

LAMPIRAN

Lampiran 1 – Daftar Riwayat Hidup

Data Pribadi

Nama : Mohammad Fachriza
 TTL : Jakarta, 20 Januari 2001
 Jenis Kelamin : Laki-Laki
 Agama : Islam
 Kewarganegaraan : Indonesia
 Alamat : Jl. Adhi Karya No.51 RT/RW 006.005 Kedoya Selatan
 Kebon Jeruk
 Nomor Telepon / HP : 085714002488
 E-mail : m.fachriza.20@gmail.com

Riwayat Pendidikan

Periode (Tahun)	Sekolah/Institusi	Jurusan	Jenjang Pendidikan
2007-2013	SDN 11 Pagi Joglo	-	SD
2013-2016	SMPN 206 Jakarta Barat	-	SMP
2016-2019	SMA 101 Jakarta Barat	MIPA	SMA

Lampiran 2 – Kode Untuk Preprocessing

```
In [1]: pip install -U scikit-learn
pip3 install sklearn
pip install nltk
pip install openpyxl
pip install pandas
pip install Sastrawi
import string
from sklearn.pipeline import Pipeline
import pandas as pd
import numpy as np
import re
import nltk
```

```
In [2]: data = pd.read_excel ('Data_Tweet.xlsx')
data.head(2000)
```

```
Out[2]:
```

	username	Text
0	@aetherusjk	lagi di fase frustrasi, paling bentar lagi depresi
1	@youngbutcher94	RT @karimmas_: Ini orang yang dulu bilang kasu...
2	@xicy	RT @TraseerNew: @frommureyes @tanyartifess Untung...
3	@makannyaindomle	depresi setelah belajar mitk minat??" https://...
4	@kkobobp	Akhirnya, terimakasih sudah membantu aku, kaak...
...
1497	@ayupurwatie	Hebat yahi Diluar bagaimana keadaan baby, dan ...
1498	@brmyao	@megworeng dulu kakak aku langsung jadi sih, t...
1499	@Legjester	@HeadShahark: Mau tukar poin jadi kesehatan me...
1500	@llisews	Mencari ketenangan bathin dan kesehatan mental...
1501	@_nvenheaven	RT @_nvenheaven: Demi menjaga kesehatan menta...

1502 rows x 2 columns

```

In [3]: df = data

In [4]: #Tahap Casefolding
def casefolding(kalimat):
    # Mengubah huruf kapital menjadi huruf kecil
    kalimat = str(kalimat).lower()
    # Menghapus username Twitter (yang dimulai dengan @)
    kalimat = re.sub(r"@[a-zA-Z0-9_]*", "", kalimat)

    #menghapus hastag
    kalimat = re.sub(r"#[a-zA-Z0-9_]*", "", kalimat)

    #Menghapus Url
    kalimat = re.sub(r"http[s]?|www|https", "", kalimat)

    # Menghapus karakter non-alphanumeric, kecuali @ dan #
    kalimat = re.sub(r"[^a-zA-Z0-9@#]", "", kalimat)

    #menghapus spasi
    kalimat = kalimat.strip()

    # Menghapus angka menggunakan regular expression
    kalimat = re.sub(r'\d+', "", kalimat)

    return kalimat

In [5]: df['Casefolding'] = df['Text'].apply(casefolding)
df.head(1500)

In [6]: from nltk.tokenize import word_tokenize

In [16]: #Tahap Tokenizing
def tokenize_text(text):
    tokens = nltk.tokenize.word_tokenize(text)
    return tokens

In [8]: df['Token'] = df['Casefolding'].apply(tokenize_text)
df.head(200)

In [9]: #Stopwords
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
Factory = StopWordRemoverFactory()
stopwords = Factory.get_stop_words()
print (stopwords)

['yang', 'untuk', 'pada', 'ke', 'para', 'namun', 'menurut', 'antara', 'dia', 'dua', 'ia', 'seperti', 'jika', 'jika', 'sehingg
a', 'kembali', 'dan', 'tidak', 'ini', 'karena', 'kepada', 'oleh', 'saat', 'harus', 'sementara', 'setelah', 'belum', 'kami', 'se
kitar', 'bagi', 'serta', 'di', 'dari', 'telah', 'sebagai', 'masih', 'hal', 'ketika', 'adalah', 'itu', 'dalam', 'bisa', 'bahwa',
'atau', 'hanya', 'kita', 'dengan', 'akan', 'juga', 'ada', 'mereka', 'sudah', 'saya', 'terhadap', 'secara', 'agar', 'lain', 'and
a', 'begitu', 'mengapa', 'kenapa', 'yaitu', 'yakni', 'daripada', 'itulah', 'lagi', 'maka', 'tentang', 'demi', 'dimana', 'keman
a', 'pula', 'sambil', 'sebelum', 'sesudah', 'supaya', 'guna', 'kah', 'pun', 'sampai', 'sedangkan', 'selagi', 'sementara', 'teta
pi', 'apakah', 'kecuali', 'sebab', 'selain', 'seolah', 'senaya', 'seterusnya', 'tanpa', 'agak', 'boleh', 'dapat', 'dsb', 'dst',
'dll', 'dahulu', 'dulunya', 'anu', 'demikian', 'tapi', 'ingin', 'juga', 'nggak', 'mari', 'nanti', 'melainkan', 'oh', 'ok', 'seh
arusnya', 'sebetulnya', 'setiap', 'setidaknya', 'sesuatu', 'pasti', 'saja', 'toh', 'ya', 'walaupun', 'tolong', 'tentu', 'amat', 'a
palagi', 'bagaimanapun']

In [10]: def stopword_text(tokens):
cleaned_tokens = []
for token in tokens:
    if token not in stopwords:
        cleaned_tokens.append(token)
return cleaned_tokens

In [11]: df['Stopwords'] = df['Token'].apply(stopword_text)
df.head(1000)

In [12]: #Proses Stopword Extend
from nltk.corpus import stopwords
nltk.download("stopwords")
list_stopwords = stopwords.words('indonesian', 'english')

list_stopwords.extend(["stlh","xf8","x9f","xa4","xa4","part","sinh","deact","rt","pls","cut","off","xf0","x9f","x90","x8f",
'bantar','sd','kadhewjbskwishshv','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','triggened','co','hayang','bunuh','alesan','urg','kamamana','mawa','peso',
'karna','neangan','psnskd','wknz','w','amp','fc','royal','gk','holaaaa','wkwk',
'anjg','goblok2in','s3lf','h4nm','nk','gi','ic','exo','jomblo','sa','tai','oink',
'anj','paska','konser','hngaimki','kontol','bajingan','this','is','trying','cloe',
'sinkkkkkkkkk','produce','bitch','encore','tds','clash','royale','anjggg','panal',
'kpop','xe2','x80','xa6','n','nhawe','play','dll','do','bnr','pens','ipuul','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','bell','maskot','lucu','kim','junmyeon','misstreatment','im','so','of',
'him','fucked','up','solo','netflixnya','sok','an','anjingg','anying','cok','ajg','anjg',
'anjng','ace','fn','aing','ajg','kontol','omg','ygy','qrtsnya','asfdgsh','haechan',
'anjrit','akut','awikwok','lol','setan','asu','dll','co','ajggg','wkwk','tai','wkwkwk',
'sayankkk','anng','bacot','hts','hngaimki','cegill','cogill','e','amuu','jet','piye',
'mameh','myet','bngst','pens','ipuul','kimak','fyp','kameshisugiru','rollecoaster','ucl',
'xd','rengingang','dragon','ball','bola','ajggggg','anyinggg','idol','jyysf','qrt','anjenggg',
'goblok','apple','tasikalaya','lo','yg','main','qrt','oliver','xf','xa','xe','xb','uu','lt',
'bod','humpubnyya','xc','it','not','dn','jiemi','ardl','sp','kj','tw','ket','anjing','pjms','jm','c
'm','ah','mo','alyaafa','dot','com','b','ra','bombardin','ken','adhd','haha','hihi','th','sm',
'aaaaa','tu','hadeu','ama','ni','kekww','tibab','taraaaaa','ekwkwkwk','wlwkwkwk','gin','hehehe','hehehehe
'wow','pi','ed','bangsal','mmc','gin','plijjj','snsjwj','ama','mewekkkk','gin','ken','iku','cukkk',
'gwuu','yesssss','kek','tin','hehehehehehe','anjrr','p','ouch','aka','bon','hahahini','wakakaka','bjir',
'anjim','hahaha','em','koar','jog','v','yoo','pen','igi','c','kww','hehe','hnhh','ber','awokwkwk','ciu',
'anjir','yaah','si','anlrrrr','awokwok','bossskuu','noje','blekping','imo','to','yo','yooongi','pdc',
'nks','lim','ff','dlu','tpi','jdi','huhu','cc','tello','ame','dog','ansharpram','ret','msti','okaayy',
'wkwkwkwk','ber','unas','ki','brantem','selamaaa','tp','wehhhh','sia','apasihhhhhhhhhhh','k','youtube',
'watch','qidtlvfy','ab','bgst','nb','dan','wkwkwk','h','gt','namjoon','un','x','f','nнарarubung','yeuhhh
yh'])

list_stopwords = set(list_stopwords)

def stopwords_removal(text):
    return [word for word in text if word not in list_stopwords]

df['Stopwords_new'] = df['Stopwords'].apply(stopwords_removal)
df.head(1000)

```

```
In [13]: #Tahap Stemming
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
stem_factory = StemmerFactory()
stemmer = stem_factory.create_stemmer()

In [14]: def stemming_text(tokens):
hasil = [stemmer.stem(token) for token in tokens]
return hasil

In [17]: df[['Stemming']] = df[['Stopwords_new']].apply(stemming_text)
df.head(1400)

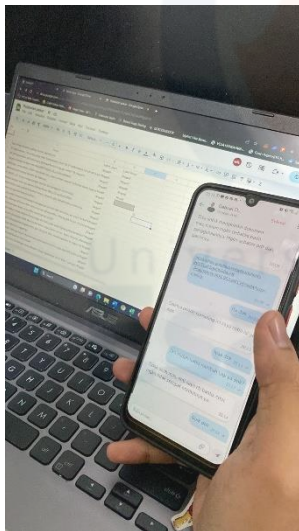
In [20]: #Tahap Normalisasi
# Definiskan fungsi untuk normalisasi teks
def normalize_text(text):
    replacements = {
        'gak': 'tidak', 'ga': 'tidak', 'yaa': 'ya', 'gua': 'aku', 'gue': 'aku', 'gw': 'aku', 'tipikal': 'khas', 'stress': 'stres', 'lu':
        'ku', 'aku', 'trigge': 'pemicu', 'resign': 'berhenti', 'maluuu': 'malu', 'gue': 'aku', 'depressi': 'depresi', 'yukk':
        'yuk', 'ayo', 'gawe': 'kerja', 'thn': 'tahun', 'depression': 'depresi', 'nang': 'nangis', 'capee': 'capek', 'sampe': 'smp',
        'rasajya': 'rasa', 'plsss': 'tolong', 'ampe': 'sampai', 'capeeeeee': 'capek', 'uda': 'udah', 'teruss': 'terus', 'bngt':
        'dari', 'dri': 'dari', 'jgan': 'jangan', 'umm': 'umm', 'overcoming': 'ngatasi', 'haruuuu': 'harus', 'tapiii': 'tapi', 'tp':
        'ketar': 'takut', 'iniiii': 'ini', 'trus': 'terus', 'samaaa': 'sama', 'akuuuu': 'aku', 'lakik': 'laki', 'org': 'orang',
        'krna': 'karena', 'kslo': 'kalo', 'sembari': 'sambil', 'lelahhhh': 'lelah', 'ksihan': 'kasihan', 'sukak': 'suka', 'g':
        'gilaaaaaaaaa': 'gila', 'gilaaa': 'gila', 'yaudahlaahh': 'yaudah', 'ajaaa': 'aja', 'makaaaaasi': 'makasih', 'ktemu':
        'banyakk': 'banyak', 'gimanaaa': 'gimana', 'sumpahhh': 'sumpah', 'makasihh': 'makasih', 'minggat': 'pergi', 'sayaaaaa':
        'acaraaaaaa': 'acara', 'bangett': 'banget', 'mumet': 'pusing', 'kebodohaan': 'bodoh',

        # Tambahkan aturan penggantian lain sesuai kebutuhan
        # Tambahkan aturan penggantian lain sesuai kebutuha
        # Tambahkan aturan penggantian lain sesuai kebutuhan
    }
    for k, v in replacements.items():
        pattern = r'\b' + k + r'\b'
        text = re.sub(pattern, v, text)
    return text

# Terapkan normalisasi pada kolom tweet
data['Normalisasi'] = data['Stemming'].apply(normalize_text)

# Lihat hasil normalisasi
data.head(1500)
```

Lampiran 3 – Pelabelan Yang Dilakukan Oleh Pakar



Merupakan proses pelabelan validasi oleh pakar yang didapat melalui halodoc.

Lampiran 4 – Algoritma *Naïve Bayes*

```

In [38]: 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("Label_Cleaned.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()

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

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

# Step 8: Calculate metrics - Confusion matrix, accuracy, precision, recall, and f1 score
confusion_mat = confusion_matrix(y_test, y_pred)
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')

# Convert metrics to percentage and round to 2 decimal places
accuracy_percent = round(accuracy * 100, 2)
precision_percent = round(precision * 100, 2)
recall_percent = round(recall * 100, 2)
f1_percent = round(f1 * 100, 2)

print("Confusion Matrix:")
print(confusion_mat)

# Step 9: Plot the confusion matrix as a heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(confusion_mat, annot=True, fmt="d", cmap="Blues", xticklabels=["Negatif", "Positif"], yticklabels=["Negatif", "Positif"])
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.title("Confusion Matrix")
plt.show()

print("")
print("=== Hasil Evaluasi Naive Bayes Classifier ===")
print("Accuracy: {}".format(accuracy_percent))
print("Precision: {}".format(precision_percent))
print("Recall: {}".format(recall_percent))
print("F1 Score: {}".format(f1_percent))

```

Lampiran 5 – Algoritma *Support Vector Machine*

```

In [39]: 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("Label_Cleaned.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)

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

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

# Step 8: Calculate metrics - Confusion matrix, accuracy, precision, recall, and f1 score
confusion_mat = confusion_matrix(y_test, y_pred)
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')

# Calculate metrics in percentage
accuracy_percent = accuracy * 100
precision_percent = precision * 100
recall_percent = recall * 100
f1_percent = f1 * 100

print("Confusion Matrix:")
print(confusion_mat)

# Step 9: Plot the confusion matrix as a heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(confusion_mat, annot=True, fmt="d", cmap="Greens", xticklabels=["Negatif", "Positif"], yticklabels=["Negatif", "Positif"])
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.title("Confusion Matrix")
plt.show()

print("")
print("=== Hasil Evaluasi Support Vector Machine ===")
print("Accuracy:", accuracy_percent, "%")
print("Precision:", precision_percent, "%")
print("Recall:", recall_percent, "%")
print("F1 Score:", f1_percent, "%")

```

Lampiran 6 – Algoritma *Random Forest*

```

In [40]: import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
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("Label_Cleaned.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)

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

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

# Step 8: Calculate metrics - Confusion matrix, accuracy, precision, recall, and f1 score
confusion_mat = confusion_matrix(y_test, y_pred)
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')

# Convert metrics to percentage and round to 2 decimal places
accuracy_percent = round(accuracy * 100, 2)
precision_percent = round(precision * 100, 2)
recall_percent = round(recall * 100, 2)
f1_percent = round(f1 * 100, 2)

print("Confusion Matrix:")
print(confusion_mat)

# Step 9: Plot the confusion matrix as a heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(confusion_mat, annot=True, fmt="d", cmap="Oranges", xticklabels=["Negatif", "Positif"], yticklabels=["Negatif", "Positif"])
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.title("Confusion Matrix")
plt.show()

print("")
print("=== Hasil Evaluasi Random Forest ===")
print("Accuracy: {}".format(accuracy_percent))
print("Precision: {}".format(precision_percent))
print("Recall: {}".format(recall_percent))
print("F1 Score: {}".format(f1_percent))

```

Lampiran 7 – Daftar Bimbingan

Bimbingan					
No	Dosen	Topik	Tanggal Bimbingan	Jenis Bimbingan	Catatan Perbaikan
1	5709 - MUNAWAR , S.TP, MM, Ph.D.	Pada tanggal 6 desember bimbingan mengenai pengajuan proposal skripsi dan membuat latar belakang penelitian	30 Jul 2023	Skripsi/Tesis/BusinessPlan Proposal	
2	5709 - MUNAWAR , S.TP, MM, Ph.D.	Pada tanggal 27 Januari bimbingan mengenai proposal skripsi dari bab 1-3	30 Jul 2023	Skripsi/Tesis/BusinessPlan Proposal	
3	5709 - MUNAWAR , S.TP, MM, Ph.D.	Pada tanggal 13 Maret 2023 bimbingan mengenai tugas akhir dan bertanya mengenai bab 4	30 Jul 2023	Skripsi/Tesis/BusinessPlan Proposal	
4	5709 - MUNAWAR , S.TP, MM, Ph.D.	Pada Tanggal 7 April 2023 Melakukan bimbingan bab 4	30 Jul 2023	Skripsi/Tesis/BusinessPlan Proposal	
5	5709 - MUNAWAR , S.TP, MM, Ph.D.	Pada tanggal 10 Juni 2023 melakukan bimbingan dan revisi bab 4	30 Jul 2023	Skripsi/Tesis/BusinessPlan Proposal	
6	5709 - MUNAWAR , S.TP, MM, Ph.D.	Pada tanggal 5 Juli 2023 bimbingan mengenai bab 4 dan memperbaiki pelabelan data	30 Jul 2023	Skripsi/Tesis/BusinessPlan Proposal	
7	5709 - MUNAWAR , S.TP, MM, Ph.D.	Pada tanggal 9 Juli 2023 bimbingan mengenai seluruh bab pada tugas akhir dan merevisi pelabelan data dan kesimpulan, saran	30 Jul 2023	Skripsi/Tesis/BusinessPlan Proposal	
8	5709 - MUNAWAR , S.TP, MM, Ph.D.	Pada tanggal 24 Juli bimbingan bab 4	30 Jul 2023	Skripsi/Tesis/BusinessPlan Proposal	
9	5709 - MUNAWAR , S.TP, MM, Ph.D.	Pada tanggal 26 Juli 2023 bimbingan revisi pada bab 4 dan 5.	30 Jul 2023	Skripsi/Tesis/BusinessPlan Proposal	
10	5709 - MUNAWAR , S.TP, MM, Ph.D.	Pada tanggal 27 Juli menyerahkan Tugas Akhir kepada pembimbing untuk di baca oleh pembimbing	30 Jul 2023	Skripsi/Tesis/BusinessPlan Proposal	