LAMPIRAN

Lampiran 1 : Pengambilan Data

```
!pip install pandas

!sudo apt-get update
!sudo apt-get install -y ca-certificates curl gnupg
!sudo mkdir -p /etc/apt/keyrings
!curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-
repo.gpg.key | sudo gpg --dearmor -o
/etc/apt/keyrings/nodesource.gpg

!NODE_MAJOR=20 && echo "deb [signed-
by=/etc/apt/keyrings/nodesource.gpg]
https://deb.nodesource.com/node_$NODE_MAJOR.x nodistro main"
| sudo tee /etc/apt/sources.list.d/nodesource.list

!sudo apt-get update
!sudo apt-get install nodejs -y

!node -v
# Crawl Data
twitter_auth_token =
'0a3c4bdc7e814643f8229138d5774eba4e9bbd59' #token bisa
diganti dengan akun twitter pribadi

filename = 'dataTELKOMSEL.csv'
search_keyword = 'telkomsel since:2022-01-01 until:2025-04-
14 lang:id'
limit = 2000

!npx -y tweet-harvest@2.6.1 -o "{filename}" -s
"{search_keyword}" --tab "LATEST" -l {limit} --token
{twitter_auth_token}

import pandas as pd

data = pd.read_csv("tweets-data/dataTELKOMSEL.csv")
data.info()
data.head(5)
```

Lampiran 2: Preprocessing

```
import pandas as pd

data = pd.read_csv("DataGabungTelkom.csv")
data.info()
data.head()
df  = pd.DataFrame(data[['created_at','full_text']])
df.info()
df.head(5)

# proses hapus data duplikat
df.info()
df.drop_duplicates(subset ="full_text", keep = 'first',
inplace = True)
df.info()

# Cleaning
import re
import string
import nltk

# Fungsi untuk menghapus URL
def remove_URL(tweet):
    if tweet is not None and isinstance(tweet, str):
        url = re.compile(r'https?://\S+|www\.\S+')
        return url.sub(r'', tweet)
    else:
        return tweet

# Fungsi untuk menghapus HTML
def remove_html(tweet):
    if tweet is not None and isinstance(tweet, str):
        html = re.compile(r'<.*?>')
        return html.sub(r'', tweet)
    else:
        return tweet

# Fungsi untuk menghapus emoji
def remove_emoji(tweet):
    if tweet is not None and isinstance(tweet, str):
        emoji_pattern = re.compile("["
            u"\U0001F600-\U0001F64F"  # emoticons
            u"\U0001F300-\U0001F5FF"  # symbols &
pictographs
```

```
 u"\U0001FA00-\U0001FA6F"  # Chess Symbols
            u"\U0001FA70-\U0001FAFF"  # Symbols and
Pictographs Extended-A
            u"\U0001F004-\U0001F0CF"  # Additional emoticons
            u"\U0001F1E0-\U0001F1FF"  # flags
                              "]+", flags=re.UNICODE)
        return emoji_pattern.sub(r'', tweet)
    else:
        return tweet

# Fungsi untuk menghapus simbol
def remove_symbols(tweet):
    if tweet is not None and isinstance(tweet, str):
        tweet = re.sub(r'[^a-zA-Z0-9\s]', '', tweet)
    return tweet

# Fungsi untuk menghapus angka
def remove_numbers(tweet):
    if tweet is not None and isinstance(tweet, str):
        tweet = re.sub(r'\d', '', tweet)
    return tweet

# Fungsi hapus username
def remove_usernames(text):
    return re.sub(r'@\w+', '', text)

df['cleaning'] = df['full_text'].apply(lambda x:
remove_URL(x))
df['cleaning'] = df['cleaning'].apply(lambda x:
remove_usernames(x))
df['cleaning'] = df['cleaning'].apply(lambda x:
remove_html(x))
df['cleaning'] = df['cleaning'].apply(lambda x:
remove_emoji(x))
df['cleaning'] = df['cleaning'].apply(lambda x:
remove_symbols(x))
df['cleaning'] = df['cleaning'].apply(lambda x:
remove_numbers(x))

df.head(5)
#case folding
def case_folding(text):
    if isinstance(text, str):
        lowercase_text = text.lower()
```

```
        return lowercase_text
    else:
        return text
df['case_folding'] = df['cleaning'].apply(case_folding)
df.head(5)
#normalisasi
import pandas as pd
import requests
from io import BytesIO
# Fungsi penggantian kata tidak baku
def replace_taboo_words(text, kamus_tidak_baku):
    if isinstance(text, str):
        words = text.split()
        replaced_words = []
        kalimat_baku = []
        kata_diganti = []
        kata_tidak_baku_hash = []
        for word in words:
            if word in kamus_tidak_baku:
                baku_word = kamus_tidak_baku[word]
                if isinstance(baku_word, str) and
all(char.isalpha() for char in baku_word):
                    replaced_words.append(baku_word)
                    kalimat_baku.append(baku_word)
                    kata_diganti.append(word)
                    kata_tidak_baku_hash.append(hash(word))
            else:
                replaced_words.append(word)
        replaced_text = ' '.join(replaced_words)
    else:
        replaced_text = ''
        kalimat_baku = []
        kata_diganti = []
        kata_tidak_baku_hash = []
    return replaced_text, kalimat_baku, kata_diganti,
kata_tidak_baku_hash
    return replaced_text, kalimat_baku, kata_diganti,
kata_tidak_baku_hash

# Baca dataset kamu (pastikan df sudah tersedia)
data =
pd.DataFrame(df[['created_at','full_text','cleaning','case_f
olding']])
```

```
data.head()
# Unduh dan baca kamus dari GitHub
url =
"https://github.com/analysisdatasentiment/kamus_kata_baku/raw/main/kamuskatabaku.xlsx"
response = requests.get(url)
file_excel = BytesIO(response.content)
kamus_data = pd.read_excel(file_excel)

# Buat dictionary dari kamus
kamus_tidak_baku_dict = dict(zip(kamus_data['tidak_baku'],
kamus_data['kata_baku']))
# Terapkan fungsi normalisasi
data[['normalisasi', 'Kata_Baku', 'Kata_Tidak_Baku',
'Kata_Tidak_Baku_Hash']] = data['case_folding'].apply(
    lambda x: pd.Series(replace_taboo_words(x,
kamus_tidak_baku_dict))
)

# Ambil kolom yang relevan
df =
pd.DataFrame(data[['created_at','full_text','cleaning','case_folding','normalisasi']])
df.head(5)

#tokenisasi
def tokenize(text):
    tokens = text.split()
    return tokens

df['tokenize'] = df['normalisasi'].apply(tokenize)

df.head(5)

#stopword
from nltk.corpus import stopwords
nltk.download('stopwords')
stop_words = stopwords.words('indonesian')
def remove_stopwords(text):
    return [word for word in text if word not in stop_words]
```

```
df['stopword removal'] = df['tokenize'].apply(lambda x:
remove_stopwords(x))

df.head(5)
#stemming
!pip install Sastrawi

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from nltk.stem import PorterStemmer
from nltk.stem.snowball import SnowballStemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

def stem_text(text):
    return [stemmer.stem(word) for word in text]

df['steming_data'] = df['stopword removal'].apply(lambda x:
' '.join(stem_text(x)))
df.head(5)

#hapus data bernilai kosong
data = df.dropna()
data.info()
```

Lampiran 3 : Pelabelan Data

```
import pandas as pd

data = pd.read_csv("Hasil_Preprocessing_Data.csv")
data.info()
data.head()
data = data.dropna()
data.info()
data = pd.DataFrame(data[['created_at','steming_data']])
data.head(5)
import pandas as pd
import requests

# Unduh kamus leksikon positif dan negatif dari GitHub
positive_url =
"https://raw.githubusercontent.com/fajri91/InSet/master/posi
tive.tsv"
negative_url =
"https://raw.githubusercontent.com/fajri91/InSet/master/nega
tive.tsv"

positive_lexicon = set(pd.read_csv(positive_url, sep="\t",
header=None)[0])
negative_lexicon = set(pd.read_csv(negative_url, sep="\t",
header=None)[0])

# Fungsi untuk menentukan sentimen dan menghitung skornya
def determine_sentiment(text):
    if isinstance(text, str):
        positive_count = sum(1 for word in text.split() if
word in positive_lexicon)
        negative_count = sum(1 for word in text.split() if
word in negative_lexicon)
        sentiment_score = positive_count - negative_count
        if sentiment_score > 0:
            sentiment = "Positif"
        elif sentiment_score < 0:
            sentiment = "Negatif"
        else:
            sentiment = "Netral"
        return sentiment_score, sentiment
    return 0, "Netral"
```

```
# Tentukan sentimen dan skor untuk setiap ulasan
data[['Score', 'Sentiment']] =
data['steming_data'].apply(lambda x:
pd.Series(determine_sentiment(x)))

# Tampilkan hasilnya
data.head(20)
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

sentiment_count = data['Sentiment'].value_counts()
sns.set_style('whitegrid')

fig, ax = plt.subplots(figsize=(6, 4))
ax = sns.barplot(x=sentiment_count.index,
y=sentiment_count.values, palette='pastel')
plt.title('Jumlah Analisis Sentimen', fontsize=14, pad=20)
plt.xlabel('Class Sentiment', fontsize=12)
plt.ylabel('Jumlah Tweet', fontsize=12)

total = len(data['Sentiment'])

for i, count in enumerate(sentiment_count.values):
    percentage = f'{100 * count / total:.2f}%'
    ax.text(i, count + 0.10, f'{count}\n({percentage})',
ha='center', va='bottom')

plt.show()
data.to_csv('Hasil_Labelling_Data.csv',encoding='utf8',
index=False)
```

Lampiran 4 : Naive Bayes Classifier

```
import pandas as pd
import numpy as np
import re
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split,
GridSearchCV
from sklearn.feature_extraction.text import
TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import confusion_matrix,
classification_report, accuracy_score

# Pastikan label bersih
data['Sentiment'] =
data['Sentiment'].str.strip().str.lower()

# TF-IDF Vectorization dengan parameter optimal
tfidf = TfidfVectorizer(
    max_features=5000,
    ngram_range=(1, 2),
    min_df=5,
    max_df=0.9,
    stop_words='english'
)
```

```
# Transformasi teks
X = tfidf.fit_transform(data['steming_data']).toarray()
y = data['Sentiment']

# Split dengan stratifikasi
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)
```

```
# Hyperparameter tuning untuk alpha
param_grid = {'alpha': [0.01, 0.1, 0.5, 1.0]}
grid = GridSearchCV(MultinomialNB(), param_grid, cv=5,
scoring='accuracy')
grid.fit(X_train, y_train)

# Ambil model terbaik
best_model = grid.best_estimator_

# Prediksi
y_pred = best_model.predict(X_test)

# Evaluasi
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)
accuracy = accuracy_score(y_test, y_pred)

print("MultinomialNB (Tuned) Results")
print("==============================")
print("Best alpha:", grid.best_params_['alpha'])
print("Confusion Matrix:")
print(conf_matrix)
print("\nClassification Report:")
print(class_report)
print(f"\nAccuracy: {accuracy:.4f}")

# Plot Confusion Matrix
labels = ['negatif', 'netral', 'positif']
plt.figure(figsize=(6, 4))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
            xticklabels=labels, yticklabels=labels)
plt.title('Confusion Matrix (MultinomialNB)')
plt.xlabel('Prediksi')
plt.ylabel('Aktual')
plt.tight_layout()
plt.show()

# Simpan hasil prediksi
results = pd.DataFrame({
    'steming_data': data.loc[y_test.index, 'steming_data'],
    'Actual': y_test,
    'Predicted': y_pred
})
```

```
results.to_csv('Hasil_pred_MultinomialNB_Tuned.csv',
encoding='utf8', index=False)
print("\nContoh hasil prediksi:")
print(results.head())
```

Lampiran 5 : Sidang Akhir

YAYASAN PENDIDIKAN WIDYA BAKTI YOGYAKARTA

UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA

Jl. Raya Janti (Majapahit) No.143, Yogyakarta, 55198, Telp (0274) 486664,
Website: www.utdi.ac.id , E-mail: info@utdi.ac.id

Hari, tanggal	:	Rabu, 9 Juli 2025
Waktu	:	13.00
Nama	:	Dita Widianti
No. Mahasiswa / Prodi	:	195410113 / Informatika

No	Hal yang harus diperbaiki	Pemberi Catatan
1.	1. Cek lagi flowchart penelitian 2. tahap preprocessing : apa yang dilakukan bukan definisi 3. tambahkan di naskah penghapusan data kosong 4. dinaskah dituliskan hasil sebelum pelabelan 5.di naskah lengkapi pelabelan	Erna Hudianti
2.	1. di naskah penjelasan lexicon base masih terbatas 2. dinaskah dituliskan hasil sebelum pelabelan 3. Naive bayes multinom belum dijelaskan, dan dijelaskan pakai library apa ?	Dini fakta sari
3.	1. Perbaiki penulisan naskah sesuai yang disarankan oleh penguji 2. Pastikan semua referensi yang di acu di naskah ada di dalam daftar pustaka	Edi Iskandar
4.		

YAYASAN PENDIDIKAN WIDYA BAKTI YOGYAKARTA

UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA

Jl. Raya Janti (Majapahit) No.143, Yogyakarta, 55198, Telp (0274) 486664,
Website: www.utdi.ac.id , E-mail: info@utdi.ac.id

KEPUTUSAN HASIL UJIAN PENDADARAN

Sesuai dengan hasil sidang pendadaran pada tanggal 9 Juli 2025 maka

Nama Mahasiswa	Dita Widianti	
NIM / Program Studi	195410113 / Informatika	
Jenjang		
	dinyatakan	LULUS
Ketua Penguji	Erna Hudianti P., S.SI, M.SI.	

SURAT KETERANGAN

PERSETUJUAN PUBLIKASI

Bahwa yang bertanda tangan di bawah ini :

Nama	: Dita Widianti
NIM	: 195410113
Jurusan	: Informatika
Jenjang	: Sarjana
Judul Tugas Akhir	: Analisis Sentimen Pada X Mengenai Pelayanan Provider Telkomsel Menggunakan Metode Naive Bayes

Menyerahkan karya ilmiah kepada pihak perpustakaan UTDI dan menyetujui untuk diunggah ke **Repository** perpustakaan UTDI sesuai dengan ketentuan yang berlaku untuk kepentingan riset dan Pendidikan.

Yogyakarta, 31 Agustus 2025

Penulis,

Dita Widianti

195410113