

# **LAMPIRAN**



PT BNI (PERSERO) MALANG  
BANK NISDA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG  
**INSTITUT TEKNOLOGI NASIONAL MALANG**

FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Karangie, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

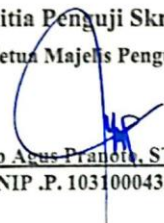
**BERITA ACARA UJIAN SKRIPSI**  
**FAKULTAS TEKNOLOGI INDUSTRI**

Nama : Cornelia Luba Tara Boro  
Nim : 2118060  
Jurusan : Teknik Informatika S-1  
Judul : Analisis Sentimen Terhadap Kampanye Pengurangan Plastik  
Pada Media Sosial Menggunakan Metode SVM

Dipertahankan Dihadapan Majelis Penguji Skripsi Jenjang Strata Satu(S-1)  
Pada


Hari : Selasa  
Tanggal : 21 Januari 2025  
Nilai : 90 (A)

Panitia Penguji Skripsi,  
Ketua Majelis Penguji


  
Yosep Agus Pranoto, ST. MT  
NIP. P. 1031000432

Anggota Penguji :

Dosen Penguji I

  
Survo Adi Wicowo, SS.T, M.T.  
NIP. P. 1031100438

Dosen Penguji II

  
Ali Mahmud, B. Eng. PhD  
NIP. P. 1031000429



PT BNI (PERSERO) MALANG  
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG  
**INSTITUT TEKNOLOGI NASIONAL MALANG**

FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang


**FORMULIR PERBAIKAN SKRIPSI**

Dalam pelaksanaan ujian skripsi jenjang Strata 1 Program Studi Teknik Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :


NAMA : Comelia Luba Tara Boro  
NIM : 2118060  
JURUSAN : Teknik Informatika S-1  
JUDUL : ANALISIS SENTIMEN TERHADAP KAMPANYE  
PENGURANGAN PLASTIK PADA MEDIA SOSIAL  
MENGUNAKAN METODE SVM

| No. | Penguji    | Tanggal         | Uraian  |
|-----|------------|-----------------|---|
| 1.  | Penguji I  | 21 Januari 2025 | <ol style="list-style-type: none"><li>1. Jawaban dari rumusan masalah</li><li>2. Kesesuaian Database dan DFD lvl 1</li><li>3. Pengujian Blackbox dan browser sesuai hak akses.</li><li>4. Pengujian Akurasi terhadap kondisi real</li></ol>               |
| 2.  | Penguji II | 21 Januari 2024 | <ol style="list-style-type: none"><li>1. Tabel dan Relasi antar Tabel</li><li>2. Penambahan teori kampanye dan Media Sosial di Bab 2</li><li>3. Pengujian User, minimal 10 user</li><li>4. Pengujian Browser (3 browser sertakan versi browser)</li></ol> |

Dosen Penguji I

  
Survo Adi W. Dowo, S.S.T, M.T.  
NIP. P.1031100438

Dosen Penguji II

  
Ali Mahmud, B.Eng, PhD  
NIP. P.1031000429

Dosen Pembimbing I

  
Ahmad Fajol, S.T, M.T  
NIP. P. 1031000431

Dosen Pembimbing II

  
Deddy Rudhistiar, S.Kom., M.Cs  
NIP.P 1032000578



PT BNI (PERSERO) MALANG  
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG  
**INSTITUT TEKNOLOGI NASIONAL MALANG**  
FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 04 Oktober 2024

Nomor : ITN-883/III.INF/TA/2024  
Lampiran : ---  
Perihal : Pembimbing Utama Skripsi

Kepada : **Yth. Bpk/Ibu Ahmad Faisol ST., MT.**  
Dosen Program Studi Teknik Informatika S-1  
Institut Teknologi Nasional  
Malang

Dengan Hormat,  
Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

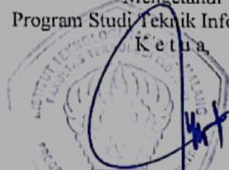
Nama : Comelia Luba Tara Boro  
Nim : 2118060  
Prodi : Teknik Informatika S-1  
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

**21 Agustus 2024 s/d 21 Pebruari 2025**

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.  
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui  
Program Studi Teknik Informatika S-1  
Ketua

  
**Yosep Agus Pranoto, ST., MT.**  
NIP.P-1031000432

Form S-4a



PT BNI (PERSERO) MALANG  
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG  
**INSTITUT TEKNOLOGI NASIONAL MALANG**  
FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 04 Oktober 2024

Nomor : ITN-883/III-INF/TA/2024  
Lampiran : ---  
Perihal : Pembimbing Pendamping Skripsi

Kepada : **Yth. Bpk/Ibu Deddy Rudhistiar S.Kom., M.Cs.**  
Dosen Pembina Program Studi Teknik Informatika S-1  
Institut Teknologi Nasional  
Malang

Dengan Hormat,  
Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

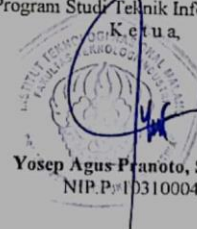
Nama : Cornelia Luba Tara Boro  
Nim : 2118060  
Prodi : Teknik Informatika S-1  
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

**21 Agustus 2024 s/d 21 Pebruari 2025**

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.  
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.











Mengetahui  
Program Studi Teknik Informatika S-1  
Ketua,

  
**Yosep Agus Pranoto, ST., MT.**  
NIP.P:1031000432


Form S-4a

## FORMULIR BIMBINGAN SKRIPSI

Nama : CORNELIA LUBA TARA BORO  
Nim : 2118060  
Masa Bimbingan : 21 AGUSTUS 2024 s.d 21 PEBRUARI 2025  
Judul Skripsi : ANALISIS SENTIMEN TERHADAP KAMPANYE PENGURANGAN PENGGUNAAN PLASTIK PADA MEDIA SOSIAL X MENGGUNAKAN METODE SUPPORT VECTOR MACHINE

| No. | Tanggal    | Uraian   | Paraf Pembimbing  |
|-----|------------|--|---|
| 1.  | 16/10/2024 | Konsultasi Perancangan SKRIPSI dan Perancangan website |    |
| 2.  | 23/10/2024 | Konsultasi Data (Pre-processing & Labelling)           |    |
| 3.  | 30/10/2024 | Konsultasi Penulisan Laporan Seminar Progress          |    |
| 4.  | 16/11/2024 | Konsultasi Pengerjaan Aplikasi                         |  |
| 5.  | 26/11/2024 | Konsultasi Pengerjaan Aplikasi                         |  |
| 6.  | 3/12/2024  | Konsultasi Penulisan Jurnal + Laporan SEMHAS           |  |
| 7.  | 4/12/2024  | Pelaporan Revisi Penulisan Jurnal + Laporan SEMHAS     |  |
| 8.  | 4/12/2024  | ACC SEMINAR HASIL                                      |  |
| 9.  | 16/12/2024 | Konsultasi Revisi Laporan SEMHAS                       |  |
| 10. | 16/12/2024 | ACC kompre   |  |











Malang, 2024

  
Ahmad Faisol, ST, MT  
NIP.P.1031000431



## FORMULIR BIMBINGAN SKRIPSI

Nama : CORNELIA LUBA TARA BORO  
 Nim : 2118060  
 Masa Bimbingan : 21 AGUSTUS 2024 s.d 21 FEBRUARI 2025  
 Judul Skripsi : ANALISIS SENTIMEN TERHADAP KAMPANYE  
 PENGURANGAN PENGGUNAAN PLASTIK PADA MEDIA  
 SOSIAL X MENGGUNAKAN METODE SUPPORT VECTOR  
 MACHINE

| No. | Tanggal    | Uraian  | Paraf Pembimbing  |
|-----|------------|---|---|
| 1.  | 12/10/2024 | Pengarahan Bimbingan, penentuan target pengerjaan SKRIPSI |    |
| 2.  | 17/10/2024 | Konsultasi Pre-processing Data                            |    |
| 3.  | 24/10/2024 | Konsultasi Labelling Dataset                              |    |
| 4.  | 31/10/2024 | Konsultasi Penulisan Laporan                              |   |
| 5.  | 09/11/2024 | Konsultasi Hasil Seminar Progress                         |  |
| 6.  | 23/11/2024 | Konsultasi Penulisan Jurnal                               |  |
| 7.  | 28/11/2024 | Konsultasi Penulisan Jurnal & Laporan SEMHAS              |  |
| 8.  | 14/01/2025 | Pelaporan Hasil Revisi SEMHAS                             |  |
| 9.  | 15/01/2025 | Pelaporan Hasil Akhir Program                             |  |
| 10. | 16/01/2025 | Konsultasi Persiapan KOMPRE                               |  |

Malang, 2024

  
Deddy Rudhastiar S.Kom., M.Cs  
 NIP.P 1032000578



**INSTITUT TEKNOLOGI NASIONAL MALANG**

**PERPUSTAKAAN PUSAT**

Jln. Bendungan Sigura-gura No 2 Malang 65145

Telp. (0341) 551431 Pcs 163-146-147 Fax. (0341) 553015 Website : library.itn.ac.id

### FORM UJI PLAGIASI UNTUK MAHASISWA

Yang bertandatangan di bawah ini, Mahasiswa Institut Teknologi Nasional Malang:

Nama : CORNELIA LUGA TARA BORO  
NIM : 2118060  
Fakultas / Jurusan : TEKNOLOGI INDUSTRI / TEKNIK INFORMATIKA S1  
Email : cornelialtboro06@gmail.com  
No. Tlp : 081299026064  
Judul/ Jml artikel : ANALISIS SENTIMEN TERHADAP KAMPANYE PENGURANGAN PLASTIK  
PADA MEDIA SOSIAL MENGGUNAKAN METODE SVM  
.....  
.....

Karya ilmiah yang bersangkutan di atas melalui proses cek plagiatsi menggunakan aplikasi trunitir dengan hasil kemiripan ( Similarity) Sebesar.....%..... %

Demikian surat keterangan ini dibuat agar dapat dipergunakan sebagaimana mestinya.

Mahasiswa

CORNELIA LUGA TARA BORO  
2118060

Malang, 7 Februari 2025

Pelaksana

Retno W. PPT



## Source Code :

### Routes.py

```
@blueprint.route('/preprocessing', methods=['POST'])
@login_required
def preprocess_dataset():
    try:
        # Ambil semua data dari tabel dataset
        dataset = Data.query.all()
        if not dataset:
            return jsonify({"message": "Dataset kosong. Silakan
unggah dataset terlebih dahulu."}), 404
        # Hapus semua data lama di tabel PreprocessedTrainingData
        PreprocessedTrainingData.query.delete()
        db.session.commit() # Konfirmasi penghapusan
        processed_count = 0
        # Proses setiap teks di dataset
        for data in dataset:
            if not data.teks or not isinstance(data.teks, str):
                blueprint.logger.warning(f"Teks tidak valid untuk
data ID={data.id}")
                continue
            # Panggil fungsi preprocess_text
            processed = preprocess_text(data.teks)
            if not processed or not isinstance(processed, dict):
                raise ValueError(f"Pre-processing gagal atau data
tidak valid untuk ID={data.id}.")
            # Tambahkan data baru ke tabel PreprocessedTrainingData
            preprocessed_entry = PreprocessedTrainingData(
                original_text=data.teks,
                cleaned_text=processed["cleaned_text"],
                case_folded_text=processed["case_folded_text"],
                tokenized_text=processed["tokenized_text"],
                normalized_text=processed["normalized_text"],
                stopword_removed_text=processed["stopword_removed_t
ext"],
                stemmed_text=processed["stemmed_text"],
                final_text=processed["final_text"],
                dataset_id=data.id
            )
            db.session.add(preprocessed_entry)
            processed_count += 1
        db.session.commit() # Konfirmasi penyimpanan data baru
        # Feedback untuk pengguna
        return jsonify({
            "message": "Proses Pre-Processing telah selesai! Semua
data telah digantikan.",
            "total_processed": processed_count,
            "role": session.get('role') # Kirim role ke front-end
        }), 200
```

```

except Exception as e:
    db.session.rollback()
    blueprint.logger.error(f"Terjadi kesalahan: {e}")
    return jsonify({"message": "Terjadi kesalahan selama proses
pre-processing.", "error": str(e)}), 500

@blueprint.route('/klasifikasi/testing', methods=['GET', 'POST'])
def klasifikasi_testing_svm():
    try:
        if request.method == 'GET':
            classification_results = KlasifikasiTesting.query.all()
            return render_template('home/datatesting.html',
classification_results=classification_results)

        elif request.method == 'POST':
            current_app.logger.debug("Starting POST request for SVM
testing...")

            # Step 1: Ambil Data Testing
            dataset = TestingData.query.all()
            if not dataset:
                current_app.logger.error("No testing data found.")
                return jsonify({'error': 'No testing data found.
Please upload testing data first.'}), 404

            # Step 2: Preprocessing Data Testing
            PreprocessedTestingData.query.delete() # Hapus data
sebelumnya
            db.session.commit()

            processed_data = []
            for data in dataset:
                if not data.teks or not isinstance(data.teks, str):
                    current_app.logger.warning(f"Skipping invalid
data ID: {data.id}")
                    continue

                # Lakukan preprocessing
                processed = preprocess_text(data.teks)
                if not processed or not isinstance(processed,
dict):
                    current_app.logger.error(f"Preprocessing failed
for data ID: {data.id}")
                    continue

                preprocessed_entry = PreprocessedTestingData(
                    original_text=data.teks,
                    cleaned_text=processed["cleaned_text"],
                    case_folded_text=processed["case_folded_text"],
                    tokenized_text=processed["tokenized_text"],
                    normalized_text=processed["normalized_text"],
                    stopword_removed_text=processed["stopword_remov
ed_text"],

```

```

        stemmed_text=processed["stemmed_text"],
        final_text=processed["final_text"],
        testing_data_id=data.id
    )
    db.session.add(preprocessed_entry)
    processed_data.append(preprocessed_entry)

    db.session.commit()
    if not processed_data:
        current_app.logger.error("No valid data found after
preprocessing.")
        return jsonify({'error': 'No valid data found after
preprocessing.'}), 400

    # Step 3: Klasifikasi
    model_path = os.path.join('models',
'svm_rbf_model.pkl')
    vectorizer_path = os.path.join('models',
'tfidf_vectorizer.pkl')

    if not os.path.exists(model_path):
        current_app.logger.error("Model file not found.")
        return jsonify({'error': 'SVM model file not
found.'}), 500

    if not os.path.exists(vectorizer_path):
        current_app.logger.error("Vectorizer file not
found.")
        return jsonify({'error': 'TF-IDF vectorizer file
not found.'}), 500

    model = joblib.load(model_path)
    vectorizer = joblib.load(vectorizer_path)

    final_texts = [data.final_text for data in
processed_data]
    if not final_texts:
        current_app.logger.error("No texts to transform
using TF-IDF.")
        return jsonify({'error': 'No valid texts for
classification.'}), 400

    tfidf_matrix = vectorizer.transform(final_texts)
    dense_matrix = tfidf_matrix.toarray()

    predictions = model.predict(dense_matrix)

    # Step 4: Simpan Hasil Klasifikasi
    KlasifikasiTesting.query.delete() # Hapus data
klasifikasi sebelumnya
    db.session.commit()

    for i, data in enumerate(processed_data):
        classification_result = KlasifikasiTesting(

```

```

        teks=data.original_text,
        hasil_klasifikasi=predictions[i]
    )
    db.session.add(classification_result)
db.session.commit()
# Mengambil data hasil klasifikasi
data = [
    {
        "id": i + 1,
        "teks": processed_data[i].original_text,
        "hasil_klasifikasi": predictions[i],
    }
    for i in range(len(predictions))
]
return jsonify({'success': True, 'data': data})

except Exception as e:
    current_app.logger.error(f"Unexpected error: {str(e)}",
exc_info=True)
    return jsonify({'error': 'Unexpected error occurred.'}),
500

```

## Models.py

```

from apps import db

class Data(db.Model):
    __tablename__ = 'dataset'
    id = db.Column(db.Integer, primary_key=True)
    teks = db.Column(db.Text, nullable=False)
    label = db.Column(db.String(50), nullable=False)
    def __init__(self, teks, label):
        self.teks = teks
        self.label = label

class TestingData(db.Model):
    __tablename__ = 'testing_data'

    id = db.Column(db.Integer, primary_key=True)
    teks = db.Column(db.Text, nullable=False)
    def __init__(self, teks):
        self.teks = teks

class PreprocessedTrainingData(db.Model):
    __tablename__ = 'preprocessed_training_data'
    id = db.Column(db.Integer, primary_key=True)
    original_text = db.Column(db.Text, nullable=False)
    cleaned_text = db.Column(db.Text, nullable=False)
    case_folded_text = db.Column(db.Text, nullable=False)
    tokenized_text = db.Column(db.Text, nullable=False)
    normalized_text = db.Column(db.Text, nullable=False)
    stopword_removed_text = db.Column(db.Text, nullable=False)
    stemmed_text = db.Column(db.Text, nullable=False)
    final_text = db.Column(db.Text, nullable=False)

```

```

dataset_id = db.Column(
    db.Integer,
    db.ForeignKey('dataset.id', ondelete='CASCADE'),
    nullable=False )
dataset = db.relationship('Data',
backref=db.backref('preprocessed_data', lazy=True))
def __init__(self, original_text, cleaned_text,
case_folded_text, tokenized_text,
        normalized_text, stopword_removed_text,
stemmed_text, final_text, dataset_id):
    self.original_text = original_text
    self.cleaned_text = cleaned_text
    self.case_folded_text = case_folded_text
    self.tokenized_text = tokenized_text
    self.normalized_text = normalized_text
    self.stopword_removed_text = stopword_removed_text
    self.stemmed_text = stemmed_text
    self.final_text = final_text
    self.dataset_id = dataset_id

class PreprocessedTestingData(db.Model):
    __tablename__ = 'preprocessed_testing_data'
    id = db.Column(db.Integer, primary_key=True)
    original_text = db.Column(db.Text, nullable=False)
    cleaned_text = db.Column(db.Text, nullable=False)
    case_folded_text = db.Column(db.Text, nullable=False)
    tokenized_text = db.Column(db.Text, nullable=False)
    normalized_text = db.Column(db.Text, nullable=False)
    stopword_removed_text = db.Column(db.Text, nullable=False)
    stemmed_text = db.Column(db.Text, nullable=False)
    final_text = db.Column(db.Text, nullable=False)
    testing_data_id = db.Column(db.Integer,
db.ForeignKey('testing_data.id', ondelete='CASCADE'),
nullable=False)
    testing_data = db.relationship('TestingData',
backref=db.backref('preprocessed_data', lazy=True))
    def __init__(self, original_text, cleaned_text,
case_folded_text, tokenized_text,
        normalized_text, stopword_removed_text,
stemmed_text, final_text, testing_data_id):
        self.original_text = original_text
        self.cleaned_text = cleaned_text
        self.case_folded_text = case_folded_text
        self.tokenized_text = tokenized_text
        self.normalized_text = normalized_text
        self.stopword_removed_text = stopword_removed_text
        self.stemmed_text = stemmed_text
        self.final_text = final_text
        self.testing_data_id = testing_data_id

class KlasifikasiTesting(db.Model):
    tablename = 'klasifikasi testing'

```

```

    id = db.Column(db.Integer, primary_key=True,
autoincrement=True, unique=True)
    teks = db.Column(db.String(255))
    hasil_klasifikasi = db.Column(db.String(50))

    def __init__(self, teks, hasil_klasifikasi):
        self.teks = teks
        self.hasil_klasifikasi = hasil_klasifikasi

```

## Preprocessing.py

```

import string
import re
import emoji
import pandas as pd
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from Sastrawi.StopWordRemover.StopWordRemoverFactory import
StopWordRemoverFactory
import os
import nltk
nltk.download('punkt')

# Muat kamus normalisasi
BASE_DIR = os.path.dirname(os.path.abspath(__file__))

normalisasi_kata_df = pd.read_csv(os.path.join(BASE_DIR, '..',
'data', 'kamus-normalisasi.csv'))
normalisasi_kata_dict = dict(zip(normalisasi_kata_df['before'],
normalisasi_kata_df['after']))

# Inisialisasi StopWordRemoverFactory dan daftar stopword hanya
sekali
factory = StopWordRemoverFactory()
stop_words = factory.get_stop_words() + [
    'a', 'ac', 'ae', 'aja', 'ajah', 'ak', 'akun', 'al', 'alah',
'ama', 'amp', 'an', 'anjur', 'are',
    'asyik', 'asyikkkk', 'atuh', 'aya', 'bafe', 'bang', 'bari',
'be', 'ben', 'bos', 'bro', 'bu', 'buibuu', 'by',
    'cabul', 'caj', 'cc', 'cha', 'cr', 'd', 'deh', 'deui', 'dg',
'dgn', 'did', 'dlh', 'dlnya', 'dok', 'dong', 'dongg',
    'dpt', 'dulur', 'dw', 'e', 'ec', 'eh', 'eiitssss', 'el',
'engak', 'engga', 'engken', 'era', 'euy', 'follow',
    'ga', 'gaes', 'gal', 'ganapti', 'gk', 'gpp', 'gt', 'gue', 'gw',
'ha', 'haha', 'hahaha', 'hai', 'heh', 'hehe',
    'hese', 'hi', 'hmm', 'hola', 'huhh', 'ibs', 'ibu', 'ih', 'iso',
'it', 'iyut', 'jadi', 'jd', 'je', 'jee', 'jk',
    'kak', 'kamu', 'kan', 'kat', 'kau', 'kie', 'kl', 'kok',
'kokedama', 'kpd', 'kuy', 'laen', 'lah', 'lahh', 'lalala',
    'lha', 'lho', 'like', 'loh', 'loo', 'lu', 'masa', 'mbak',
'mbok', 'nah', 'nahh', 'nak', 'ndeso', 'nenek',
    'ngga', 'nih', 'njir', 'no', 'nya', 'oke', 'pak', 'perlis',
'pgn', 'pih', 'pria', 'pulak', 'r', 'rijig',

```



```

'sa', 'sama', 'saya', 'selengkapnya', 'seorang', 'share', 'si',
'siah', 'siej', 'sih', 'sini', 'sj', 'so', 'sok',
'sono', 'spt', 'subscribe', 'sy', 'ta', 'taek', 'tak', 'tau',
'td', 'tdk', 'teba', 'teh', 'teu', 'thd', 'the',
'tho', 'tol', 'tp', 'tu', 'tuda', 'tuh', 'type', 'uea', 'via',
'vt', 'wae', 'wah', 'we', 'weee', 'whahaha',
'wkwk', 'wkwkwk', 'wkwkwkwk', 'wkwkwkkw', 'ws', 'wwf', 'yaa',
'yaaa', 'yaaaa', 'yah', 'yapp', 'yapss', 'yg',
'yhuu', 'yo', 'yuk', 'yukkk', 'yuuk', 'yuukk', 'zr',
'wuih', 'hiks'
]

# Fungsi cleaning text
def cleaning_text(text):
    text = re.sub(r'@\S+', '', text)
    text = re.sub(r'^RT[\s]+', '', text)
    hashtags_with_plastik = re.findall(r'#\S*plastik\S*', text,
flags=re.IGNORECASE)
    text = re.sub(r'#\S+', '', text)
    text += " " + " ".join([ht.replace('#', '') for ht in
hashtags_with_plastik])
    text = re.sub(r'http\S+', '', text)
    text = re.sub(r'[0-9]+', '', text)
    text = emoji.demojize(text)
    text = text.translate(str.maketrans(string.punctuation, ' '*
len(string.punctuation)))
    text = re.sub(r'\s+', ' ', text).strip()
    return text

# Fungsi case folding
def case_folding(text):
    return text.lower()

# Fungsi tokenizing
def tokenizing(text):
    return word_tokenize(text)

# Fungsi normalisasi
def normalisasi(text):
    return [normalisasi_kata_dict.get(word, word) for word in text]

# Fungsi stopword removal
def stopword_removal(text):
    return [word for word in text if word not in stop_words]

# Fungsi stemming
def stemming(text):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    return [stemmer.stem(word) for word in text if isinstance(word,
str)]

# Fungsi untuk menjalankan semua tahap pre-processing
def preprocess_text(original_text):

```

```

cleaned_text = cleaning_text(original_text)
case_folded_text = case_folding(cleaned_text)
tokenized_text = tokenizing(case_folded_text)
normalized_text = normalisasi(tokenized_text)
stopword_removed_text = stopword_removal(normalized_text)
stemmed_text = stemming(stopword_removed_text)
final_text = ' '.join(stemmed_text)
return {
    "original_text": original_text,
    "cleaned_text": cleaned_text,
    "case_folded_text": case_folded_text,
    "tokenized_text": ' '.join(tokenized_text),
    "normalized_text": ' '.join(normalized_text),
    "stopword_removed_text": ' '.join(stopword_removed_text),
    "stemmed_text": ' '.join(stemmed_text),
    "final_text": final_text
}

```

## Klasifikasitraining.html

```

{% extends "layouts/base.html" %}
{% block title %}Training Data Klasifikasi{% endblock %}
{% block stylesheets %}
<style>
    .main-content {margin-left: 7px;padding: 4px;}
    h3, h4 {color: #ff527b;}
    .proses-section, .hasil-section {background-color: #f9f9f9;
        border-radius: 8px;padding: 20px;margin-top: 20px;flex: 1;
        box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);}
    .dataset-section {background-color: #f9f9f9;border-radius: 8px;
        padding: 20px;margin-top: 20px;}
    .d-flex-container {display: flex;gap: 20px;flex-wrap: wrap;}
    .button-container {display: flex;justify-content: flex-
start; gap: 15px;margin-top: 10px; }

    .btn-preprocess {display: inline-block;padding: 10px 20px;
        font-size: 16px;font-weight: bold; border: none;
        border-radius: 5px;cursor: pointer;transition: background-color
0.3s ease; background-color: #5dae60;color: white; }

    .btn-preprocess:hover { background-color: #45a049;}
    .btn-download {display: inline-block;padding: 10px 20px;
        font-size: 16px;font-weight: bold; border: none;
        border-radius: 5px; cursor: pointer;transition: background-
color 0.3s ease; background-color: #007bff; color: white;}

    .btn-download:hover { background-color: #0056b3;}

    .btn-download:active { background-color: #004085;}

    .nav-tabs .nav-link.active {background-color: #007bff;
        color: white;}

    .table td, .table th {

```

```

vertical-align: top;word-wrap: break-word; max-width: 300px;
white-space: normal; /* Perubahan dari nowrap ke normal */
overflow: hidden; /* Masih diperlukan jika Anda tidak ingin
scrollbar */text-overflow: ellipsis;word-break: break-word; /*
Menangani kata panjang */
/* Kolom "No" lebih kecil */#classification-results-table th:nth-
child(1), #classification-results-table td:nth-child(1) {
width: 70px; /* Lebar kolom No */ text-align: center; }
/* Kolom "Teks" lebih lebar */
#classification-results-table th:nth-child(2),
#classification-results-table td:nth-child(2) {
width: auto; /* Sisa ruang diberikan ke kolom ini */
word-wrap: break-word; white-space: normal; }
.custom-alert { background-color: #ff527b; color: #ffffff;
padding: 10px 15px; border-radius: 8px; border: 1px solid
#ff527b; font-size: 16px; margin: 10px 0; line-height: 1.4; }

.custom-link { color: #FFDA76; text-decoration: underline;
font-weight: bold; }
.custom-link:hover { color: #ffda76d7;text-decoration: none;
}
</style>
{% endblock stylesheets %}
{% block content %}
<div class="main-content">
<div class="container">
<!-- Proses Section -->
<div class="d-flex justify-content-between">
<div class="proses-section flex-grow-1 me-3">
<h3>Training Klasifikasi SVM</h3>
<div class="custom-alert">
<p>Pastikan sudah <a href="/upload.html" class="custom-
link">Import Data</a> lalu <a href="/preprocessing" class="custom-
link">Pre-Processing Data</a>.</p>
</div>
<!-- <div id="flash-message-container" class="flash-message
d-none">
<span id="flash-message-text"></span>
<button class="close-btn"
onclick="hideFlashMessage()">x</button>
</div> -->

<div class="form-group">
<label for="split-type" class="form-label">Pilih Jenis
Split Data:</label>
<select id="split-type" class="form-select">
<option value="70-30">70% Training, 30%
Testing</option>
<option value="75-25">75% Training, 25%
Testing</option>
<option value="80-20" selected>80% Training, 20%
Testing</option>
<option value="85-15">85% Training, 15%
Testing</option>

```

```

        <option value="90-10">90% Training, 10%
Testing</option>
    </select>
</div>
    <div class="button-container">
        <button id="train-svm" class="btn btn-preprocess"
onclick="startTraining()">
            <span id="train-svm-text">Training Dataset</span>
        </button>
        <button id="download-btn" class="btn btn-
download">Download Hasil</button>
    </div>
</div>

<!-- Hasil Section -->
<div class="hasil-section flex-grow-1">
    <h3>Hasil Klasifikasi Training</h3>
    <div id="training-results" class="d-none">
        <p><strong>Rasio Split Dataset : </strong> <span
id="split_type">-</span></p>
        <p><strong>Tingkat Akurasi:</strong> <span id="accuracy-
value">-</span></p>
        <p><strong>Confusion Matrix:</strong></p>
        <table class="table table-bordered">
            <thead>
                <tr>
                    <th>Actual/Predicted</th>
                    <th>Negative</th>
                    <th>Neutral</th>
                    <th>Positive</th>
                </tr>
            </thead>
            <tbody id="confusion-matrix"></tbody>
        </table>
    </div>
    <div id="no-results" class="custom-alert">
        Belum ada hasil klasifikasi. Silakan lakukan training
terlebih dahulu.
    </div>
</div>
</div>

<!-- Dataset Section -->
<div class="dataset-section">
    <h4>Hasil Training Klasifikasi</h4>
    <div class="table-responsive">
        <table class="table table-bordered" id="classification-
results-table">
            <thead>
                <tr>
                    <th>No</th>
                    <th>Teks</th>
                    <th>Label Asli</th>
                    <th>Hasil Prediksi</th>
                </tr>
            </thead>
        </table>
    </div>
</div>

```



```

        zeroRecords: "Tidak ditemukan data yang sesuai",
    },
});

// Ambil data hasil klasifikasi dari server
fetchClassificationResults(dataTable);

// Event listener untuk tombol training
$("#train-svm").on("click", function () {
    startTraining();
});

});

// Fungsi untuk mengambil hasil klasifikasi dari backend
function fetchClassificationResults(dataTable) {
$.ajax({
    url: "/klasifikasi/results",
    type: "GET",
    success: function (response) {
        if (response.results_testing &&
response.results_testing.length > 0) {
            updateTable(dataTable, response.results_testing);
            updateConfusionMatrix(response.confusion_matrix);
            $("#accuracy-value").text((response.accuracy *
100).toFixed(2) + "%");
            $("#split_type").text(response.split_type || "-");
            $("#training-results").removeClass("d-none");
            $("#no-results").addClass("d-none");
        } else {
            $("#training-results").addClass("d-none");
            $("#no-results").removeClass("d-none");
        }
    },
    error: function (xhr) {
        console.error("Error fetching classification results:",
xhr.responseJSON || xhr.statusText);
        $("#training-results").addClass("d-none");
        $("#no-results").removeClass("d-none");
    },
});
}

// Fungsi untuk memulai proses training
function startTraining() {
    if ($("#train-svm").prop("disabled")) {
        return; // Jika tombol sedang disabled, jangan proses ulang
    }
    $("#train-svm").prop("disabled", true).text("Processing...");

    const splitType = $("#split-type").val(); // Mendapatkan rasio
split dari dropdown

$.ajax({
    url: "/klasifikasi/train",

```



```

type: "POST",
contentType: "application/json",
data: JSON.stringify({ split_type: splitType }),
success: function (response) {
    $("#train-svm").prop("disabled", false).text("Training
Dataset");

    if (response.accuracy) {
        $("#accuracy-value").text((response.accuracy *
100).toFixed(2) + "%");
    }

    if (response.split_type) {
        $("#split_type").text(response.split_type); // Perbarui
split type di hasil }
        if (response.confusion_matrix) {
            updateConfusionMatrix(response.confusion_matrix); }
        if (response.results_testing) {
            updateTable("#dataTable", response.results_testing); }
        showFlashMessage(response.flash_message, true);
        $("#training-results").removeClass("d-none");
        $("#no-results").addClass("d-none"); },
error: function (xhr) {
    $("#train-svm").prop("disabled", false).text("Training
Dataset");
    const errorMessage = xhr.responseJSON?.error || "Terjadi
kesalahan pada server.";
    showFlashMessage(`Proses training gagal! Error:
${errorMessage}`, false); }); }
// Fungsi untuk memperbarui confusion matrix
function updateConfusionMatrix(matrix) {
    const tbody = $("#confusion-matrix").empty();
    const labels = ["Negative", "Neutral", "Positive"];
    matrix.forEach((row, index) => {
        tbody.append(`
            <tr>
                <th>${labels[index]}</th>
                ${row.map((value) => `<td>${value}</td>`).join("")}
            </tr>
        `);
    });
}
// Fungsi untuk memperbarui tabel hasil klasifikasi
function updateTable(dataTable, data) {
    dataTable.clear();
    data.forEach((item, index) => {
        dataTable.row.add([
            index + 1,
            item.teks,
            item.label_asli,
            item.prediksi,
            item.status,
        ]);
    });
}

```

```
    dataTable.draw();
}

$('#download-btn').click(function () {
    window.location.href = "/klasifikasi/download";
});
</script>
{% endblock javascripts %}
```