultextension=".csv",
filetypes=[("CSV files", "*.csv")])

```

```

if file_path_export:
    df.to_csv(file_path_export, index=False)
    messagebox.showinfo("Ekspor Sukses", "Data has been exported successfully!")

def reset():
    try:
        self.pred.destroy()
    except:
        messagebox.showinfo("Peringatan", "Entry belum di isi")

# SEARCH
image = Image.open('gambarnya/searcc.png')

# Resize the image - make sure to provide a tuple for the size (width, height)
resized_image = image.resize((40, 40), Image.Resampling.NEAREST) # or use any
other resampling filter as needed

# Convert to a format Tkinter can use
photo_search = ImageTk.PhotoImage(resized_image)
self.search = Button(self.window, image=photo_search, bg='white', relief=FLAT,
command=prediksi)
self.search.image = photo_search
self.search.place(x=550, y=495)

# FILE
image = Image.open('gambarnya/file.png')

# Resize the image - make sure to provide a tuple for the size (width, height)
resized_image = image.resize((40, 40), Image.Resampling.NEAREST) # or use any
other resampling filter as needed

# Convert to a format Tkinter can use
photo_file = ImageTk.PhotoImage(resized_image)
self.file = Button(self.window, image=photo_file, bg='white', relief=FLAT,
command=predik_csv)
self.file.image = photo_file
self.file.place(x=595, y=494, height=47)
self.btnreset=customtkinter.CTkButton(master=self.window,text=' Reset
',text_font=('Courier New', 20), corner_radius=0,text_color="black", width=150, height=40,
fg_color="white",hover_color="#444352", command=reset).place(x=320, y=550)

```

```
self.btnreset=customtkinter.CTkButton(master=self.window,text=' Export  
,text_font=('Courier New', 20), corner_radius=0,text_color="black", width=150, height=40,  
fg_color="white",hover_color="#444352", command=export_to_csv).place(x=488, y=550)
```

```
window = Tk()  
root = News_title(window)  
window.mainloop()
```