

# LAPORAN PRAKTIKUM APLIKASI WEB

OLEH: MUHAMMAD EFFLIN RIZQALLAH LIMBONG (22537144007)

---

## MODUL

### TOPIK:

NLP Dashboard Pertemuan 10



```
78 // Trim the path and get the real path to the document root
79 $DOCUMENT_ROOT = trim(preg_replace('/\\\\\\/', '/', $image_src), '/');
80
81 $SESSION['CAPTCHA']['config'] = serialize($captcha_config);
82
83 return array(
84     'code' => $captcha_config['code'],
85     'image_src' => $image_src
86 );
87
88 // If the function exists, use it
89 if (function_exists('hex2rgb')) {
90     $hex_str = hex2rgb($hex_str, $return_string);
91     $rgb_array = preg_replace("/[0-9A-Fa-f]/", '', $hex_str); // Gets a proper hex string
92     if (strlen($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) : $hex_str;
105 }
106 // Draw the image
107 if (isset($image_src)) {
108     // Draw the image
109     // Draw the image
110     // Draw the image
```

# TABLE OF CONTENTS

<b><i>Week #10</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>15</b>
<b>D. Kesimpulan</b>	<b>15</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 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](https://github.com/flinrzqlh/apk_web_streamlit)

Untuk lokasi project berada di awworks > pertemuan9.

### B. DASHBOARD NLP STREAMLIT

#### 1. 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('labelNLTKutbk.csv')

# Streamlit App Title with Emoji

st.title("📊 NLP Dashboard")

# Sidebar for user input with Emojis

st.sidebar.title("⚙️ Options")

analysis_type = st.sidebar.selectbox(

    "🔍 Choose Analysis Type",

    ("📁 View Dataset", "📋 Basic Info", "📝 Text
Analysis", "😊 Sentiment Distribution", "☁ Word Cloud",
"🔮 Sentiment Analysis")

)

# Option for Viewing the Dataset

if analysis_type == "📁 View Dataset":

    st.subheader("📁 Preprocessed Dataset")

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

    st.dataframe(df)

```

```

st.write("Dataset columns:")

st.write(df.columns)

# Option 1: Show basic dataset info

if analysis_type == "📋 Basic Info":

    st.subheader("📋 Basic Information")

    st.write("This section provides general information  
about the dataset, including its size, data types, and any  
missing values.")

    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")

```

```

    st.write("In this section, we explore the text data by
analyzing word lengths, providing insights into the
overall text structure.")

    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

elif analysis_type == "😊 Sentiment Distribution":

```

```

st.subheader("😊 Sentiment Distribution")

    st.write("Sentiment distribution shows the frequency
of different sentiment categories (e.g., positive,
negative, neutral) in the dataset. The graph below
illustrates the distribution of sentiments across the
dataset.")

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

elif analysis_type == "☁ Word Cloud":

```



```

st.subheader("☁ Word Cloud")

    st.write("The word cloud provides a visual
representation of the most common words in the dataset.
The larger the word, the more frequently it appears.")

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")

    st.write("This section uses TextBlob to calculate the polarity and subjectivity of the text data. Polarity indicates the sentiment (positive or negative), while subjectivity indicates how subjective the text is.")

    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!")
```

## 2. Kode pada file config.toml:

```
[theme]

primaryColor = "#3F2739"    # This is the color of the sidebar

backgroundColor = "#616247FF" # Light pink for the main background

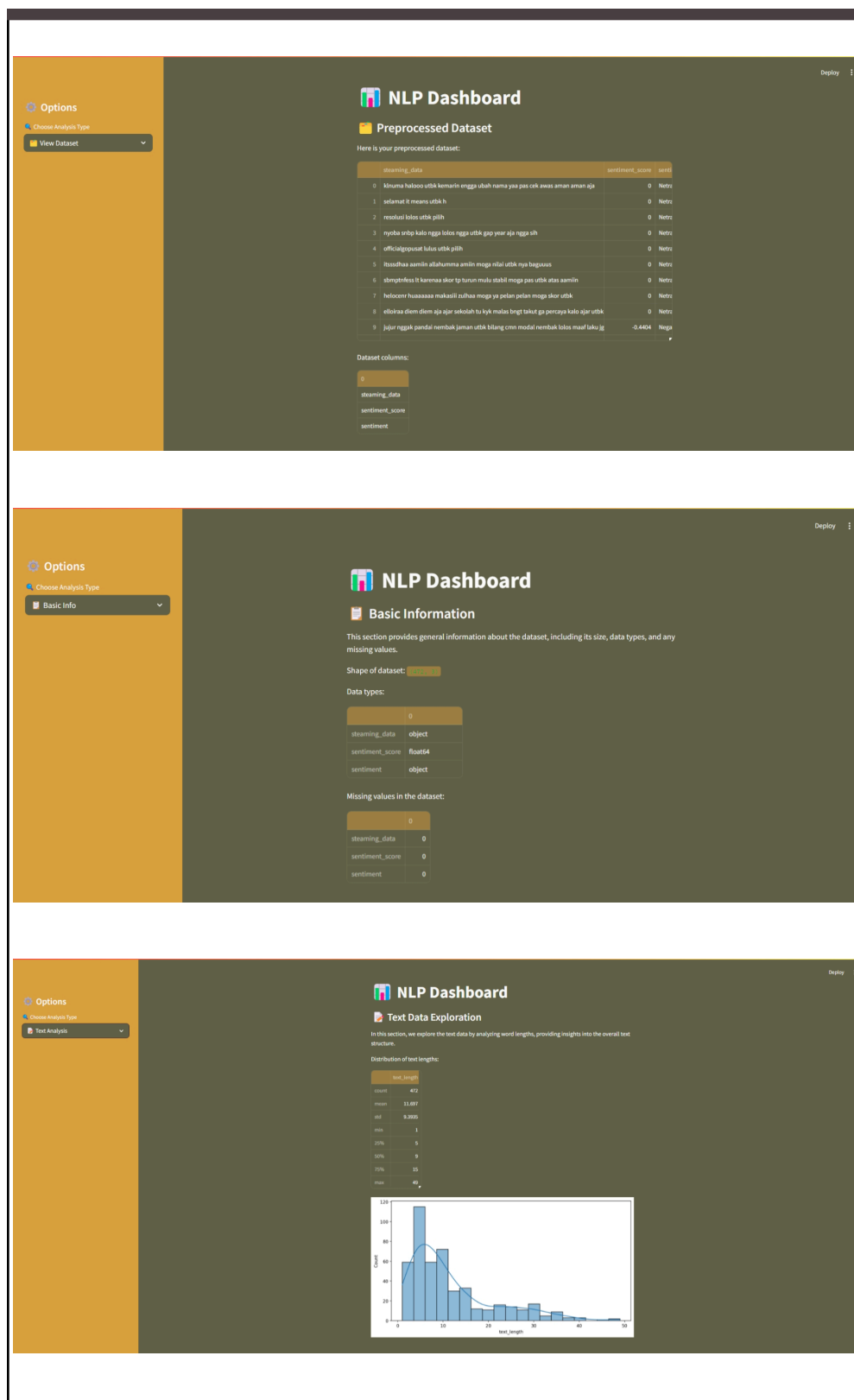
secondaryBackgroundColor = "#DAA03DFF" # Optional lighter shade for other elements

textColor = "#ffffff" # Main text color (black)

font = "sans serif" # Default font family
```

## 3. Screenshot:

Screenshot Halaman Streamlit:





### **C. Kendala yang Dialami**

Untuk praktikum ini, kami 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 kami 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. Selain dari itu, kami juga mendekorasi aplikasi dengan menggunakan file config.toml

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