

**Analisis Sentiment Netizen Indonesia pada Berita Viral di Akun Instagram Folkative Menggunakan Model Indonesian *Bidirectional Representation Encoder from Transformers* (IndoBERT)**

**Valentino Bryan Fortunando<sup>1</sup>, Darius Andana Haris<sup>2</sup>**

<sup>1,2</sup>Sistem Informasi Universitas Tarumanagara

valentino.825220101@stu.untar.ac.id<sup>1</sup>, darius@fti.untar.ac.id<sup>2</sup>

**ABSTRACT**

*This study aims to analyze the sentiments of Indonesian netizens toward viral news posted on the Instagram account Folkative during the period from January to July 2025, with one viral news item selected from each month. The analysis was conducted to identify patterns of public opinion, which were classified into four categories: positive, negative, almost positive, and almost negative, using the IndoBERT model. The research data were collected through web scraping using Instans Data Scraper based on criteria requiring posts to have more than 200,000 likes and 1,000 comments. The pre-processing stage consisted of case folding, removal of non-text characters, filtering, and duplicate elimination. IndoBERT was utilized to perform sentiment classification and evaluate its performance through accuracy, precision, recall, and F1-score metrics. The use of IndoBERT for classifying the four sentiment categories revealed that the model was capable of recognizing linguistic patterns containing emotions, sarcasm, and complex opinions. The findings also provide an overview of the dynamics of Indonesian public opinion toward viral content, demonstrating IndoBERT's strong potential as an accurate sentiment analysis solution. With its bidirectional context understanding, the model is highly suitable for analyzing Instagram comments, which are generally informal. Therefore, this study adopts the IndoBERT model to analyze public sentiment toward viral news posted on the Instagram account Folkative during the January–July 2025 period.*

**Keywords:** IndoBERT, Instagram, Python, NLP, deep learning

**ABSTRAK**

Pada penelitian ini bertujuan untuk menganalisis sentiment netizen Indonesia terhadap berita viral pada akun Instagram folkative periode Januari hingga Juli 2025 yang diambil satu berita setiap bulannya, analisis ini dilakukan untuk mengidentifikasi kecenderungan opini public yang diklasifikasikan ke dalam empat kategori yaitu positif, negative, hampir positif, dan hampir negatif dengan menggunakan model IndoBERT. Data penelitian melalui hasil scraping menggunakan instans data scraper dengan kriteria berita yang memiliki lebih dari 200.000 likes dan 1.000 komentar, proses pre-processing dilakukan melalui tahap case folding, penghapusan karakter non-teks, filtering serta penghapusan duplikasi. Model IndoBERT ini digunakan untuk klasifikasi sentiment serta mengukur performa menggunakan nilai accuracy, precision, recall, dan F1-score. Model IndoBERT digunakan untuk mengklasifikasikan ke empat kategori, hasil penelitian ini menunjukkan bahwa model IndoBERT mampu mengenali pola Bahasa yang mengandung emosi, sarkasme maupun opini yang kompleks, dan juga memberikan Gambaran mengenai dinamika opini public Indonesia terhadap isi viral serta menunjukkan porenasi pada model IndoBERT sebagai Solusi analisis sentiment yang akurat. Model ini memiliki kemampuan memahami konteks secara dua arah, sehingga cocok untuk menganalisis komentar Instagram yang umumnya

bersifat informal, oleh karena itu penelitian ini memanfaatkan model IndoBERT untuk menganalisis sentiment komentar public pada berita viral di akun Instagram periode Januari hingga Juli 2025

**Kata kunci:** *IndoBERT, Instagram, Python, NLP, deep learning*

## **PENDAHULUAN**

Perkembangan teknologi digital mengubah cara masyarakat dalam memperoleh dan menyebarkan informasi, terutama melalui media sosial, Instagram menjadi salah satu platform paling populer di kalangan Masyarakat. Akun Instagram Folkative menempati posisi penting sebagai sumber berita viral yang kerap memicu diskusi hangat dikalangan warganet, kolom komentar pada setiap unggahan mencerminkan beragam reaksi masyarakat dalam berita viral mulai dari dukungan, kritik, hingga ujaran kebencian yang mencerminkan sentiment public terhadap isi yang sedang berkembang, dengan besarnya volume data komentar yang dihasilkan dari unggahan berita viral membuka peluang untuk dianalisis sentiment netizen Indonesia guna memahami persepsi *public* secara mandala, dengan pemanfaatan model Indonesian *Bidirectional Encoder Representations From Transformers* (IndoBERT) menjadi relevan karena mampu memahami konteks Bahasa Indonesia secara dua arah (*bidirectional*) dan menghasilkan teks yang sangat akurat dibandingkan metode tradisional, model IndoBERT ini merupakan pengembangan dari model BERT yang telah dilatih secara khusus menggunakan korpus besar Bahasa Indonesia, sehingga unggul dalam memahami struktur kalimat, kosakata, serta konteks linguistic local. Penelitian ini menganalisis sentiment komentar netizen Indonesia terhadap berita viral pada akun Instagram Folkative selama periode Januari-Juli 2025, penelitian ini juga menyandingkan *Social Network Analysis* (SNA) untuk mengidentifikasi pola hubungan dan pengaruh akun terverifikasi terhadap pembentukan opini public. Model IndoBERT ini merupakan salah satu model NLP yang berbasis arsitektur Transformers yang telah diadaptasi secara khusus untuk Bahasa Indonesia

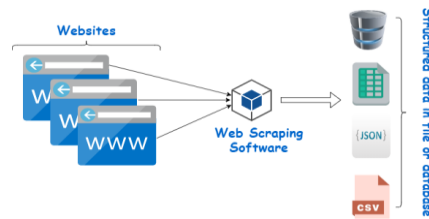
## **METODE PENELITIAN**

Metode yang digunakan dalam analisis sentiment dengan model IndoBERT ini adalah Metodologi yang dipakai dalam penelitian ini adalah *Mixed Methods* melalui pendekatan secara kualitatif dan kuantitatif, pendekatan kualitatif digunakan untuk menganalisis data komentar (sentimen) positif, negative, hampir positif, dan hampir negatif serta melakukan analisis dengan melihat komentar yang paling banyak disukai sehingga berpengaruh pada berita tersebut menggunakan algoritma berbasis model Indonesian *Bidirectional Encoder Representation from Transformers* (IndoBERT), sementara kuantitatif digunakan menghitung metrik centrality (*Degree Centrality, Betweenness Centrality, Closeness Centrality, dan Eigenvector Centrality*) yang mengukur pengaruh dan peran akun besar yang sudah verified menggunakan *Social Network Analysis* (SNA) untuk melihat opini publik serta menghitung evaluasi performa (*accuracy, F1-score, precision, dan recall*) untuk melihat nilai evaluasi

performa pada model *IndoBERT*. Metode pengumpulan data dilakukan dengan metode web scraping yaitu dengan pengambilan data secara otomatis di internet, data komentar diperoleh dari berbagai berita viral pada akun instagram folkative dalam rentang waktu Januari 2025-Juli 2025 yang bersifat publik dan dapat diakses yang menggunakan instans data scraper.

### 1. Scraping Data

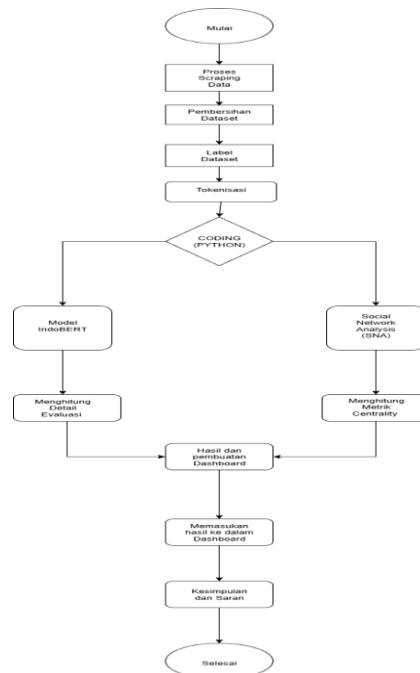
Tahapan Scraping Data untuk proses pengambilan data dalam jumlah yang besar dari berbagai komentar di akun folkative yang disimpan dalam format CSV



**Gambar 1. Proses Scraping Data**

### 2. Alur Penelitian

Pada penelitian ini dilakukan proses pengambilan data komentar dari akun Instagram folkative dari Januari sampai Juli 2025 yang diambil satu berita setiap bulannya, yang setelah itu dilakukan pembersihan data, pebelan dataset dan tokenisasi pada dataset, setelah itu dilakukan 2 analisis yaitu menggunakan model *IndoBERT* yang di sandingkan dengan Social Network Analysis.



**Gambar 2. Alur Penelitian**

### 3. Penghapusan Case Folding

Tahapan penghapusan tujuannya supaya semua huruf dalam teks komentar di ubah menjadi huruf kecil (*lowercase*) proses ini menghilangkan perbedaan leksikal yang disebabkan oleh kapitalisasi, selain itu tujuan ini mengurangi kompleksitas kosakata dan mencegah model yang tidak relevan, dengan menerapkan lowercase seluruh bentuk penulisan diseragamkan sehingga memiliki makna yang identic sebagai proses representasi Tunggal, dengan Langkah ini mengurangi redundansi kosakata yang sering muncul pada teks informasi media sosial, serta membantu menghindari ambiguitas leksikal yang tidak relevan dan mengurangi resiko model yang salah, dengan model IndoBERT kerja lebih optimal Ketika kosakata yang diproses dalam bentuk yang seragam. Tahap lowercase adalah Langkah yang fundamental untuk mempersiapkan data yang bersih dan siap untuk diproses, sehingga data dalam analisis sentiment ini menjadi lebih akurat dan relevan.

```
[3]: import pandas as pd
import os

# Nama file hasil penggabungan dari langkah sebelumnya
input_file = 'df_step1.pkl'
output_file = 'df_step2_casefolded.pkl'

print("--- 2. Memuat Data Gabungan dan Melakukan Case Folding ---")

try:
    # Memuat DataFrame yang telah digabungkan dari file pickle
    df_step1 = pd.read_pickle(input_file)
    print(f"Berhasil memuat DataFrame dari '{input_file}' dengan {len(df_step1)} baris.")

    # Pastikan kolom 'text' ada
    if 'text' in df_step1.columns:
        # Melakukan Case Folding pada kolom 'text'
        # Mengubah semua karakter dalam kolom 'text' menjadi huruf kecil (Lowercase)
        df_step1['text'] = df_step1['text'].astype(str).str.lower()

        print("\nCase folding (mengubah ke huruf kecil) 'text'.")

        # Tampilkan beberapa baris pertama setelah case folding untuk verifikasi
        print("\n--- Hasil Case Folding (5 baris pertama) ---")
        print(df_step1.head())

        # Simpan DataFrame
        df_step1.to_pickle(output_file)

        print(f"DataFrame hasil case folding telah disimpan ke '{output_file}'")
    else:
        print("Kolom 'text' tidak ditemukan dalam DataFrame. Case folding dibatalkan.")

except FileNotFoundError:
    print(f"ERROR: File '{input_file}' tidak ditemukan.")
except Exception as e:
    print(f"Terjadi kesalahan saat memproses data: {e}")
```

**Gambar 3. Penghapusan Case Folding**

#### 4. Penghapusan Karakter Non-Teks

Tujuan penghapusan karakter-non teks yaitu menghapus angka, tanda baca, URL, dan Emoji supaya data semakin relevan dan berfungsi mempercepat waktu pelatihan dan meningkatkan efisiensi model. Penghapusan karakter non-teks ini bertujuan untuk memastikan bahwa data yang akan diolah dengan model IndoBERT benar-benar informasi yang actual dan relevan dan tidak mengurangi noise pada dataset. Bagian penghapusan karakter Non-Teks merupakan Langkah yang sangat krusial untuk meningkatkan kualitas pada data sehingga mempercepat dalam proses pelatihan dan membantu model dalam melakukan analisis.

```
[7]: import pandas as pd
import re
import os

# Nama file hasil case folding
input_file = 'df_step1_casefolded.pkl'
output_file = 'df_step1_filtered.pkl'

print("... 3. Memuat Data Case Folding dan Melakukan Pembersihan/Filtering ...")

try:
    # Memuat DataFrame yang sudah di-case folding
    df_step1 = pd.read_pickle(input_file)
    print(f"Berhasil memuat DataFrame dari '{input_file}' dengan {len(df_step1)} baris.")

    # Pastikan kolom 'text' ada
    if 'text' in df_step1.columns:

        # --- Fungsi Pembersihan teks ---
        def clean_text(text):
            # 1. Menghapus URL (http/https)
            text = re.sub(r'(http|https)', '', text, flags=re.MULTILINE)

            # 2. Menghapus Mention (@user)
            text = re.sub(r'@[a-zA-Z0-9_]+', '', text)

            # 3. Menghapus karakter khusus, angka, dan emoji
            text = re.sub(r'[\d\W\p{Emoji}]', '', text)

            # 4. Menghapus spasi ganda
            text = re.sub(r'\s+', ' ', text).strip()

            return text

        # Menerapkan fungsi pembersihan ke kolom 'text'
        df_step1['text_clean'] = df_step1['text'].apply(clean_text)

        # Hapus kolom 'text' yang lama dan ganti namanya
        df_step1.drop('text', axis=1, inplace=True)
        df_step1.rename(columns={'text_clean': 'text'}, inplace=True)

        print("\nPembersihan (menghapus URL, mention, emoji, dan karakter khusus).")

        df_step1.dropna(subset=['text'], inplace=True)
        df_step1 = df_step1[df_step1['text'].str.strip().astype(bool)]

        print(f"Jumlah baris setelah menghapus baris kosong: {len(df_step1)}")

        # Tampilkan beberapa baris pertama setelah pembersihan untuk verifikasi
        print("\n... Hasil Pembersihan teks (5 baris pertama) ...")
        print(df_step1.head())

        # Simpan DataFrame yang sudah bersih ke file pickle baru
        df_step1.to_pickle(output_file)

        print(f"DataFrame hasil filtering telah disimpan ke '{output_file}'")
    else:
        print("Kolom 'text' tidak ditemukan dalam DataFrame. Pembersihan dibatalkan.")

except FileNotFoundError:
    print(f"ERROR: File '{input_file}' tidak ditemukan.")
except Exception as e:
    print(f"Terjadi kesalahan saat memproses data: {e}")
```

**Gambar 4. Penghapusan Karakter Non-Teks**

#### 5. Filtering dan Duplikasi

Tujuan ini yaitu menghapus komentar yang tidak relevan seperti komentar spam atau komentar duplikat sehingga data yang dihasilkan bersih dan siap untuk di analisis. Proses filtering ini bertujuan untuk menghapus komentar yang tidak relevan yang tidak memiliki konten berarti atau baris kosong yang secara tidak sengaja kosong akibat proses pembersihan data. Dan duplikasi bertujuan untuk memastikan



```
[6]: import pandas as pd
import os

# Nama file hasil deduplikasi dari Langkah sebelumnya
input_file = 'df_step4_deduplicated.pkl'
output_dir = 'DATA_BERSIH/' # Direktori untuk menyimpan file hasil

print("... 5. Memuat Data Bersih dan Memisahkan per Bulan ...")

try:
    # Memuat DataFrame yang sudah bersih
    df_step4 = pd.read_pickle(input_file)
    print(f"Berhasil memuat DataFrame dari '{input_file}' dengan {len(df_step4)} baris.")

    if not os.path.exists(output_dir):
        os.makedirs(output_dir)
        print(f"Direktori '{output_dir}' dibuat.")

    if 'month' in df_step4.columns:

        # Mendapatkan daftar bulan
        unique_months = df_step4['month'].unique()
        print(f"\nBulan unik yang ditemukan: {list(unique_months)}")

        # 2. Loop melalui setiap bulan dan simpan
        for month in unique_months:
            # Filter data untuk bulan tertentu
            df_month = df_step4[df_step4['month'] == month]

            file_name = f'DATA_BERSIH_{month.upper()}.csv'
            output_path = os.path.join(output_dir, file_name)

            df_month[['text']].to_csv(output_path, index=False, encoding='utf-8')

            print(f"Berhasil menyimpan {len(df_month)} baris ke '{output_path}'")

        print("\nSemua data bersih telah berhasil dipisahkan dan disimpan ke dalam file CSV masing-masing.")
        print(f"File-file disimpan di direktori: {output_dir}")

    else:
        print("ERROR: kolom 'month' tidak ditemukan. Penisahan per bulan dibatalkan.")
        print("Pastikan langkah penggabungan awal (Langkah 1) menyertakan kolom 'month'.")

except FileNotFoundError:
    print("ERROR: file '{input_file}' tidak ditemukan.")
except Exception as e:
    print(f"Terjadi kesalahan saat memproses data: {e}")
```

**Gambar 6. Final Dataset**

## 7. Proses Label

Sesudah dilakukan pre-processing pada dataset langkah selanjutnya yaitu labeling pada model *IndoBERT* supaya teks dapat dipahami oleh model *IndoBERT* dengan melakukan inferensi teks dari ke empat klasifikasi tersebut, dan supaya dapat dimengerti secara luas dari berbagai komentar yang ada dan dapat dibaca dari empat klasifikasi tersebut

```
[4]: sentiment_keywords = {
  "Positif": [
    "bagus", "keren", "indah", "nyaman", "bersih", "puas", "mantap",
    "recommended", "ramah", "murah", "cepat", "senang", "suka", "menarik",
    "luar biasa", "amazing", "worth it", "bikin betah", "memuaskan",
    "top", "kualitas oke", "rapi", "bagus banget", "perfect", "sesuai ekspektasi",
    "harga bersahabat", "tidak mengecewakan", "fungsi baik", "layak dicoba",
    "wow", "juara", "friendly", "praktis", "efisien", "bagus parah"
  ],
  "Negatif": [
    "jelek", "kotor", "mahal", "parah", "mengecewakan", "buruk", "lama",
    "pelan", "padat", "penuh", "tidak rekomendasi", "kapok", "kacau",
    "bau", "rusak", "berbahaya", "penipuan", "tidak nyaman", "payah",
    "aneh", "zank", "sampah", "gak jelas", "merugikan", "tragis",
    "ngecewain", "tidak sesuai", "overprice", "bikin emosi", "ribet",
    "kurang ajar", "pelayanan buruk", "menakutkan"
  ],
  "Hampir Positif": [
    "lumayan bagus", "cukup nyaman", "boleh lah", "tidak terlalu buruk",
    "bisa dibilang bagus", "hampir sempurna", "masih oke lah",
    "cukup memuaskan", "overall baik", "standar tapi oke", "lumayan lah",
    "not bad", "better than expected", "cukup layak", "bisa diterima",
    "expected lah", "ya bolehlah", "masih enak dipakai"
  ],
  "Hampir Negatif": [
    "biasa saja", "tidak terlalu bagus", "standar", "kurang memuaskan",
    "agak mahal", "agak kotor", "kurang ramah", "hampir mengecewakan",
    "tidak seburuk itu", "boleh lah tapi kurang", "so-so", "b aja",
    "lumayan mengecewakan", "tidak sesuai harapan", "perlu diperbaiki",
    "agak ribet", "kurang worth it", "kalau bisa dihindari"
  ]
}

# daftar semua aspek yang ada
all_aspects = list(sentiment_keywords.keys())
counter = 0 # untuk round robin

def aspect_extraction(text):
  global counter
  for aspect, keywords in sentiment_keywords.items():
    if any(keyword in text for keyword in keywords):
      return aspect

  # jika tidak ketemu keyword -> dibagi rata
  aspect = all_aspects[counter % len(all_aspects)]
  counter += 1
  return aspect

# terapkan ke dataframe
data['label'] = data['text'].astype(str).apply(aspect_extraction)
data.head(20)
```

**Gambar 7. Proses Labeling**

## 8. Proses Tokenisasi

Setelah proses label pada model IndoBERT selesai, tahap selanjutnya yaitu tokenisasi pada model IndoBERT fungsi dari tokenisasi yaitu mengubah teks menjadi urutan token yang bisa dipahami oleh model IndoBERT. Proses tokenisasi ini merupakan fundamental dalam pemrosesan Bahasa alami yang sangat berfungsi untuk mengubah teks mentah menjadi bagian bagian kecil (token) yang dapat dipahami oleh model IndoBERT , secara keseluruhan tokenisasi berperan penting dalam memastikan model IndoBERT dapat memahami konteks secara optimal, dengan proses tokenisasi yang tepat seperti **Gambar 8**, representasi teks menjadi lebih terstruktur dan konsisten

```
[35]: # ===== Import libraries =====
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
import torch

from datasets import Dataset
from transformers import (
    AutoTokenizer,
    AutoModelForSequenceClassification,
    DataCollatorWithPadding,
    TrainingArguments,
    Trainer,
    TrainerCallback,
    set_seed
)

[37]: # ===== Settings (sesuaikan jika perlu) =====
CSV_PATH = "Hasil_Pelebaran_Data_JULI.csv" # dari developer note
TEXT_COL = "text"
LABEL_COL = "label"
MODEL_NAME = "indobenchmark/indobert-base-p1" # pilihan MODEL = 1
SPLIT = 0.2
RANDOM_STATE = 42
MAX_LENGTH = 128
NUM_EPOCHS = 3
PER_DEVICE_TRAIN_BATCH_SIZE = 16
PER_DEVICE_EVAL_BATCH_SIZE = 32
LEARNING_RATE = 2e-5
WEIGHT_DECAY = 0.01
OUTPUT_DIR = "./indobert-finetuned"

# reproducibility
set_seed(RANDOM_STATE)
torch.backends.cudnn.deterministic = True
torch.backends.cudnn.benchmark = False

# ===== Load CSV and inspect =====
df = pd.read_csv(CSV_PATH)
print("Ukuran dataset (baris, kolom):", df.shape)
print("Contoh baris:")
display(df[[TEXT_COL, LABEL_COL]].head(5))

# Drop na if any in text/label
df = df.dropna(subset=[TEXT_COL, LABEL_COL]).reset_index(drop=True)

# ===== Encode Labels (LabelEncoder) =====
le = LabelEncoder()
df["label_id"] = le.fit_transform(df[LABEL_COL].astype(str))
label_names = list(le.classes_)
num_labels = len(label_names)
print("Label mapping:", dict(zip(label_names, range(num_labels))))
print("Jumlah kelas:", num_labels)
```

**Gambar 8. Proses Tokenisasi**

## 9. Rumus Perhitungan Metriks

### a) Accuracy

Accuracy dihitung dengan membagi seluruh prediksi yang benar (*True Positive + True Negative*) dengan total seluruh data. Accuracy menunjukkan seberapa akurat model secara keseluruhan

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

### b) F1-score

F1-Score dihitung sebagai harmonisasi antara *Precision* dan *Recall*. Nilai ini dihasilkan dari dua dikali hasil perkalian Precision dan Recall, lalu dibagi dengan jumlah antara *Precision* dan *Recall*

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

c) *Precision*

*Precision* dihitung dengan membagi jumlah prediksi positif yang benar (*True Positive*) dengan total semua prediksi positif (*True Positive + False Positive*). *Precision* menunjukkan tingkat ketepatan model dalam memberikan prediksi positif.

$$Precision = \frac{TP}{TP+FP}$$

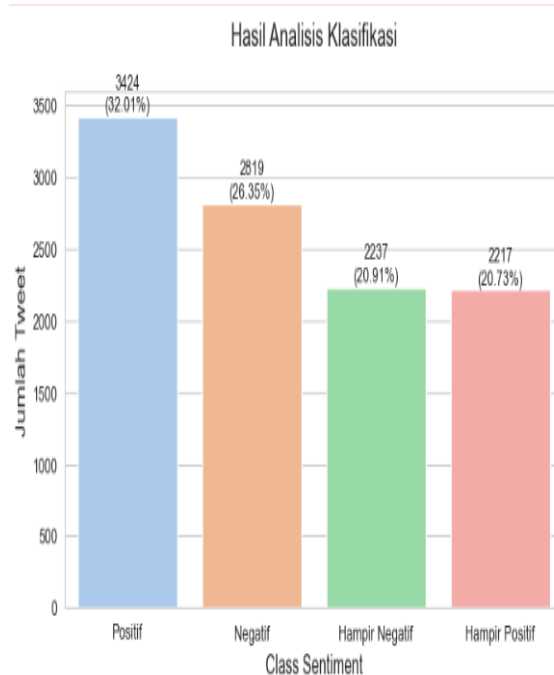
d) *Recall*

*Recall* dihitung dengan membagi jumlah kasus positif yang berhasil terdeteksi model (*True Positive*) dengan seluruh jumlah kasus positif sebenarnya (*True Positive + False Negative*)

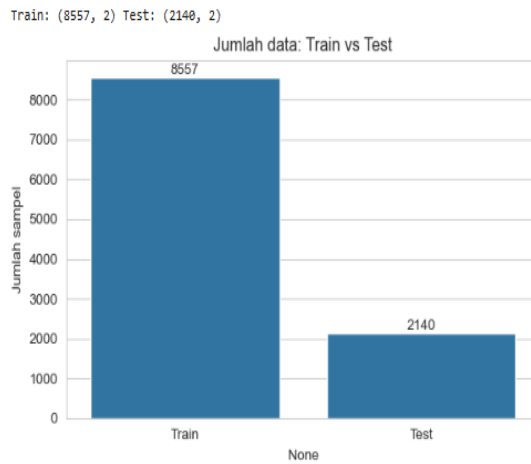
$$Recall = \frac{TP}{TP+FN}$$

## HASIL DAN PEMBAHASAN

Hasil Percobaan menggunakan data gabung pada Gambar 9 menggunakan model IndoBERT dengan melihat jumlah tweet yang dilakukan oleh netizen Indonesia pada berita viral di akun Instagram Folkative.



Gambar 9. Hasil Analisis Data Gabung



**Gambar 10. Jumlah data Train vs Test**

Pada Gambar 10 merupakan visualisasi dalam bentuk diagram batang yang membandingkan jumlah sampel (data) yang dialokasikan untuk fase pelatihan (Train) dan pengujian (Test).

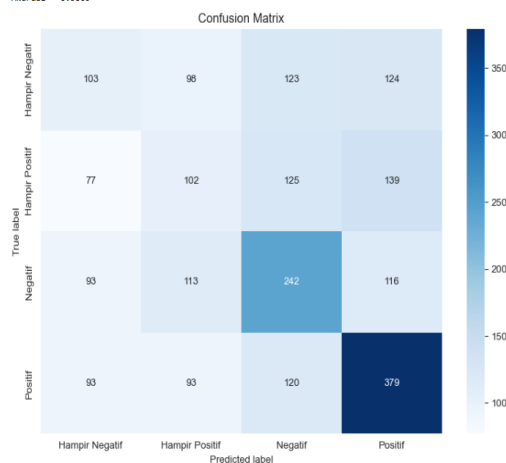
```

==== Classification Report ====
              precision    recall  f1-score   support

 Hampir Negatif    0.2814    0.2299    0.2531     448
 Hampir Positif    0.2512    0.2382    0.2403     443
 Negatif           0.3967    0.4291    0.4123     564
 Positif           0.5900    0.5533    0.5253     685

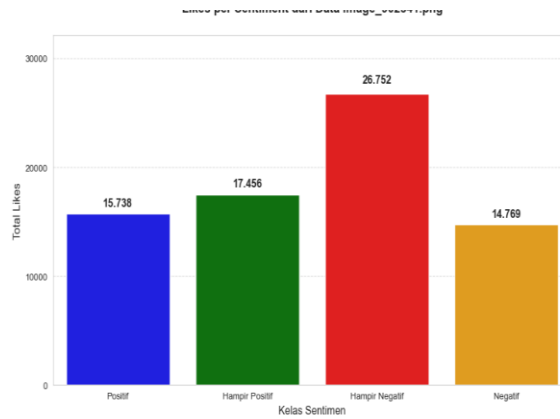
 accuracy          0.3860    0.3860    0.3860    2140
 macro avg         0.3573    0.3696    0.3577    2140
 weighted avg      0.3755    0.3860    0.3795    2140

 Akurasi = 0.3860
    
```

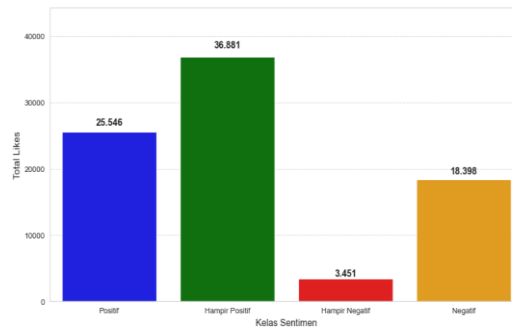


**Gambar 11. Klasifikasi Report**

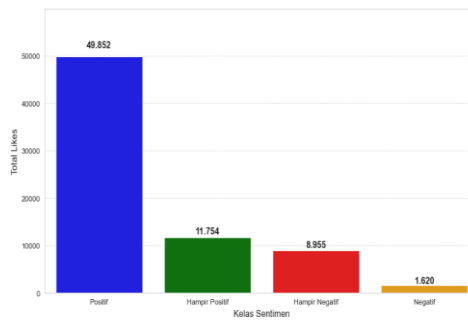
Pada Gambar 11 adalah evaluasi kinerja fundamental dari model IndoBERT yang bertugas untuk melakukan klasifikasi dengan pengelompokan data kedalam suatu kategori.



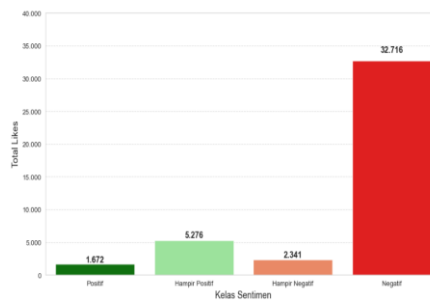
**Gambar 12 Hasil Likes per sentiment bulan Januari**



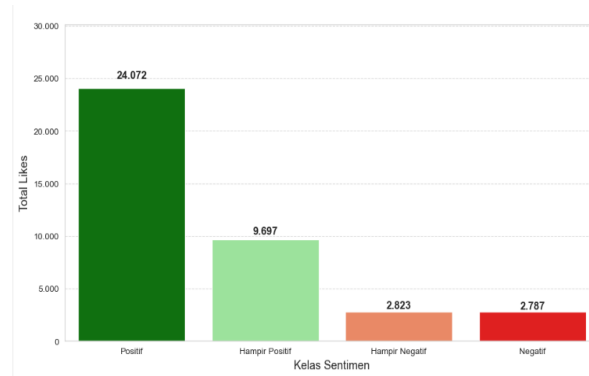
**Gambar 13. Hasil Likes per sentiment bulan Februari**



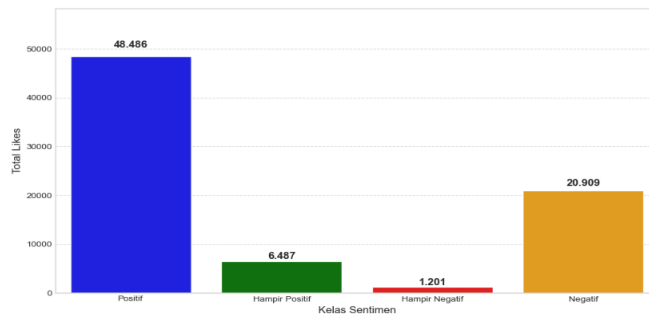
**Gambar 14. Hasil Likes per sentiment bulan Maret**



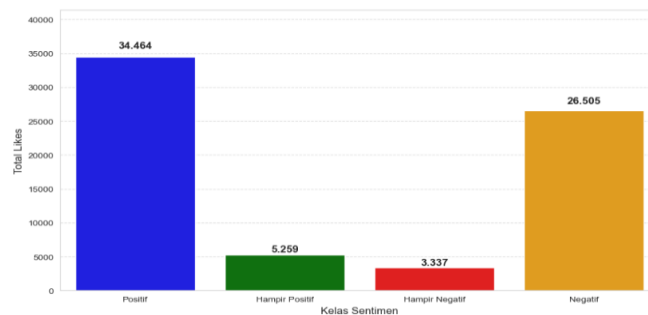
**Gambar 15. Hasil Likes per sentiment bulan April**



**Gambar 16. Hasil Likes per sentiment bulan Mei**



**Gambar 17. Hasil Likes per sentiment bulan Juni**



**Gambar 18. Hasil Likes per sentiment bulan Juli**

## **KESIMPULAN DAN SARAN**

Kesimpulan dari penelitian ini yaitu menunjukkan bahwa penggunaan model IndoBERT mampu mengklasifikasikan sentiment pada berita viral di akun Instagram folkative secara akurat dalam empat kategori yaitu positif, negative, hampir positif, dan hampir negative, dengan mengidentifikasi kecenderungan opini public yang diklasifikasikan de dalam empat kategori dengan menggunakan pendekatan mixed methods. Keunggulan penelitian ini yaitu pada model memahami konteks Bahasa Indonesia secara dua arah dan visualisasi hasil yang interaktif dengan menggunakan dashboard streamlit. Namun penelitian ini masih sangat terbatas pada periode data bulan Januari-Juli 2025 dan hanya berfokus pada satu akun Instagram yang belum sepenuhnya mewakili opini public secara luas, untuk penelitian selanjutnya

penelitian ini dapat diperluas dengan melibatkan lebih banyak akun berita dan memanfaatkan varian model transformers lain agar hasil lebih akurat.

## DAFTAR PUSTAKA

- Baharuddin, F., & Naufal, M. F. (2023). Fine-tuning IndoBERT for Indonesian exam question classification based on Bloom's Taxonomy. *Journal of Information Systems Engineering and Business Intelligence*, 9(2), 253-263.
- Jayadianti, H., Kaswidjanti, W., Utomo, A. T., Saifullah, S., Dwiyanto, F. A., & Drezewski, R. (2022). Sentiment analysis of Indonesian reviews using fine-tuning IndoBERT and R-CNN. *ILKOM Jurnal Ilmiah*, 14(3), 348-354.
- Merdiansah, R., Siska, S., & Ridha, A. A. (2024). Analisis sentimen pengguna X Indonesia terkait kendaraan listrik menggunakan IndoBERT. *Jurnal Ilmu Komputer dan Sistem Informasi (JIKOMSI)*, 7(1), 221-228.
- Prabowo, A., & Sanjaya, F. I. (2024). Penerapan Metode Transfer Learning Pada Indobert Untuk Analisis Sentimen Teks Bahasa Jawa Ngoko Lugu. *Jurnal Sistem Informasi dan Sistem Komputer*, 9(2), 205-217.
- Sayarizki, P., & Nurrahmi, H. (2024). Implementation of indobert for sentiment analysis of indonesian presidential candidates. *Indonesian Journal on Computing (Indo-JC)*, 9(2), 61-72.
- Valentino Bryan Fortunando, Mahasiswa S1 Program Studi Sistem Informasi, Fakultas Teknologi Informasi, Universitas Tarumanagara.