LAMPIRAN

CARA MENJALANKAN PROGRAM

1. Buka terminal *comman prompt* dan masuk ke direktori tempat file aplikasi berada.

```
C:\Users\DICKO>cd C:\Skripsi
```

2. Aktifkasi *environtment* sebagai tempat menampung *libraries* dan pekerjaan untuk penelitian ini.

```
C:\Skripsi>python -m venv venv
```

```
C:\Skripsi>venv\Scripts\activate
```

3. Jalankan aplikasi web

```
(venv) C:\Skripsi>streamlit run app.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.100.40:8501
```

4. *Browser* otomatis terbuka dan aplikasi sudah berjalan.

LISTING PROGRAM

a. Import libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import string
import re
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import nltk
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn import svm
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.naive_bayes import MultinomialNB # Import untuk Naive Bayes
```

b. Proses crawling

```
filename = 'pragib1.csv'
search_keyword = 'siapBEKERJA untukRAKYAT lang:id'
limit = 1000

!npx --yes tweet-harvest@latest -o "{filename}" -s "{search_keyword}" -l {limit} --token {twitter_auth_token}
```

c. Proses prepocessing

```
#-----CASE FOLDING-----#
import pandas as pd


df['Case Folding'] = df['sentimen'].str.lower()
print('Hasil Case Folding : \n')
print(df['Case Folding'])
import nltk
```

```
from nltk.tokenize import word_tokenize

nltk.download('punkt_tab')
df['Tokenizing'] = df['Case Folding'].apply(word_tokenize)
# Menampilkan hasil
print('Hasil Tokenizing:')
print(df[['Tokenizing']])


!pip install Sastrawi
from Sastrawi.StopWordRemover.StopWordRemoverFactory import
StopWordRemoverFactory

factory = StopWordRemoverFactory()
stopword = factory.create_stop_word_remover()

# Filtering
df['Filtering'] = df['Tokenizing'].apply(lambda tokens: [token
for token in tokens if token not in factory.get_stop_words() and
token.isalnum()])

# Menampilkan hasil
print('Hasil Filtering:')
print(df[['Filtering']])
import re
def normalize_text(text):
    # Case folding
    text = text.lower()

    # Hapus karakter khusus (kecuali spasi)
    text = re.sub(r"[^a-zA-Z0-9 ]", "", text)

    # Hapus angka
    text = re.sub(r"\d+", "", text)

    return text
# Terapkan normalisasi pada kolom 'sentimen'
df['Normalized_Tweet'] = df['sentimen'].apply(normalize_text)

# Menampilkan hasil
print(df[['sentimen', 'Normalized_Tweet']].head())
!pip install Sastrawi
```

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from Sastrawi.StopWordRemover.StopWordRemoverFactory import
StopWordRemoverFactory

# Create stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()
# Create stop word remover
stop_factory = StopWordRemoverFactory()
stopword = stop_factory.create_stop_word_remover()

positive_wOrds=['baik','bagus','senang','percaya','hebat','selam
at','betul','yakin','benar','jujur','terima','setuju','maksimal'
,'amanah','dukung','salut','pantas','berhasil','layak','oke','ef
ektif','sukses','keren','sepakat','suka', 'positif']
negative_words=['buruk','tidak','fufufafa','parah','jelek','rusa
k','kecewa','lemes','bohong','sedih','sebal','tolak','kesal','be
nci','goblok','bodoh','panas','gila','tai','anjing','masalah','h
ancur','batal','gagal','kampret','kosong', 'negatif']

# contoh te
text = "Saya sangat senang dengan pelantikan ini."
# Stemming
stemmed_text = stemmer.stem(text)
# Stop word removal (filtering)
filtered_text = stopword.remove(stemmed_text)

# tampilan hasil
print("Original Text:", text)
print("Stemmed Text:", stemmed_text)
print("Filtered Text:", filtered_text)

df['Stemming'] = df['sentimen'].apply(lambda x: stemmer.stem(x))
print(df[['sentimen', 'Stemming']].head())
```

d. Proses Labeling

```
def labeling_data(text):
    score = 0
    for word in text:
        if word in lexicon:
            if lexicon[word] == 'positif':
                score += 1
```

```
            elif lexicon[word] == 'negatif':
                score -= 1
    if score > 0:
        return 'positif'
    elif score < 0:
        return 'negatif'
    else:
        return 'netral'

df['Label'] = df['Stemming'].apply(labeling_data)

# Menampilkan hasil
print('Hasil Labeling:')
print(df['sentimen'])
```

e. Proses Ekstraksi Fitur

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)

#------ EKSTRAKSI FITUR (CountVectorizer) ------#
from sklearn.feature_extraction.text import CountVectorizer

vectorizer = CountVectorizer()
corpus = [' '.join(tokens) for tokens in df['Stemmed_Tokens']]
tfidf_df = vectorizer.fit_transform(corpus)
```

f. Proses Klasifikasi Naive Bayes

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score

# Inisialisasi model Naive Bayes
model = MultinomialNB()
# Melatih model menggunakan data latih
model.fit(X_train, y_train)
# Melakukan prediksi pada data uji
y_pred = model.predict(X_test)
# Evaluasi performa model (misalnya, menggunakan accuracy)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

g. File app.py

```
import streamlit as st
import prediksi
```

```
import visualisasi
import confusionmatrix
import pandas as pd
# Set page config
st.set_page_config(
    page_title="Analisis Sentimen",
    layout="wide",
    initial_sidebar_state="expanded"
)

# Mapping label ke deskripsi teks
label_to_sentiment = {
    0: "Negatif",
    1: "Positif",
    2: "Netral"
}

# Pilihan halaman di sidebar
pages = ["Halaman Beranda", "Visualisasi Data", "Confusion Matrix"]
choice = st.sidebar.selectbox("Pilih Halaman", pages)

if choice == "Halaman Beranda":
    st.title("Selamat Datang di Halaman Beranda")
    st.write("Aplikasi ini membantu menganalisis sentimen dengan model yang
dilatih.")

    # Form input untuk teks yang akan dianalisis
    user_input = st.text_area("Masukkan teks untuk dianalisis")
    if st.button("Analisis"):
        model, vectorizer = prediksi.load_model()
        prediction_label = model.predict(vectorizer.transform([user_input]))[0]
        prediction_text = label_to_sentiment.get(prediction_label, "Tidak
ahui")  # Konversi label ke teks
        st.write(f"Hasil analisis sentimen: **{prediction_text}**")

elif choice == "Visualisasi Data":
    # Panggil fungsi halaman_visualisasi_data dari visualisasi.py
    visualisasi.halaman_visualisasi_data()



elif choice == "Confusion Matrix":
    confusionmatrix.show_confusion_matrix()
```

h. Halaman Beranda

```
import streamlit as st
import pickle
import seaborn as sns
```

```
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix

# Fungsi untuk memuat model
@st.cache_data
def load_model():
    model = pickle.load(open("skripsianalisis.pkl", "rb"))
    vectorizer = pickle.load(open("vectorizer.pkl", "rb"))
    return model, vectorizer

def show_confusion_matrix():
    model, vectorizer = load_model()

    # Misalkan data testing dan hasil prediksi sudah ada
    # Gantilah dengan data dan prediksi yang sesuai
    y_test = [0, 1, 2, 0, 1, 1, 2, 2, 0, 1]  # Label aktual
    y_pred = model.predict(vectorizer.transform([
        "Teks positif 1", "Teks negatif 1", "Teks netral 1",
        "Teks positif 2", "Teks negatif 2", "Teks positif 3",
        "Teks netral 2", "Teks netral 3", "Teks negatif 3", "Teks positif 4"
    ]))  # Prediksi berdasarkan model

    # Buat confusion matrix
    cm = confusion_matrix(y_test, y_pred)

    # Visualisasikan confusion matrix dengan heatmap dengan ukuran lebih kecil
    fig, ax = plt.subplots(figsize=(5, 3))  # Ukuran plot lebih kecil
    sns.heatmap(cm, annot=True, fmt='d', cmap='Wistia',
                xticklabels=['Negatif', 'Positif', 'Netral'],
                yticklabels=['Negatif', 'Positif', 'Netral'],
                ax=ax)
    ax.set_title('Confusion Matrix')
    ax.set_xlabel('Prediksi')
    ax.set_ylabel('Aktual')

    # Tampilkan plot menggunakan st.pyplot untuk ukuran lebih kecil
    st.pyplot(fig, use_container_width=False)  # Mengatur lebar container
```

i. Halaman Visualisasi Data

```
import streamlit as st
```

```
from PIL import Image

def halaman_visualisasi_data():
    st.sidebar.success("Kamu sedang berada di menu Sebaran
Data")
    st.markdown("<h1 style='text-align: center;'>Visualisasi
Sebaran Sentimen</h1>", unsafe_allow_html=True)
    # Path ke gambar Anda (ganti dengan path sebenarnya)
    bar_chart_path = "bar.jpg"
    pie_chart_path = "pie.jpg"

    # Buka gambar menggunakan PIL
    bar_chart_image = Image.open(bar_chart_path)
    pie_chart_image = Image.open(pie_chart_path)

    # Tampilkan gambar berdampingan menggunakan st.columns
    col1, col2 = st.columns(2)
    with col1:
        st.image(bar_chart_image, caption='Diagram Batang',
use_container_width=True) # Perbaikan di sini
    with col2:
        st.image(pie_chart_image, caption='Diagram Lingkaran',
use_container_width=True) # Perbaikan di sini

    st.markdown(
        """
        Dengan menggunakan diagram batang dan diagram
lingkaran, bisa diketahui sebaran sentimen data tweet
        tentang Pelantikan Prabowo Subianto dan Gibran
Rakabuming Raka.
        Data sentimen negatif berada di urutan ke-2 yaitu berjumlah
358 data.
        Data sentimen positif memiliki jumlah paling banyak yaitu
1708 data.
        Data sentimen netral memiliki data paling sedikit yaitu 934
data.
        """
    )
```

j. Halaman Confusion Matrix

```
import streamlit as st
from PIL import Image
from prediksi import load_model

def show_confusion_matrix():
    st.markdown("<h1 style='text-align: center;'>Confusion
```

```
        Matrix</h1>", unsafe_allow_html=True)  # Hapus atau
        pindahkan ke app.py

            confusion_matrix_path = "cf.jpg"

            # Buka gambar menggunakan PIL
            confusion_matrix_image =
        Image.open(confusion_matrix_path)

            # Tampilkan gambar confusion matrix
    st.image(confusion_matrix_image, caption='Confusion Matrix',
width=500)
```