

ANALISIS SENTIMEN PADA MEDIA SOSIAL YOUTUBE TERHADAP PARIWISATA LUMPUR LAPINDO DENGAN MENGGUNAKAN ALGORITMA K-NEAREST NEIGHBOR (K-NN) DAN TERM FREQUENCY –INVERS DOCUMENT FREQUENCY

Muhammad Iqbal Nahariqi

181080200225

Dosen Pembimbing :

Yulian Findawati , ST., M.MT.

Fakultas Sains dan Teknologi Program Studi Informatika

Universitas Muhammadiyah Sidoarjo

Juli 2025

Pendahuluan

Kemajuan teknologi di era industri telah mendorong perubahan besar dalam berbagai aspek kehidupan, termasuk meningkatnya penggunaan media sosial seperti YouTube. Platform ini memungkinkan pengguna mengunggah video, memberi komentar, dan berinteraksi secara luas, namun juga berpotensi disalahgunakan. Untuk mendukung pengembangan sistem yang terintegrasi, YouTube menyediakan API yang memudahkan pengembang dalam mengakses dan mengelola konten. Dalam penelitian ini, dilakukan analisis sentimen terhadap komentar pengguna terhadap pariwisata Lumpur Lapindo. Analisis sentimen bertujuan mengidentifikasi opini publik, yang diklasifikasikan ke dalam sentimen positif dan negatif. Proses klasifikasi menghasilkan model yang dapat memprediksi kategori data baru berdasarkan pola tertentu, sehingga mendukung pemahaman terhadap persepsi masyarakat.

Rumusan Masalah

- Berdasarkan pada permasalahan yang diangkat dalam latar belakang, maka dirumuskan pertanyaan penelitian sebagai berikut: “Mengetahui sentimen komentar pengguna YouTube berpengaruh terhadap pariwisata Lumpur Lapindo?”

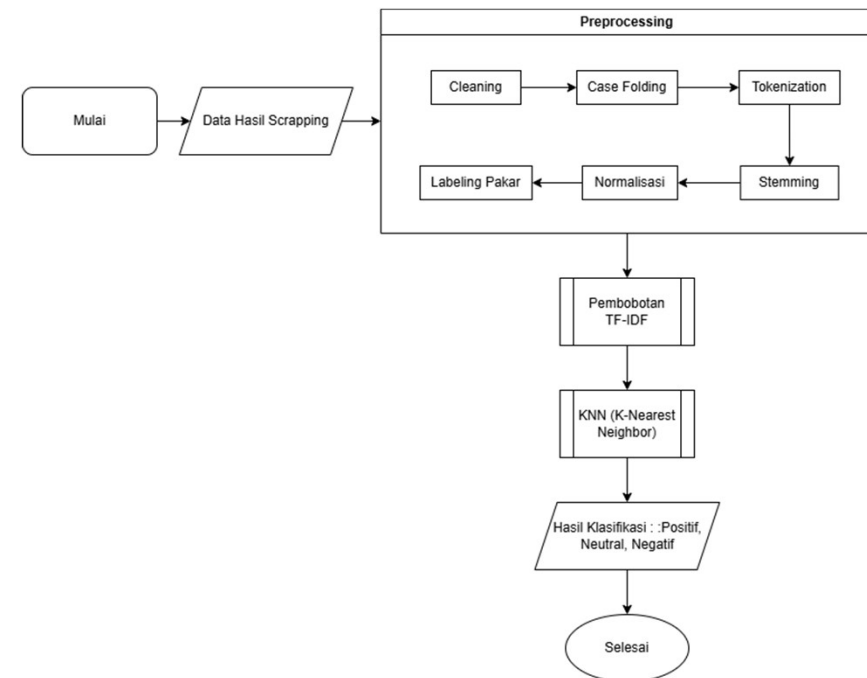
Metode

- Penelitian ini menggunakan metode dan tujuan yang akan dicapai yakni sebagai berikut:
 1. Menjelaskan pengaruh terhadap hasil evaluasi pada algoritma K-Nearest Neighbor (KNN) dan term frequency – invers document frequency (TF-IDF).
 2. Mendapatkan perbandingan hasil pengujian dari klasifikasi emosi seseorang berdasarkan akurasi, precision, recall serta f-measure dari penerapan algoritma K-Nearest Neighbor (KNN) dan term frequency – invers document frequency (TF-IDF).

Metodologi

Tahapan-tahapan dalam metode penelitian ini dijelaskan secara rinci melalui flowchart berikut :

Tahap awal dari sistem yaitu diawali dengan *Scrapping* komentar/ulasan dari sosial media Youtube, setelah itu dilanjutkan dengan Preprocessing, pada proses ini akan didapatkan *term* dari sebuah data, selanjutnya akan dilakukan pembobotan dengan menggunakan metode N-gram, dari proses pembobotan ini didapatkan matriks dari sebuah data yang akan dilakukan perhitungan pada proses selanjutnya yaitu menggunakan metode KNN (K-Nearest Neighbor). Untuk menentukan klasifikasi dilakukan perhitungan dengan menggunakan metode Cosine Similarity dan didapatkan hasilnya.



Hasil dan Pembahasan

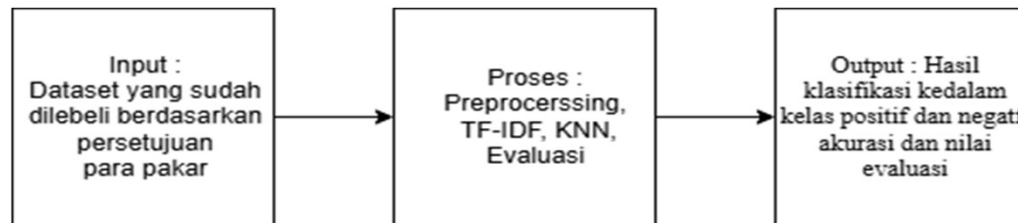
- Dataset yang digunakan dalam penelitian ini terdiri dari 520 komentar YouTube yang berkaitan dengan video “Pariwisata Lumpur Lapindo” dan mencakup enam atribut, yaitu Nama, Komentar, L_Pakar1, L_Pakar2, L_Pakar3, dan L_FIX. Namun, hanya lima fitur yang digunakan dalam implementasi program, yakni Komentar, L_Pakar1, L_Pakar2, L_Pakar3, dan L_FIX. Komentar diberi label oleh tiga pakar yang merupakan mahasiswa Program Studi Psikologi dari Universitas XXX, dan label akhir ditentukan melalui metode voting. Atribut “Nama” menunjukkan identitas akun yang memberikan komentar, sementara “Komentar” berisi opini, kritik, atau tanggapan terhadap pariwisata Lumpur Lapindo. Kolom L_Pakar1, L_Pakar2, dan L_Pakar3 masing-masing berisi label hasil penilaian dari tiap pakar, dan L_FIX merupakan hasil akhir dari ketiga penilaian tersebut. Dataset ini diklasifikasikan menjadi dua kelas, yaitu 169 data berlabel Positif dan 351 data berlabel Negatif. Dan berikut contoh table dataset ulasan :

Nama	Komentar	L_Pakar 1	L_Pakar 2	L_Pakar 3	L_FIX
Niko_ Channel	Yang kemaren banyak tanya tentang sejarah dan kisah Lumpur Lapindo, ini sudah saya buat video khusus cerita asli dari pak Aksan loh, ini link videonya: https://youtu.be/6YYVVGlxCfU Ada juga yang tanya di mana	Negatif	Negatif	Negatif	Negatif

Hasil dan Pembahasan

- **Implementasi**

Implementasi sistem dilakukan menggunakan bahasa pemrograman Python dengan tahapan meliputi preprocessing berupa tokenisasi teks komentar, pembobotan menggunakan metode TF-IDF, klasifikasi menggunakan algoritma K-Nearest Neighbor (K-NN), serta evaluasi untuk mengukur performa model. Ilustrasi dari rancangan Implementasi akan ditunjukkan pada gambar berikut:



Hasil dan Pembahasan

- **Pengujian dan Analisis**

Pada proses ini akan dijelaskan proses pengujian berdasarkan rancangan yang telah dibuat, serta akan dijelaskan proses analisis dari pengujian yang akan dilakukan.

- **Kebutuhan software**

1. Sistem Operasi Windows 11
2. Google Chrome Browser
3. Microsoft Office 2016
4. Microsoft Excel 2016
5. Bahasa Pemrograman *Python*
6. Library yang digunakan adalah *Sastrawi, Numpy, Pandas, re, string, math, scipy, dan taudataNlpTm*

Hasil dan Pembahasan

- **Pembahasan Program**

Tahap berikutnya meliputi perhitungan bobot kata menggunakan metode Term Frequency-Inverse Document Frequency (TF-IDF) serta penentuan klasifikasi dengan menerapkan algoritma K-Nearest Neighbor (K-NN).

```
[19] !pip install Sastrawi
Requirement already satisfied: Sastrawi in /usr/local/lib/python3.11/dist-packages (1.0.1)

[20] from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import re

[21] # Buat stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

[22] def preprocess(text):
    text = text.lower()
    text = re.sub(r'[\a-z\s]', '', text) # hanya huruf dan spasi
    return stemmer.stem(text)

[23] processed_docs = [preprocess(doc) for doc in komentar_list]

[24] # TF-IDF
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(processed_docs)

# Split dan klasifikasi KNN
X_train, X_test, y_train, y_test = train_test_split(X, label_fix_list, test_size=0.7, random_state=42)
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)

# Evaluasi
y_pred = knn.predict(X_test)
print(classification_report(y_test, y_pred))
```

Hasil dan Pembahasan

- **Pembahasan Program**

Pada hasil penelitian Sebanyak 70% dari keseluruhan data yang digunakan pada tahap test sekaligus validasi, pada uji yang dilakukan pada data test ini menentukan nilai yg di dapat nilai *Precision* Positif (1) 67% dan Negatif (0) 85%, untuk nilai *Recall* Positif 70% dan Negatif 83%, untuk nilai rata rata (f1-score) Positif 69% dan negatif 84% seperti sebagaimana ditunjukkan pada gambar di bawah ini

	precision	recall	f1-score	support
0	0.85	0.83	0.84	242
1	0.67	0.70	0.69	122
accuracy			0.79	364
macro avg	0.76	0.77	0.76	364
weighted avg	0.79	0.79	0.79	364

Hasil dan Pembahasan

• Classification Report Heatmap

- Pada fase evaluasi ini, penelitian dilakukan dengan penerapan berbagai metrik evaluasi, seperti peta panas (heatmap), *akurasi*, *presisi*, *recall*, serta *skor f1-score*, guna menilai kinerja model secara komprehensif. Pendekatan ini memberikan pandangan terhadap kemampuan model dalam melakukan klasifikasi secara tepat dan efisien dalam identifikasi klasifikasi. Berikut merupakan gambar Script dan Hasil Evaluasi *Classification Report Heatmap*.

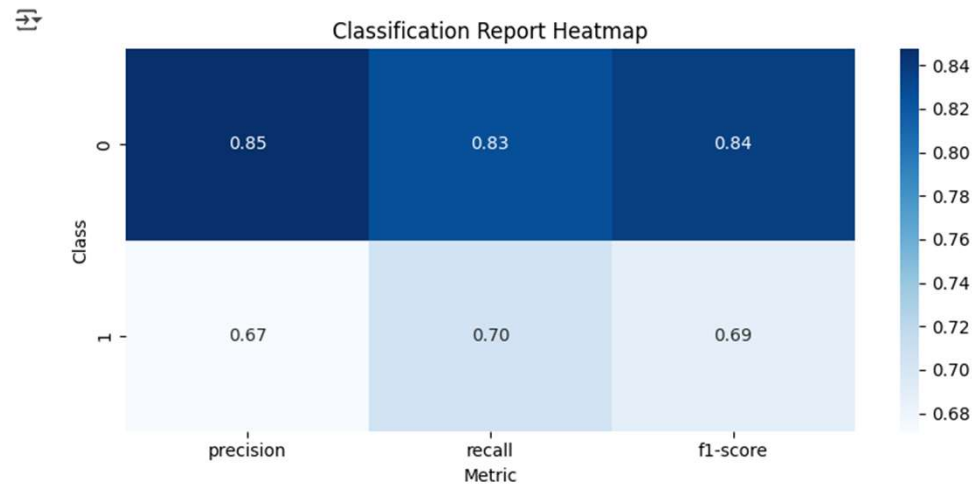
```
# Classification report sebagai dictionary
report_dict = classification_report(y_test, y_pred, output_dict=True)

# Buat DataFrame dan hilangkan 'accuracy', 'macro avg', 'weighted avg'
df_report = pd.DataFrame(report_dict).drop(columns=['accuracy', 'macro avg', 'weighted avg'], errors='ignore').T

# Plot Heatmap
plt.figure(figsize=(8, 4))
sns.heatmap(df_report.iloc[:, :3], annot=True, cmap='Blues', fmt=".2f") # hanya precision, recall, f1-score
plt.title("Classification Report Heatmap")
plt.xlabel("Metric")
plt.ylabel("Class")
plt.tight_layout()
plt.show()
```

Hasil dan Pembahasan

- Hasil Classification Report Heatmap



Hasil dan Pembahasan

- **Confusion Matrix Heatmap**

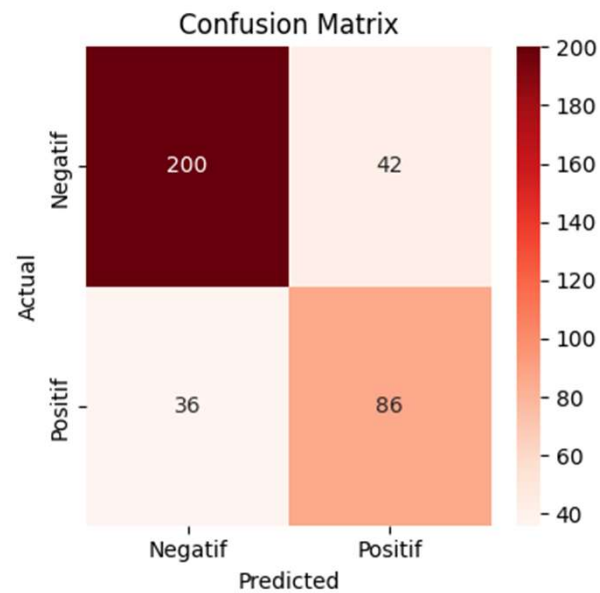
Hasil dari confusion matrix menunjukkan bahwa jumlah True Negative (TN) adalah 200 data, False Positive (FP) sebanyak 42 data, True Positive (TP) sebanyak 86 data, dan False Negative (FN) sebanyak 36 data. Nilai akurasi, presisi, recall, dan f1-score. Kemudian dihitung secara manual berdasarkan data berikut:

```
# Tambahkan Confusion Matrix
conf_matrix = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(4, 4))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Reds', xticklabels=['Positif', 'Negatif'], yticklabels=['Positif', 'Negatif'])
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.tight_layout()
plt.show()
```

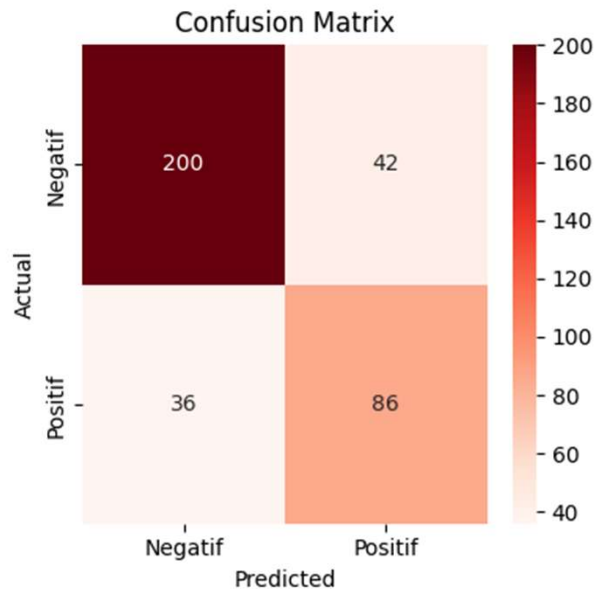
Hasil dan Pembahasan

- Hasil Confusion Matrix Heatmap



Hasil dan Pembahasan

- Hasil Confusion Matrix Heatmap



Perhitungan nilai manual dari Precision, Accuracy, Recall, f1-score dibawah ini:

$$\text{Accuracy} = \frac{TN+TP}{TN+FP+FN+TP} = \frac{200+86}{200+42+36+86} = \frac{286}{364} = 0,78*100 = 78\%$$

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{86}{86+42} = \frac{86}{128} = 0,67*100 = 67\%$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{86}{86+36} = \frac{86}{122} = 0,70*100 = 70\%$$

$$\text{F1-score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} = 2 * \frac{0,67*0,70}{0,67+0,70} = 68\%$$

Kesimpulan

Dari hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa algoritma K-Nearest Neighbor (K-NN) efektif dalam mengklasifikasikan sentimen komentar dengan tingkat akurasi yang memuaskan. Selain itu, penerapan metode TF-IDF sebagai teknik ekstraksi fitur turut berkontribusi signifikan dalam meningkatkan performa model klasifikasi, dengan metode pembobotan TF-IDF mampu melakukan klasifikasi sentimen komentar YouTube terhadap video “Pariwisata Lumpur Lapindo” dengan tingkat akurasi sebesar 78%. Proses pelabelan data dilakukan oleh tiga pakar yang berasal dari Program Studi Psikologi dan menghasilkan label final melalui metode voting. Dari hasil evaluasi terhadap data uji, diperoleh nilai precision sebesar 67%, recall sebesar 70%, dan f1-score sebesar 68% untuk sentimen positif, serta nilai yang lebih tinggi untuk sentimen negatif, menunjukkan bahwa sistem ini cenderung lebih akurat dalam mengenali komentar bernada negatif.

Hasil evaluasi ini menunjukkan bahwa sistem yang dibangun mampu mengidentifikasi sentimen komentar secara cukup baik, meskipun masih terdapat ruang untuk peningkatan akurasi, khususnya pada klasifikasi sentimen positif. Penggunaan algoritma K-NN terbukti efektif untuk tugas klasifikasi berbasis teks seperti ini. Oleh karena itu, sistem ini dapat dijadikan sebagai dasar bagi pengembangan sistem analisis sentimen yang lebih kompleks di masa depan, misalnya dengan menambahkan fitur Bahasa alami (semantic analysis), penggunaan algoritma berbasis machine learning, atau pengujian terhadap penggunaan dataset yang lebih besar dan beragam diperlukan guna mencapai hasil yang lebih optimal serta meningkatkan kemampuan generalisasi model.

Ucapan Terima Kasih

Ucapan Terima Kasih