LAMPIRAN

Lampiran 1

```
[ ] def clean_text(text):

        text = re.sub(r"([@#][A-Za-z0-9]+)|(\w+:\/\/\S+)", " ", text)
        text = re.sub(r'@[A-Za-z0-9]+', '', text) # remove mentions
        text = re.sub(r'#[A-Za-z0-9]+', '', text) # remove hashtag
        text = re.sub(r'RT[\s]', '', text) # remove RT
        text = re.sub(r"http\S+", '', text) # remove link
        text = re.sub(r'[0-9]+', '', text) # remove numbers

        text = text.replace('\n', ' ') # replace new line into space
        text = text.translate(str.maketrans('', '', string.punctuation)) # remove all punctuations
        text = text.strip(' ') # remove characters space from both left and right text
        text = text.lower()
        return text.replace("http://", " ").replace("https://", " ")
        data.drop_duplicates(subset=['tweet'], inplace=True)
        return text

    tweet_cyberbully['clean'] = [clean_text(i) if isinstance(i, str) else "" for i in tweet]
    tweet = tweet_cyberbully['clean']
```

```
[ ] from nltk.tokenize import regexp_tokenize
    regexp = r'\w+|\$[\d.]+|\S+'
    tweet_cyberbully['Token'] = tweet_cyberbully['clean'].apply(lambda text: regexp_tokenize(text, regexp))

    tweet_cyberbully
```

	Date	Tweet	username	tweet_url	clean	Token
0	Tue Sep 26 03:30:03 +0000 2023	Dewasa ini bullying atau perundungan tak hanya...	rumahzakat	https://twitter.com/rumahzakat/status/17065114...	dewasa ini bullying atau perundungan tak hanya...	[dewasa, ini, bullying, atau, perundungan, tak...
1	Fri Sep 08 10:44:45 +0000 2023	Perundungan atau bullying adalah perilaku tida...	Diah_nvt	https://twitter.com/Diah_nvt/status/1700097837...	perundungan atau bullying adalah perilaku tida...	[perundungan, atau, bullying, adalah, perilaku...
2	Tue Sep 05 06:52:31 +0000 2023	@seikhamey @Nabilataqiyyah_ Banyak yg relate y...	Lavyna_	https://twitter.com/Lavyna_/status/16989522293...	banyak yg relate ya sebenernya di masa banyak ...	[banyak, yg, relate, ya, sebenernya, di, masa,...
3	Fri Aug 25 21:10:00 +0000 2023	Jenis-Jenis Perundungan Dunia Maya https://t.c...	okezonenews	https://twitter.com/okezonenews/status/1695181...	jenisjenis perundungan dunia maya	[jenisjenis, perundungan, dunia, maya]

```
[ ] import nltk
    from nltk.corpus import stopwords
    nltk.download("stopwords")
    stopwords = stopwords.words('indonesian')
    txt_stopwords = pd.read_csv('/content/sample_data/stopwords.txt', names=["stopwords"], header=None)

    additional_stopwords = ["yg", "dg", "rt", "dgn", "ny", "d", 'klo', 'kalo', 'amp', 'biar', 'bikin', 'bilang',
                            'gak', 'ga', 'krn', 'nya', 'nih', 'sih', 'si', 'tau', 'tdk', 'tuh', 'utk', 'ya',
                            'jd', 'jgn', 'sdh', 'aja', 'n', 't', 'nyg', 'hehe', 'pen', 'u', 'nan', 'loh', 'rt',
                            '&amp', 'yah', "stlh", "xf0", "x9f", "xa4", "xa4", "part", "sih", "deact", "rt", "pls",
                            "cut", "off", "xf0", 'x9f', 'x99', 'x8f', 'bentar', 'sd', 'kahdhewjbswkwishshw', '1', 'pasc
                            'ahhhh', 't', 'ya', 'walach', 'abiez', 'time', 'true', "nya", 'wkwkw', 'happy', 'ending', 'y
                            'should', 'know', 'how', 'much', 'i', 'love', 'you', 'my', 'baby', 'my', 'proud', 'boy', 'it
                            'yuuji', 'what', 's', 'wrong', 'with', 'me', '9', '10', 'y', '2', '2ny', 'triggered', 'co',
                            'bunuh', 'alesan', 'urg', 'kamamana', 'mawa', 'peso', 'karna', 'neangan', 'psnskd', 'wkzn',
                            'fc', 'royal', 'gk', 'holaaaa', 'wkwk', 'anjg', 'goblok2in', 's3lf', 'h4rm', 'nk', 'gi', 'i
                            'jomblo', 'sa', 'tai', 'oink', 'anj', 'pasca', 'konser', 'hrgaimki', 'kontol', 'bajingan',
                            'is', 'trying', 'cloe', 'ainkkkkkkkkk', 'produce', 'bitch', 'encore', 'tds', 'clash', 'royal
                            'anjgggg', 'panal', 'kpop', 'xe2', 'x80', 'xa6', 'n', 'nhawe', 'play', 'dll', 'doi', 'bnr',
                            'ipuel', 'etmin', 'waiji', 'bola', 'kimak', 'fyp', 'angst', 'tolol', 'jkt48', 'buddies', 'h
                            'kemenkes', '1500', '567', 'marshanda', 'selesaikan', 'charger', 'ipad', 'ipadnya', 'gem',
                            'x98', 'x8c', 'xa5', 'xb2', 'cos', 'jakarta', 'beli', 'maskot', 'lucu', 'kim', 'junmyeon',
                            'misstreatment', 'im', 'so', 'of', 'him', 'fucked', 'up', 'solo', 'netflixnya', 'sok', 'an',
                            'anying', 'cok', 'ajg', 'anjg', 'anjng', 'ace', 'fr', 'aing', 'ajg', 'kontol', 'omg', 'ygy',
                            'asfdgsh', 'haechan', 'anjrit', 'akut', 'awikwok', 'lol', 'setan', 'asu', 'dll', 'co', 'ajg
                            'tai', 'wkwkwk', 'sayankkk', 'anhing', 'bacot', 'hts', 'hrgaimki', 'cegil', 'cogil', 'e', 'a
                            'piye', 'maneh', 'nyet', 'bngst', 'pens', 'ipuel', 'kimak', 'fyp', 'kareshisugiru', 'rollerc
                            'xd', 'rengginang', 'dragon', 'ball', 'bola', 'ajggggg', 'anyinggg', 'idol', 'yysf', 'qrt',
                            'goblok', 'apple', 'tasikmalaya', 'lo', 'yg', 'main', 'qrt', 'oliver', 'xf', 'xa', 'xe', 'xb
                            'bod', 'humpubnnya', 'xc', 'it', 'not', 'dr', 'jiemi', 'ardi', 'sp', 'kj', 'tw', 'ket', 'an
                            'jm', 'd', 'm', 'ah', 'mo', 'alywafa', 'dot', 'com', 'b', 'ra', 'bombardir', 'ken', 'adhd',
                            'th', 'sm', 'aaaaaa', 'tu', 'hadeu', 'ama', 'ni', 'kekwkw', 'tibab', 'taraaaaa', 'ekwkwkwk',
```

```
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_cyberbully['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])

print(term_dict)
print("------------------------")
```

```
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_cyberbully['Normalisasi'] = tweet_cyberbully['Token'].apply(normalized_term)

tweet_cyberbully['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)
```

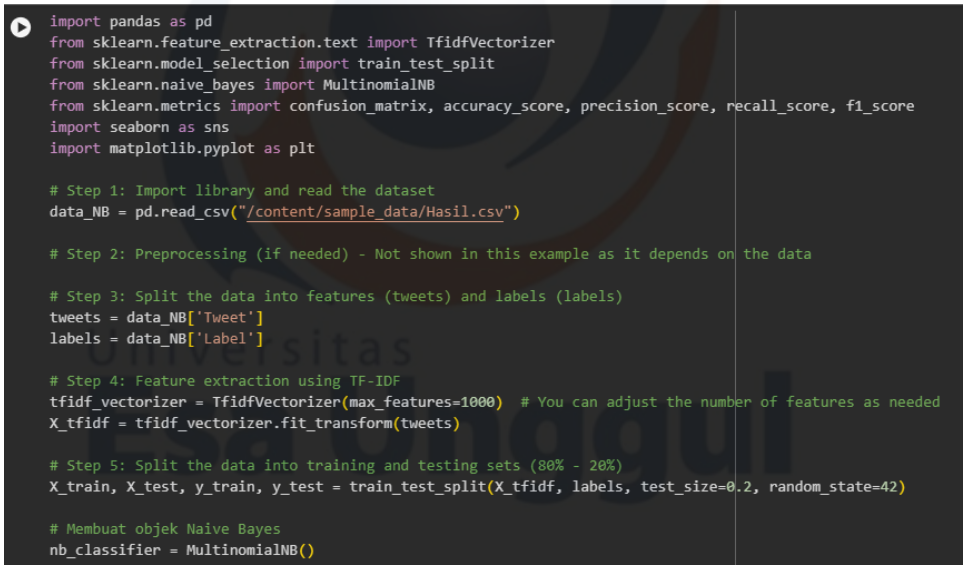
Lampiran 2 Pelabelan

Lampiran 3

```python
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

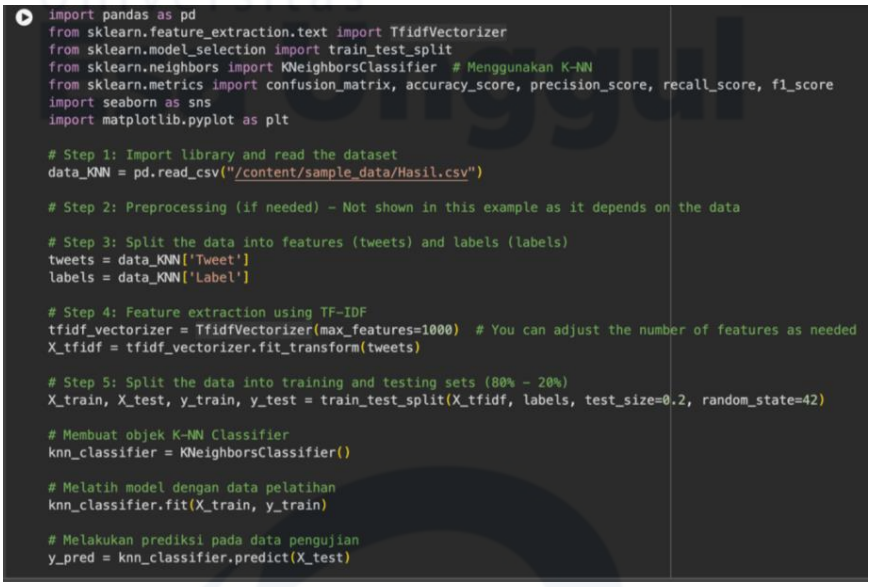
Lampiran 4



```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 Classifier

Lampiran 5



```python
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier  # Menggunakan K-NN
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_KNN = 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_KNN['Tweet']
labels = data_KNN['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 K-NN Classifier
knn_classifier = KNeighborsClassifier()

# Melatih model dengan data pelatihan
knn_classifier.fit(X_train, y_train)

# Melakukan prediksi pada data pengujian
y_pred = knn_classifier.predict(X_test)
```

Lampiran 5 Algoritma K-Nearest Neighbor

Lampiran 6

```python
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier  # Menggunakan Decision Tree
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_DT = 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_DT['Tweet']
labels = data_DT['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=0.3, random_state=42)

# Membuat objek Decision Tree Classifier
dt_classifier = DecisionTreeClassifier(random_state=42)

# Melatih model dengan data pelatihan
dt_classifier.fit(X_train, y_train)

# Melakukan prediksi pada data pengujian
y_pred = dt_classifier.predict(X_test)
```

Lampiran 6 Algoritma Decision Tree

Lampiran 7

Bimbingan

No	Dosen	Topik	Tanggal Bimbingan	Jenis Bimbingan	Catatan Perbaikan
1	5709 - MUNAWAR , S.TP, MM, Ph.D.	9 Januari 2023, diadakan bimbingan untuk judul proposal penelitian	6 Des 2023	Skripsi/Tesis/BusinessPlan Proposal	
2	5709 - MUNAWAR , S.TP, MM, Ph.D.	pada 16 Januari 2023, diadakan bimbingan konsep proposal penelitian. yang mencakup garis besar ide penelitian, tujuan, dan kerangka berfikir	6 Des 2023	Skripsi/Tesis/BusinessPlan Proposal	
3	5709 - MUNAWAR , S.TP, MM, Ph.D.	pada 23 Januari 2023, bimbingan untuk Bab 1 Proposal Penelitian	6 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	6 Des 2023	Skripsi/Tesis/BusinessPlan Proposal	
5	5709 - MUNAWAR , S.TP, MM, Ph.D.	pada 13 februari 2023, dilakukan penambahan 2 algoritma baru untuk dibandingkan dalam proposal penelitian	6 Des 2023	Skripsi/Tesis/BusinessPlan Proposal	
6	5709 - MUNAWAR , S.TP, MM, Ph.D.	pada 20 Februari 2023, bimbingan untuk bab 1 sampai dengan bab 3 dalam proposal penelitian	6 Des 2023	Skripsi/Tesis/BusinessPlan Proposal	
7	5709 - MUNAWAR , S.TP, MM, Ph.D.	pada 27 Februari 2023, diadakan bimbingan revisi bab 1 sampai dengan bab 3	6 Des 2023	Skripsi/Tesis/BusinessPlan Proposal	
8	5709 - MUNAWAR , S.TP, MM, Ph.D.	pada 13 maret 2023, diadakan pertemuan untuk arahan terkait revisi bab 1 sampai dengan bab 3	6 Des 2023	Skripsi/Tesis/BusinessPlan Proposal	
9	5709 - MUNAWAR , S.TP, MM, Ph.D.	pada 25 juni 2023, dilakukan penandatanganan untuk sidang Seminar Proposal	6 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 5	6 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 5	6 Des 2023	Skripsi/Tesis/BusinessPlan Proposal	
12	5709 - MUNAWAR , S.TP, MM, Ph.D.	pada 27 November 2023 diadakan pertemuan bimbingan untuk membahas revisi bab 4 dan 5, serta dilakukan tanda tangan untuk daftar sidang TA oleh dosen pembimbing	6 Des 2023	Skripsi/Tesis/BusinessPlan Proposal	

Lampiran 7 Daftar Bimbingan

Lampiran 8

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	: MUHAMMAD FARRAS HEYKAL MOUSTHAFA
NIM	20190801250
Program Studi	: Teknik Informatika / ~~Sistem Informasi~~ *
Judul	: ANALISIS SENTIMEN CYBERBULLYING DI MEDIA SOSIAL X (TWITTER) MENGGUNAKAN METODE KLASIFIKASI NAÏVE BAYES CLASSIFER (NBC), K-NEAREST NEIGHBOR (KNN), DAN DECISION TREE CLASSIFIER.
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

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

Lampiran 8 Lembar Pengajuan Sidang