

Kelompok 9:

- Kevin Erdianto Simon - 2702223683
- Vincent Jeremy Mandang - 2702378363
- Ivan Feri Budiman - 2702377606

Kelas: LA01

Jurusan: Computer Science

Mata Kuliah: Natural Language Processing

# Laporan Evaluasi dan Perbandingan Model Sentiment Analysis pada Dataset Twitter

Link Google Collab

<https://colab.research.google.com/drive/1zbBre9An7tz82pSyyTK7IjqUBFjwxx1M?usp=sharing>

## Pendahuluan

Dalam project ini, dilakukan experiment untuk mengklasifikasikan tweet content ke dalam tiga kelas sentimen yaitu negative, neutral, dan positive, berdasar dataset [Twitter Dataset for Sentiment Analysis](#) dari Kaggle.

Project kami akan membandingkan performa antara model RoBERTa, BERT, DistilBERT, dan metode klasik TF-IDF + Logistic Regression yang akan dievaluasi performanya menggunakan metrik yang sesuai seperti Accuracy, F1-Score, dan ROC AUC untuk konteks klasifikasi multiclass.

## Dataset

- Sumber: <https://www.kaggle.com/dunyajasim/twitter-dataset-for-sentiment-analysis>
- Ukuran: ~4.7k tweet
- Kolom: File Name, caption, LABEL
- Sentiment: Neutral, Positive, Negative

## Model

- BERT (bert-base-uncased) → Model Terbaik berdasarkan performa
- RoBERTa (roberta-base)
- DistilBERT (distilbert-base-uncased)
- TF-IDF + Logistic Regression → sebagai baseline klasik

## Metodologi dan Kontrol Variabel

- Digunakan data split yang sama untuk train/validation pada semua model.
- Semua model transformer dilatih menggunakan parameter umum: 2 epochs, batch size 16/64, dan weight decay 0.01.
- Model klasik menggunakan TF-IDF Vectorizer dengan Logistic Regression tanpa oversampling/undersampling.

## Evaluasi Model

Model	Accuracy	F1 Score	ROC AUC
BERT (Top Performer)	0.7885	0.7873	0.9258
RoBERTa	0.7433	0.7411	0.8999
DistilBERT	0.7823	0.7805	0.9231
TF-IDF + LogisticRegression	0.6869	0.6878	0.8437

BERT berhasil menjadi model terbaik secara keseluruhan. Ia menghasilkan metrik tertinggi baik pada Accuracy, F1 Score, maupun ROC AUC, menunjukkan bahwa arsitektur model ini mampu memahami konteks kalimat secara mendalam, bahkan ketika struktur kalimat tidak eksplisit (misalnya tweet informal, slang, atau sarcasm).

DistilBERT, meskipun versi ringan dari BERT, menunjukkan performa kompetitif yang nyaris menyamai BERT.

RoBERTa, meskipun memiliki arsitektur lebih baru dan sering unggul dalam banyak benchmark, dalam eksperimen ini justru sedikit tertinggal. Hal ini mungkin disebabkan oleh kebutuhan fine-tuning RoBERTa yang lebih kompleks.

TF-IDF + Logistic Regression tetap menjadi baseline yang baik, namun memiliki performa yang jauh tertinggal dibanding model transformer karena tidak mempertimbangkan konteks atau urutan kata, hanya frekuensi kata yang berdiri sendiri.

Adapun hal yang masih dapat dikembangkan diantaranya:

- Memberikan data yang lebih berkualitas untuk model (hapus URL, tag mention, emoji noise. Lalu lakukan text normalization)
- Fine-tune hyperparameter seperti learning rate schedule, penambahan epoch
- Menambahkan variasi data

## Arsitektur BERT

**Tabel Hyperparameter BERT**

Hyperparameter	Nilai	Keterangan
Model Dasar	bert-base-uncased	Versi dasar dari BERT
Learning Rate	2e-5	Umum digunakan untuk fine-tuning BERT
Batch Size	16 atau 32	Disesuaikan dengan kapasitas GPU
Epochs	3–5	Rentang ideal untuk menghindari overfitting
Optimizer	AdamW	Optimizer khusus dengan weight decay
Panjang Maksimum	128 token	Disesuaikan dengan panjang rata-rata tweet
Dropout	0.1	Untuk regularisasi

Model BERT (Bidirectional Encoder Representations from Transformers) dipilih untuk project NLP ini karena memiliki performa yang baik diantara model-model lainnya dalam mengklasifikasi sentiment pada dataset yang digunakan. BERT membaca kalimat secara bidirectional, sehingga dapat memahami konteks kata berdasarkan kata-kata sebelum dan sesudahnya. Dibanding RoBERTa yang butuh lebih banyak data/fine-tuning atau DistilBERT yang ringan tapi sedikit lebih lemah, BERT menawarkan keseimbangan antara akurasi dan stabilitas.

Tingkatan arsitektur BERT:

1. Input Layer

BERT menerima input dalam bentuk token-token dan menambahkan representasi sebagai berikut:

- [CLS] token di awal setiap input: digunakan untuk klasifikasi.
- [SEP] token untuk memisahkan dua segmen (kalimat A dan B) — tapi hanya kalimat tunggal pada kasus ini.
- Position Embedding: menyandikan posisi tiap token dalam kalimat.
- Segment Embedding: membedakan dua kalimat (tidak digunakan di klasifikasi tunggal).
- Token Embedding: hasil tokenisasi dari model bert-base-uncased.

2. Transformer Encoder Stack (12 Layer)

Setiap layer Transformer memiliki dua komponen utama:

- a. Multi-Head Self-Attention

- o Membagi hidden state menjadi 12 "kepala" untuk menangkap konteks berbeda dalam kalimat.
- o Attention memperbolehkan setiap token memperhatikan token lain dalam kalimat secara bidirectional.

- b. Feed Forward Network (FFN)

- Layer fully connected dua tahap: Linear > ReLU > Linear

Setiap layer juga mempunyai:

- Residual Connection
- Layer Normalization
- Dropout

3. Output dari [CLS] Token

- Vektor representasi dari token [CLS] yang keluar dari layer ke-12 digunakan sebagai representasi kalimat.
- Vektor ini diteruskan ke layer klasifikasi Linear + Softmax untuk menghasilkan skor dari 3 kelas (negatif, netral, positif).