

LAPORAN PRAKTIKUM APLIKASI WEB

OLEH: MUHAMMAD EFFLIN RIZQALLAH LIMBONG (22537144007)

MODUL

TOPIK:

NLP Dashboard Pertemuan 9



```
78 $SESSION['CAPTCHA']['config'] = serialize($captcha_config);
79
80 return array(
81     'code' => $captcha_config['code'],
82     'image_src' => $image_src
83 );
84
85 }
86
87
88 // function_exists('hex2rgb') ) {
89 //     function hex2rgb($hex_str, $return_string = false, $separator = ',') {
90 //         $hex_str = preg_replace("/[A0-9A-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

<i>Week #4</i>	<i>1</i>
A. Penjelasan Tugas Praktikum	3
B. Langkah-langkah dan Screenshot	4
C. Kendala yang Dialami	18
D. Kesimpulan	18

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 melakukan Crawling Data, Pre-Processing Data, dan Labelling Data. Kemudian membuat NLP dashboard melalui platform Streamlit.

B. Langkah-langkah dan Screenshot

A. CRAWL DATA, PRE-PROCESSING DATA, DAN LABELLING DATA

1. Kode Program Crawl-Data_Pre-Processing-Data_Labelling-Data.ipynb

Untuk kode program ini, akan saya lampirkan dalam link github repository saya dibawah sebab kode programnya yang berbentuk ipynb dan sangat panjang.

Link GitHub Repository:

https://github.com/flinrzqlh/apk_web_streamlit

Untuk lokasi project berada di awworks > pertemuan9.

B. DASHBOARD NLP STREAMLIT

2. Kode Program dashboardNLP.py:

```
import streamlit as st

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from wordcloud import WordCloud

from textblob import TextBlob    # for sentiment
analysis

# Load preprocessed data
```

```
df = pd.read_csv('labelNLTK.csv')

# Streamlit App Title

st.title("NLP Dashboard")

# Display the dataframe

st.write("Here is your preprocessed dataset:")

st.dataframe(df)

# Show the column names

st.write("Dataset columns:")

st.write(df.columns)

# Sidebar for user input

st.sidebar.title("Options")

analysis_type = st.sidebar.selectbox(

    "Choose Analysis Type",
```

```

        ("Basic Info", "Text Analysis", "Sentiment
Distribution", "Word Cloud", "Sentiment Analysis")
    )

# Option 1: Show basic dataset info

if analysis_type == "Basic Info":

    st.subheader("Basic Information")

    st.write("Shape of dataset:", df.shape)

    st.write("Data types:")

    st.write(df.dtypes)

    st.write("Missing values in the dataset:")

    st.write(df.isnull().sum())

# Option 2: Perform simple text analysis

elif analysis_type == "Text Analysis":

    st.subheader("Text Data Exploration")

```

```

        # Example of analyzing word lengths in a
        specific column (replace 'steaming_data' with the
        actual column name)

        if 'steaming_data' in df.columns:

            df['text_length'] =
df['steaming_data'].apply(lambda x:
len(str(x).split()))

            st.write("Distribution of text lengths:")

            st.write(df['text_length'].describe())

            # Plotting text length distribution

            plt.figure(figsize=(10, 5))

            sns.histplot(df['text_length'], kde=True)

            st.pyplot(plt)

        else:

            st.error("Column 'steaming_data' not found
in the dataset!")

# Option 3: Sentiment Distribution (replace
'sentiment' with the actual sentiment column)

```

```

elif analysis_type == "Sentiment Distribution":

    st.subheader("Sentiment Distribution")

    if 'sentiment' in df.columns:

        sentiment_counts = df['sentiment'].value_counts()

        st.write(sentiment_counts)

        # Plotting sentiment distribution

        plt.figure(figsize=(10, 5))

        sns.countplot(data=df, x='sentiment')

        st.pyplot(plt)

    else:

        st.error("Column 'sentiment' not found in the dataset!")

# Option 4: Generate Word Cloud (replace '...' with the actual text column)

elif analysis_type == "Word Cloud":

```



```

st.subheader("Word Cloud")

if 'steaming_data' in df.columns:

    text = "
".join(df['steaming_data'].astype(str))

    # Generate a word cloud

    wordcloud = WordCloud(width=800,
height=400,
background_color="white").generate(text)

    # Display the word cloud

    plt.figure(figsize=(10, 5))

    plt.imshow(wordcloud,
interpolation="bilinear")

    plt.axis("off")

    st.pyplot(plt)

else:

    st.error("Column 'steaming_data' not found
in the dataset!")

```

```

# Option 5: Sentiment Analysis using TextBlob

elif analysis_type == "Sentiment Analysis":

    st.subheader("Sentiment Analysis")

    if 'steaming_data' in df.columns:

        df['polarity'] =
df['steaming_data'].apply(lambda x:
TextBlob(str(x)).sentiment.polarity)

        df['subjectivity'] =
df['steaming_data'].apply(lambda x:
TextBlob(str(x)).sentiment.subjectivity)

    st.write("Sentiment Analysis Results
(Polarity & Subjectivity):")

    st.write(df[['steaming_data', 'polarity',
'subjectivity']])

    # Visualizing sentiment polarity

    plt.figure(figsize=(10, 5))

```

```

        sns.histplot(df['polarity'], kde=True,
bins=20, color="blue", label="Polarity")

        plt.legend()

        st.pyplot(plt)

    # Visualizing word cloud based on positive
sentiment

        positive_text = " ".join(df[df['polarity']
> 0]['steaming_data'].astype(str))

        wordcloud_pos = WordCloud(width=800,
height=400,
background_color="white").generate(positive_text)

        st.write("Positive Sentiment Word Cloud:")

        plt.figure(figsize=(10, 5))

                                plt.imshow(wordcloud_pos,
interpolation="bilinear")

        plt.axis("off")

        st.pyplot(plt)

else:

```

```
st.error("Column 'steaming_data' not found  
in the dataset!")
```

```
# Footer
```

```
st.write("NLP Dashboard created using Streamlit!")
```

3. Screenshot:

Screenshot Halaman Streamlit:

NLP Dashboard

Here is your preprocessed dataset:

	steaming_data	sentiment_score	sentiment
29	ta h jadwal interview iisma ku mock interview katingku tau gak interview boom i aln	-0.5926	Negatif
30	ultitaskign joki minimal sih	0	Netral
31	yungstress kali ultahnya ldr ya	0	Netral
32	a sangkal rasio anak vo terima apply iisma kaya gimana coba pikir faktor kaya pakai	0	Netral
33	adiah neng pergi iisma psikobagiduo	0	Netral
34	in marah klo pacar yg kayak anjg anjg setia bangga	0	Netral
35	wardeez iisma iismavo lulus yg ucap terima kasih skrng blm yuk skrng ucapin dicorr	0	Netral
36	egabutanagata yhh makasih orang yg bikin kompetitif akademik semester wont be t	-0.2183	Negatif
37	egabutanagata km udh ga sayang ya dig udh jarang main sm gaada foto	0	Netral
38	smafess uni	0	Netral

Dataset columns:

0
steaming_data
sentiment_score
sentiment

Basic Information

Shape of dataset: (319, 3)

Data types:

	0
steaming_data	object
sentiment_score	float64
sentiment	object

Missing values in the dataset:

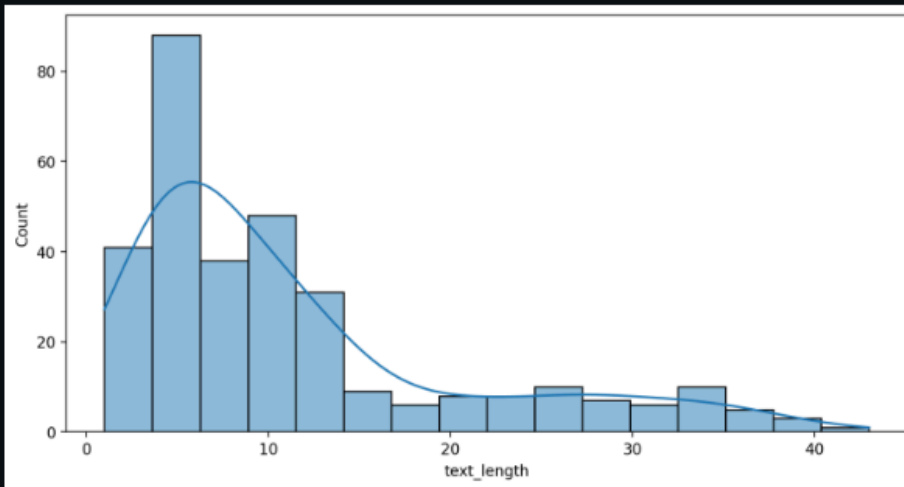
	0
steaming_data	0
sentiment_score	0
sentiment	0

NLP Dashboard created using Streamlit!

Text Data Exploration

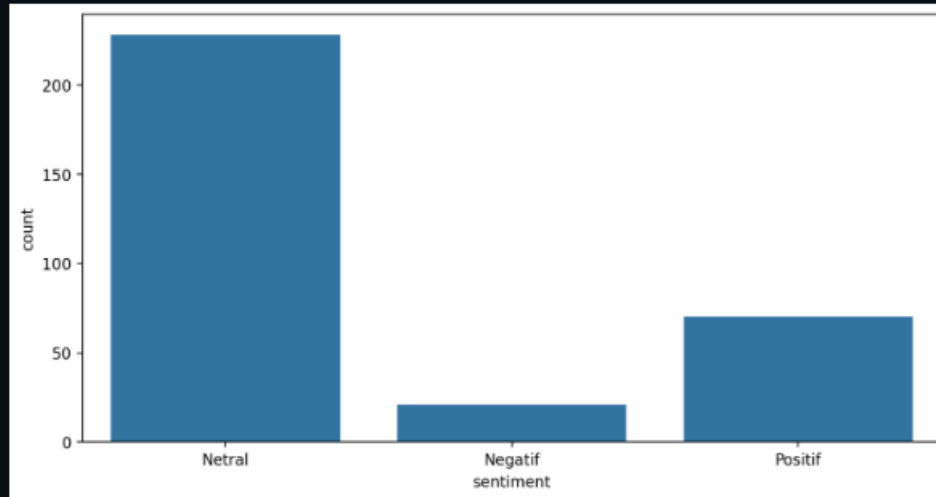
Distribution of text lengths:

	text_length
count	319
mean	11.348
std	9.3702
min	1
25%	5
50%	8
75%	14
max	43



NLP Dashboard created using Streamlit!

sentiment	count
Netral	228
Positif	70
Negatif	21



Word Cloud



NLP Dashboard created using Streamlit!

C. Kendala yang Dialami

Untuk praktikum ini, saya menghabiskan waktu 10 jam (satu malam penuh tanpa tidur) untuk mempelajari cara melakukan Crawling data, Pre-Processing Data, dan Labelling Data, serta membuat web dashboard NLPnya melalui Streamlit. Untuk kendala yang saya alami adalah masih tidak pahamnya secara lebih dalam dan lebih rinci mengenai cara melakukan pengolahan data (crawling, preprocessing, dan labelling).

D. Kesimpulan

Pada praktikum ini, kami berhasil melakukan pengolahan data hasil crawling dari Twitter dengan keyword "IISMA" dan membuat dashboard NLP dengan platform Streamlit. Proses ini melibatkan beberapa langkah penting, di antaranya:

1. **Crawling Data:** Pada tahap ini, kami berhasil mengumpulkan data dari Twitter berdasarkan keyword "IISMA." Data yang diperoleh mencakup berbagai tweet yang kemudian digunakan sebagai dasar untuk analisis lebih lanjut. Pengambilan data ini dilakukan dengan menggunakan API Twitter dan beberapa filter untuk mendapatkan data yang relevan dan sesuai dengan kriteria yang ditentukan.
2. **Preprocessing Data:** Setelah mendapatkan data mentah, kami melakukan preprocessing atau pembersihan data. Proses ini meliputi penghapusan karakter khusus, tanda baca, angka, dan link yang tidak relevan. Selain itu, kami juga melakukan normalisasi teks dengan mengubah kata-kata menjadi bentuk dasar melalui stemming dan tokenisasi. Tahapan ini bertujuan untuk memastikan bahwa data yang digunakan dalam analisis selanjutnya berada dalam bentuk yang siap untuk diolah.
3. **Labelling Data:** Setelah data dibersihkan, kami melakukan proses labelling untuk menentukan sentimen dari setiap tweet yang diperoleh. Sentimen ini dikategorikan ke dalam tiga kelas utama, yaitu positif, negatif, dan netral. Kami menggunakan model sentiment analysis yang sederhana untuk memberikan label berdasarkan skor sentimen yang diperoleh dari analisis sentimen. Label ini digunakan untuk memahami bagaimana persepsi publik terkait keyword "IISMA."

4. Pembuatan Dashboard NLP: Setelah berhasil melakukan pengolahan dataset. Kami lanjut dengan membuat dashboard NLP agar dapat menampilkan visualisasi dataset berupa sentiment analysis, wordcloud, dan lain sebagainya.

Secara keseluruhan, proses ini menunjukkan bahwa data yang berhasil kami crawl dari Twitter dapat diolah dengan baik melalui berbagai tahap preprocessing dan labelling.