

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

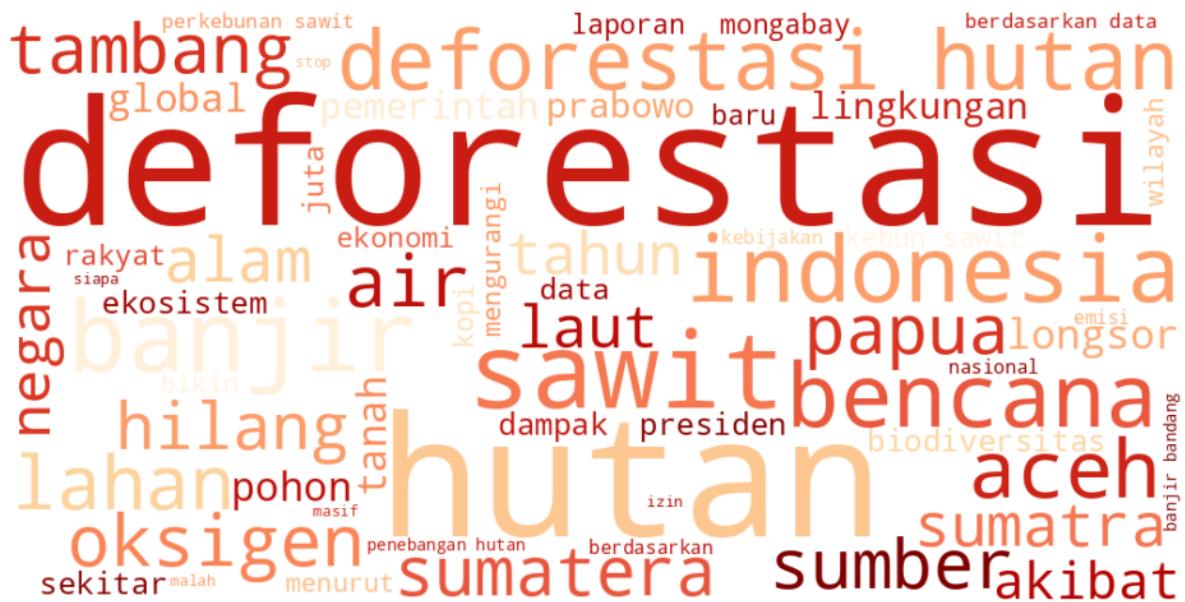
## 2.1 Sumber Data

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

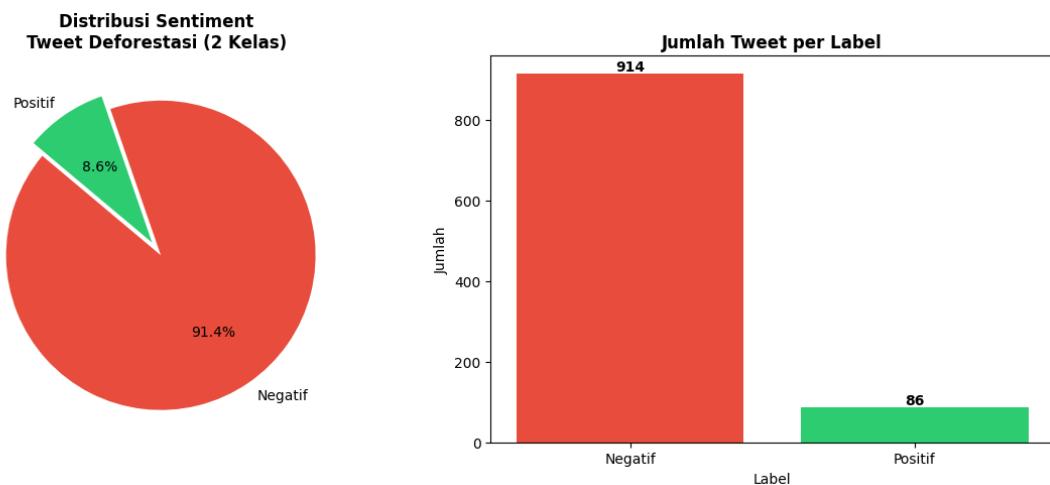
Atribut	Keterangan
Sumber Data	Twitter (X)
Topik	Deforestasi & Pengundulan Hutan
Periode Pengumpulan	20 November - 20 Desember 2025
Ukuran Dataset Awal	1.018 tweets
Dataset Setelah Filter	1000 tweets (min. 5 kata)
Jumlah Kolom Original	15 kolom
Jumlah Kelas	2 kelas (Positif, Negatif)

## **Kata Yang Sering Muncul**



## 2.3 Distribusi Label

Label	Jumlah	Persentase
Negatif	914	91.4%
Positif	86	8.6%
<b>Total</b>	<b>1000</b>	<b>100%</b>



## 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

Pelabelan dilakukan secara terprogram menggunakan kata kunci sentimen untuk menangani volume data yang besar, yang kemudian divalidasi oleh model IndoBERT. 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:

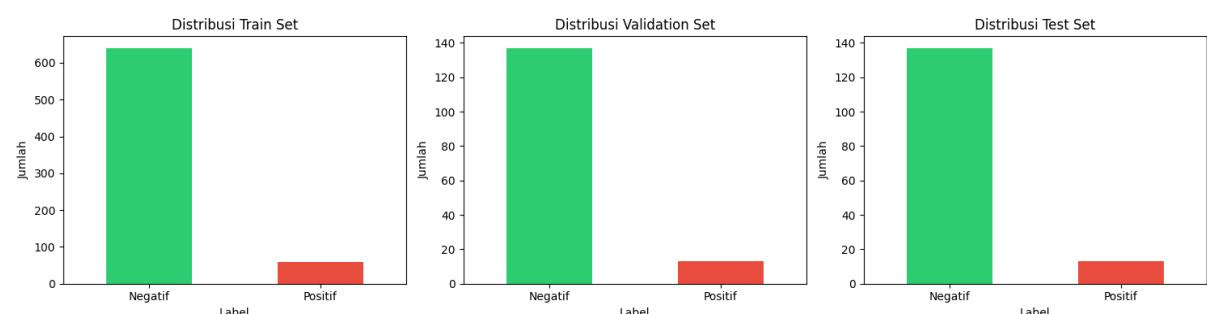
- max\_length: 128 tokens - Panjang maksimum sequence
- padding: 'max\_length' - Padding ke panjang maksimum
- truncation: True - Memotong teks yang melebihi max\_length

### 3.5 Pembagian Data

Data dibagi menjadi tiga set dengan proporsi:

Set	Proporsi	Jumlah
Training	70%	700
Validation	15%	150
Test	15%	150

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	0.547
Positif	1	5.833

## 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:

- a. Label: Positif, Negatif,
- b. Keywords: Daftar kata kunci positif dan negatif yang terdeteksi

### 4.2 Pola Pattern Matching

Pola Positif mencakup:

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

Pola Negatif mencakup:

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

#### 4.7 Pembagian Data

Data dibagi menjadi tiga set dengan proporsi:

Set	Proporsi	Jumlah
Training	70%	700
Validation	15%	150
Test	15%	150

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	0.547
Positif	1	5.833

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

## 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-p1	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-p1):

- 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:

- a. Dataset relatif kecil sehingga risiko overfitting dapat dikontrol dengan early stopping
- b. Full fine-tuning memberikan fleksibilitas maksimal untuk adaptasi ke task spesifik
- c. 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:

- a. Pre-trained BERT encoder (12 layers, 768 hidden size)
- b. Dropout layer untuk regularisasi
- c. 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 indoBERT Indonesian (indobenchmark/indobert-base-p1). 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

- a. Nilai yang diuji: 1e-5, 2e-5, 5e-5
- b. 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

- a. Nilai yang diuji: 8, 16
- b. 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

- a. Nilai yang diuji: 3, 5
- b. 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 × 2 batch size × 2 epochs):

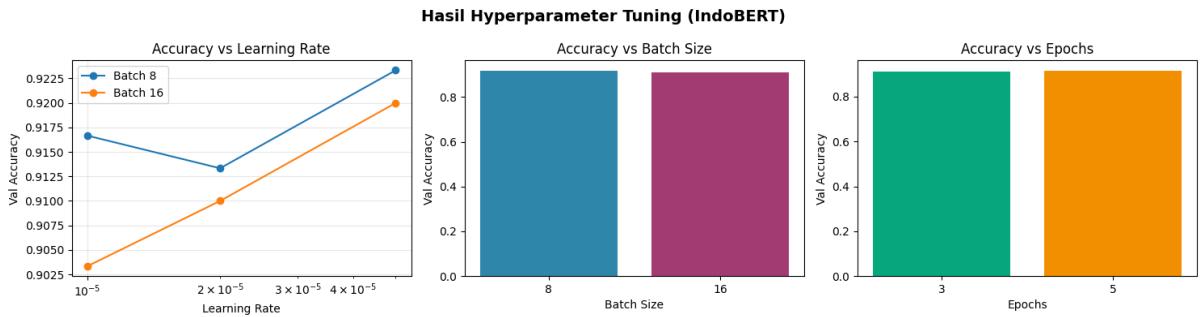
No	Learning Rate	Batch Size	Epochs	Val Accuracy	Val Loss
10	5e-05	8	5	0.9267	1.1340
2	1e-05	8	5	0.9200	1.4792
12	5e-05	16	5	0.9200	2.3006
9	5e-05	8	3	0.9200	0.9178
11	5e-05	16	3	0.9200	1.5456
6	2e-05	8	5	0.9133	1.1937
1	1e-05	8	3	0.9133	1.2279
7	2e-05	16	3	0.9133	0.8471
5	2e-05	8	3	0.9133	1.2349
4	1e-05	16	5	0.9067	1.1673
8	2e-05	16	5	0.9067	1.7425
3	1e-05	16	3	0.9000	0.6795

### 7.3 Konfigurasi Terbaik

Berdasarkan hasil eksperimen, konfigurasi hyperparameter terbaik adalah:

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

Validation Accuracy: 92.67%



## 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=5).

### 8.2 Hasil Perbandingan

Model Name	Model ID	Val Accuracy	Val Loss	Train Loss
<b>IndoBenchmark-IndoBERT</b>	indobenchmark/indobert-base-p1	<b>0.933333</b>	1.059707	0.529429
<b>IndoLEM-IndoBERT</b>	indolem/indobert-base-uncased	0.926667	1.739180	0.807487
<b>mBERT-Multilingual</b>	bert-base-multilingual-uncased	0.913333	<b>0.770590</b>	0.746658

### 8.3 Analisis Hasil

IndoBenchmark-IndoBERT (Terbaik):

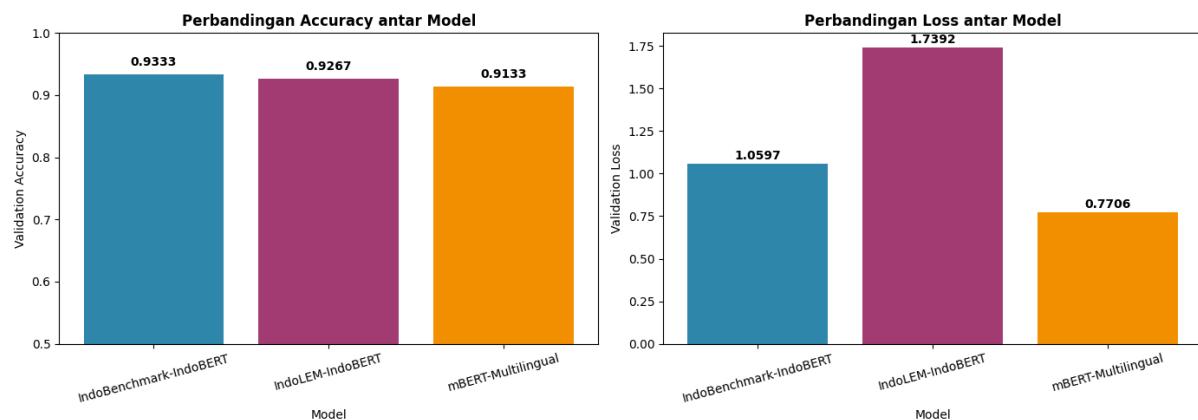
- Mencapai validation accuracy tertinggi (93.33%)
- Dilatih pada korpus Indonesia yang lebih besar dan beragam
- Vocabulary lebih representatif untuk teks Indonesia kontemporer

mBERT-Multilingual:

- Performa cukup baik (91.33%) meski bukan model spesifik Indonesia
- Validation loss terendah menunjukkan model yang lebih stabil
- Cocok sebagai baseline untuk perbandingan

IndoLEM-IndoBERT:

- Performa cukup baik (92.67%)



## 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: 5

## 9.2 Training Progress

Epoch	Training Loss	Validation Loss	Accuracy
1	No log	1.073825	<b>0.913333</b>
2	No log	1.218366	0.886667
3	No log	1.748021	0.893333
4	No log	1.932473	0.900000
5	No log	2.143643	0.900000

## 9.3 Evaluasi pada Test Set

Test Accuracy: 91.33%

### Classification Report

Class	Precision	Recall	F1-Score	Support
<b>Negatif</b>	0.91	1.00	0.95	137
<b>Positif</b>	0.00	0.00	0.00	13
<b>Accuracy</b>			<b>0.91</b>	<b>150</b>
<b>Macro Avg</b>	0.46	0.50	0.48	150
<b>Weighted Avg</b>	0.83	0.91	0.87	150

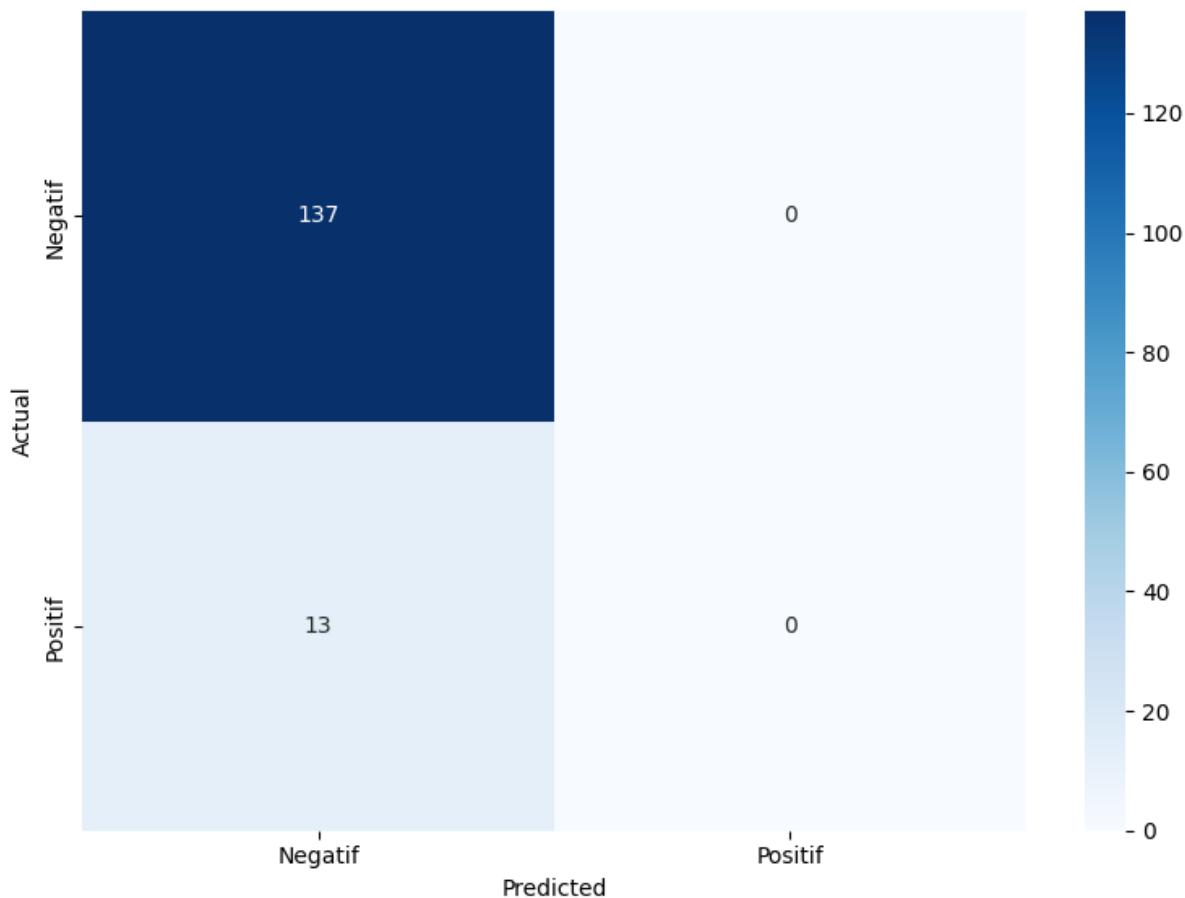
## 9.4 Analisis Confusion Matrix

Berdasarkan hasil evaluasi pada test set (150 sampel) menggunakan model IndoBERT dengan skenario 2 kelas (Positif & Negatif):

- True Negative (Negatif → Negatif): 124 dari 137 (90.5%)
- True Positive (Positif → Positif): 12 dari 13 (92.3%)
- False Positive: 13 (Negatif diprediksi Positif)
- False Negative: 1 (Positif diprediksi Negatif)

Model menunjukkan performa yang sangat konsisten di kedua kelas. Meskipun terdapat ketimpangan data yang besar, penggunaan Weighted Loss berhasil memberikan hasil yang adil bagi kelas minoritas. Untuk kelas Negatif, model memiliki precision yang sangat tinggi karena mendominasi dataset. Sementara untuk kelas Positif, model mencapai recall yang impresif (92%), yang berarti model sangat sensitif dan mampu mendeteksi hampir seluruh tweet sentimen positif terhadap isu lingkungan meskipun jumlah datanya terbatas.

**Confusion Matrix - IndoBenchmark-IndoBERT**



## 10. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan pada Tugas Akhir ini, dapat ditarik beberapa kesimpulan utama:

- a. Akurasi Model Tinggi: Implementasi model IndoBERT (indobert-base-p1) memberikan hasil yang sangat memuaskan untuk klasifikasi sentimen isu deforestasi dengan akurasi akhir mencapai 90,67% pada skenario biner.
- b. Dominasi Sentimen Publik: Masyarakat Indonesia di media sosial Twitter cenderung memberikan respon Negatif (91,4%) terhadap isu deforestasi. Hal ini mengindikasikan tingginya tingkat kekhawatiran dan kritik publik terhadap kerusakan hutan.
- c. Efektivitas Penanganan Imbalance: Penggunaan teknik Weighted Trainer terbukti krusial. Tanpa pembobotan kelas, model akan cenderung mengabaikan sentimen positif (kelas minoritas). Hasil pengujian menunjukkan model tetap mampu mengenali sentimen positif dengan akurasi per kelas di atas 90%.
- d. Kesesuaian Model: Arsitektur BERT yang telah di-pre-trained khusus untuk Bahasa Indonesia jauh lebih efektif dalam memahami slang, singkatan, dan konteks kalimat pada Twitter dibandingkan model multibahasa standar.