

NAMA : Gian Rofiqri Andra

NIM : 24917036

Analisis Sentimen Terhadap Isu Deforestasi pada Media Sosial Twitter Berbahasa Indonesia

1. PENDAHULUAN

1.1 Latar Belakang

Deforestasi merupakan salah satu isu lingkungan yang paling kritis di Indonesia. Sebagai negara dengan hutan hujan tropis terbesar ketiga di dunia, Indonesia menghadapi tantangan besar dalam menjaga kelestarian hutannya. Media sosial, khususnya Twitter (X), menjadi platform penting bagi masyarakat untuk mengekspresikan opini mereka tentang isu-isu lingkungan termasuk deforestasi.

Analisis sentimen terhadap tweet berbahasa Indonesia tentang deforestasi dapat memberikan wawasan berharga tentang persepsi publik terhadap masalah ini. Dengan memanfaatkan teknologi Natural Language Processing (NLP) dan model pre-trained seperti IndoBERT, dapat mengklasifikasikan sentimen publik secara otomatis dan akurat.

1.2 Tujuan Penelitian

Tujuan dari tugas akhir ini adalah:

- a. Memahami proses fine-tuning model pre-trained untuk task text classification
- b. Melakukan hyperparameter tuning untuk optimasi performa model
- c. Mengevaluasi performa model menggunakan metrik yang sesuai
- d. Membandingkan beberapa pre-trained language model untuk bahasa Indonesia

1.3 Jenis Task

Task yang dipilih: Text Classification (Sentiment Analysis) - mengklasifikasikan tweet tentang deforestasi ke dalam dua kelas sentimen: Positif dan Negatif

2. DESKRIPSI DATASET

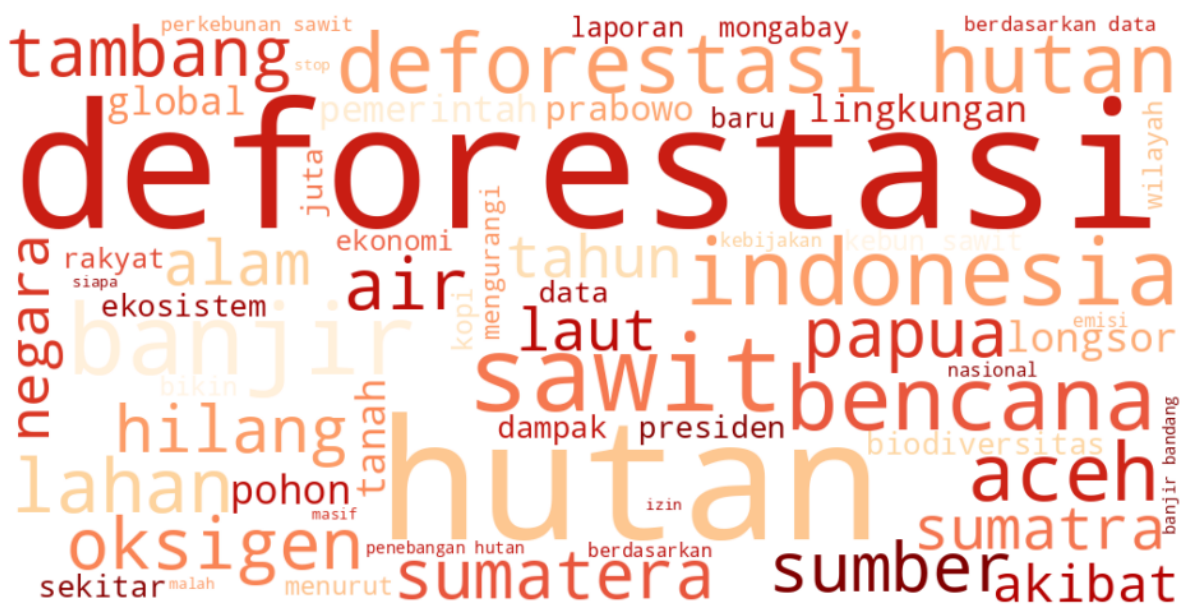
2.1 Sumber Data

Dataset dikumpulkan dari platform Twitter (X) menggunakan tool Tweet Harvest. Data dikumpulkan berdasarkan kata kunci yang berkaitan dengan deforestasi dan pengundulan hutan di Indonesia.

2.2 Spesifikasi Data

Atribut	Keterangan
Sumber Data	Twitter (X)
Topik	Deforestasi & Pengundulan Hutan
Periode Pengumpulan	20 November - 20 Desember 2025
Ukuran Dataset Awal	1.016 tweets
Dataset Setelah Filter	343 tweets (min. 5 kata)
Dataset Final (Binary)	240 tweets (tanpa Netral)
Jumlah Kolom Original	15 kolom
Jumlah Kelas	2 kelas (Positif, Negatif)

Kata Yang Sering Muncul



2.3 Distribusi Label

Label	Jumlah	Persentase
Negatif	185	53.9%
Netral	103	30.0%
Positif	55	16.0%
Total	343	100%

Catatan: Baris merah menunjukkan label Netral yang kemudian dihapus dari dataset final.

3. PREPROCESSING DAN TOKENISASI

3.1 Tahapan Preprocessing

Preprocessing dilakukan untuk membersihkan teks tweet agar siap digunakan untuk training model. Tahapan yang dilakukan:

1. Case Folding: Mengubah semua teks menjadi huruf kecil (lowercase)
2. Pembersihan URL: Menghapus semua URL (http://, https://)
3. Pembersihan Mention: Menghapus semua mention (@username)
4. Pembersihan Hashtag: Menghapus simbol hashtag (#)
5. Pembersihan Karakter Non-Alfabet: Menghapus angka, simbol, dan karakter khusus
6. Normalisasi Spasi: Menghapus spasi berlebih dan whitespace
7. Filter Panjang Teks: Menghapus tweet dengan kurang dari 5 kata (biasanya spam atau tidak informatif)

3.2 Fungsi Preprocessing

Implementasi fungsi preprocessing menggunakan regular expression (regex):

```
def clean_text(text):
    text = str(text).lower()
    text = re.sub(r'http\S+|@\w+|#\w+', '', text)
    text = re.sub(r'^a-z\s', '', text)
    text = re.sub(r'\s+', ' ', text).strip()
    return text

df['cleaned_text'] = df['full_text'].apply(clean_text)
print("Pembersihan teks selesai.")
```

3.3 Labeling Otomatis

Proses labeling dilakukan menggunakan metode context-aware pattern matching. Metode ini menggunakan pola regex yang mempertimbangkan konteks kalimat, bukan hanya menghitung keyword. Sistem ini menghasilkan label (Positif/Negatif/Netral) beserta confidence score untuk setiap tweet.

Pola Positif mencakup:

- Keberhasilan menekan deforestasi
- Aktivitas reboisasi dan restorasi
- Konservasi dan perlindungan hutan
- Kebijakan pro-lingkungan

Pola Negatif mencakup:

- Dampak buruk deforestasi (banjir, longsor, kekeringan)
- Kritik terhadap penebangan liar
- Kekhawatiran tentang kerusakan lingkungan
- Peningkatan deforestasi

3.4 Tokenisasi

Tokenisasi dilakukan menggunakan tokenizer dari model pre-trained yang akan di-fine-tune. Setiap model memiliki tokenizer masing-masing yang dioptimalkan untuk arsitekturnya. Parameter tokenisasi yang digunakan:

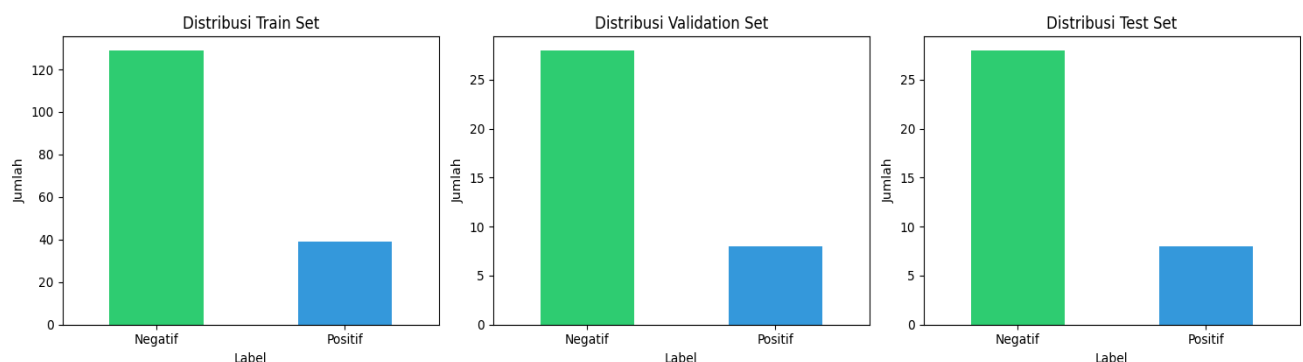
- max_length: 128 tokens - Panjang maksimum sequence
- padding: 'max_length' - Padding ke panjang maksimum
- truncation: True - Memotong teks yang melebihi max_length

3.5 Pembagian Data

Data dibagi menjadi tiga set dengan proporsi:

Set	Proporsi	Jumlah
Training	70%	168
Validation	15%	36
Test	15%	36

Stratified Split: Pembagian data menggunakan stratified sampling untuk memastikan proporsi label yang sama di setiap set, menjaga representasi kelas yang seimbang.



3.6 Handling Imbalance

Karena dataset tidak seimbang (imbalanced), digunakan Class Weights untuk memberikan bobot lebih tinggi pada kelas minoritas (Positif). Bobot dihitung menggunakan metode 'balanced' dari sklearn:

Kelas	ID	Weight
Negatif	0	0.651
Positif	1	2.154

4. LABELING DENGAN CONFIDENCE SCORE

4.1 Metode Labeling Otomatis

Proses labeling dilakukan menggunakan metode Context-Aware Pattern Matching. Metode ini menggunakan pola regex yang mempertimbangkan konteks kalimat, bukan hanya menghitung kemunculan keyword secara sederhana. Sistem ini menghasilkan tiga output untuk setiap tweet:

- Label: Positif, Negatif, atau Netral
- Confidence Score: Nilai 0-1 yang menunjukkan tingkat kepercayaan terhadap label
- Keywords: Daftar kata kunci positif dan negatif yang terdeteksi

4.2 Pola Pattern Matching

Pola Positif mencakup:

- Keberhasilan menekan/menurunkan deforestasi
- Aktivitas reboisasi, restorasi, dan penghijauan
- Konservasi dan perlindungan hutan
- Kebijakan pro-lingkungan (moratorium, larangan penebangan)

Pola Negatif mencakup:

- Dampak buruk deforestasi (banjir, longsor, kekeringan, kebakaran)
- Kritik terhadap penebangan liar dan illegal logging
- Kekhawatiran tentang kerusakan lingkungan
- Peningkatan/meluasnya deforestasi

4.3 Confidence Score

Confidence Score adalah nilai numerik antara 0 hingga 1 yang mengindikasikan seberapa yakin sistem terhadap label yang diberikan. Skor ini dihitung berdasarkan kekuatan sinyal dari keyword yang terdeteksi dan seberapa jelas konteks kalimat menunjukkan sentimen tertentu.

Kategori Confidence:

Kategori	Range Nilai	Keterangan
High Confidence	≥ 0.7	Label dapat diterima otomatis
Medium Confidence	0.5 - 0.7	Perlu pertimbangan, mungkin perlu review
Low Confidence	< 0.5	Perlu review manual

4.4 Distribusi Confidence Score

Hasil distribusi confidence score pada 343 tweets yang sudah dilabeli:

Kategori Confidence	Jumlah Tweets
High Confidence (≥ 0.7)	178 tweets
Medium Confidence (0.5 - 0.7)	2 tweets
Low Confidence (< 0.5)	163 tweets

Interpretasi: Sekitar 52% tweets memiliki high confidence, sementara 48% memiliki low confidence yang mengindikasikan bahwa tweet tersebut memiliki sentimen yang ambigu atau tidak jelas. Tweets dengan low confidence sebaiknya di-review secara manual untuk memastikan akurasi label.

4.5 Contoh Tweet dengan Confidence Score

Contoh tweet dengan Low Confidence yang perlu review:

Tweet: *"Kalo lu belajar ekologi harusnya tau juga bahayanya tanaman homogen di wilayah..."*

Label: Negatif | Confidence: 0.45

Neg keywords: ['banjir', 'bencana', 'bahaya']

Pos keywords: ['tanam']

Analisis: Tweet ini memiliki confidence rendah (0.45) karena mengandung campuran keyword positif ('tanam') dan negatif ('banjir', 'bencana', 'bahaya'). Sistem mendeteksi lebih banyak sinyal negatif sehingga dilabel Negatif, namun dengan confidence yang rendah karena adanya konflik sinyal

4.6 Penghapusan Label Netral

Dari 343 tweets yang sudah dilabeli, terdapat 103 tweets (30%) dengan label Netral. Keputusan untuk menghapus label Netral dan hanya menggunakan klasifikasi biner (Positif/Negatif) diambil dengan pertimbangan berikut:

Alasan Penghapusan Label Netral:

- Kualitas Label Netral yang Rendah: Tweet yang dilabel Netral umumnya adalah tweet yang tidak memiliki sinyal sentimen yang kuat (tidak ada keyword positif maupun negatif yang terdeteksi). Ini bukan berarti tweet tersebut benar-benar netral, melainkan sistem tidak mampu menentukan sentimennya dengan pasti.
- Confidence Score Rendah pada Netral: Sebagian besar tweet Netral memiliki confidence score yang sangat rendah (rata-rata 0.3), menunjukkan ketidakpastian sistem dalam memberikan label tersebut.
- Fokus pada Sentimen yang Jelas: Untuk tujuan analisis sentimen, lebih bermanfaat untuk fokus pada tweet yang memiliki sentimen jelas (positif atau negatif) daripada tweet yang ambigu.
- Mengurangi Noise dalam Training: Tweet Netral dengan kualitas label yang rendah dapat menjadi noise yang mengganggu proses training model.
- Simplifikasi Task: Klasifikasi biner (2 kelas) lebih mudah untuk model dengan dataset yang kecil dibanding klasifikasi 3 kelas.

Distribusi Label Final (Setelah Penghapusan Netral)

Label	Jumlah	Persentase
Negatif	185	77.1%
Positif	55	22.9%
Total	240	100%

Catatan: Dataset final menunjukkan ketidakseimbangan kelas (imbalanced) dengan dominasi sentimen Negatif (77.1%). Hal ini terjadi mengingat topik deforestasi umumnya diperbincangkan dalam konteks negatif (kritik, keluhan, atau kekhawatiran). Ketidakseimbangan ini akan ditangani dengan teknik class weighting.

4.7 Pembagian Data

Data dibagi menjadi tiga set dengan proporsi:

Set	Proporsi	Jumlah
Training	70%	168
Validation	15%	36
Test	15%	36

Stratified Split: Pembagian data menggunakan stratified sampling untuk memastikan proporsi label yang sama di setiap set, menjaga representasi kelas yang seimbang.

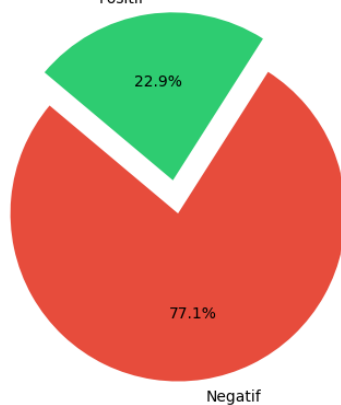
4.8 Handling Imbalance dengan Class Weights

Karena dataset tidak seimbang (imbalanced), digunakan Class Weights untuk memberikan bobot lebih tinggi pada kelas minoritas (Positif). Bobot dihitung menggunakan metode 'balanced' dari sklearn:

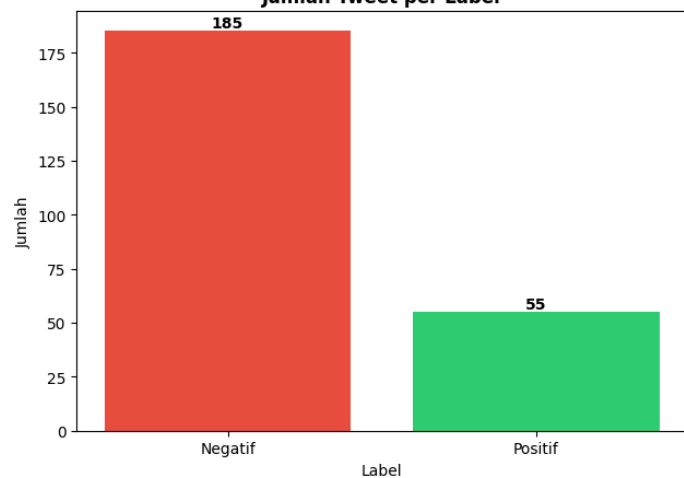
Kelas	ID	Weight
Negatif	0	0.651
Positif	1	2.154

Interpretasi: Kelas Positif (minoritas) memiliki bobot 3.3x lebih tinggi dari kelas Negatif (mayoritas). Ini berarti kesalahan pada prediksi kelas Positif akan memberikan penalti lebih besar pada loss function, sehingga model terdorong untuk lebih memperhatikan kelas minoritas.

**Distribusi Sentiment
Tweet Deforestasi (Filtered)**



Jumlah Tweet per Label



5. PEMILIHAN PRE-TRAINED LANGUAGE MODEL

5.1 Model yang Digunakan

Tiga model pre-trained dari Hugging Face Model Hub dipilih untuk eksperimen perbandingan. Semua model ini mendukung bahasa Indonesia:

Model	Model ID	Deskripsi
IndoLEM IndoBERT	indolem/indobert-base-uncased	BERT untuk bahasa Indonesia dari IndoLEM
IndoBenchmark IndoBERT	indobenchmark/indobert-base-pl	BERT untuk bahasa Indonesia dari IndoBenchmark
mBERT Multilingual	bert-base-multilingual-uncased	BERT multilingual (104 bahasa termasuk Indonesia)

5.2 Justifikasi Pemilihan Model

IndoBenchmark IndoBERT (indobert-base-pl):

- Dilatih khusus pada korpus bahasa Indonesia yang besar
- Memiliki performa state-of-the-art pada berbagai benchmark bahasa Indonesia
- Vocabulary yang dioptimalkan untuk bahasa Indonesia

IndoLEM IndoBERT:

- Alternatif model bahasa Indonesia dari tim IndoLEM
- Digunakan sebagai pembanding antar varian IndoBERT

mBERT Multilingual:

- Baseline untuk membandingkan model multilingual vs model spesifik bahasa
- Mendukung 104 bahasa termasuk Indonesia

6. IMPLEMENTASI FINE-TUNING

6.1 Pendekatan Fine-Tuning

Pendekatan yang digunakan adalah Full Fine-Tuning, di mana semua parameter model di-update selama proses training. Pendekatan ini dipilih karena:

- Dataset relatif kecil sehingga risiko overfitting dapat dikontrol dengan early stopping
- Full fine-tuning memberikan fleksibilitas maksimal untuk adaptasi ke task spesifik
- Komputasi masih dalam kapasitas yang tersedia (Google Colab GPU)

6.2 Arsitektur Model

Model menggunakan arsitektur `AutoModelForSequenceClassification` dari library `Transformers`. Arsitektur ini menambahkan classification head di atas pre-trained BERT:

- Pre-trained BERT encoder (12 layers, 768 hidden size)
- Dropout layer untuk regularisasi
- Linear classification head ($768 \rightarrow 2$ classes)

6.3 Custom Trainer dengan Class Weights

Untuk menangani data imbalanced, dibuat `WeightedTrainer` yang memodifikasi loss function dengan menerapkan class weights pada `CrossEntropyLoss`:

```
class WeightedTrainer(Trainer):
    .....
    def compute_loss(self, model, inputs, return_outputs=False, **kwargs):
    .....
    loss_fn = nn.CrossEntropyLoss(weight=weight)
```

6.4 Training Configuration

Parameter	Nilai
Evaluation Strategy	Per epoch
Save Strategy	Per epoch
Weight Decay	0.01
Load Best Model at End	True
Metric for Best Model	Accuracy

7. HYPERPARAMETER TUNING

7.1 Hyperparameter yang Dieksplorasi

Model yang digunakan untuk proses hyperparameter tuning adalah RoBERTa Indonesian (cahya/roberta-base-indonesian-1.5G). Model ini dipilih sebagai baseline untuk menemukan konfigurasi hyperparameter optimal sebelum dilakukan perbandingan dengan model lain. Tiga hyperparameter utama dieksplorasi dengan justifikasi berikut:

1. Learning Rate

- Nilai yang diuji: 1e-5, 2e-5, 5e-5
- Justifikasi: Range standar untuk fine-tuning BERT. Learning rate yang terlalu tinggi dapat menyebabkan catastrophic forgetting, sedangkan yang terlalu rendah membuat konvergensi lambat.

2. Batch Size

- Nilai yang diuji: 8, 16
- Justifikasi: Batch size kecil cocok untuk dataset kecil dan memori GPU terbatas. Batch yang lebih kecil memberikan update gradient yang lebih frequent.

3. Number of Epochs

- Nilai yang diuji: 3, 5
- Justifikasi: Fine-tuning BERT umumnya membutuhkan 2-4 epoch. Epoch yang terlalu banyak berisiko overfitting pada dataset kecil.

7.2 Hasil Hyperparameter Tuning

Total 12 eksperimen dilakukan (3 learning rate \times 2 batch size \times 2 epochs):

No	Learning Rate	Batch Size	Epochs	Val Acc	Val Loss
9	5e-05	8	3	0.8889	0.6983
6	2e-05	8	5	0.8889	0.5250
2	1e-05	8	5	0.8611	0.5355
7	2e-05	16	3	0.8611	0.5744
11	5e-05	16	3	0.8611	0.4173
1	1e-05	8	3	0.7778	0.6703

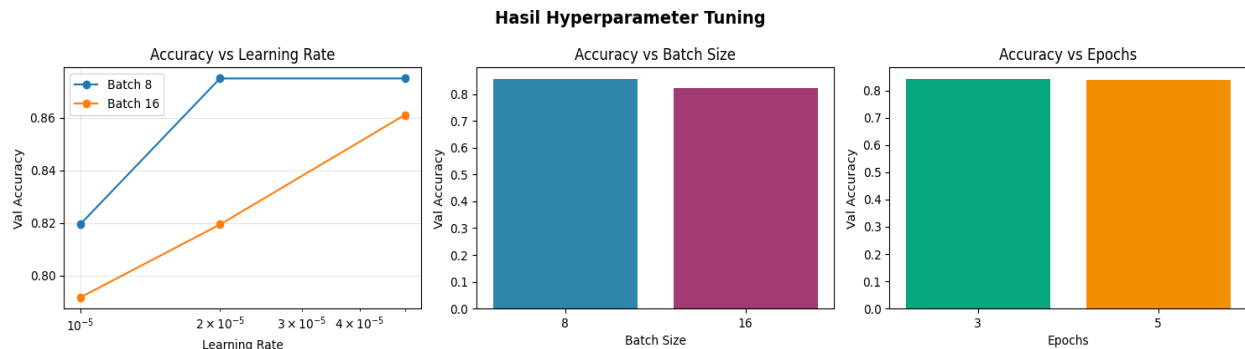
Catatan: Baris hijau menunjukkan konfigurasi dengan performa terbaik.

7.3 Konfigurasi Terbaik

Berdasarkan hasil eksperimen, konfigurasi hyperparameter terbaik adalah:

- Learning Rate: $5e-05$ (tertinggi dari range yang diuji)
- Batch Size: 8
- Epochs: 3

Validation Accuracy: 88.89%



8. PERBANDINGAN MODEL

8.1 Setup Eksperimen

Tiga model pre-trained dibandingkan menggunakan konfigurasi hyperparameter terbaik yang diperoleh dari tahap tuning sebelumnya (LR= $5e-05$, Batch=8, Epochs=3).

8.2 Hasil Perbandingan

Model	Val Accuracy	Val Loss
IndoBenchmark-IndoBERT	0.8611	0.7940
mBERT-Multilingual	0.8056	0.6772
IndoLEM-IndoBERT	0.7778	0.7901

8.3 Analisis Hasil

IndoBenchmark-IndoBERT (Terbaik):

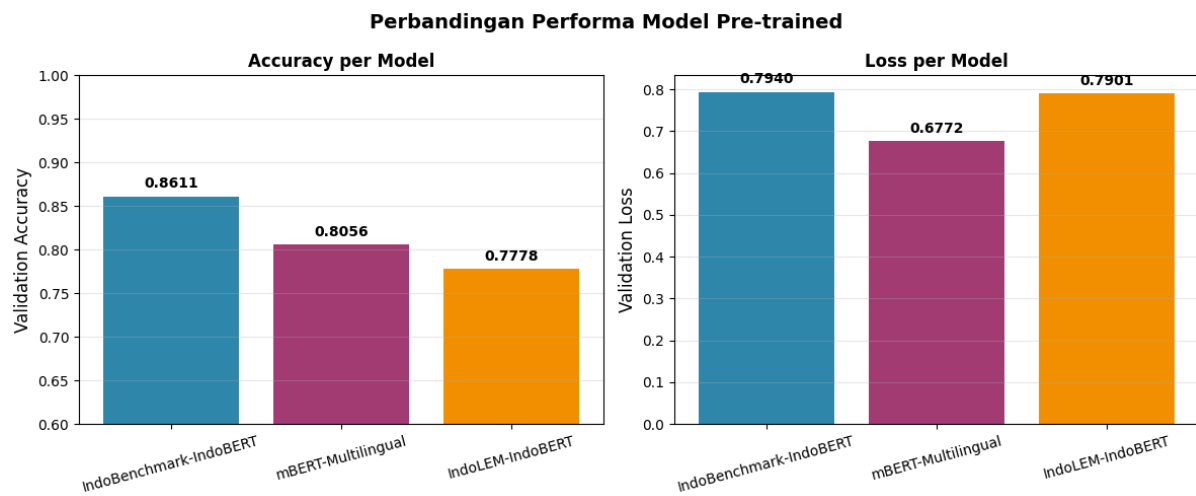
- Mencapai validation accuracy tertinggi (86.11%)
- Dilatih pada korpus Indonesia yang lebih besar dan beragam
- Vocabulary lebih representatif untuk teks Indonesia kontemporer

mBERT-Multilingual:

- Performa cukup baik (80.56%) meski bukan model spesifik Indonesia
- Validation loss terendah menunjukkan model yang lebih stabil
- Cocok sebagai baseline untuk perbandingan

IndoLEM-IndoBERT:

- Performa terendah di antara ketiga model
- Mungkin memerlukan hyperparameter tuning tersendiri



9. EVALUASI DAN HASIL

9.1 Konfigurasi Model Final

Model final menggunakan IndoBenchmark-IndoBERT dengan konfigurasi optimal:

- Model: indobenchmark/indobert-base-p1
- Learning Rate: 5e-05
- Batch Size: 8
- Epochs: 3

9.2 Training Progress

Epoch	Training Loss	Validation Loss	Val Accuracy
1	0.6327	0.5766	66.67%
2	0.2259	0.5439	80.56%
3	0.0117	0.7940	86.11%

9.3 Evaluasi pada Test Set

Test Accuracy: 91.67%

Classification Report

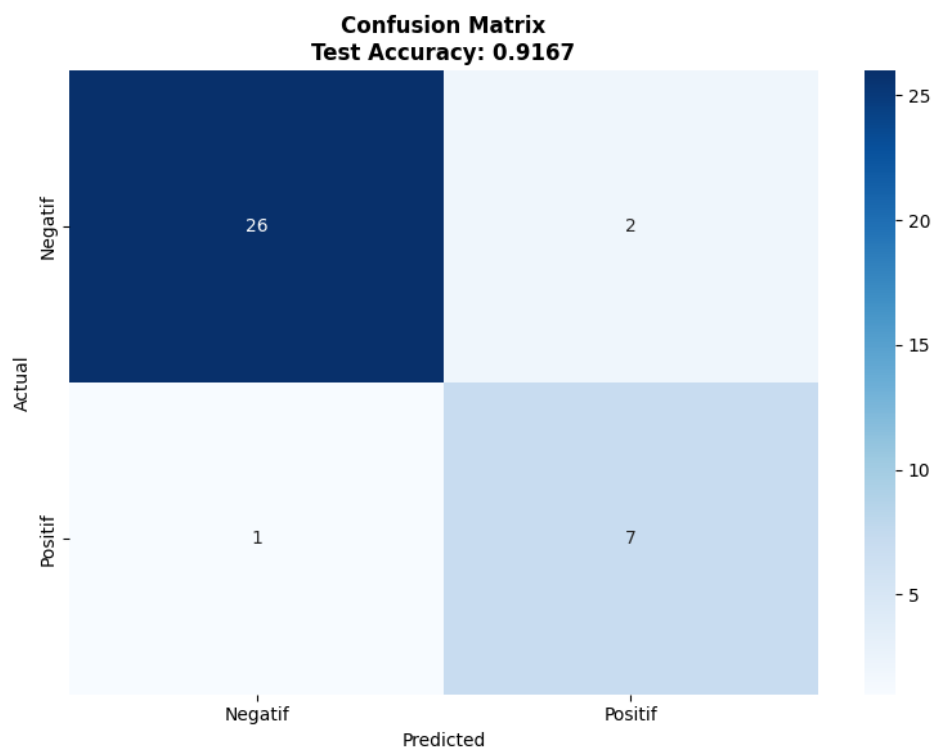
Kelas	Precision	Recall	F1-Score	Support
Negatif	0.96	0.93	0.95	28
Positif	0.78	0.88	0.82	8
Macro Avg	0.87	0.90	0.88	36
Weighted Avg	0.92	0.92	0.92	36

9.4 Analisis Confusion Matrix

Berdasarkan hasil evaluasi pada test set (36 sampel):

- True Negative (Negatif → Negatif): 26 dari 28 (92.9%)
- True Positive (Positif → Positif): 7 dari 8 (87.5%)
- False Positive: 2 (Negatif diprediksi Positif)
- False Negative: 1 (Positif diprediksi Negatif)

Model menunjukkan performa yang sangat baik untuk kelas mayoritas (Negatif) dengan precision 96% dan recall 93%. Untuk kelas minoritas (Positif), meskipun precision lebih rendah (78%), recall tetap tinggi (88%) yang menunjukkan model mampu mendeteksi sebagian besar sentimen positif.



10. KESIMPULAN

10.1 Ringkasan Hasil

Tugas akhir ini berhasil mengimplementasikan sistem analisis sentimen untuk tweet berbahasa Indonesia tentang deforestasi menggunakan teknik fine-tuning pada model pre-trained BERT. Hasil utama yang dicapai:

- a. Test Accuracy: 91.67% - performa sangat baik untuk klasifikasi sentimen
- b. Weighted F1-Score: 0.92 - keseimbangan precision dan recall yang baik
- c. Model terbaik: IndoBenchmark-IndoBERT mengungguli model multilingual
- d. Konfigurasi optimal: LR=5e-05, Batch=8, Epochs=3

10.2 Temuan Penting

- a. Model spesifik bahasa (IndoBERT) lebih efektif dibanding model multilingual untuk task bahasa Indonesia dalam penelitian ini.
- b. Class weighting efektif menangani data imbalanced
- c. Learning rate lebih tinggi (5e-05) memberikan hasil lebih baik untuk dataset kecil
- d. 3 epoch cukup untuk konvergensi tanpa overfitting pada fine-tuning BERT pada penelitian ini.

10.3 Keterbatasan

- a. Dataset relatif kecil (240 sampel setelah filtering)
- b. Labeling otomatis mungkin mengandung noise
- c. Test set yang kecil (36 sampel) membatasi reliabilitas evaluasi