

Analisis Sentimen Tweet Bahasa Indonesia (Emoji-Aware)

Proyek ini menyajikan pipeline lengkap untuk analisis sentimen tweet berbahasa Indonesia. Fokus utamanya adalah penggunaan machine learning klasik (TF-IDF + Logistic Regression / Naive Bayes) dengan preprocessing berbasis Sastrawi yang secara strategis **mempertahankan emoji sebagai sinyal emosi**.

Metode

Machine Learning Klasik (TF-IDF + LR/NB).

Fitur Kunci

Preprocessing Emoji-Aware dan Stemming Sastrawi.

Tujuan

Klasifikasi sentimen (Positif, Netral, Negatif) pada data Twitter.





Deskripsi dan Analisis Dataset

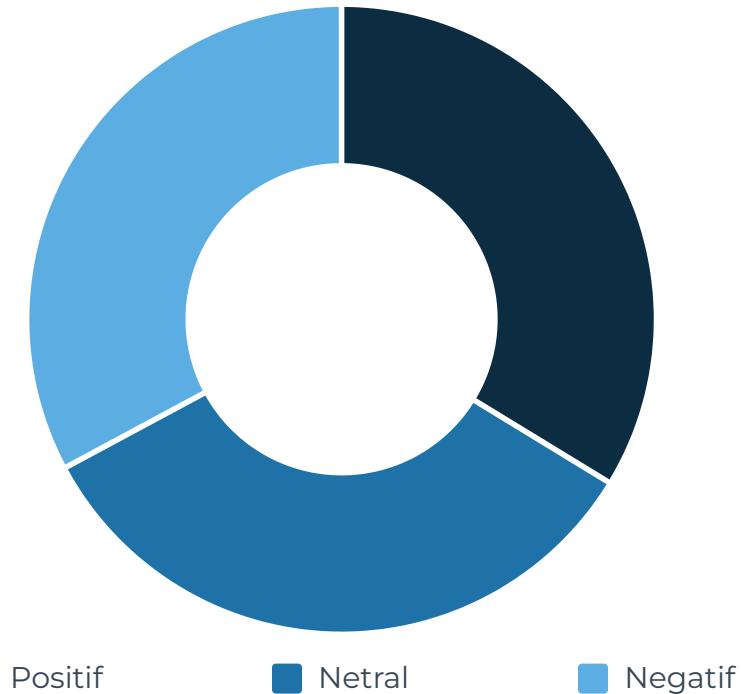
Dataset yang digunakan terdiri dari 1,815 baris data tweet. Analisis awal menunjukkan distribusi sentimen yang seimbang, menjadikannya ideal untuk pelatihan model klasifikasi.

Statistik Dataset

- Jumlah Data: **1,815** baris
- Jumlah Kolom: 3 (Unnamed: 0, sentimen, tweet)
- Rata-rata Panjang Tweet: **194.7 karakter**
- Median Panjang Tweet: **200 karakter**
- Panjang Maksimum: **668** karakter

Dataset memiliki variasi panjang teks yang wajar (sebagian besar 130–250 karakter) dan distribusi sentimen yang seimbang.

Distribusi Sentimen





Tahapan Preprocessing (Emoji-Aware)

Pipeline preprocessing dirancang khusus untuk Bahasa Indonesia dan bertujuan untuk mempertahankan sinyal emosi yang terkandung dalam emoji, yang sering diabaikan dalam proses normalisasi teks.



1. Lowercasing

Mengubah semua teks menjadi huruf kecil untuk konsistensi.



2. Pembersihan Teks

Menghapus URL, namun **mempertahankan hashtag (#) dan mention (@)** sebagai fitur penting.



3. Konversi Emoji

Menggunakan `emoji.demojize()` untuk mengubah emoji menjadi token teks (misal: 😊 → emoji_sob).

4. Tokenisasi Emoji

Mengelompokkan token emoji ke kategori umum: `emoji_positive` (😂 😃 😍) atau `emoji_negative` (😡 😭 😢).

5. Hapus Stopword

Menghapus kata-kata umum Bahasa Indonesia menggunakan `Sastrawi.StopWordRemoverFactory`.

6. Stemming

Menggunakan `Sastrawi.StemmerFactory` untuk mengembalikan kata ke bentuk dasarnya.

7. Normalisasi

Normalisasi spasi dan pembersihan akhir.

Contoh Preprocessing dan Ekstraksi Fitur

Proses konversi emoji menjadi token teks dan pengelompokannya memastikan bahwa sinyal emosi tidak hilang, melainkan diubah menjadi fitur yang dapat diproses oleh model.

Contoh Transformasi Teks

Asli	"Aduh... kecewa banget 😢"
Setelah Preprocessing	aduh kecewa banget emoji_negative

Ekstraksi Fitur – TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) digunakan untuk mengubah teks yang telah diproses menjadi representasi numerik (vektor fitur).

- Parameter: `ngram_range=(1,2)` (Unigram dan Bigram)
- Token Emoji dan Hashtag: **Dipertahankan** dalam vektor fitur.
- Skala: Menggunakan skala logaritmik (`sublinear_tf=True`).
- Filter: `min_df=2, max_df=0.95, max_features=30000`.



Model Klasifikasi yang Digunakan

Dua model Machine Learning klasik dipilih untuk klasifikasi sentimen. Data dibagi menjadi 80% untuk pelatihan dan 20% untuk pengujian menggunakan `train_test_split` dengan stratifikasi.



Multinomial Naive Bayes (MNB)

Dipilih sebagai **baseline** karena kecepatannya dan efisiensi yang baik untuk klasifikasi teks, berdasarkan asumsi independensi fitur.



Logistic Regression (LR)

Model klasifikasi linear yang kuat, menggunakan regularisasi L2. Model ini sering memberikan performa yang lebih baik daripada Naive Bayes pada data teks yang kompleks.

Pipeline yang diterapkan adalah: **TF-IDF → Classifier**.



Hasil Evaluasi Kuantitatif

Evaluasi model menunjukkan bahwa Logistic Regression memberikan performa terbaik pada data pengujian, terutama dalam menangani label sentimen Negatif dan Netral.

MultinomialNB	0.8467	0.590	0.590	Baseline stabil
LogisticRegression	0.9222	0.624	0.625	Performa terbaik ✓

0.625

Akurasi LR

Akurasi tertinggi pada set pengujian, menunjukkan kemampuan generalisasi yang lebih baik.

0.9222

F1 Train LR

Nilai F1 yang tinggi pada data latih, menunjukkan model belajar dengan baik.

3.5%

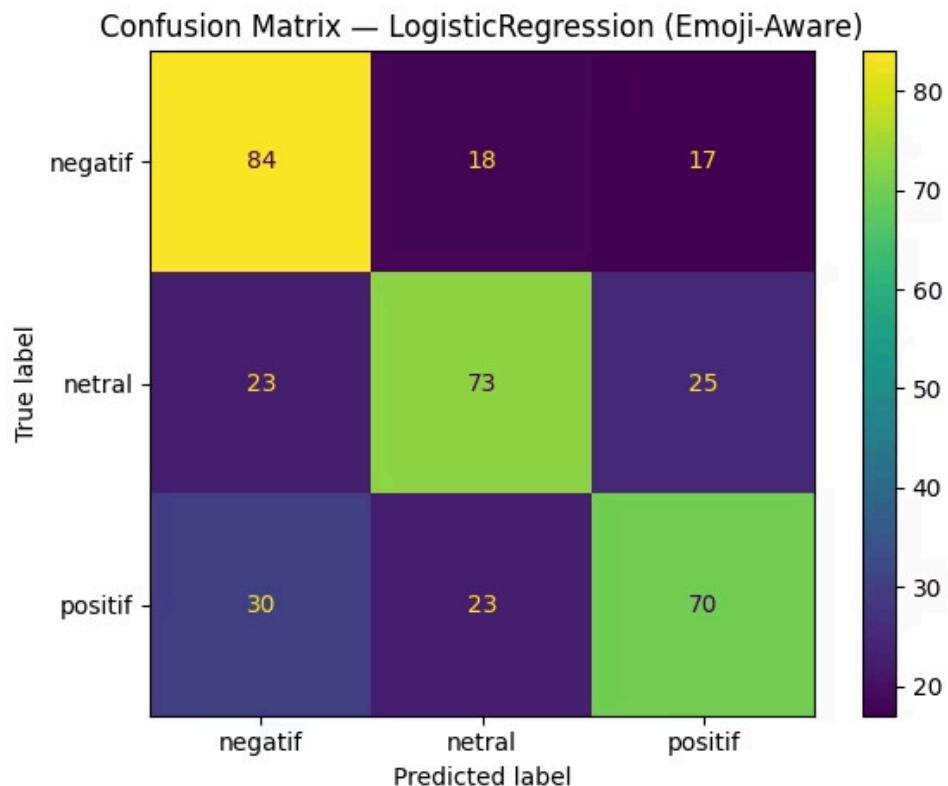
Peningkatan F1

Peningkatan Macro F1 Test sebesar 3.5% dibandingkan Multinomial Naive Bayes.

Visualisasi Performa: Logistic Regression

Model Logistic Regression (LR) yang terpilih menunjukkan performa yang cukup baik, meskipun terdapat tantangan dalam membedakan sentimen Netral dan Negatif, seperti yang terlihat pada Confusion Matrix.

✖️ Confusion Matrix (LR Emoji-Aware)



📈 F1-Score Logistic Regression

== LogisticRegression ==

Macro F1 (train): 0.9222

Macro F1 (test) : 0.6244

	precision	recall	f1-score	support
negatif	0.613	0.706	0.656	119
netral	0.640	0.603	0.621	121
positif	0.625	0.569	0.596	123
accuracy			0.625	363
macro avg	0.626	0.626	0.624	363
weighted avg	0.626	0.625	0.624	363

F1-Score per kelas menunjukkan performa yang paling kuat pada kelas Positif, sejalan dengan hasil Confusion Matrix.

Matrix menunjukkan bahwa LR cenderung lebih akurat dalam mengklasifikasikan sentimen Positif, namun masih terjadi kebingungan antara label Netral dan Negatif.

Perbandingan F1-Score dengan Naive Bayes

Perbandingan visual F1-Score antara Logistic Regression dan Multinomial Naive Bayes menegaskan keunggulan LR dalam klasifikasi sentimen Bahasa Indonesia pada dataset ini.

F1-Score Multinomial Naive Bayes

==== MultinomialNB ===				
	precision	recall	f1-score	support
negatif	0.559	0.639	0.596	119
netral	0.660	0.562	0.607	121
positif	0.565	0.569	0.567	123
accuracy			0.590	363
macro avg	0.595	0.590	0.590	363
weighted avg	0.595	0.590	0.590	363

Naive Bayes menunjukkan F1-Score yang lebih rendah secara keseluruhan (Macro F1: 0.590), terutama pada kelas Netral dan Negatif.

Keunggulan Logistic Regression

Akurasi Lebih Tinggi

LR mencapai akurasi 0.625, mengungguli MNB.

Regulasi L2

Regularisasi membantu mencegah overfitting dan meningkatkan generalisasi model.

Fokus pada Batas Keputusan

Sebagai model diskriminatif, LR lebih efektif dalam menentukan batas keputusan antar kelas sentimen.



Inferensi dan Penyimpanan Model

Model terbaik (Logistic Regression) dan asset terkait telah disimpan untuk memudahkan deployment dan inferensi pada teks baru.

Contoh Inferensi Teks Baru

Text : Mantap banget acaranya! 😁🔥 #keren

Clean : mantap banget acaranya emoji_positive keren

Pred : **positif** | prob: 0.91

Inferensi dapat dijalankan melalui skrip Python (utils/inference_emoji.py) atau langsung menggunakan fungsi predict_sentiment.

```
from utils.inference_emoji import predict_sentiment  
predict_sentiment("Aduh kecewa banget")
```



Rencana Pengembangan dan Dependensi

Proyek ini memiliki potensi pengembangan lebih lanjut, terutama dalam eksplorasi model yang lebih canggih dan optimasi fitur.

Rencana Pengembangan Selanjutnya



Transformer Berbasis IndoBERT

Implementasi model berbasis transformer untuk meningkatkan akurasi klasifikasi sentimen secara signifikan.



Visualisasi Frekuensi Emoji

Analisis mendalam mengenai korelasi antara frekuensi emoji spesifik dengan masing-masing kelas sentimen.



Dashboard Real-Time

Pembangunan dashboard menggunakan Streamlit atau FastAPI untuk analisis sentimen secara langsung.



Optimasi TF-IDF

Eksplorasi grid search dan n-gram yang lebih luas untuk penyempurnaan ekstraksi fitur.



Dependensi Utama



pandas, numpy

Manipulasi dan pemrosesan data.



scikit-learn

Model ML, TF-IDF, dan evaluasi.



Sastrawi

Stemming dan stopword Bahasa Indonesia.



emoji

Konversi emoji menjadi token teks.

