

# LAPORAN PRAKTIKUM APLIKASI WEB

OLEH: MUHAMMAD EFFLIN RIZQALLAH LIMBONG (22537144007)

---

## MODUL ?

### TOPIK:

Streamlit Machine Learning



```
78 // Trim the path and get the real path ($SERVER['DOCUMENT_ROOT']) . '?_CAPTCHA&t=' . ur
79 $SESSION['_CAPTCHA']['config'] = serialize($captcha_config);
80
81 return array(
82     'code' => $captcha_config['code'],
83     'image_src' => $image_src
84 );
85
86 }
87
88 // function_exists('hex2rgb') ) {
89 // function hex2rgb($hex_str, $return_string = false, $separator = ',') {
90 //     $hex_str = preg_replace("/[0-9A-Fa-f]/", '', $hex_str); // Gets a proper hex string
91 //     $rgb_array = array();
92 //     if (strlen($hex_str) == 6) {
93 //         $color_val = hexdec($hex_str);
94 //         $rgb_array['r'] = 0xFF & ($color_val >> 0x10);
95 //         $rgb_array['g'] = 0xFF & ($color_val >> 0x08);
96 //         $rgb_array['b'] = 0xFF & $color_val;
97 //     } elseif (strlen($hex_str) == 3) {
98 //         $rgb_array['r'] = hexdec(str_repeat(substr($hex_str, 0, 1), 2));
99 //         $rgb_array['g'] = hexdec(str_repeat(substr($hex_str, 1, 1), 2));
100 //         $rgb_array['b'] = hexdec(str_repeat(substr($hex_str, 2, 1), 2));
101 //     } else {
102 //         return false;
103 //     }
104 //     return $return_string ? implode($separator, $rgb_array) : $rgb_array;
105 // }
106 // }
107
108 // Draw the image
109 // if (isset($image_src)) {
110 //     $img = imagecreatefrompng($image_src);
111 //     imagepng($img, $image_src);
112 // }
```

# TABLE OF CONTENTS

<b><i>Week #4</i></b>	<b><i>1</i></b>
<b>A. Penjelasan Tugas Praktikum</b>	<b>3</b>
<b>B. Langkah-langkah dan Screenshot</b>	<b>4</b>
<b>C. Kendala yang Dialami</b>	<b>19</b>
<b>D. Kesimpulan</b>	<b>20</b>

## **A. Penjelasan Tugas Praktikum**

Streamlit merupakan library yang dimiliki Python yang bersifat open source. Streamlit sendiri dikeluarkan pada bulan Oktober tahun 2019. Ada beberapa Package yang terkandung pada Streamlit, diantaranya adalah Flask dan Django. Streamlit biasanya digunakan untuk membuat aplikasi web (Web Apps). Selain dari itu, Streamlit bersifat open source sehingga mudah untuk dibagikan ke pengguna-pengguna lain. Streamlit dapat memudahkan pengguna untuk mengubah data script menjadi aplikasi berbasis web yang interaktif.

Ketika Streamlit dieksekusi atau dijalankan, streamlit akan membuat server lokal. Aplikasi yang dibuat akan tampil di tab browser secara default. Tab browser tersebut adalah tempat dimana pengguna dapat membuat chart, text, widget, table, dan lain-lain.

Tugas praktikum pada pertemuan ini adalah menerapkan sentiment analysis dalam streamlit untuk pernyataan-pernyataan Bahasa Indonesia.

## B. Langkah-langkah dan Screenshot

### A. SENTIMENT ANALYSIS DALAM BAHASA INGGRIS

#### 1. Kode Program:

```
import streamlit as st

from textblob import TextBlob

import pandas as pd

import altair as alt

from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer


def main():

    st.title("Sentiment Analysis NLP App")

    st.subheader("Streamlit Projects")

    menu = ["Home", "About"]

    choice = st.sidebar.selectbox("Menu", menu)

    if choice == "Home":
```

```

        st.subheader("Home")

        with st.form("nlpForm"):

            raw_text = st.text_area("Enter Text
Here")

            submit_button =
st.form_submit_button(label='Analyze')

        # layout

        col1,col2 = st.columns(2)

        if submit_button:

            with col1:

                st.info("Results")

                sentiment =
TextBlob(raw_text).sentiment

                st.write(sentiment)

            # Emoji

            if sentiment.polarity > 0:

```

```

                                st.markdown("Sentiment::
Positive :smiley: ")

                                elif sentiment.polarity < 0:

                                st.markdown("Sentiment::
Negative :angry: ")

                                else:

                                st.markdown("Sentiment::
Neutral 😐 ")

                                # Dataframe

                                result_df =
convert_to_df(sentiment)

                                st.dataframe(result_df)

                                # Visualization

                                c =
alt.Chart(result_df).mark_bar().encode(

                                x='metric',

                                y='value',

                                color='metric')

```

```

st.altair_chart(c, use_container_width=True)

    with col2:

        st.info("Token Sentiment")

                                token_sentiments =
analyze_token_sentiment(raw_text)

        st.write(token_sentiments)

    else:

        st.subheader("About")

def convert_to_df(sentiment):

                                sentiment_dict =
{'polarity':sentiment.polarity,'subjectivity':senti
ment.subjectivity}

                                sentiment_df =
pd.DataFrame(sentiment_dict.items(), columns=['metri
c', 'value'])

    return sentiment_df

```

```

def analyze_token_sentiment(docx):

    analyzer = SentimentIntensityAnalyzer()

    pos_list = []

    neg_list = []

    neu_list = []

    for i in docx.split():

                                                res =
analyzer.polarity_scores(i) ['compound']

        if res > 0.1:

            pos_list.append(i)

            pos_list.append(res)

        elif res <= -0.1:

            neg_list.append(i)

            neg_list.append(res)

        else:

            neu_list.append(i)

```



```
        result =  
{ 'positives': pos_list, 'negatives': neg_list, 'neutral'  
  ':neu_list'  
}  
  
    return result  
  
if __name__ == '__main__':  
  
    main()
```

2. Screenshot:

# Home

Enter Text Here

I HATE YOU

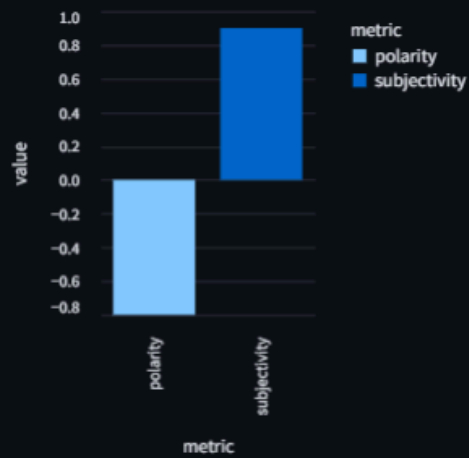
Analyze

Results

```
{
  "polarity": -0.8
  "subjectivity": 0.9
}
```

Sentiment:: Negative 😡

	metric	value
0	polarity	-0.8
1	subjectivity	0.9



Token Sentiment

```
{
  "positives": []
  "negatives": [
    0 : "HATE"
    1 : -0.5719
  ]
  "neutral": [
    0 : "I"
    1 : "YOU"
  ]
}
```

## B. SENTIMENT ANALYSIS DALAM BAHASA INDONESIA:

Untuk versi Bahasa Indonesianya digunakan `GoogleTranslator` dari `deep-translator` untuk menerjemahkan teks bahasa Indonesia ke bahasa Inggris secara otomatis sebelum analisis sentimen dilakukan.

Sebelum menerjemahkan teks, tokenisasi teks asli dalam Bahasa Indonesia dan simpan dalam variabel `original\_tokens`. Kemudian lakukan analisis sentimen pada teks yang telah diterjemahkan ke Bahasa Inggris. Selanjutnya tampilkan hasil token sentimen menggunakan token asli dari `original\_tokens`.

Dengan cara ini, token yang ditampilkan akan tetap menggunakan Bahasa Indonesia, sementara analisis sentimen dilakukan berdasarkan teks terjemahan.

### 1. Kode Program:

```
import streamlit as st

from textblob import TextBlob

import pandas as pd

import altair as alt

from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

from deep_translator import GoogleTranslator #
import library translator
```

```

def main():

    st.title("Aplikasi NLP Sentiment Analysis")

    st.subheader("Project Streamlit")

    menu = ["Home", "About"]

    choice = st.sidebar.selectbox("Menu", menu)

    if choice == "Home":

        st.subheader("Home")

        with st.form("nlpForm"):

            raw_text = st.text_area("Ketikkan Teks  
Anda Disini!")

            submit_button = st.form_submit_button(label='Analisis')

        # layout

        col1, col2 = st.columns(2)

```

```

        if submit_button:

            with coll:

                st.info("Hasil")

                # Simpan token asli sebelum
diterjemahkan

                original_tokens = raw_text.split()

                # Translate text to English

                translated_text =
GoogleTranslator(source='auto',
target='en').translate(raw_text)

                # baris kode dibawah digunakan
untuk menampilkan hasil terjemahan di app (tidak
perlu)

                # st.write("Translated Text:",
translated_text)

                # Sentiment analysis on translated
text

```

```

sentiment =
TextBlob(translated_text).sentiment

st.write(sentiment)

# Emoji

if sentiment.polarity > 0:

    st.markdown("Sentiment::
Positif :smiley: ")

elif sentiment.polarity < 0:

    st.markdown("Sentiment::
Negatif :angry: ")

else:

    st.markdown("Sentiment:: Netral
😐 ")

# Dataframe

result_df =
convert_to_df(sentiment)

st.dataframe(result_df)

```

```

# Visualization

c =
alt.Chart(result_df).mark_bar().encode(

    x='metric',

    y='value',

    color='metric')

st.altair_chart(c,
use_container_width=True)

with col2:

    st.info("Token Sentiment")

# Gunakan token asli saat
menampilkan hasil sentimen

token_sentiments =
analyze_token_sentiment(translated_text,
original_tokens) # Pass original tokens

st.write(token_sentiments)

else:

    st.subheader("About")

```

```

def convert_to_df(sentiment):

    sentiment_dict = {'polarity':
sentiment.polarity,
sentiment.subjectivity}

    sentiment_df =
pd.DataFrame(sentiment_dict.items(),
columns=['metric', 'value'])

    return sentiment_df

def analyze_token_sentiment(translated_text,
original_tokens):

    analyzer = SentimentIntensityAnalyzer()

    pos_list = []

    neg_list = []

    neu_list = []

    # Tokenize translated text and compare with
original tokens

    for i, token in
enumerate(translated_text.split()):

```



```

res =
analyzer.polarity_scores(token) ['compound']

    if res > 0.1:

        pos_list.append(original_tokens[i]) #
Use original token

        pos_list.append(res)

    elif res <= -0.1:

        neg_list.append(original_tokens[i]) #
Use original token

        neg_list.append(res)

    else:

        neu_list.append(original_tokens[i]) #
Use original token

    result = {'positives': pos_list, 'negatives':
neg_list, 'neutral': neu_list}

    return result

if __name__ == '__main__':

    main()

```



*Screenshot:*



### C. Kendala yang Dialami

Untuk praktikum ini, saya mengalami kendala dalam melakukan sentiment analysis dalam bahasa indonesia. Saya tidak bisa menggunakan package

yang memang menggunakan alat-alat machine learning seperti IndoBERT dan IndoNLU yang merupakan model transformer, dan lain-lain sebagainya disebabkan saya masih belum mengerti. Maka dari itu saya hanya dapat melakukan sentiment analysis Bahasa Indonesia menggunakan Google Translator dari deep-translator.

#### **D. Kesimpulan**

Melalui praktikum ini, peserta praktikum telah berhasil memahami konsep dan mempraktikkan sendiri sebuah Aplikasi NLP Sentiment Analysis yang dikembangkan menggunakan Streamlit memberikan kemampuan untuk menganalisis sentimen dari teks yang dimasukkan oleh pengguna dengan fitur:

1. Input Teks: Pengguna dapat memasukkan teks dalam berbagai bahasa. Teks tersebut kemudian diterjemahkan ke dalam bahasa Inggris menggunakan Google Translator untuk memastikan akurasi analisis sentimen.
2. Analisis Sentimen: Aplikasi ini memanfaatkan TextBlob untuk menghitung polaritas dan subjektivitas dari teks yang diterjemahkan. Polaritas memberikan indikasi apakah sentimen tersebut positif, negatif, atau netral.
3. Visualisasi Data: Hasil analisis disajikan dalam bentuk dataframe yang menunjukkan nilai polaritas dan subjektivitas. Selain itu, aplikasi ini juga menampilkan visualisasi menggunakan Altair, yang memberikan representasi grafis dari hasil analisis.
4. Token Sentiment: Aplikasi ini juga melakukan analisis sentimen pada token-token individual dari teks asli, memungkinkan pengguna untuk memahami sentimen masing-masing kata dalam konteks.
5. Feedback Visual: Aplikasi menyediakan umpan balik visual berupa emoji yang mencerminkan sentimen keseluruhan dari teks, meningkatkan interaksi pengguna dengan aplikasi.

Secara keseluruhan, aplikasi ini menunjukkan bagaimana teknologi NLP dapat diintegrasikan dalam platform web untuk memberikan analisis sentimen yang intuitif dan informatif. Implementasi Streamlit sebagai framework memungkinkan pengembangan yang cepat dan interaktif, menjadikannya alat yang berguna untuk proyek berbasis NLP.