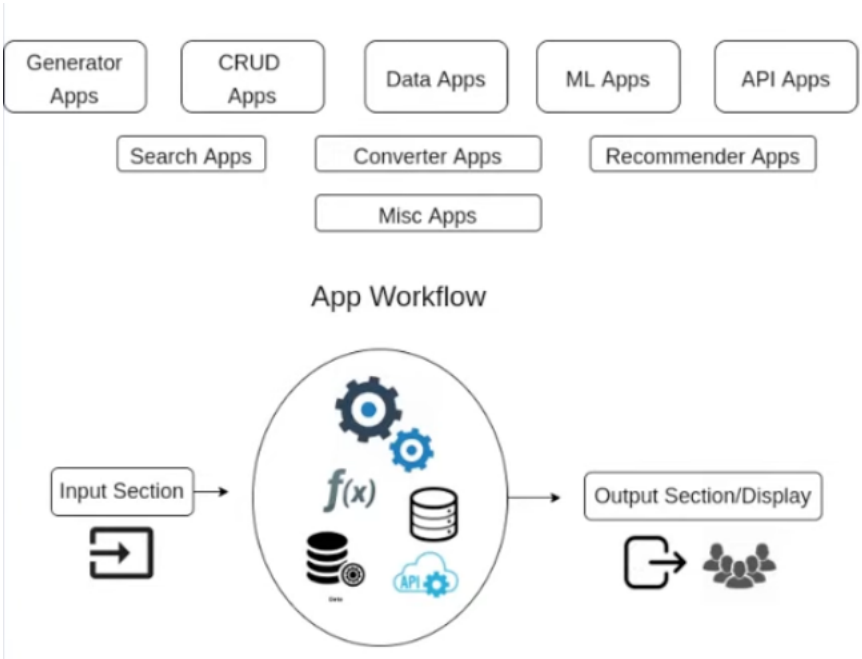
Simple Sentiment Analysis using TextBlob and vaderSentiment



1. Import library

```
import streamlit as st
from textblob import TextBlob
import pandas as pd
import altair as alt
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
```

2. Function untuk main page

```
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")


if __name__ == '__main__':
    main()
```

3. Function untuk menyimpan data teks menjadi dataframe

```
def convert_to_df(sentiment):
    sentiment_dict =
{'polarity':sentiment.polarity,'subjectivity':sentiment.subjectivity}
    sentiment_df =
pd.DataFrame(sentiment_dict.items(),columns=['metric','value'])
    return sentiment_df
```

4. Function untuk analisis token

```
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:
            neu_list.append(i)
            neu_list.append(res)
        else:
            neu_list.append(i)

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

Kesimpulan:

1. Bagaimana cara kerja analisis sentimen?
2. Bagaimana alur melakukan analisis sentimen pada data teks?
3. Jelaskan masing-masing kegunaan library yang digunakan!
4. Analisislah masing-masing function yang telah digunakan!