

Studi Kasus Modul 9. Analisis Sentimen Media Sosial

Studi Kasus: Opini Publik terhadap Layanan Transportasi Online di Indonesia

Tujuan Pembelajaran

Setelah menyelesaikan modul ini, mahasiswa mampu:

1. Menjelaskan konsep dasar dan pentingnya analisis sentimen dalam NLP.
2. Mengimplementasikan tahapan analisis sentimen secara komprehensif pada data media sosial.
3. Menerapkan preprocessing teks berbahasa Indonesia (tokenisasi, stopwords, stemming, dsb).
4. Membangun model klasifikasi sentimen menggunakan *machine learning*.
5. Mengevaluasi hasil model dan menafsirkan maknanya secara kontekstual.

A. Pendahuluan

Analisis sentimen (*Sentiment Analysis*) adalah proses untuk mengekstraksi dan mengklasifikasikan opini, sikap, atau emosi dalam teks menjadi kategori seperti **positif**, **negatif**, atau **netral**. Pada media sosial, analisis ini membantu memahami persepsi publik terhadap produk, layanan, atau kebijakan tertentu. Dalam modul ini, kita akan melakukan analisis sentimen terhadap tweet publik tentang **Gojek** dan **Grab** — dua layanan transportasi online terbesar di Indonesia.

B. Konteks dan Sumber Data

Data yang digunakan bersumber dari media sosial **Twitter**, dikumpulkan dengan pustaka `snsrape`, atau dapat menggunakan dataset publik dari:

Kaggle: [Indonesian Twitter Sentiment Dataset](#)

Contoh isi dataset:

ID	Tanggal	Tweet	Label
1	2024-08-15	Driver gojeknya ramah banget, pengiriman cepat pula!	Positif
2	2024-08-17	Grab sekarang sering error, susah banget pesen!	Negatif
3	2024-08-18	Tarif gojek naik ya, tapi tetap worth it sih.	Netral
4	2024-08-19	Gojek app-nya crash terus, tolong dong diperbaiki.	Negatif
5	2024-08-20	Suka banget sama fitur GoFood, makin cepat!	Positif

C. Tahapan Analisis Sentimen

1. Data Collection

```
import snsrape.modules.twitter as sntwitter
import pandas as pd

tweets = []
query = "Gojek OR Grab lang:id since:2024-08-01 until:2024-08-31"
for i, tweet in enumerate(sntwitter.TwitterSearchScraper(query).get_items()):
```

```

if i > 500:
    break
tweets.append([tweet.date, tweet.content])

df = pd.DataFrame(tweets, columns=["tanggal", "tweet"])
df.head()

```

2. Text Preprocessing

Membersihkan teks dari elemen yang tidak relevan, menormalkan kata, dan menghapus stopwords.

```

import re
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from nltk.corpus import stopwords
import nltk
nltk.download('stopwords')

factory = StemmerFactory()
stemmer = factory.create_stemmer()
stop_words = set(stopwords.words('indonesian'))

def clean_text(text):
    text = text.lower()
    text = re.sub(r'http\S+|www\S+', '', text)
    text = re.sub(r'^a-zA-Z\s', '', text)
    text = re.sub(r'\s+', ' ', text)
    tokens = text.split()
    tokens = [word for word in tokens if word not in stop_words]
    tokens = [stemmer.stem(word) for word in tokens]
    return ' '.join(tokens)

df['clean_tweet'] = df['tweet'].apply(clean_text)
df.head(5)

```

Contoh hasil preprocessing:

Tweet Asli	Setelah Preprocessing
"Driver gojeknya ramah banget, pengiriman cepat pula!"	driver gojek ramah cepat pula
"Grab sekarang sering error, susah banget pesen!"	grab sering error susah pesen

3. Feature Extraction (TF-IDF)

```

from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer(max_features=2000)
X = tfidf.fit_transform(df['clean_tweet']).toarray()

```

4. Model Training – Naive Bayes

```
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report, confusion_matrix

X_train, X_test, y_train, y_test = train_test_split(X, df['label'], test_size=0.2,
random_state=42)

model = MultinomialNB()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

print(classification_report(y_test, y_pred))
```

5. Evaluasi Model

Hasil evaluasi dari dataset nyata (contoh):

	precision	recall	f1-score	support
negatif	0.86	0.84	0.85	200
netral	0.78	0.80	0.79	150
positif	0.91	0.89	0.90	250
accuracy			0.86	600

Interpretasi:

- ✓ Model memiliki **akurasi 86%**, cukup baik untuk dataset sosial media.
- ✓ Tweet positif paling mudah dikenali (ciri: kata “cepat”, “bagus”, “ramah”).
- ✓ Kesalahan terbesar ada pada tweet netral, yang sering ambigu.

D. Visualisasi Hasil

1. Distribusi Sentimen

```
import matplotlib.pyplot as plt

sentiment_count = df['label'].value_counts()
sentiment_count.plot(kind='bar', color=['green','orange','red'])
plt.title('Distribusi Sentimen Pengguna terhadap Layanan Gojek & Grab')
plt.xlabel('Sentimen')
plt.ylabel('Jumlah Tweet')
plt.show()
```

Contoh Hasil Visualisasi:

Sekitar:

- ✓ 55% tweet positif
- ✓ 30% negatif
- ✓ 15% netral

Mayoritas pengguna puas, tetapi masih terdapat keluhan terhadap error aplikasi dan tarif.

2. Word Frequency Analysis

```
from collections import Counter

all_words = ' '.join(df['clean_tweet']).split()
word_freq = Counter(all_words).most_common(10)
word_freq
```

Hasil (Top 10 kata):

```
[('gojek', 420), ('grab', 310), ('driver', 290),
 ('cepat', 180), ('error', 160), ('aplikasi', 145),
 ('bagus', 130), ('mahal', 120), ('layanan', 110), ('puas', 100)]
```

Interpretasi:

- ✓ Kata “driver”, “cepat”, dan “puas” muncul dominan dalam tweet positif.
- ✓ Kata “error”, “mahal” muncul dalam tweet negatif.
- ✓ Isu utama negatif: bug aplikasi dan tarif naik.

E. Contoh Grafik Word Cloud

(ditampilkan dalam file Word)

Menggunakan pustaka wordcloud:

```
from wordcloud import WordCloud

text = ' '.join(df['clean_tweet'])
wc = WordCloud(width=800, height=400,
background_color='white').generate(text)
plt.imshow(wc, interpolation='bilinear')
plt.axis('off')
plt.show()
```

Interpretasi Visual:

Kata yang paling besar pada *word cloud* menunjukkan topik yang sering dibicarakan publik, misalnya “gojek”, “driver”, dan “cepat”.

F. Analisis Ringkas

Aspek	Temuan
Model Terbaik	Naive Bayes + TF-IDF
Akurasi	86%
Sentimen Dominan	Positif (55%)

Aspek	Temuan
Isu Negatif	Error aplikasi, tarif tinggi
Insight	Publik umumnya puas dengan layanan cepat, namun terganggu bug aplikasi.

G. Lembar Kerja Mahasiswa (Isian)

No	Tweet Asli	Setelah Preprocessing	Prediksi Model	Analisis Manual	Cocok / Tidak
1
2
...

H. Rubrik Penilaian

Komponen	Kriteria	Bobot
Kode & Implementasi	Preprocessing, TF-IDF, Model Naive Bayes berjalan dengan benar	30%
Analisis Hasil	Interpretasi tepat dan logis	25%
Visualisasi	Informatif dan rapi (grafik & word cloud)	15%
Laporan	Penulisan sistematis, maksimal 2 halaman	20%
Diskusi & Refleksi	Pemahaman mendalam atas hasil	10%
Total		100%

9. Refleksi dan Diskusi

1. Apakah model lebih baik menggunakan data mentah atau yang sudah distemming?
2. Apa yang menyebabkan model salah memprediksi tweet netral?
3. Jika dataset diperbesar 10x lipat, bagaimana pengaruhnya terhadap akurasi model?
4. Bagaimana jika digunakan model *deep learning* seperti IndoBERT?

10. Pengembangan Lanjutan

- Implementasi **Aspect-Based Sentiment Analysis (ABSA)**.
- Analisis tren waktu: perubahan opini publik mingguan.
- Visualisasi interaktif dengan **Streamlit dashboard**.