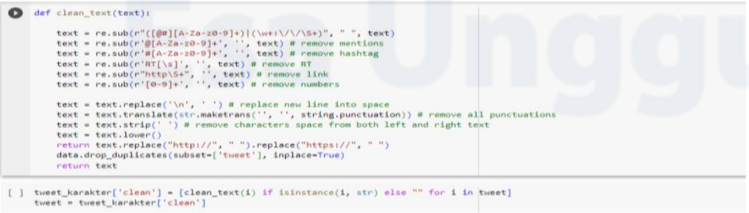
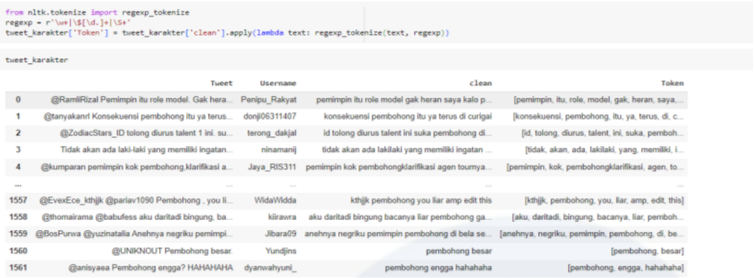
LAMPIRAN

Lampiran 1



```python
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
```

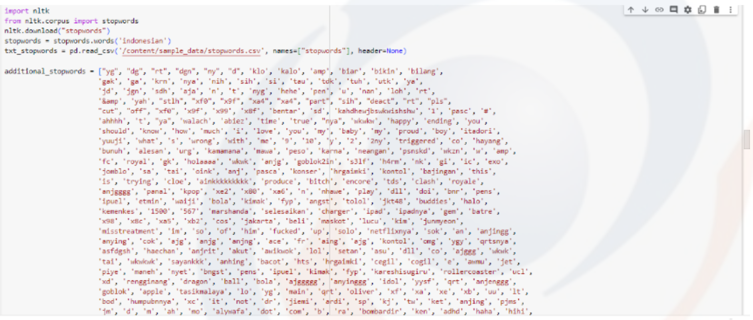
```python
[ ] tweet_karakter['clean'] = [clean_text(i) if isinstance(i, str) else "" for i in tweet]
    tweet = tweet_karakter['clean']
```



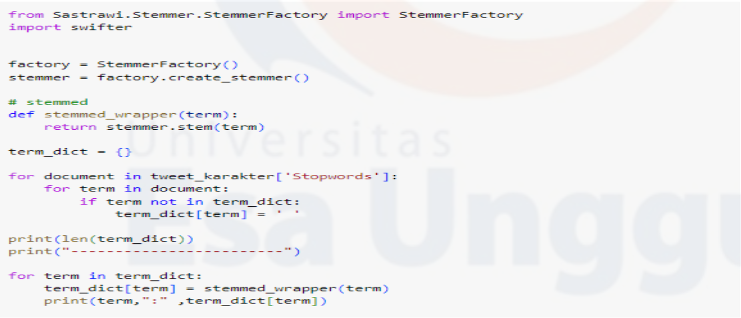
```python
from nltk.tokenize import regexp_tokenize
regexp = r'\w+|\$[\d.]+|\S+'
tweet_karakter['Token'] = tweet_karakter['clean'].apply(lambda text: regexp_tokenize(text, regexp))
```

```python
tweet_karakter
```

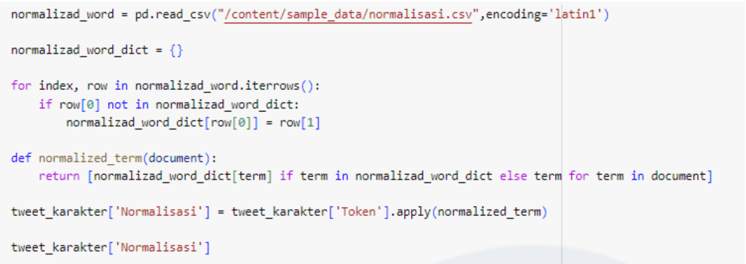
	Tweet	Username	clean	Token
0	@RamliRizal Pemimpin itu role model. Gak hera...	Penipu_Rakyat	pemimpin itu role model gak heran saya kalo p...	[pemimpin, itu, role, model, gak, heran, saya,...
1	@tanyakanrl Konsekuensi pembohong itu ya terus...	donji06311407	konsekuensi pembohong itu ya terus di curigai	[konsekuensi, pembohong, itu, ya, terus, di, c...
2	@ZodiacStars_ID tolong diurus talent 1 ini. su...	terong_dakjal	id tolong diurus talent ini suka pembohong di...	[id, tolong, diurus, talent, ini, suka, pemboh...
3	Tidak akan ada laki-laki yang memiliki ingatan...	ninamanij	tidak akan ada lakilaki yang memiliki ingatan ...	[tidak, akan, ada, lakilaki, yang, memiliki, i...
4	@kumparan pemimpin kok pembohong,klarifikasi a...	Jaya_RIS311	pemimpin kok pembohongklarifikasi agen tournya...	[pemimpin, kok, pembohongklarifikasi, agen, to...
...	...	...	...	...
1557	@EvexEce_kthjjk @pariav1090 Pembohong , you li...	WidaWidda	kthjjk pembohong you liar amp edit this	[kthjjk, pembohong, you, liar, amp, edit, this]
1558	@thomairama @babufess aku daritadi bingung, ba...	kiirawra	aku daritadi bingung bacanya liar pembohong ga...	[aku, daritadi, bingung, bacanya, liar, pemboh...
1559	@BosPurwa @yuzinatalia Anehnya negriku pemimpi...	Jibara09	anehnya negriku pemimpin pembohong di bela se...	[anehnya, negriku, pemimpin, pembohong, di, be...
1560	@UNIKNOUT Pembohong besar.	Yundjins	pembohong besar	[pembohong, besar]
1561	@anisyaea Pembohong engga? HAHAHAHA	dyanwahyuni_	pembohong engga hahahaha	[pembohong, engga, hahahaha]



```python
import nltk
from nltk.corpus import stopwords
nltk.download("stopwords")
stopwords = stopwords.words('indonesian')
txt_stopwords = pd.read_csv('/content/sample_data/stopwords.csv', 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', 'you',
                        'should', 'know', 'how', 'much', 'i', 'love', 'you', 'my', 'baby', 'my', 'proud', 'boy', 'itadori',
                        'yuuji', 'what', 's', 'wrong', 'with', 'me', '9', '10', 'y', '2', '2ny', 'triggered', 'co', 'hayang',
                        'bunuh', 'alesan', 'urg', 'kamamana', 'mawa', 'peso', 'karna', 'neangan', 'psnskd', 'wkzn', 'w', 'amp',
                        'fc', 'royal', 'gk', 'holaaaa', 'wkwk', 'anjg', 'goblok2in', 's3lf', 'h4rm', 'nk', 'gi', 'ic', 'exo',
                        'jomblo', 'sa', 'tai', 'oink', 'anj', 'pasca', 'konser', 'hrgaimki', 'kontol', 'bajingan', 'this',
                        'is', 'trying', 'cloe', 'ainkkkkkkkkk', 'produce', 'bitch', 'encore', 'tds', 'clash', 'royale',
                        'anjgggg', 'panal', 'kpop', 'xe2', 'x80', 'xa6', 'n', 'nhawe', 'play', 'dll', 'doi', 'bnr', 'pens',
                        'ipuel', 'etmin', 'waiji', 'bola', 'kimak', 'fyp', 'angst', 'tolol', 'jkt48', 'buddies', 'halo',
                        'kemenkes', '1500', '567', 'marshanda', 'selesaikan', 'charger', 'ipad', 'ipadnya', 'gem', 'batre',
                        'x98', 'x8c', 'xa5', 'xb2', 'cos', 'jakarta', 'beli', 'maskot', 'lucu', 'kim', 'junmyeon',
                        'misstreatment', 'im', 'so', 'of', 'him', 'fucked', 'up', 'solo', 'netflixnya', 'sok', 'an', 'anjingg',
                        'anying', 'cok', 'ajg', 'anjg', 'anjng', 'ace', 'fr', 'aing', 'ajg', 'kontol', 'omg', 'ygy', 'qrtsnya',
                        'asfdgsh', 'haechan', 'anjrit', 'akut', 'awikwok', 'lol', 'setan', 'asu', 'dll', 'co', 'ajggg', 'wkwk',
                        'tai', 'wkwkwk', 'sayankkk', 'anhing', 'bacot', 'hts', 'hrgaimki', 'cegil', 'cogil', 'e', 'awmu', 'jet',
                        'piye', 'maneh', 'nyet', 'bngst', 'pens', 'ipuel', 'kimak', 'fyp', 'kareshisugiru', 'rollercoaster', 'ucl',
                        'xd', 'rengginang', 'dragon', 'ball', 'bola', 'ajggggg', 'anyinggg', 'idol', 'yysf', 'qrt', 'anjenggg',
                        'goblok', 'apple', 'tasikmalaya', 'lo', 'yg', 'main', 'qrt', 'oliver', 'xf', 'xa', 'xe', 'xb', 'uu', 'lt',
                        'bod', 'humpubnnya', 'xc', 'it', 'not', 'dr', 'jiemi', 'ardi', 'sp', 'kj', 'tw', 'ket', 'anjing', 'pjms',
                        'jm', 'd', 'm', 'ah', 'mo', 'alywafa', 'dot', 'com', 'b', 'ra', 'bombardir', 'ken', 'adhd', 'haha', 'hihi',
```



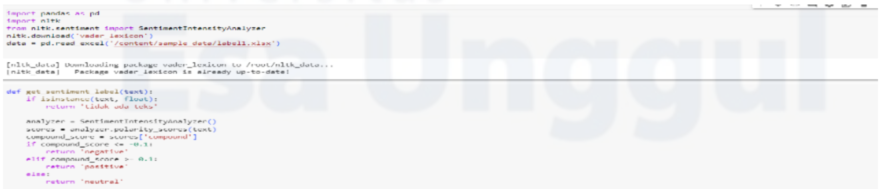
```python
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import swifter


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

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

term_dict = {}

for document in tweet_karakter['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])
```



```python
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_karakter['Normalisasi'] = tweet_karakter['Token'].apply(normalized_term)

tweet_karakter['Normalisasi']
```

Lampiran 1 Preprocessing

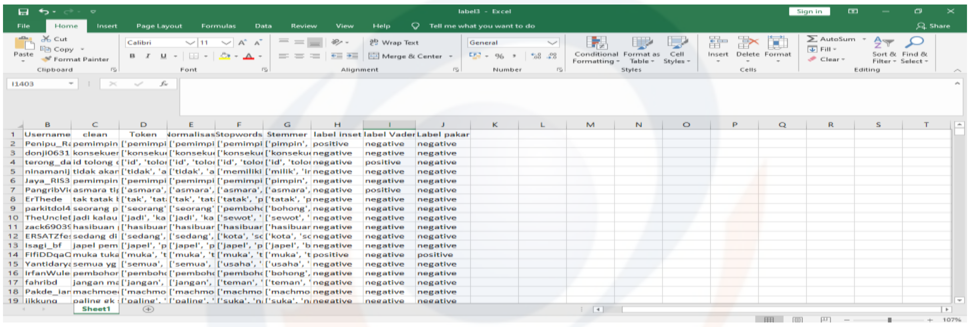
Lampiran 2



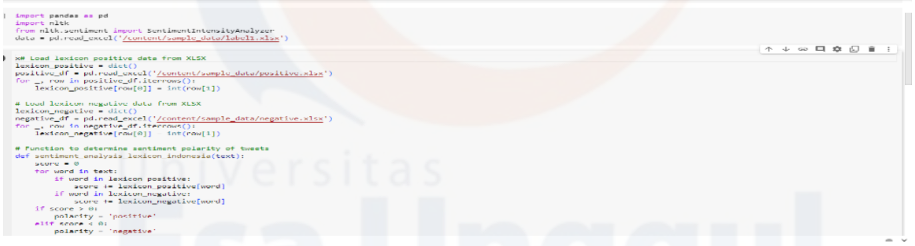
```
import pandas as pd
import nltk
from nltk.sentiment import SentimentIntensityAnalyzer
nltk.download('vader_lexicon')
data = pd.read_excel('/content/sample_data/label1.xlsx')
```

```
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
```

```
def get_sentiment_label(text):
    if isinstance(text, float):
        return 'tidak ada teks'

    analyzer = SentimentIntensityAnalyzer()
    scores = analyzer.polarity_scores(text)
    compound_score = scores['compound']
    if compound_score <= -0.1:
        return 'negative'
    elif compound_score >= 0.1:
        return 'positive'
    else:
        return 'neutral'
```



	B	C	D	E	F	G	H	I	J
1	Username	clean	Token	Normalisas	Stopwords	Stemmer	label inset	label Vader	Label pakar
2	Penipu_Ra	pemimpin	['pemimpi	['pemimpi	['pemimpi	['pimpin',	positive	negative	negative
3	donji0631	konsekuer	['konsekue	['konsekue	['konsekue	['konsekue	negative	negative	negative
4	terong_da	id tolong c	['id', 'tolor	['id', 'tolor	['id', 'tolor	['id', 'tolor	negative	positive	negative
5	ninamanij	tidak akan	['tidak', 'a	['tidak', 'a	['memiliki	['milik', 'ir	negative	negative	negative
6	Jaya_RIS3	pemimpin	['pemimpi	['pemimpi	['pemimpi	['pimpin',	negative	negative	negative
7	PangribVi	asmara tip	['asmara',	['asmara',	['asmara',	['asmara',	negative	positive	negative
8	ErThede	tak tatak k	['tak', 'tat	['tak', 'tat	['tatak', 'p	['tatak', 'p	negative	negative	negative
9	parkitdol4	seorang p	['seorang'	['seorang'	['pembohc	['bohong',	negative	negative	negative
10	TheUnclet	jadi kalau	['jadi', 'ka	['jadi', 'ka	['sewot', '	['sewot', '	negative	negative	negative
11	zack69039	hasibuan	['hasibuar	['hasibuar	['hasibuar	['hasibuar	negative	negative	negative
12	ERSATZfe	sedang di	['sedang',	['sedang',	['kota', 'sc	['kota', 'sc	negative	negative	negative
13	Isagi_bf	japel pem	['japel', 'p	['japel', 'p	['japel', 'p	['japel', 'b	negative	negative	negative
14	FIfiDDqaC	muka tuka	['muka', 't	['muka', 't	['muka', 't	['muka', 't	positive	negative	positive
15	Yantidary	semua yg	['semua',	['semua',	['usaha', '	['usaha', '	negative	negative	negative
16	IrfanWule	pembohor	['pembohc	['pembohc	['pembohc	['bohong',	negative	negative	negative
17	fahribd	jangan ma	['jangan',	['jangan',	['teman', '	['teman', '	negative	negative	negative
18	Pakde_Iar	machmoe	['machmo	['machmo	['machmo	['machmo	negative	negative	negative
19	jikkung	paling ek	['paling', '	['paling', '	['suka', 'n	['suka', 'n	negative	negative	negative

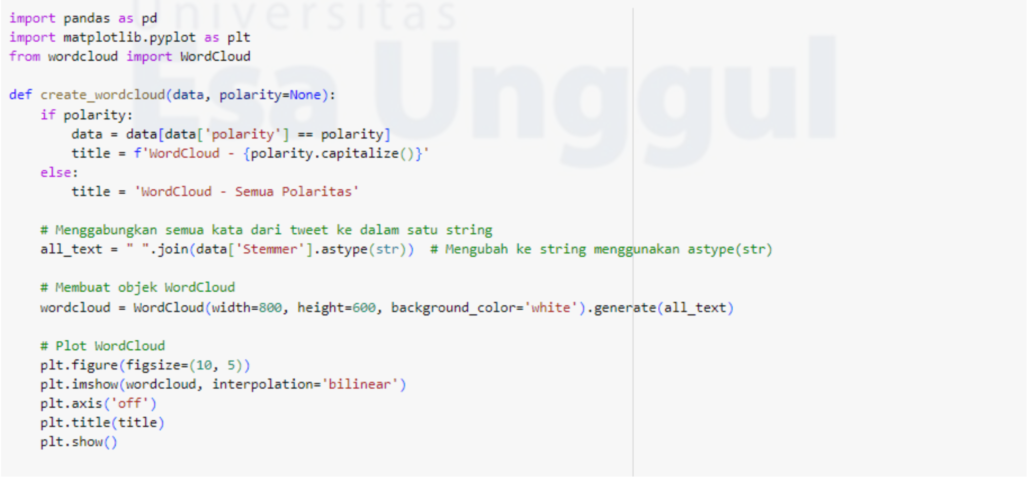


```
import pandas as pd
import nltk
from nltk.sentiment import SentimentIntensityAnalyzer
data = pd.read_excel('/content/sample_data/label1.xlsx')
```

```
x# Load lexicon positive data from XLSX
lexicon_positive = dict()
positive_df = pd.read_excel('/content/sample_data/positive.xlsx')
for _, row in positive_df.iterrows():
    lexicon_positive[row[0]] = int(row[1])

# Load lexicon negative data from XLSX
lexicon_negative = dict()
negative_df = pd.read_excel('/content/sample_data/negative.xlsx')
for _, row in negative_df.iterrows():
    lexicon_negative[row[0]] = int(row[1])

# Function to determine sentiment polarity of tweets
def sentiment_analysis_lexicon_indonesia(text):
    score = 0
    for word in text:
        if word in lexicon_positive:
            score += lexicon_positive[word]
        if word in lexicon_negative:
            score += lexicon_negative[word]
    if score > 0:
        polarity = 'positive'
    elif score < 0:
        polarity = 'negative'
```

Lampiran 2 Pelabelan

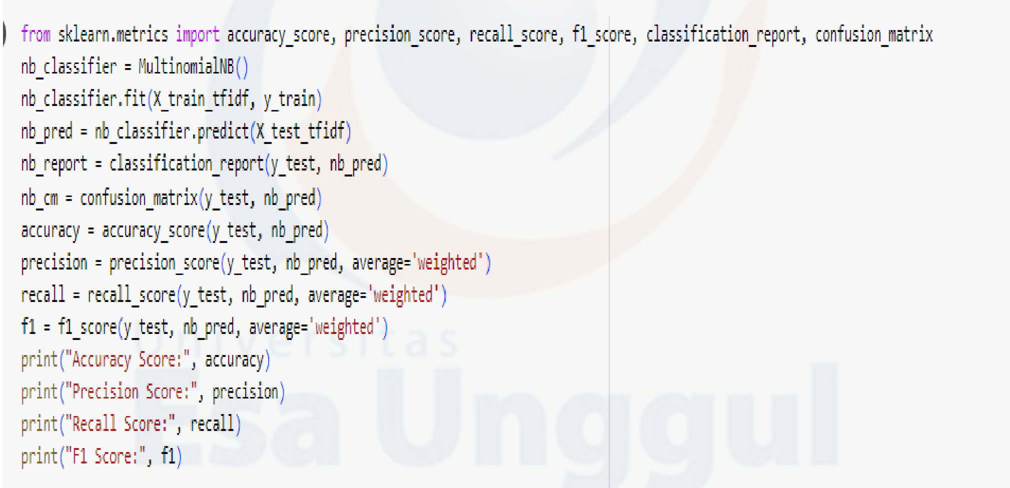
Lampiran 3



```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['Stemmer'].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

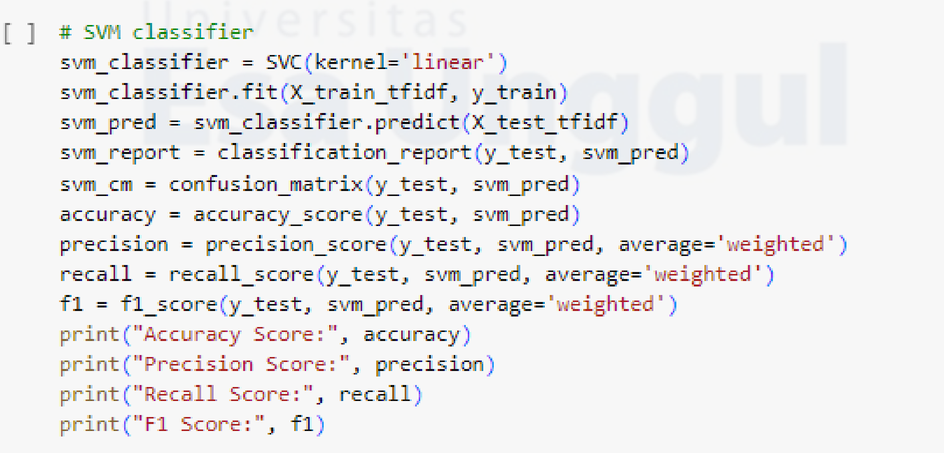
Lampiran 4



```python
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, classification_report, confusion_matrix
nb_classifier = MultinomialNB()
nb_classifier.fit(X_train_tfidf, y_train)
nb_pred = nb_classifier.predict(X_test_tfidf)
nb_report = classification_report(y_test, nb_pred)
nb_cm = confusion_matrix(y_test, nb_pred)
accuracy = accuracy_score(y_test, nb_pred)
precision = precision_score(y_test, nb_pred, average='weighted')
recall = recall_score(y_test, nb_pred, average='weighted')
f1 = f1_score(y_test, nb_pred, average='weighted')
print("Accuracy Score:", accuracy)
print("Precision Score:", precision)
print("Recall Score:", recall)
print("F1 Score:", f1)
```

Lampiran 4 Algoritma Naive Bayes

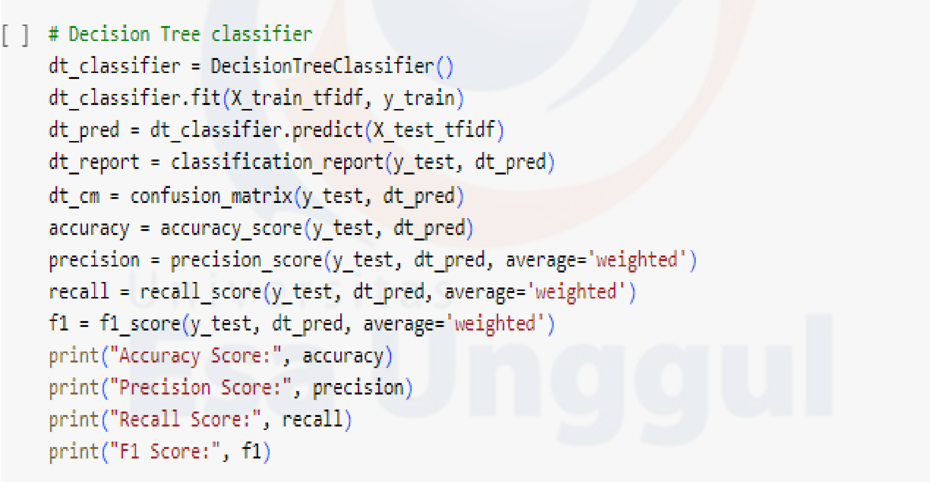
Lampiran 5



```
[ ]  # SVM classifier
     svm_classifier = SVC(kernel='linear')
     svm_classifier.fit(X_train_tfidf, y_train)
     svm_pred = svm_classifier.predict(X_test_tfidf)
     svm_report = classification_report(y_test, svm_pred)
     svm_cm = confusion_matrix(y_test, svm_pred)
     accuracy = accuracy_score(y_test, svm_pred)
     precision = precision_score(y_test, svm_pred, average='weighted')
     recall = recall_score(y_test, svm_pred, average='weighted')
     f1 = f1_score(y_test, svm_pred, average='weighted')
     print("Accuracy Score:", accuracy)
     print("Precision Score:", precision)
     print("Recall Score:", recall)
     print("F1 Score:", f1)
```

Lampiran 5 Algoritma Support Vector Machine

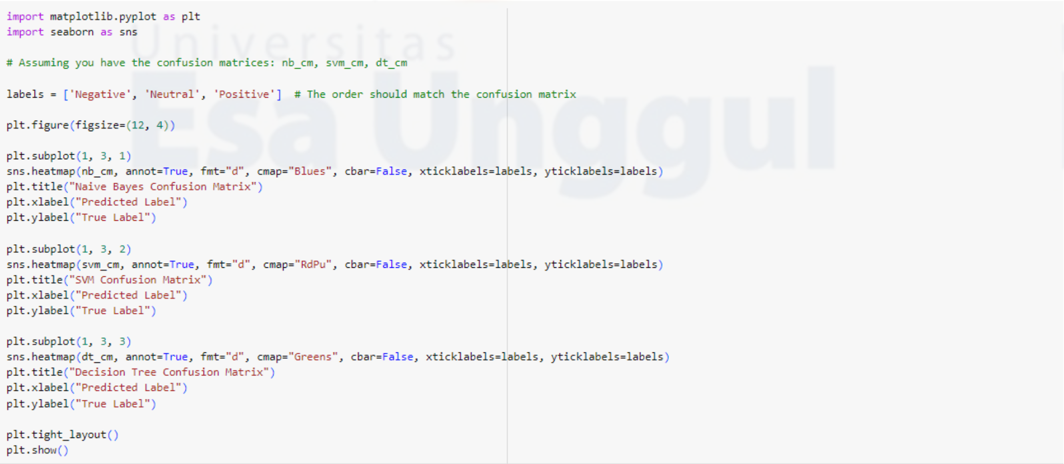
Lampiran 6



```
[ ]  # Decision Tree classifier
     dt_classifier = DecisionTreeClassifier()
     dt_classifier.fit(X_train_tfidf, y_train)
     dt_pred = dt_classifier.predict(X_test_tfidf)
     dt_report = classification_report(y_test, dt_pred)
     dt_cm = confusion_matrix(y_test, dt_pred)
     accuracy = accuracy_score(y_test, dt_pred)
     precision = precision_score(y_test, dt_pred, average='weighted')
     recall = recall_score(y_test, dt_pred, average='weighted')
     f1 = f1_score(y_test, dt_pred, average='weighted')
     print("Accuracy Score:", accuracy)
     print("Precision Score:", precision)
     print("Recall Score:", recall)
     print("F1 Score:", f1)
```

Lampiran 6 Algoritma Decision Tree

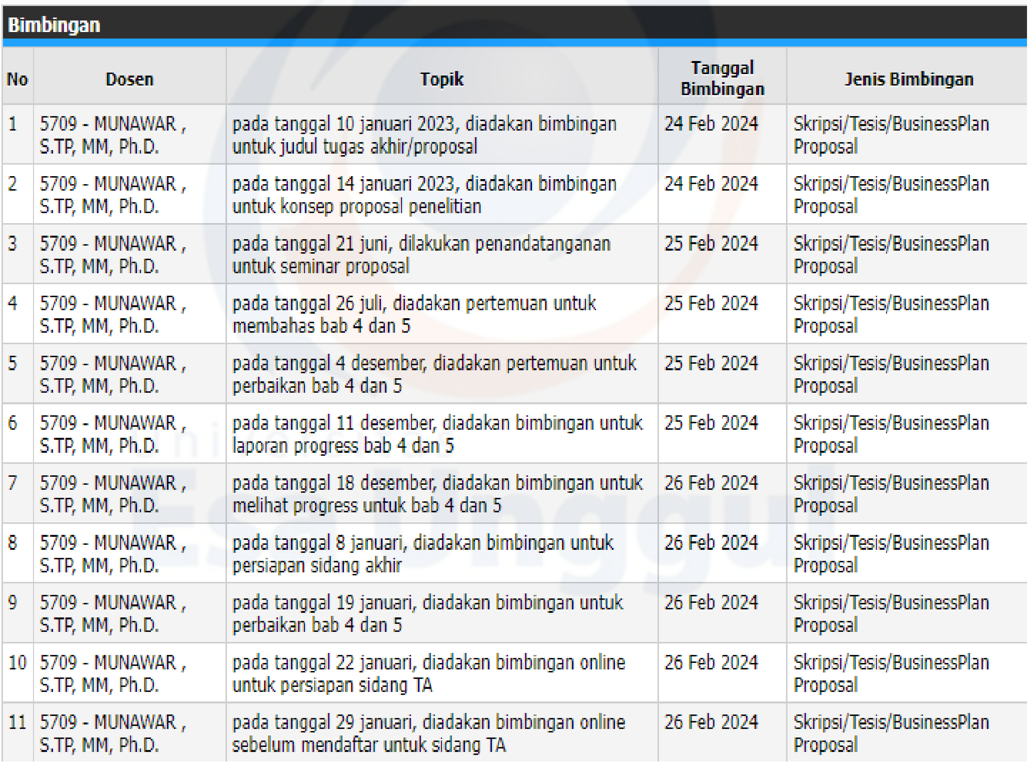
Lampiran 7



```python
import matplotlib.pyplot as plt
import seaborn as sns

# Assuming you have the confusion matrices: nb_cm, svm_cm, dt_cm

labels = ['Negative', 'Neutral', 'Positive']  # The order should match the confusion matrix

plt.figure(figsize=(12, 4))

plt.subplot(1, 3, 1)
sns.heatmap(nb_cm, annot=True, fmt="d", cmap="Blues", cbar=False, xticklabels=labels, yticklabels=labels)
plt.title("Naive Bayes Confusion Matrix")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")

plt.subplot(1, 3, 2)
sns.heatmap(svm_cm, annot=True, fmt="d", cmap="RdPu", cbar=False, xticklabels=labels, yticklabels=labels)
plt.title("SVM Confusion Matrix")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")

plt.subplot(1, 3, 3)
sns.heatmap(dt_cm, annot=True, fmt="d", cmap="Greens", cbar=False, xticklabels=labels, yticklabels=labels)
plt.title("Decision Tree Confusion Matrix")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")

plt.tight_layout()
plt.show()
```

Lampiran 7 Confusion Matrix

Lampiran 8

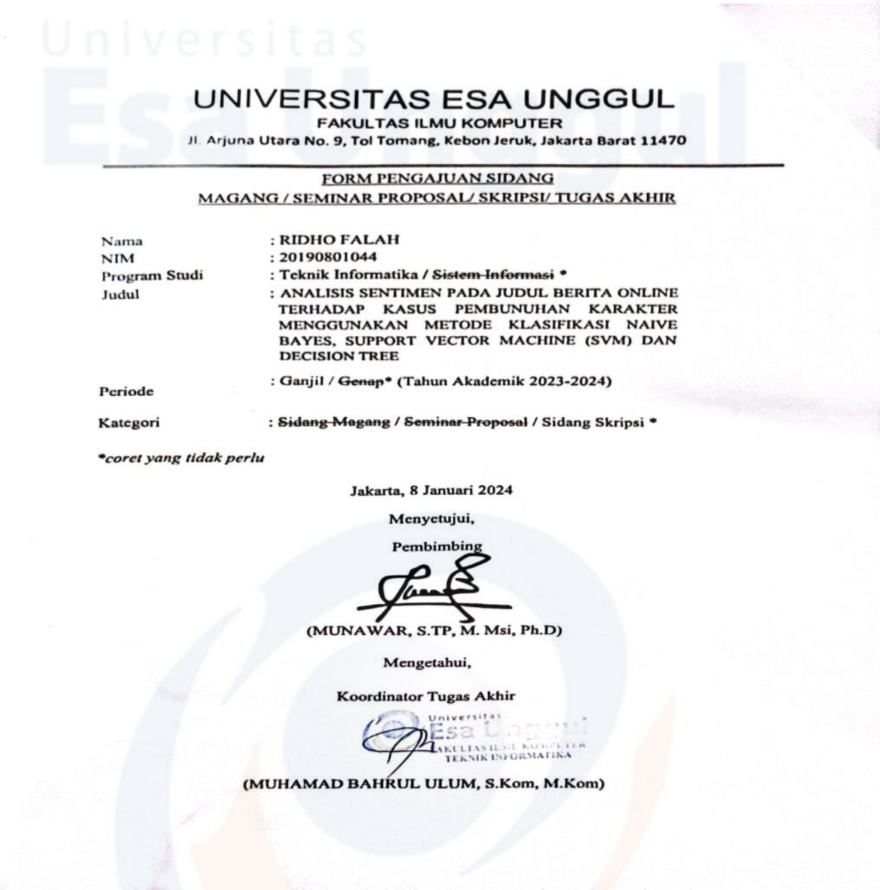


Bimbingan

No	Dosen	Topik	Tanggal Bimbingan	Jenis Bimbingan
1	5709 - MUNAWAR , S.TP, MM, Ph.D.	pada tanggal 10 januari 2023, diadakan bimbingan untuk judul tugas akhir/proposal	24 Feb 2024	Skripsi/Tesis/BusinessPlan Proposal
2	5709 - MUNAWAR , S.TP, MM, Ph.D.	pada tanggal 14 januari 2023, diadakan bimbingan untuk konsep proposal penelitian	24 Feb 2024	Skripsi/Tesis/BusinessPlan Proposal
3	5709 - MUNAWAR , S.TP, MM, Ph.D.	pada tanggal 21 juni, dilakukan penandatanganan untuk seminar proposal	25 Feb 2024	Skripsi/Tesis/BusinessPlan Proposal
4	5709 - MUNAWAR , S.TP, MM, Ph.D.	pada tanggal 26 juli, diadakan pertemuan untuk membahas bab 4 dan 5	25 Feb 2024	Skripsi/Tesis/BusinessPlan Proposal
5	5709 - MUNAWAR , S.TP, MM, Ph.D.	pada tanggal 4 desember, diadakan pertemuan untuk perbaikan bab 4 dan 5	25 Feb 2024	Skripsi/Tesis/BusinessPlan Proposal
6	5709 - MUNAWAR , S.TP, MM, Ph.D.	pada tanggal 11 desember, diadakan bimbingan untuk laporan progress bab 4 dan 5	25 Feb 2024	Skripsi/Tesis/BusinessPlan Proposal
7	5709 - MUNAWAR , S.TP, MM, Ph.D.	pada tanggal 18 desember, diadakan bimbingan untuk melihat progress untuk bab 4 dan 5	26 Feb 2024	Skripsi/Tesis/BusinessPlan Proposal
8	5709 - MUNAWAR , S.TP, MM, Ph.D.	pada tanggal 8 januari, diadakan bimbingan untuk persiapan sidang akhir	26 Feb 2024	Skripsi/Tesis/BusinessPlan Proposal
9	5709 - MUNAWAR , S.TP, MM, Ph.D.	pada tanggal 19 januari, diadakan bimbingan untuk perbaikan bab 4 dan 5	26 Feb 2024	Skripsi/Tesis/BusinessPlan Proposal
10	5709 - MUNAWAR , S.TP, MM, Ph.D.	pada tanggal 22 januari, diadakan bimbingan online untuk persiapan sidang TA	26 Feb 2024	Skripsi/Tesis/BusinessPlan Proposal
11	5709 - MUNAWAR , S.TP, MM, Ph.D.	pada tanggal 29 januari, diadakan bimbingan online sebelum mendaftar untuk sidang TA	26 Feb 2024	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 : RIDHO FALAH
NIM : 20190801044
Program Studi : Teknik Informatika / ~~Sistem Informasi~~ *
Judul : ANALISIS SENTIMEN PADA JUDUL BERITA ONLINE TERHADAP KASUS PEMBUNUHAN KARAKTER MENGGUNAKAN METODE KLASIFIKASI NAIVE BAYES, SUPPORT VECTOR MACHINE (SVM) DAN DECISION TREE
Periode : Ganjil / ~~Genap~~* (Tahun Akademik 2023-2024)
Kategori : ~~Sidang Magang~~ / ~~Seminar Proposal~~ / Sidang Skripsi *

**coret yang tidak perlu*

Jakarta, 8 Januari 2024
Menyetujui,
Pembimbing

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

Mengetahui,
Koordinator Tugas Akhir

Universitas
Esa Unggul
FAKULTAS ILMU KOMPUTER
TEKNIK INFORMATIKA

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

Lampiran 9 Lembar Pengajuan Sidang