```
[ ]  from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
     import swifter


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

     # stemmed
     def stemmed_wrapper(term):
         return stemmer.stem(term)

     term_dict = {}

     for document in tweet_pelecehan['Stopwords']:
         for term in document:
             if term not in term_dict:
                 term_dict[term] = ' '

     print(len(term_dict))
     print("------------------------")

     for term in term_dict:
         term_dict[term] = stemmed_wrapper(term)
         print(term,":" ,term_dict[term])
```

```
[ ]  normalizad_word = pd.read_csv("/content/sample_data/normalisasi.csv",encoding='latin1')

     normalizad_word_dict = {}

     for index, row in normalizad_word.iterrows():
         if row[0] not in normalizad_word_dict:
             normalizad_word_dict[row[0]] = row[1]

     def normalized_term(document):
         return [normalizad_word_dict[term] if term in normalizad_word_dict else term for term in document]

     tweet_pelecehan['Normalisasi'] = tweet_pelecehan['Token'].apply(normalized_term)

     tweet_pelecehan['Normalisasi']
```

Lampiran 1 Preprocessing

Lampiran 2

```
import pandas as pd
import nltk

# Baca file kosakata positif dalam format TSV
df_positif = pd.read_csv('/content/sample_data/positive1.tsv', delimiter='\t')

# Baca file kosakata negatif dalam format TSV
df_negatif = pd.read_csv('/content/sample_data/negative1.tsv', delimiter='\t')
```

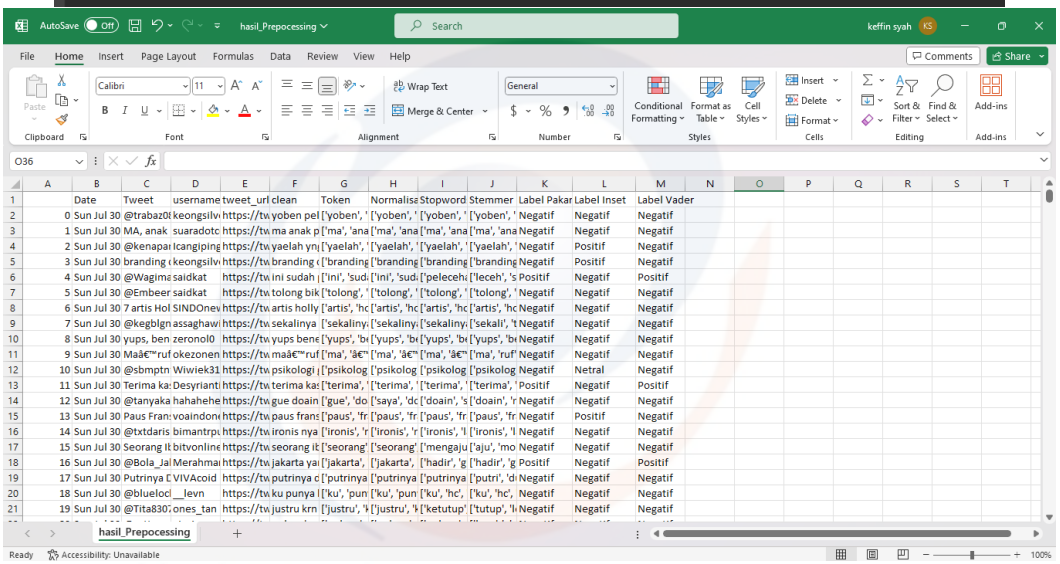
```
[ ]  #Menggabungkan Data Kosakata
     df_kosakata = pd.concat([df_positif, df_negatif], ignore_index=True)
```

```
[ ]  #Membuat Kamus Kata Sentimen
     kamus_sentimen = dict(zip(df_kosakata['word'], df_kosakata['weight']))
```

```
[ ]  def melabelkan_teks(teks):
         kata_kunci = teks.split()  # Pisahkan kata-kata dalam teks
         sentimen = []  # Menyimpan label sentimen untuk setiap kata

         for kata in kata_kunci:
             if kata in kamus_sentimen:
```

```
import googletrans
from googletrans import Translator

[ ]  df = pd.read_csv('/content/sample_data/Normalisasi.csv')
     df['Tweet'] = df['Tweet'].astype(str)
     df.head(1500)
```

```
df.loc[df['Compound_Scores'] < 0, 'Sentiments'] = 'Negatif'
df.loc[df['Compound_Scores'] > 0, 'Sentiments'] = 'Positif'
df.loc[df['Compound_Scores'] == 0, 'Sentiments'] = 'Netral'

df.head(11000)
```



Lampiran 2 Pelabelan

Lampiran 3

```
import pandas as pd
import matplotlib.pyplot as plt
from wordcloud import WordCloud

def create_wordcloud(data, polarity=None):
    if polarity:
        data = data[data['polarity'] == polarity]
        title = f'WordCloud - {polarity.capitalize()}'
    else:
        title = 'WordCloud - Semua Polaritas'

    # Menggabungkan semua kata dari tweet ke dalam satu string
    all_text = " ".join(data['Tweet'].astype(str))  # Mengubah ke string menggunakan astype(str)

    # Membuat objek WordCloud
    wordcloud = WordCloud(width=800, height=600, background_color='white').generate(all_text)

    # Plot WordCloud
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis('off')
    plt.title(title)
    plt.show()
```

Lampiran 3 Wordcloud

Lampiran 4

```python
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score, f1_score
import seaborn as sns
import matplotlib.pyplot as plt

# Step 1: Import library and read the dataset
data_NB = pd.read_csv("/content/sample_data/Hasil.csv")

# Step 2: Preprocessing (if needed) - Not shown in this example as it depends on the data

# Step 3: Split the data into features (tweets) and labels (labels)
tweets = data_NB['Tweet']
labels = data_NB['Label']

# Step 4: Feature extraction using TF-IDF
tfidf_vectorizer = TfidfVectorizer(max_features=1000)  # You can adjust the number of features as needed
X_tfidf = tfidf_vectorizer.fit_transform(tweets)

# Step 5: Split the data into training and testing sets (80% - 20%)
X_train, X_test, y_train, y_test = train_test_split(X_tfidf, labels, test_size=0.2, random_state=42)

# Membuat objek Naive Bayes
nb_classifier = MultinomialNB()
```

Lampiran 4 Algoritma Naïve Bayes Classifer

Lampiran 5

```python
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score, f1_score
import seaborn as sns
import matplotlib.pyplot as plt

# Step 1: Import library and read the dataset
data_SV = pd.read_csv("/content/sample_data/Hasil.csv")

# Step 2: Preprocessing (if needed) - Not shown in this example as it depends on the data

# Step 3: Split the data into features (tweets) and labels (labels)
tweets = data_SV['Tweet']
labels = data_SV['Label']

# Step 4: Feature extraction using TF-IDF
tfidf_vectorizer = TfidfVectorizer(max_features=1000)  # You can adjust the number of features as needed
X_tfidf = tfidf_vectorizer.fit_transform(tweets)

# Step 5: Split the data into training and testing sets (80% - 20%)
X_train, X_test, y_train, y_test = train_test_split(X_tfidf, labels, test_size=0.2, random_state=42)

# Membuat objek SVM
svm_classifier = SVC(kernel='linear', random_state=42)
```

Lampiran 5 Algoritma Support Vector Machine

Lampiran 6

```python
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier  # Menggunakan Random Forest
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score, f1_score
import seaborn as sns
import matplotlib.pyplot as plt

# Step 1: Import library and read the dataset
data_RF = pd.read_csv("/content/sample_data/Hasil.csv")

# Step 2: Preprocessing (if needed) - Not shown in this example as it depends on the data

# Step 3: Split the data into features (tweets) and labels (labels)
tweets = data_RF['Tweet']
labels = data_RF['Label']

# Step 4: Feature extraction using TF-IDF
tfidf_vectorizer = TfidfVectorizer(max_features=1000)  # You can adjust the number of features as needed
X_tfidf = tfidf_vectorizer.fit_transform(tweets)

# Step 5: Split the data into training and testing sets (80% - 20%)
X_train, X_test, y_train, y_test = train_test_split(X_tfidf, labels, test_size=0.2, random_state=42)

# Membuat objek Random Forest Classifier
rf_classifier = RandomForestClassifier(random_state=42)
```

Lampiran 6 Algoritma Random Forest

Lampiran 7

```python
import nltk
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
from nltk.tokenize import word_tokenize, sent_tokenize
from mlxtend.frequent_patterns import apriori, association_rules

df = pd.read_csv('/content/sample_data/HasilPositif.csv', usecols=['final_data'])
df['final_data'] = df['final_data'].str.split()
te = TransactionEncoder()
te_ary = te.fit_transform(df['final_data'])
df_encoded = pd.DataFrame(te_ary, columns=te.columns_)
frequent_itemsets = apriori(df_encoded, min_support=0.1, use_colnames=True)
association_rules_df = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.1)
association_rules_filtered = association_rules_df[['antecedents', 'consequents', 'support', 'confidence', 'lift']]
association_rules_filtered.head(50)
```

```python
import pandas as pd
import re
import numpy as np
from nltk.probability import FreqDist
import matplotlib.pyplot as plt
from collections import Counter

df = pd.read_csv('/content/sample_data/HasilNegatif.csv')

cnt = Counter(" ".join(df.final_data).split()).most_common(20)
cnt
```

Lampiran 7 Asosiaasi Rule

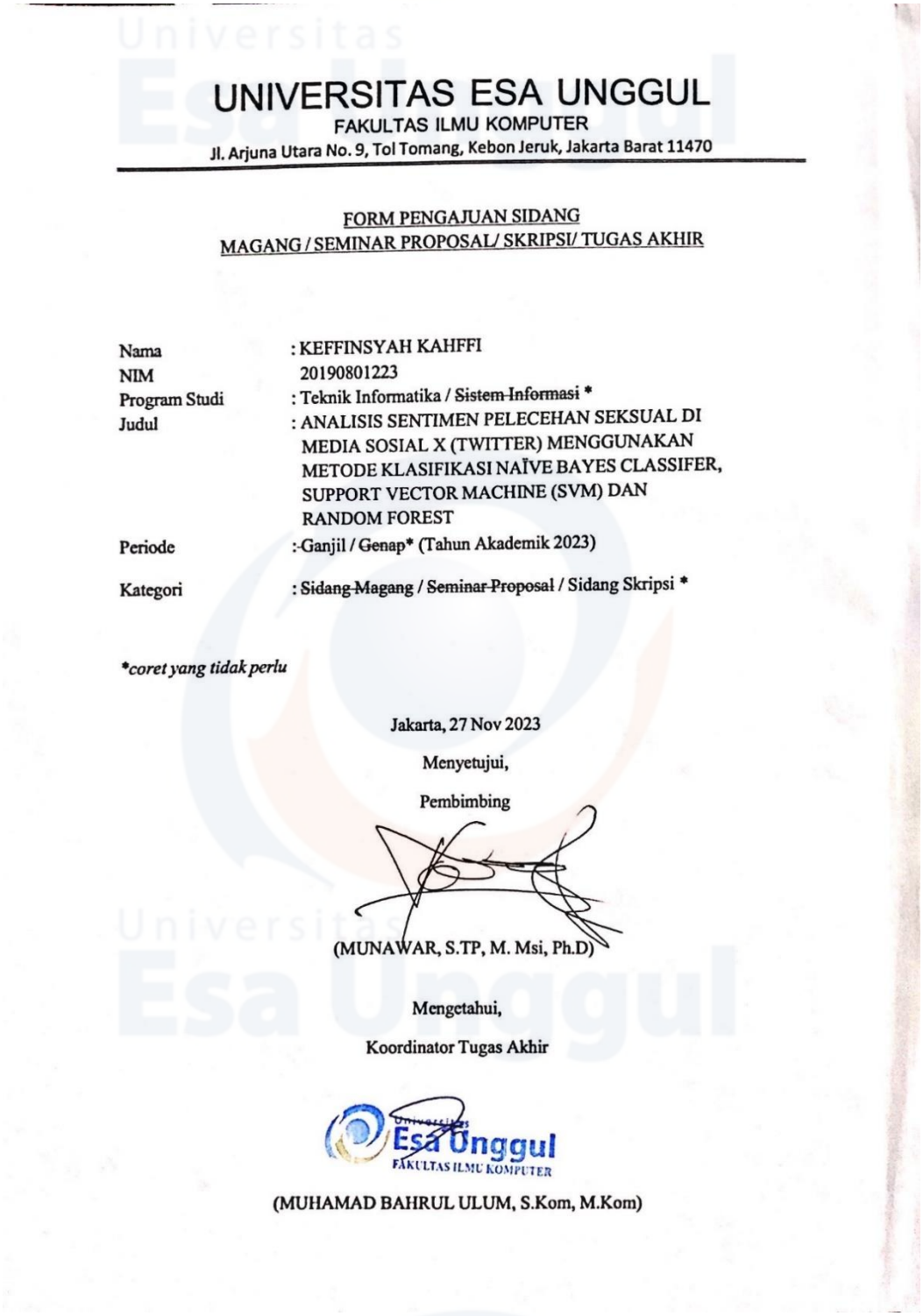
Lampiran 8

Bimbingan

No	Dosen	Topik	Tanggal Bimbingan	Jenis Bimbingan	Catatan Perbaikan
1	5709 - MUNAWAR , S.TP, MM, Ph.D.	Pada tanggal 10 januari 2023, diadakan bimbingan untuk judul proposal penelitian	5 Des 2023	Skripsi/Tesis/BusinessPlan Proposal	
2	5709 - MUNAWAR , S.TP, MM, Ph.D.	Pada 16 januari 2023, diadakan bimbingan untuk konsep proposal penelitian. Konsep proposal mencakup garis besar ide penelitian, tujuan, dan kerangka berfikir.	5 Des 2023	Skripsi/Tesis/BusinessPlan Proposal	
3	5709 - MUNAWAR , S.TP, MM, Ph.D.	Pada 23 januari 2023 bimbingan untuk Bab 1 dalam proposal penelitian	5 Des 2023	Skripsi/Tesis/BusinessPlan Proposal	
4	5709 - MUNAWAR , S.TP, MM, Ph.D.	pada 6 februari 2023, dilakukan perbaikan pada kerangka berfikir dalam proposal penelitian	5 Des 2023	Skripsi/Tesis/BusinessPlan Proposal	
5	5709 - MUNAWAR , S.TP, MM, Ph.D.	Pada 13 februari 2023, dilakukan penambahan 2 algoritma baru sehingga totalnya menjadi 3 algoritma untuk dibandingkan dalam proposal penelitian	5 Des 2023	Skripsi/Tesis/BusinessPlan Proposal	
6	5709 - MUNAWAR , S.TP, MM, Ph.D.	Pada 20 februari 2023, diadakan bimbingan untuk Bab 1, Bab 2, Bab 3 dalam proposal penelitian	5 Des 2023	Skripsi/Tesis/BusinessPlan Proposal	
7	5709 - MUNAWAR , S.TP, MM, Ph.D.	Pada 27 februari 2023, diadakan bimbingan untuk membahas revisi dari Bab 1,2,dan 3	5 Des 2023	Skripsi/Tesis/BusinessPlan Proposal	
8	5709 - MUNAWAR , S.TP, MM, Ph.D.	Pada 6 maret 2023, diadakan pertemuan untuk arahan terkait revisi Bab 1,2,dan 3	5 Des 2023	Skripsi/Tesis/BusinessPlan Proposal	
9	5709 - MUNAWAR , S.TP, MM, Ph.D.	Pada 25 juni 2023 dilakukan penandatanganan untuk sidang seminar proposal	5 Des 2023	Skripsi/Tesis/BusinessPlan Proposal	
10	5709 - MUNAWAR , S.TP, MM, Ph.D.	Pada 6 november 2023 diadakan sesi bimbingan untuk membahas isi dari Bab 4 dan Bab 5	5 Des 2023	Skripsi/Tesis/BusinessPlan Proposal	
11	5709 - MUNAWAR , S.TP, MM, Ph.D.	Pada 13 november 2023 diadakan pertemuan bimbingan untuk membahas revisi isi dari Bab 4 dan Bab 5	5 Des 2023	Skripsi/Tesis/BusinessPlan Proposal	
12	5709 - MUNAWAR , S.TP, MM, Ph.D.	Pada 20 november 2023 diadakan pertemuan bimbingan untuk membahas revisi Bab 4 dan Bab 5. Dan dilakukan tanda tangan untuk daftar sidang tugas akhir	5 Des 2023	Skripsi/Tesis/BusinessPlan Proposal	

Lampiran 8 Daftar Bimbingan

Lampiran 9



UNIVERSITAS ESA UNGGUL
FAKULTAS ILMU KOMPUTER
Jl. Arjuna Utara No. 9, Tol Tomang, Kebon Jeruk, Jakarta Barat 11470

FORM PENGAJUAN SIDANG
MAGANG / SEMINAR PROPOSAL/ SKRIPSI/ TUGAS AKHIR

Nama : KEFFINSYAH KAHFFI
NIM 20190801223
Program Studi : Teknik Informatika / ~~Sistem Informasi~~ *
Judul : ANALISIS SENTIMEN PELECEHAN SEKSUAL DI MEDIA SOSIAL X (TWITTER) MENGGUNAKAN METODE KLASIFIKASI NAÏVE BAYES CLASSIFER, SUPPORT VECTOR MACHINE (SVM) DAN RANDOM FOREST
Periode : ~~Ganjil / Genap~~* (Tahun Akademik 2023)
Kategori : ~~Sidang Magang~~ / ~~Seminar Proposal~~ / Sidang Skripsi *

**coret yang tidak perlu*

Jakarta, 27 Nov 2023

Menyetujui,

Pembimbing

(MUNAWAR, S.TP, M. Msi, Ph.D)

Mengetahui,

Koordinator Tugas Akhir

Universitas Esa Unggul
FAKULTAS ILMU KOMPUTER

(MUHAMAD BAHRUL ULUM, S.Kom, M.Kom)

Lampiran 9 Lembar Pengajuan Sidang