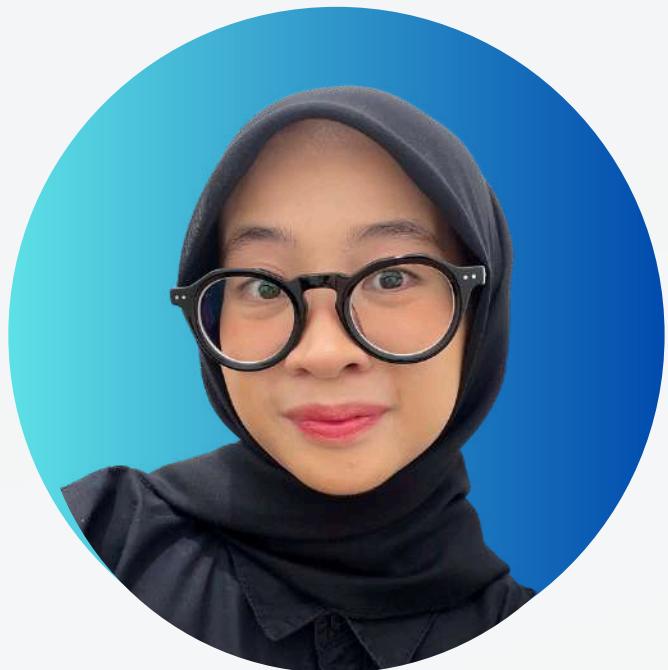


Topic Level Sentiment Analysis dengan Basis Transformer terhadap Data Ibu Kota Nusantara (IKN) dalam cuitan X

KELOMPOK 6

DRA. NORA HARIADI, M.SI.
PROF. DR.RER.NAT. HENDRI MURFI, S.SI., M.KOM
From Sawyer Merritt

OUR MEMBERS



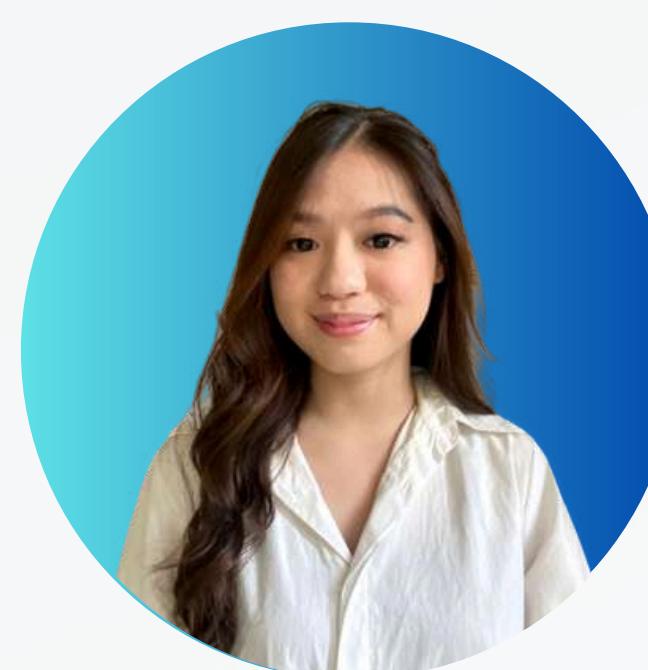
Reizka Fathia

2206052755



Fakhri Rayhan A

2206048814



Cecilia Susanto

2206052881



Ria Mulyadi

2206048556

TABLE OF CONTENTS

01

Pendahuluan

02

Data

03

Metode

04

Simulasi
Analisis

05

Kesimpulan

06

Daftar
Pustaka

See what's
happening in the
world right now.

01

Pendahuluan



PENDAHULUAN

Latar Belakang

Pemindahan Ibu Kota Negara (IKN) dari Jakarta ke Kalimantan Timur merupakan langkah strategis untuk mengatasi permasalahan Jakarta seperti kepadatan, kemacetan, dan banjir. IKN dirancang sebagai kota masa depan yang modern, berkelanjutan, dan berlandaskan nilai Pancasila (Kurniawan et al., 2024).

Kebijakan ini **menimbulkan beragam respon** masyarakat. Sebagian mendukung visi pembangunan IKN, tetapi tidak sedikit pula yang mengkhawatirkan dampak lingkungan, sosial, dan anggaran (Yusuf et al., 2024).

Media sosial X menjadi ruang aktif masyarakat menyampaikan opini, baik berupa sentimen positif maupun negatif terkait IKN (Yusuf et al., 2024).

PENDAHULUAN

Rumusan Masalah, Tujuan

Rumusan Masalah

Bagaimana **persebaran sentimen** (positif dan negatif) masyarakat terhadap topik Ibu Kota Negara (IKN) di platform Twitter?

Topik-topik apa saja yang paling sering dibahas masyarakat terkait IKN dalam cuitan di Twitter?

Bagaimana keterkaitan **antara sentimen dan topik-topik** yang muncul dalam percakapan mengenai IKN?

TUJUAN

Mengevaluasi **IndoBERTweet** kinerja model dalam publik mengklasifikasikan sentimen terhadap IKN.

Mengidentifikasi dan mengelompokkan topik-topik utama terkait IKN dari data cuitan menggunakan metode **BERTopic**.

Menyajikan **distribusi dan hubungan** antara sentimen dan topik yang ditemukan untuk memahami opini publik secara lebih mendalam.

PENDAHULUAN

Batasan Masalah

Batasan Masalah

1. Sentimen yang diklasifikasikan dibatasi pada dua kelas: **positif** dan **negatif**, sesuai label yang tersedia dalam dataset.
2. Data yang digunakan terbatas pada **ulasan teks dalam Bahasa Indonesia**, tanpa mempertimbangkan ulasan berbahasa asing lainnya.
3. Pengelompokan topik dilakukan menggunakan **BERTopic** dengan hasil yang bergantung pada kualitas embedding dan parameter clustering, sehingga interpretasi topik bersifat subjektif.

02

Metode



METODE

Analisis Sentimen

Analisis sentimen merupakan bagian dari Natural Language Processing (NLP) yang bertujuan mengidentifikasi dan mengekstraksi opini atau emosi dalam teks. Teknik ini mengklasifikasikan sentimen menjadi tiga kategori utama: **positif, negatif, dan netral.**

Dalam praktiknya, analisis sentimen digunakan untuk memahami opini publik dan preferensi konsumen, terutama dari data berukuran besar seperti media sosial dan ulasan online.

Menurut Medhat et al. (2014), perusahaan memanfaatkan teknik ini untuk memantau reputasi merek, menilai efektivitas kampanye pemasaran, dan mengolah umpan balik pelanggan secara efisien tanpa harus membaca ribuan teks secara manual.

Seiring berkembangnya era big data, peran analisis sentimen menjadi semakin penting dalam pengambilan keputusan berbasis data.



METODE

Pemodelan Topik

Pemodelan topik adalah metode unsupervised untuk **mengidentifikasi topik-topik utama** dari sekumpulan dokumen teks tanpa memerlukan label sebelumnya. Algoritma populer seperti LDA (Latent Dirichlet Allocation) bekerja dengan membagi dokumen menjadi distribusi kata yang mencerminkan topik. Namun, LDA memiliki keterbatasan, terutama dalam menangani data berukuran kecil atau teks pendek seperti tweet, serta cenderung menghasilkan banyak topik outlier dan kesulitan dalam interpretasi hasil.

Untuk mengatasi hal ini, metode modern seperti BERTopic menggunakan representasi teks berbasis embedding dari model bahasa pretrained (misal BERT) dan mengelompokkan dokumen berdasarkan kemiripan vektornya. Pendekatan ini memberikan hasil topik yang lebih relevan dan mudah dipahami, dengan sedikit kebutuhan tuning parameter.



METODE

Pra-Pemrosesan Data

Data Cleaning

Menghapus elemen tidak penting seperti angka, simbol, URL, tag HTML, emoji, dan lambang hashtag. **Membersihkan noise** dalam teks agar model fokus pada informasi yang bermakna.

Text Cleaning

Text Cleaning adalah proses **mengurangi "noise"** dan **meningkatkan kualitas data** sehingga model dapat belajar dari informasi yang relevan saja

Stripping

Menghapus spasi berlebih, tab, dan baris baru agar format teks lebih rapi. Membantu proses tokenisasi menjadi lebih efisien dan konsisten.

Tokenizing (BERT Tokenizer)

BERT tokenizer adalah proses **mengubah teks mentah menjadi representasi numerik** yang dapat diproses oleh model BERT. Proses ini menggunakan algoritma khusus bernama WordPiece, yang dirancang untuk menangani kata-kata umum maupun yang jarang muncul (out-of-vocabulary) dengan efisien.

METODE

Exploratory Data Analysis (EDA)

- **EDA adalah tahap awal untuk mengeksplorasi dan memahami karakteristik data sebelum membangun model.** Konsep ini diperkenalkan oleh Tukey (1977) untuk menggali pola, anomali, dan struktur data secara visual maupun statistik
- Mengapa EDA penting?
 - Memahami sebaran label sentimen (positif, negatif, netral)
 - Mengetahui panjang rata-rata teks & outlier
 - Mengungkap kata-kata atau frasa dominan (dengan frekuensi atau TF-IDF)
 - Mendeteksi bias data, misalnya ketimpangan jumlah kelas
 - Menentukan langkah preprocessing dan model yang sesuai
- Teknik yang digunakan:
 - Word Cloud → visualisasi kata paling sering muncul
 - Donut Chart → distribusi kelas sentimen
 - Histogram distribusi panjang teks → melihat persebaran jumlah kata per dokumen berdasarkan kelas sentimen

- Subbidang machine learning yang menggunakan jaringan saraf (neural networks) dalam untuk **mempelajari representasi fitur secara otomatis** dari data mentah.
- Mengandalkan lapisan transformasi non-linear; pelatihan dilakukan dengan backpropagation dan optimisasi (SGD/Adam).
- **Model Populer:** RNN, LSTM, Transformer – efektif untuk data teks berurutan dan kompleks.
- Transfer Learning: Model BERT yang dapat di fine-tuned untuk tugas spesifik (analisis sentimen).
- **Kelebihan:** Tidak memerlukan rekayasa fitur manual, mampu menangkap makna kontekstual.
- **Kekurangan:** Bersifat black box, sulit dijelaskan secara interpretatif.

- **Mengubah data kategorikal (teks) menjadi angka** agar dapat diproses oleh model machine learning. Pilihan encoding sangat mempengaruhi performa model dan harus disesuaikan dengan tipe data. Pemilihan encoding yang tidak tepat membuat model keliru menangkap pola yang tidak ada.
- **Label Encoding:** Setiap kategori diberi angka unik; cepat dan sederhana, tapi bisa menimbulkan ilusi urutan yang salah.
- **One-Hot Encoding:** Mengubah kategori menjadi vektor biner (0/1); mencegah bias urutan dan cocok untuk klasifikasi.

METODE

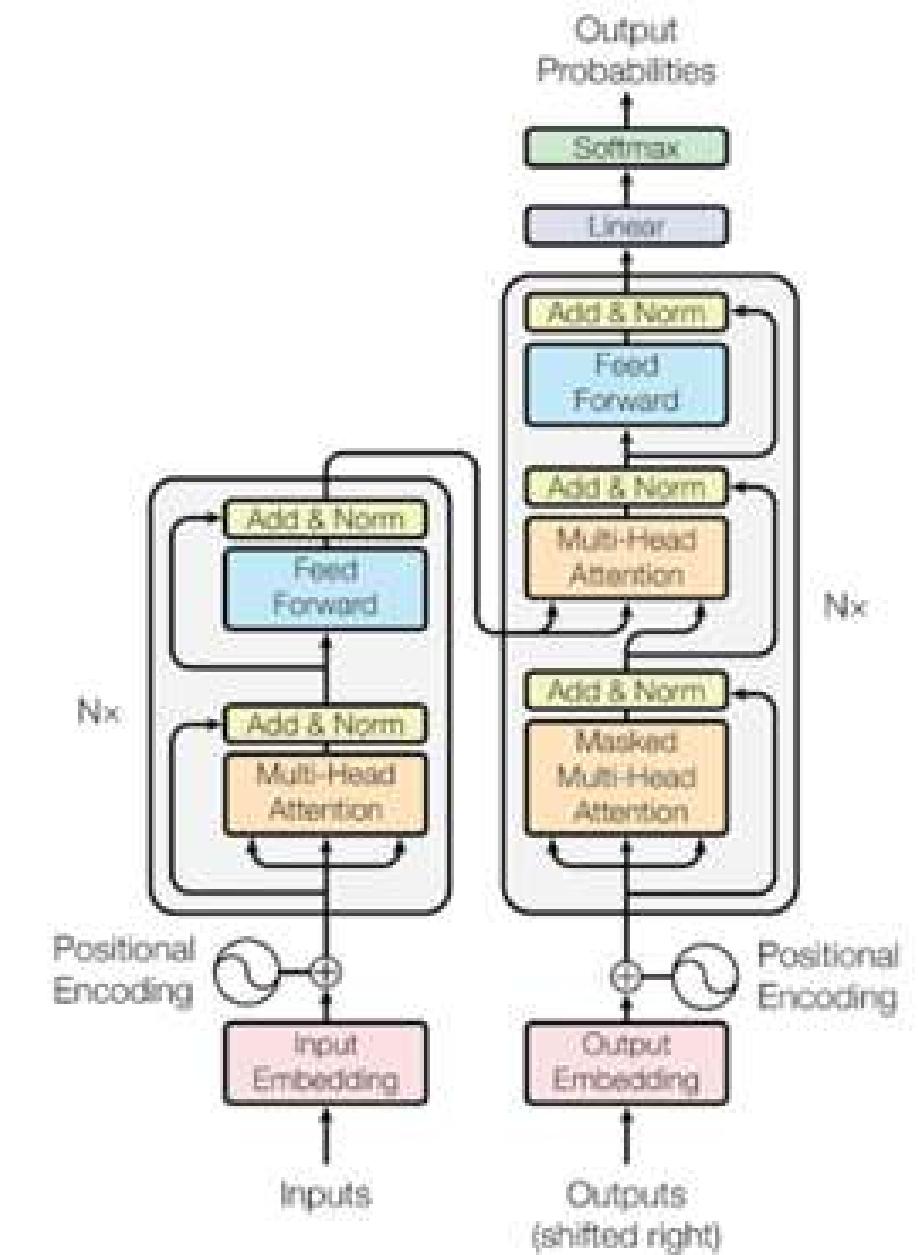
Bidirectional Encoder Representations from Transformers (BERT)

- Model NLP berbasis Transformer (Devlin et al., 2018), menggunakan representasi kata dua arah (bidirectional contextual embeddings) yang **memahami konteks dari kiri dan kanan sekaligus**. Berbeda dari Word2Vec, GloVe, dan ELMo yang statis, BERT menghasilkan representasi kata dinamis tergantung konteks.
- **Dua tahap utama BERT:**
 - a. Pretraining:
 - Masked Language Modeling (MLM): prediksi kata yang disembunyikan dari konteks.
 - Next Sentence Prediction (NSP): prediksi apakah dua kalimat saling berurutan secara logis.
 - b. Fine-tuning:
 - Penyesuaian pada tugas spesifik (mis. klasifikasi, QA) menggunakan token [CLS].
 - Semua parameter diperbarui agar model beradaptasi dengan data baru.
- Kekuatan transfer learning: Pengetahuan dari pretraining bisa digunakan ulang di banyak tugas tanpa pelatihan dari nol.
- Tantangan BERT:
 - Kurang robust terhadap serangan adversarial (mis. TextFooler).
 - Solusi: retraining dengan data hasil serangan untuk meningkatkan ketahanan (Koroteev, 2021).
- Aplikasi luas: Klasifikasi teks, named entity recognition, sentiment analysis, question answering, dll.

IndoBERT

IndoBERT: Model Bahasa Indonesia Berbasis BERT

- Arsitektur: BERT-base uncased
 - 12 hidden layers (768 dimensi)
 - 12 attention heads
 - Feed-forward 3.072 dimensi
- Keunggulan:
 - Max Multi-Head Attention: mengambil nilai maksimum dari setiap attention head untuk memperkuat representasi
 - Menangkap informasi dari berbagai subruang teks secara lebih efektif
- Performa:
 - Unggul dibanding model tradisional seperti CNN, RNN, dan LSTM
 - Efektif untuk berbagai tugas NLP berbahasa Indonesia



METODE

IndoBERT

BERTopic adalah metode topik modeling berbasis machine learning yang memanfaatkan model bahasa BERT untuk merepresentasikan teks secara kontekstual.

Cara Kerja:

1. Mengubah dokumen teks menjadi embedding (vektor) menggunakan BERT.
2. Melakukan clustering pada embedding untuk mengelompokkan dokumen ke dalam topik-topik serupa.
3. Menemukan kata-kata kunci yang paling representatif dari setiap topik.
4. Menyediakan dokumen-dokumen perwakilan untuk memperjelas isi topik.

METODE

IndoBERTweet

IndoBERTweet adalah sebuah pre-trained language model berbasis BERT yang dirancang khusus untuk **memahami bahasa Indonesia dalam konteks media sosial, khususnya Twitter (sekarang X)**. Model ini dikembangkan untuk menangani karakteristik unik dari teks media sosial yang sering kali informal, tidak baku, singkat, dan mengandung banyak simbol atau slang.

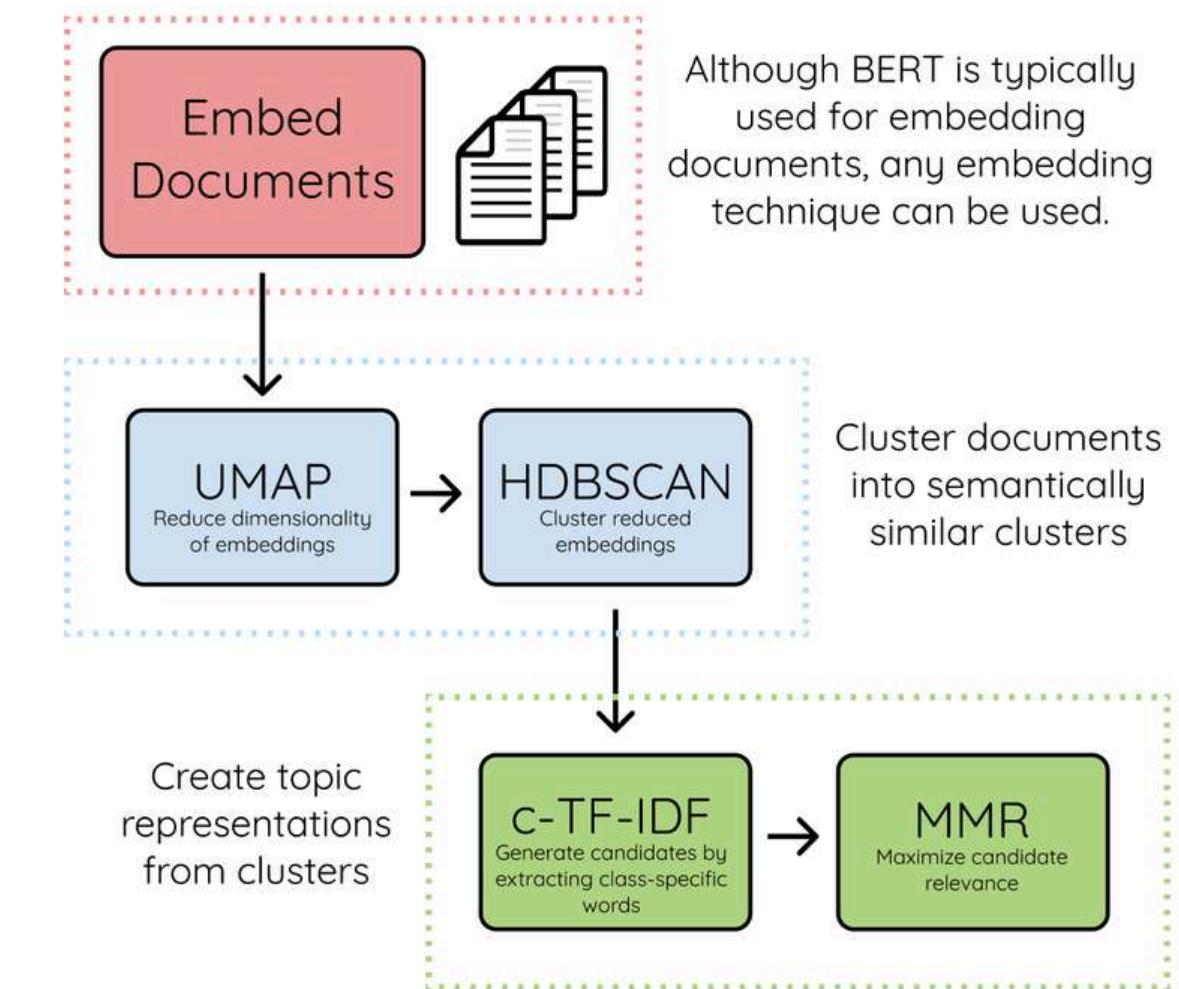
Arsitektur:

- IndoBERTweet menggunakan arsitektur Transformer, seperti BERT dan BERTweet.
- Mengadopsi RoBERTa-base sebagai kerangka awal.
- Dilatih pada data Twitter berbahasa Indonesia dalam jumlah besar agar memahami struktur dan gaya bahasa khas media sosial.

METODE

BERTopic

- Teknik topic modeling modern berbasis transformer yang mengelompokkan dokumen ke dalam topik-topik tematik secara otomatis dan interpretatif. BERTopic dapat digunakan untuk berbagai bahasa (multilingual BERT) serta cocok untuk dokumen pendek seperti tweet
- Diperkenalkan oleh Grootendorst (2022), BERTopic menggabungkan:
 - Embedding semantik (BERT/SBERT)
 - Reduksi dimensi (UMAP)
 - Klasterisasi (HDBSCAN)
 - Representasi topik (c-TF-IDF)
- Alur Kerja BERTopic
 - Document Embedding dengan BERT atau Sentence-BERT → Vektor representasi semantik.
 - Dimensionality Reduction menggunakan UMAP → Memudahkan visualisasi dan clustering.
 - Clustering dengan HDBSCAN → Menghasilkan klaster topik.
 - Topic Representation dengan c-TF-IDF → Ekstraksi kata kunci dari tiap topik.



METODE

Komponen Matematis BERTopic

1. Standard TF-IDF (untuk pemahaman awal)

Digunakan untuk merepresentasikan topik, bukan dokumen.

$$TF(t, d) = \frac{\text{jumlah } t \text{ dalam } d}{\text{jumlah total kata di } d}$$

$$IDF(t) = \log \left(\frac{N}{DF(t)} \right)$$

$$TF - IDF(t, d) = TF(t, d) \times IDF(t)$$

Contoh:

- "Earthquake" muncul 35 kali dalam 100 kata \rightarrow TF = 0.35
- Muncul di 500 dari 100.000 dokumen \rightarrow IDF \approx 5.298
- TF-IDF = $0.35 \times 5.298 = 1.854$

2. Class-based TF-IDF (c-TF-IDF)

Digunakan untuk merepresentasikan topik, bukan dokumen.

$$w_{t,c} = tf_{t,c} \cdot \log \left(1 + \frac{A}{tf_t} \right)$$

Dengan:

- $tf_{t,c}$: frekuensi kata ttt dalam topik/klaster ccc
- tf_t : total frekuensi kata ttt dalam seluruh topik
- A : rata-rata panjang dokumen dalam klaster

Tujuannya: Menonjolkan kata-kata penting yang spesifik pada topik tertentu, bukan sekadar frekuensi tinggi.

3. Embedding BERT

BERTopic menggunakan BERT atau SBERT untuk mengubah teks menjadi vektor berdimensi tinggi:

$$\mathbf{v}_d = \text{BERT}(d) \in \mathbb{R}^n$$

Biasanya $n=768$ (dimensi output BERT base), dan hasilnya adalah embedding kontekstual: kata “bank” pada konteks keuangan dan sungai akan memiliki representasi vektor berbeda.

4. Reduksi Dimensi: UMAP

UMAP (Uniform Manifold Approximation and Projection) memetakan vektor dari dimensi tinggi ke ruang berdimensi rendah:

$$f : \mathbb{R}^{768} \rightarrow \mathbb{R}^{d_{\text{low}}}$$

Tujuan UMAP:

- Mempertahankan struktur lokal (tetangga dekat tetap dekat).
- Mempermudah klasterisasi dan visualisasi.

UMAP meminimalkan fungsi loss jarak probabilistik antara high-dimensional graph dan low-dimensional embedding.

5. Clustering: HDBSCAN

HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) adalah perluasan dari DBSCAN yang:

- Tidak memerlukan jumlah klaster (k) sebagai input.
- Berdasarkan density estimation: jika titik-titik cukup rapat, mereka membentuk klaster.

Secara matematis, HDBSCAN membangun minimum spanning tree (MST) dari graf jarak antar titik, lalu memotongnya berdasarkan parameter `min_samples` dan `min_cluster_size`.

Output-nya: klaster C_1, C_2, \dots, C_k , serta kemungkinan beberapa data tidak termasuk klaster (noise).

6. Probabilitas Keanggotaan Topik

Meskipun hasil utama adalah klasifikasi hard (satu dokumen = satu topik), BERTopic juga dapat menghitung probabilitas keanggotaan:

$$P(topic_i | d) = similarity(v_d, \mu_i)$$

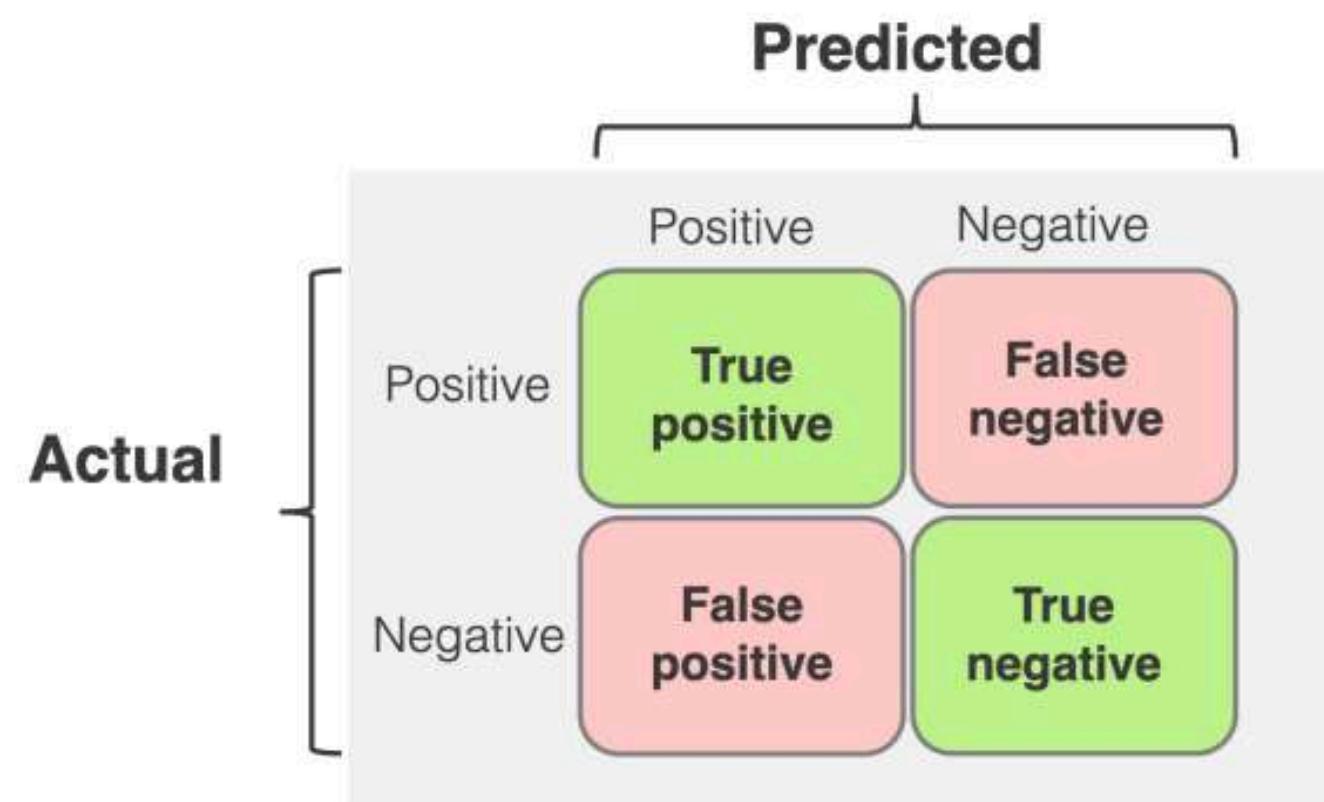
- v_d : vektor dokumen
- μ_i : centroid vektor topik i

METODE

Evaluasi Model Klasifikasi

CONFUSION MATRIX

- *Confusion matrix*: Alat evaluasi dalam tugas klasifikasi, terutama dalam machine learning dan deep learning. Matriks ini membandingkan label aktual (ground truth) dengan label yang diprediksi oleh model dalam bentuk tabel dua dimensi (TP, TN, FP, FN).
- Dengan: TP (Prediksi positif (benar)), TN (Prediksi negatif (benar)), FP (Prediksi positif (salah)), FN (Prediksi negatif (salah)). Gambar berikut menunjukkan struktur *Confusion Matrix*.



METODE

Evaluasi Model Klasifikasi

Accuracy

Proporsi prediksi benar dari seluruh prediksi. Mengukur keakuratan total model.

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

Precision

Proporsi prediksi positif yang benar-benar positif.

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

Recall

Proporsi instance positif yang berhasil ditemukan model.

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

METODE

Evaluasi Model Klasifikasi

F1-Score

Harmonik rata-rata Precision dan Recall.

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

Macro Average

Rata-rata metrik dari semua kelas, tanpa mempertimbangkan jumlah instance.

$$Precision_{macro} = \frac{1}{n} \sum_{i=1}^n Precision_i \quad Recall_{macro} = \frac{1}{n} \sum_{i=1}^n Recall_i$$

$$Macro - F1 = \frac{2 \times Precision_{macro} \times Recall_{macro}}{Precision_{macro} + Recall_{macro}}$$

METODE

Evaluasi Model Klasifikasi

Weighted Average

Rata-rata metrik yang memperhitungkan jumlah instance (support) pada setiap kelas.

Support_i = jumlah instance aktual (ground truth) dari kelas i

$$Precision_{weighted} = \frac{\sum_{i=1}^n (Precision_i \times Support_i)}{\sum_{i=1}^n Support_i}$$

$$Recall_{weighted} = \frac{\sum_{i=1}^n (Recall_i \times Support_i)}{\sum_{i=1}^n Support_i}$$

$$Weighted - F1 = \frac{\sum_{i=1}^n (F1_i \times Support_i)}{\sum_{i=1}^n Support_i}$$

03

Data



DATA

Gambar disamping menggambarkan dataset yang diambil dari platform Kaggle dan berisi **1.464 baris** dan **2 kolom** yang terdiri dari teks cuitan (**tweet**) dan label **sentimen** yang menyertainya.

Kolom tweet berisi teks dari pengguna Twitter yang membahas isu pemindahan ibu kota, sementara kolom sentiment menunjukkan klasifikasi sentimen dari masing-masing tweet, yaitu apakah bernada positive atau negative.

Dataset ini sangat relevan untuk analisis sentimen publik terhadap kebijakan IKN dan dapat digunakan lebih lanjut dalam pemodelan klasifikasi teks atau studi opini masyarakat berbasis media sosial.

	tweet	sentiment	grid icon	bar chart icon	edit icon
0	@jokowi saya sangat setuju pak bahkan lebih se...	positive			
1	@hnurwahid @FPKSDPRRI Saya setuju ibu kota pin...	positive			
2	@MardaniAliSera @FPKSDPRRI Saya dan mayoritas ...	positive			
3	cocok ibu kota pindah ke kalimantan apalagi gu...	positive			
4	@geedeulbeyou1 Jadi kepada lo yang gak setuju ...	positive			
...
1459	IKN yg ditundabukan pemilu https://t.co/GzXNzp...	negative			
1460	@Opposih6890bio @maryshelparaiso Ulama Jawa Ti...	positive			
1461	Mantap IKN Nusantara meningkatkan perekonomian...	positive			
1462	Tiga manfaat besar IKN Nusantara\n\n_Warganet ...	positive			
1463	@OICoffe @maryshelparaiso Masyarakat Jawa Timu...	positive			

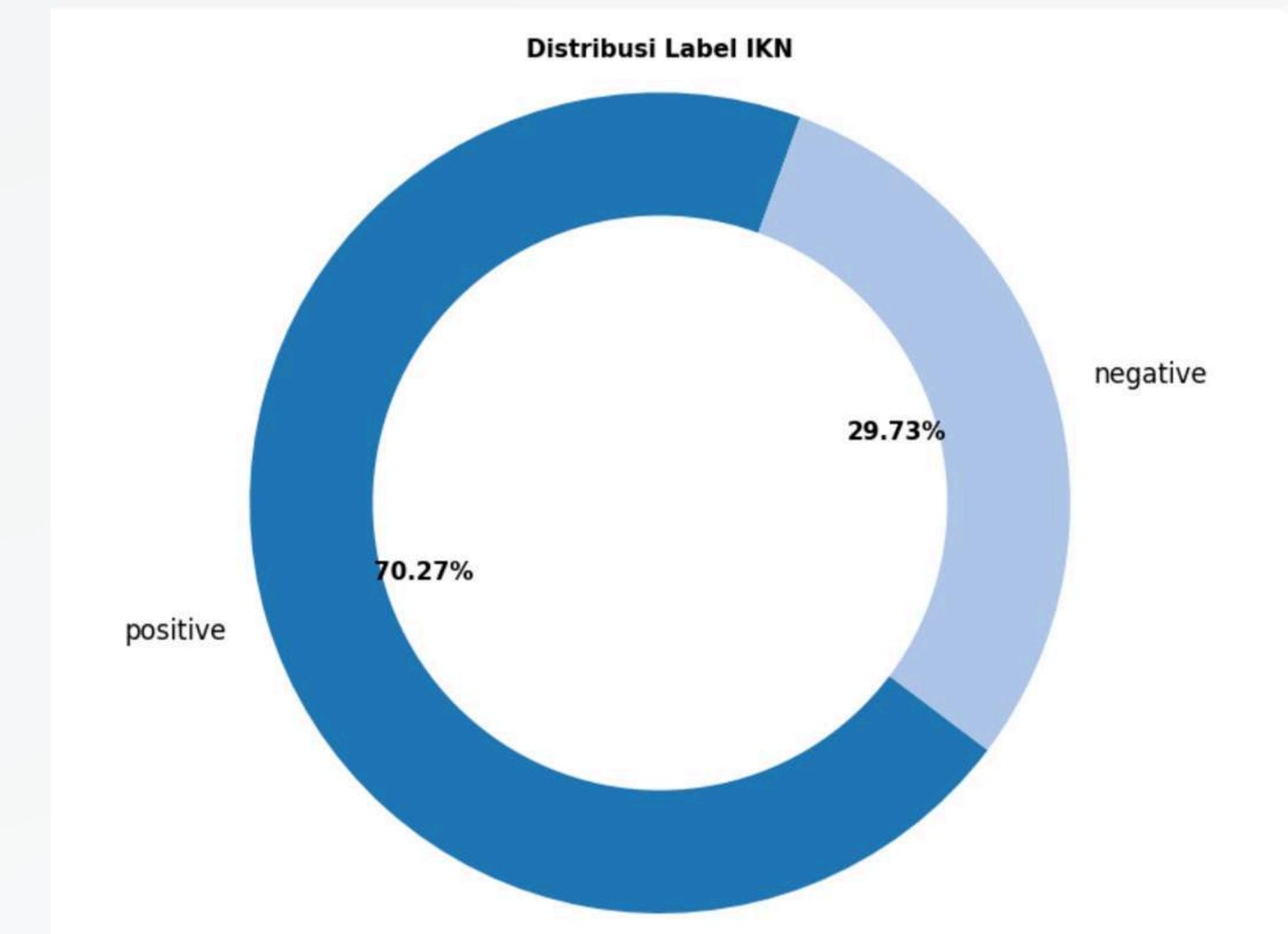
1464 rows × 2 columns

DATA

Diagram di samping menunjukkan distribusi sentimen masyarakat terhadap Ibu Kota Nusantara (IKN) berdasarkan data yang telah dianalisis. Hasilnya menunjukkan bahwa:

- **70.27%** dari total opini tergolong **positif**, mengindikasikan mayoritas masyarakat mendukung atau memiliki pandangan baik terhadap rencana pemindahan IKN.
- Sementara itu, **29.73%** opini diklasifikasikan sebagai **negatif**, menunjukkan masih adanya kelompok masyarakat yang meragukan atau menolak rencana tersebut.

Distribusi ini memberikan gambaran awal tentang persepsi publik terhadap kebijakan pemindahan IKN, yang dapat menjadi dasar bagi pengambilan keputusan lanjutan dan strategi komunikasi publik.



DATA

EDA

WordCloud

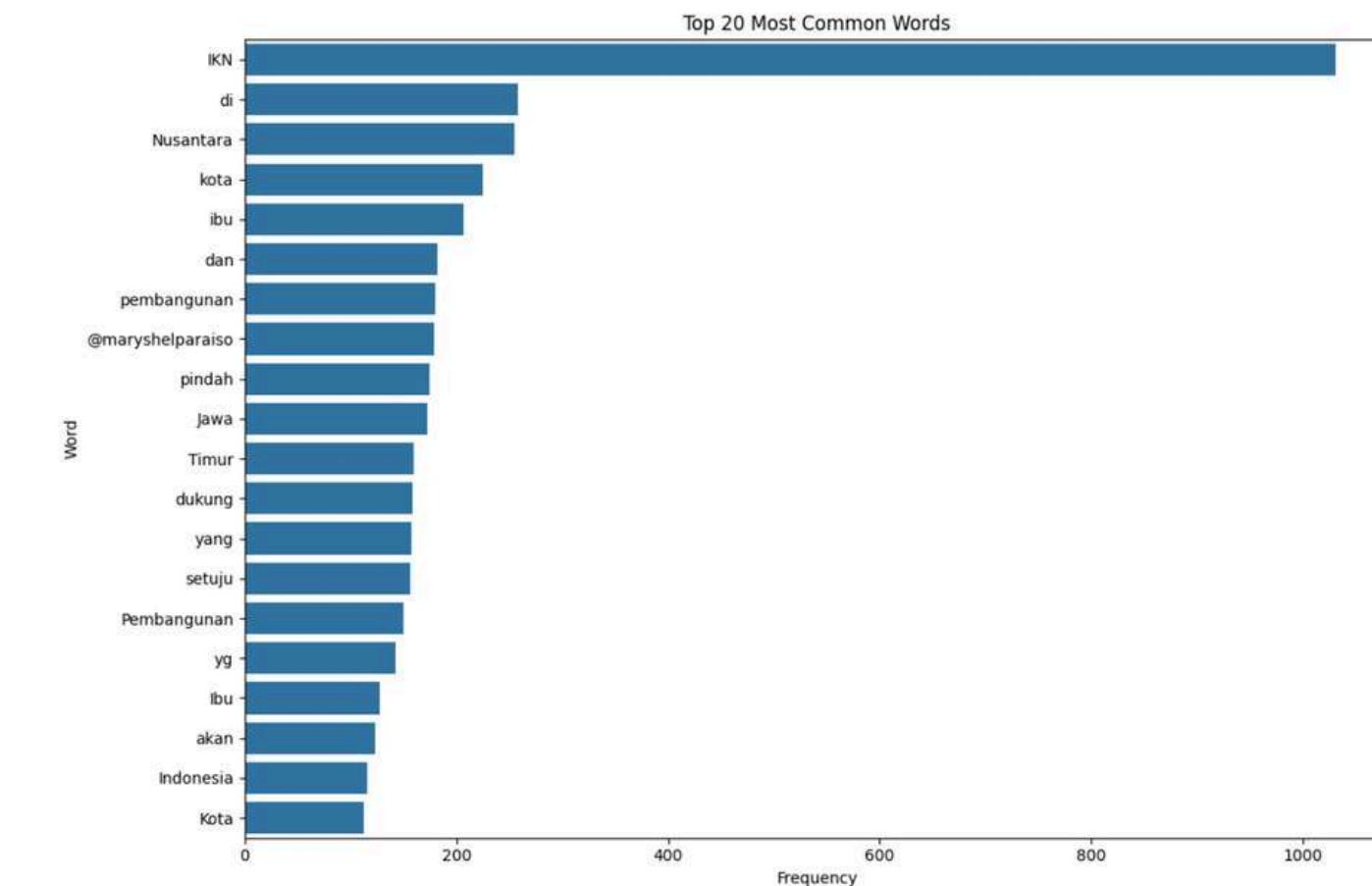
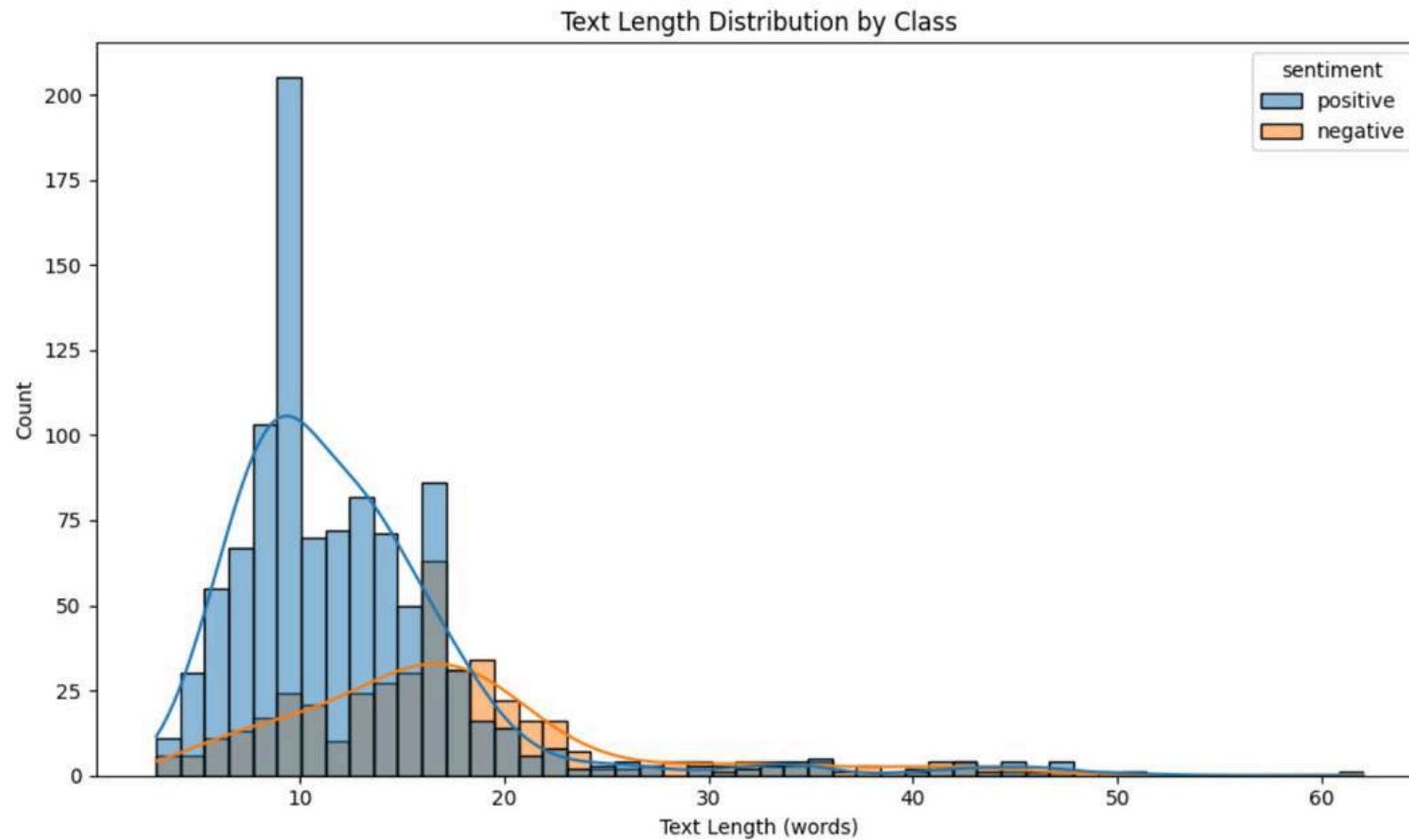


Word cloud pada gambar menggambarkan **perbedaan fokus kata** antara ulasan secara keseluruhan, ulasan positif, dan ulasan negatif terkait pemindahan Ibu Kota Negara (IKN). Secara umum, kata-kata seperti "ibu", "kota", "IKN", dan "pembangunan" mendominasi semua kelas. Pada kelas positif, muncul kata-kata yang mencerminkan dukungan seperti "dukung", "pembangunan", "nusantara", dan "Indonesia", menandakan harapan terhadap pemerataan pembangunan dan kemajuan. Sebaliknya, kelas negatif menunjukkan kata-kata seperti "tolak", "tidak", "proyek", "uang", dan "oligarki", yang merefleksikan kekhawatiran publik terhadap proyek IKN.

DATA

EDA

Hasil Visualisasi



Majoritas cuitan berada di kisaran 5–20 kata. Baik sentimen positif maupun negatif memiliki pola distribusi yang mirip, namun sentimen positif cenderung memiliki jumlah kata sedikit lebih banyak dibanding negatif. Lalu Kata "IKN", "Nusantara", dan "kota" merupakan kata yang paling dominan, yang menunjukkan topik utama. Kata-kata lain seperti "pembangunan", "pindah", dan "dukung" mengindikasikan bahwa sebagian besar percakapan terkait dengan aspek pembangunan dan sikap terhadap pemindahan ibu kota.

04

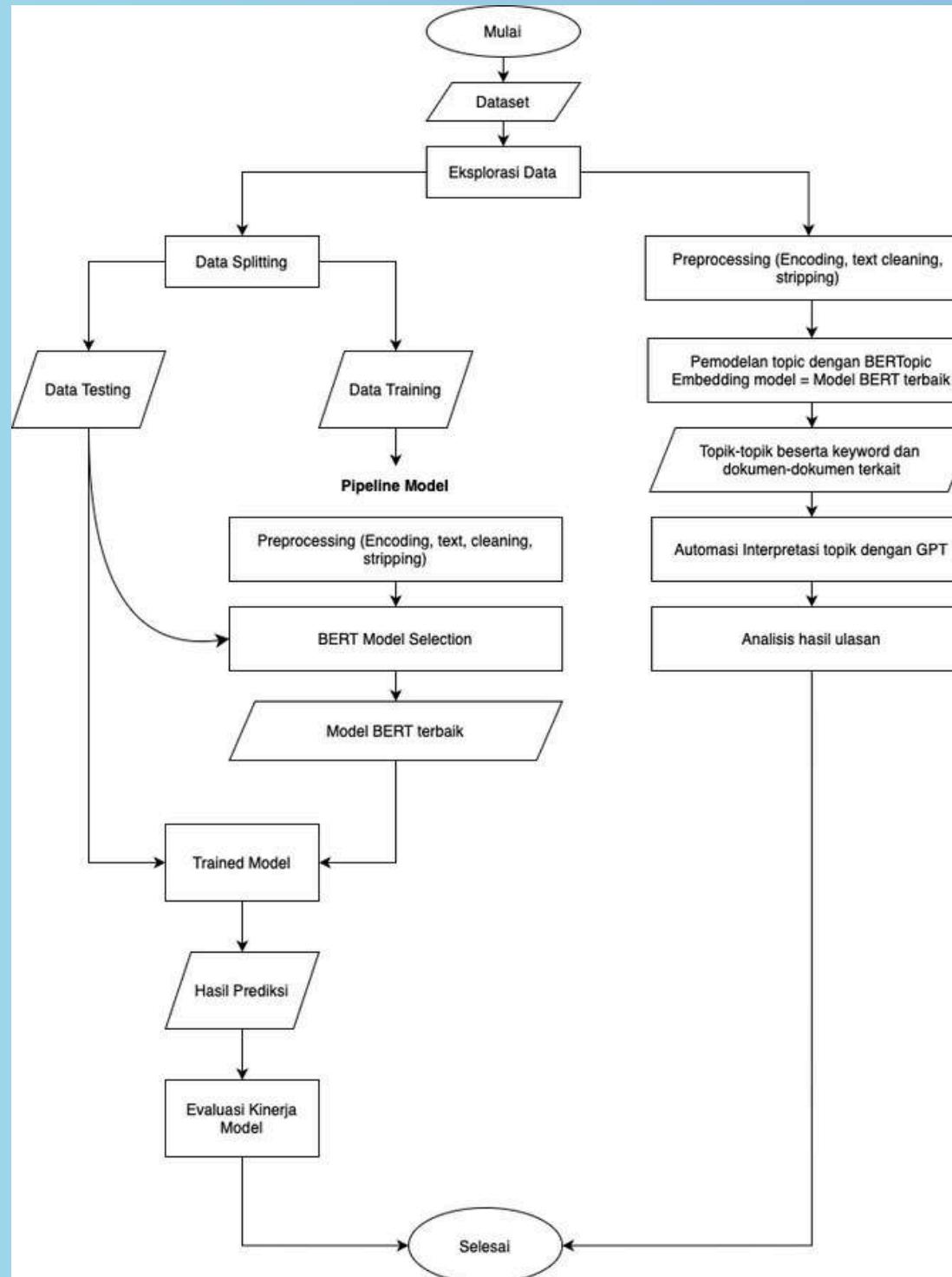
Simulasi Analisis



SIMULASI ANALISIS

33

Research Diagram



Preprocessing

Import Library dan Membaca Dataset

```
!pip install bertopic  
!pip install sentence-transformers  
!pip install demoji  
  
import os  
import re  
import demoji  
import kagglehub  
import pandas as pd  
import plotly.express as px  
from bertopic import BERTopic  
import pandas as pd  
from sentence_transformers import SentenceTransformer  
  
path = kagglehub.dataset_download("gevabriel/ibu-kota-nusantara")  
print("Path to dataset files:", path)  
  
Path to dataset files: /kaggle/input/ibu-kota-nusantara  
  
[ # Missing Value  
] print("Missing value:")  
print(df.isnull().sum())  
  
↳ Missing value:  
tweet      0  
sentiment   0  
dtype: int64  
  
[ # Remove duplicates  
] df_no_duplicates = df.drop_duplicates(subset=["tweet"])  
print(f"Removed {df.shape[0]} - {df_no_duplicates.shape[0]} duplicate entries")  
df = df_no_duplicates  
  
↳ Removed 4 duplicate entries
```

1. Import Library

Pertama, menginstalasi pustaka yang dibutuhkan seperti bertopic, sentence-transformers, dan demoji, yang masing-masing digunakan untuk ekstraksi topik berbasis model transformer, pembuatan embedding kalimat, serta pembersihan emoji dari teks. Setelah itu, berbagai library penting diimpor, mulai dari pandas untuk manipulasi data, plotly untuk visualisasi, hingga kagglehub untuk mengakses dataset secara langsung dari Kaggle.

2. Handling Missing Values and Duplicates

Melakukan pemeriksaan terhadap nilai yang hilang. Kemudian, data duplikat dihapus berdasarkan kolom tweet. Jumlah data yang dihapus juga ditampilkan agar pengguna tahu seberapa banyak tweet yang tidak unik.

Preprocessing

Import Dataset dan EDA

```
[# Missing Value
] print("Missing value:")
print(df.isnull().sum())

→ Missing value:
tweet      0
sentiment   0
dtype: int64

[# Remove duplicates
] df_no_duplicates = df.drop_duplicates(subset=["tweet"])
print(f"Removed {df.shape[0] - df_no_duplicates.shape[0]} duplicate entries")
df = df_no_duplicates

→ Removed 4 duplicate entries
```

3. Import Dataset

Mengimpor dataset dari **Kaggle** menggunakan library kagglehub, menyimpannya dalam format CSV, kemudian memuatnya ke dalam *df pandas* untuk dianalisis. Kita memeriksa struktur data awal dengan menampilkan 10 baris pertama, mengecek tipe data dan missing values menggunakan *info()*, serta melihat statistik deskriptif untuk memahami distribusi label sentimen. Dataset ini terdiri dari kolom **tweet** yang berisi teks dan **sentiment** sebagai label klasifikasi.

4. EDA

Melakukan eksplorasi terhadap struktur data dengan pengecekan **missing value** menggunakan *df.isnull().sum()*, lalu **menghapus duplikat** pada kolom teks dengan *df.drop_duplicates()*. Setelah itu, dicetak jumlah data yang dihapus (sebanyak 4 entri). Terakhir, dibuat **fungsi color_sentiment()** yang memberi warna latar berbeda pada nilai kolom sentiment diterapkan ke 10 data awal dengan *.style.applymap()* agar hasilnya lebih mudah dibaca secara visual.

SIMULASI ANALISIS

EDA

Hasil Visualisasi

```
# Text length analysis
df['text_length'] = df[text_column].apply(lambda x: len(str(x).split()))
plt.figure(figsize=(10, 6))
sns.histplot(data=df, x='text_length', hue=label_column, kde=True, bins=50)
plt.title('Text Length Distribution by Class')
plt.xlabel('Text Length (words)')
plt.ylabel('Count')
plt.tight_layout()
plt.show()
```

```
# Word frequency analysis (top 20 words)
all_words = ' '.join(df[text_column].fillna('')).split()
word_freq = Counter(all_words)
common_words = pd.DataFrame(word_freq.most_common(20), columns=['Word', 'Frequency'])
plt.figure(figsize=(12, 8))
sns.barplot(x='Frequency', y='Word', data=common_words)
plt.title('Top 20 Most Common Words')
plt.tight_layout()
plt.show()
```

5. Text Length Analysis

Mengukur panjang teks dalam jumlah kata. Ini membantu untuk memahami apakah teks yang lebih panjang atau lebih pendek memiliki hubungan dengan label atau kelas tertentu dalam analisis sentimen. Lalu Menggunakan `sns.histplot` untuk membuat histogram yang menggambarkan distribusi panjang teks berdasarkan kelas (label) dengan menambahkan kernel density estimate (KDE) untuk memperhalus distribusi.

6. Word Frequency Analysis

Dilakukan dengan penggabungan semua kata, pemecahan teks ke dalam kata, penghitungan frekuensi kata, dan mengambil 20 Kata Teratas. Untuk mendapatkan daftar kata-kata yang paling sering muncul dalam dataset.

SIMULASI ANALISIS

EDA

Hasil Visualisasi

```
# Label Distribution
label_counts = df[label_column].value_counts()
labels = label_counts.index
sizes = label_counts.values

# Donut Chart
fig, ax = plt.subplots(figsize=(8,6))
colors = plt.cm.tab20.colors

wedges, texts, autotexts = ax.pie(
    sizes,
    labels=labels,
    autopct='%.1f%%',
    startangle=70,
    wedgeprops=dict(width=0.4),
    textprops=dict(color="black"),
    colors=colors
)

for text in texts:
    text.set_fontsize(12)
for autotext in autotexts:
    autotext.set_fontsize(11)
    autotext.set_weight("bold")

centre_circle = plt.Circle((0,0),0.70,fc='white')
fig.gca().add_artist(centre_circle)

ax.set_title('Distribusi Label IKN', fontsize=11, fontweight='bold')
plt.axis('equal')
plt.tight_layout()
plt.show()
```

7. Donut Chart

Menghitung ulang distribusi label dengan fungsi `value_counts()`, tetapi kali ini digunakan untuk keperluan visualisasi lebih lanjut (dalam bentuk donut chart). Membuat donut chart dengan fungsi `pie`. Chart ini akan menggambarkan distribusi label dalam bentuk grafik yang menunjukkan persentase dari setiap kelas dengan ukuran irisan yang proporsional.

Visualisasi dengan donut chart menunjukkan sentimen positif terhadap IKN mendominasi dengan 70,27% dari total cuitan yang dianalisis. Sentimen negatif mencakup 29,73%, menunjukkan bahwa meskipun mayoritas bersikap positif, masih ada proporsi signifikan yang menunjukkan kritik atau ketidaksetujuan.

Preprocessing

Data Cleaning

```
# Preprocessing Data Function

# Data Cleaning For BERT (IndoBERT)
def clean_text_bert(text):
    if not isinstance(text, str):
        return ""

    # Remove URLs
    text = re.sub(r"http\S+|www\S+|https\S+", "", text)

    # Remove emails
    text = re.sub(r"\S+@\S+", "", text)

    # Remove @username (Twitter mentions)
    text = re.sub(r'@[\w]+', '', text)

    # Remove hashtags (keep the text after #)
    text = re.sub(r"#[\w]+", r"<hashtag><hashtag>", text)

    # Replace numbers with spaces
    text = re.sub(r"\d+", " ", text)

    # Remove emojis using demoji
    text = demoji.replace_with_desc(text, sep=" ")

    # Hilangkan karakter encoding error
    text = re.sub(r"\u([\u0080-\u00ff]{2})", "", text)
    text = re.sub(r"\u([\u00a0-\u00ff]{2})\u([\u0080-\u00ff]{1})", "", text)

    # Hilangkan simbol atau karakter aneh
    text = re.sub(r"[\u2000-\u200f]", "", text)

    # Hapus kata bisa encoding seperti 'aa', 'aa', dst
    text = re.sub(r"\b([a-z]{1,2})\b", "", text)

    # Hapus kata sangat pendek yang tidak bermakna (<2 huruf)
    text = " ".join([word for word in text.split() if len(word) > 2])

    # Keep punctuation, numbers, and other structural elements for BERT

    return text

# Stripping
def strip_text(text):
    return " ".join(text.split()) if isinstance(text, str) else ""

# Complete preprocessing pipeline for BERT
def preprocess_text_bert(text, use_stem=False):
    #text = case_folding(text)
    text = clean_text_bert(text)
    text = strip_text(text)
    return text
```

8. Text Preprocessing

- Menghapus URL, email, dan mention (@username) untuk mengurangi noise dari media sosial.
- Mengubah hashtag menjadi format khusus (<hashtag>teks<hashtag>) agar tetap dikenali konteksnya.
- Menghapus angka dan karakter encoding error yang tidak relevan untuk analisis teks.
- Menghapus emoji dan simbol aneh menggunakan demoji agar model fokus pada teks bermakna.
- Menghapus kata sangat pendek (≤ 2 huruf) yang umumnya tidak bermakna secara semantik.
- Melakukan normalisasi spasi dengan fungsi strip_text untuk merapikan teks akhir.

SIMULASI ANALISIS

Model Selection

Model Selection

```
for model_name in model_names:  
    print(f"\n🔍 Evaluating model: {model_name}")  
  
    tokenizer = AutoTokenizer.from_pretrained(model_name)  
    model = AutoModel.from_pretrained(model_name)  
  
    # Embedding CLS  
    X_train_emb = get_cls_embeddings(texts_train, tokenizer, model)  
    X_test_emb = get_cls_embeddings(texts_test, tokenizer, model)  
  
    # Klasifikasi pakai Logistic Regression  
    clf = LogisticRegression(max_iter=1000)  
    clf.fit(X_train_emb, labels_train)  
    y_pred = clf.predict(X_test_emb)  
  
    acc = accuracy_score(labels_test, y_pred)  
    print("Accuracy:", acc)  
    print(classification_report(labels_test, y_pred))  
  
    results.append((model_name, acc))
```

📊 Model Selection Summary:

indobenchmark/indobert-base-p1: Accuracy = 0.9075
indolem/indobertweet-base-uncased: Accuracy = 0.9144
google-bert/bert-base-uncased: Accuracy = 0.8733

9. Model Selection

Tiga model BERT diuji untuk klasifikasi sentimen menggunakan embedding [CLS] dan Logistic Regression. Hasilnya menunjukkan bahwa model berbasis Bahasa Indonesia memberikan performa terbaik.

- **IndoBERTweet unggul** dengan akurasi **91,44%**, cocok untuk teks informal.
- IndoBERT Base menyusul dengan 90,75%, kuat untuk Bahasa Indonesia umum.
- BERT Base (Inggris) tertinggal dengan 87,33%, menunjukkan pentingnya model berbasis bahasa lokal.

Modelling

IndoBERTweet

```
import torch
import torch.nn as nn
import numpy as np
from transformers import AutoTokenizer, AutoModel, get_scheduler
from torch.optim import AdamW
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from torch.utils.data import Dataset, DataLoader
import matplotlib.pyplot as plt

# Konstanta
EPOCHS = 10
BATCH_SIZE = 32

# 1. Load IndoBERTweet dan Tokenizer
model_name = "indolem/indobertweet-base-uncased"
tokenizer = AutoTokenizer.from_pretrained(model_name)
bert = AutoModel.from_pretrained(model_name)

# 2. Preprocessing Data
texts = []
labels = []

for i in range(len(X_train)):
    text = X_train[i]
    label = y_train[i]
    texts.append(text)
    labels.append(label)

# 3. Dataset Class
class BERTDataset(Dataset):
    def __init__(self, texts, labels, tokenizer, max_len=128):
        self.texts = texts.tolist()
        self.labels = labels.tolist()
        self.tokenizer = tokenizer
        self.max_len = max_len

    def __len__(self):
        return len(self.texts)

    def __getitem__(self, idx):
        encoding = self.tokenizer(
            self.texts[idx],
            padding='max_length',
            truncation=True,
            max_length=self.max_len,
            return_tensors='pt'
        )

        return (
            encoding['input_ids'].squeeze(0),
            encoding['attention_mask'].squeeze(0),
            torch.tensor(self.labels[idx], dtype=torch.long)
        )

# 4. Data Split
X_train_final, X_val, y_train_final, y_val = train_test_split(
    X_train, y_train, test_size=0.1, stratify=y_train, random_state=42
)

# 5. Dataset & Dataloader
train_dataset = BERTDataset(X_train_final, y_train_final, tokenizer)
val_dataset = BERTDataset(X_val, y_val, tokenizer)
test_dataset = BERTDataset(X_test, y_test, tokenizer)

train_dataloader = DataLoader(train_dataset, batch_size=BATCH_SIZE, shuffle=True)
val_dataloader = DataLoader(val_dataset, batch_size=BATCH_SIZE, shuffle=False)
test_dataloader = DataLoader(test_dataset, batch_size=BATCH_SIZE, shuffle=False)
```

10. Fine-Tuning IndoBERTweet

Proses dimulai dengan memuat model pralatih IndoBERTweet dan tokenizer-nya. Model ini disiapkan untuk pelatihan dengan menggunakan optimizer AdamW yang dilengkapi weight decay untuk mencegah overfitting, serta fungsi loss CrossEntropyLoss karena tugasnya adalah klasifikasi teks. Selanjutnya, dibuat model klasifikasi khusus yang memanfaatkan output token CLS dari BERT, dilanjutkan dengan dropout yang diperbesar untuk meningkatkan regularisasi, dan diakhiri dengan lapisan linear untuk prediksi. Data teks kemudian diproses menggunakan class BERTDataset yang bertugas men-tokenisasi dan mengubah data menjadi format tensor yang bisa dibaca model. Dataset ini dibungkus dalam DataLoader agar data dapat diproses dalam batch saat pelatihan dan evaluasi. Model dan data dipindahkan ke perangkat yang tersedia (GPU jika ada, atau CPU). Dengan pengaturan ini, proses training dan evaluasi dapat berjalan efisien dan optimal.

SIMULASI ANALISIS

41

Modelling

IndoBERTweet

```
# 6. Model, Optimizer, Loss, Scheduler
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model = BERTClassifier(bert).to(device)

optimizer = AdamW(model.parameters(), lr=1e-5, weight_decay=0.01)
criterion = nn.CrossEntropyLoss()

num_training_steps = len(train_dataloader) * EPOCHS
scheduler = get_scheduler(
    name="linear",
    optimizer=optimizer,
    num_warmup_steps=0,
    num_training_steps=num_training_steps,
)

# 7. Train & Eval Functions
def train_epoch(model, dataloader):
    model.train()
    losses = []
    preds_all, labels_all = [], []

    for batch in dataloader:
        input_ids, attention_mask, labels = [b.to(device) for b in batch]
        optimizer.zero_grad()
        outputs = model(input_ids, attention_mask)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
        scheduler.step()

        losses.append(loss.item())
        preds = torch.argmax(outputs, dim=1)
        preds_all.extend(preds.detach().cpu().numpy())
        labels_all.extend(labels.detach().cpu().numpy())

    acc = accuracy_score(labels_all, preds_all)
    return np.mean(losses), acc
```

```
def eval_epoch(model, dataloader):
    model.eval()
    losses = []
    preds_all, labels_all = [], []

    with torch.no_grad():
        for batch in dataloader:
            input_ids, attention_mask, labels = [b.to(device) for b in batch]
            outputs = model(input_ids, attention_mask)
            loss = criterion(outputs, labels)

            losses.append(loss.item())
            preds = torch.argmax(outputs, dim=1)
            preds_all.extend(preds.detach().cpu().numpy())
            labels_all.extend(labels.detach().cpu().numpy())

    acc = accuracy_score(labels_all, preds_all)
    return np.mean(losses), acc

def plot_metrics(train_losses, val_losses, test_losses, val_accs, test_accs):
    epochs = range(len(train_losses))
    plt.figure(figsize=(12,5))

    plt.subplot(1,2,1)
    plt.plot(epochs, train_losses, label='Train Loss')
    plt.plot(epochs, val_losses, label='Val Loss')
    plt.plot(epochs, test_losses, label='Test Loss')
    plt.legend()
    plt.title('Loss per Epoch')

    plt.subplot(1,2,2)
    plt.plot(epochs, val_accs, label='Val Accuracy')
    plt.plot(epochs, test_accs, label='Test Accuracy')
    plt.legend()
    plt.title('Accuracy per Epoch')

    plt.show()
```

11. Fine-Tuning

Proses ini merupakan implementasi pipeline pelatihan dan evaluasi model NLP menggunakan PyTorch, dengan tambahan visualisasi performa. Pertama, get_scheduler digunakan untuk mengatur penjadwalan learning rate secara linear berdasarkan jumlah total langkah pelatihan (num_training_steps). Fungsi train_epoch menjalankan proses pelatihan model dalam satu epoch, mencakup forward pass, perhitungan loss, backpropagation, serta update parameter menggunakan optimizer dan scheduler. Di sisi lain, eval_epoch digunakan untuk evaluasi model tanpa proses training, menggunakan torch.no_grad() agar lebih efisien. Kedua fungsi ini juga menghitung akurasi dan rata-rata loss untuk setiap epoch. Setelah proses pelatihan selesai, fungsi plot_metrics digunakan untuk menampilkan grafik loss (train, val, test) dan akurasi (val, test) per epoch, sehingga memudahkan analisis performa model dari waktu ke waktu. Ukuran batch yang digunakan selama proses ini ditetapkan sebesar 32 melalui BATCH_SIZE = 32.

Modelling

IndoBERTweet

```
# 8. Training Loop + Early Stopping
train_losses, val_losses, test_losses = [], [], []
val_accs, test_accs = [], []

patience = 2
counter = 0
best_val_loss = float('inf')

for epoch in range(EPOCHS):
    train_loss, train_acc = train_epoch(model, train_dataloader)
    val_loss, val_acc = eval_epoch(model, val_dataloader)
    test_loss, test_acc = eval_epoch(model, test_dataloader)

    train_losses.append(train_loss)
    val_losses.append(val_loss)
    test_losses.append(test_loss)
    val_accs.append(val_acc)
    test_accs.append(test_acc)

    print(f'Epoch {epoch+1}/{EPOCHS} | Train Loss: {train_loss:.4f} | Val Loss: {val_loss:.4f} |')
    print(f'Val Acc: {val_acc:.4f} | Test Acc: {test_acc:.4f}')

    if val_loss < best_val_loss:
        best_val_loss = val_loss
        torch.save(model.state_dict(), 'best_model.pt')
        counter = 0
    else:
        counter += 1
        if counter >= patience:
            print("Early stopping triggered!")
            break
```

12. Fine-Tuning

Proses ini menunjukkan proses pelatihan model machine learning dengan penambahan validasi dan early stopping menggunakan PyTorch. Pertama, data pelatihan (X_{train}) dibagi lagi sebanyak 10% menjadi data validasi (X_{val}) menggunakan `train_test_split`. Proses ini penting untuk mengevaluasi performa model di luar data latih. Selanjutnya, loop pelatihan berjalan selama jumlah epoch yang ditentukan (10 epoch), di mana pada setiap epoch, model dilatih menggunakan `train_epoch` dan dievaluasi menggunakan `eval_epoch` pada data validasi dan data uji. Hasil loss dan akurasi disimpan dalam list untuk keperluan visualisasi atau pelacakan performa.

Selain itu, diterapkan mekanisme early stopping dengan parameter `patience = 2`. Jika model tidak menunjukkan perbaikan loss validasi selama dua epoch berturut-turut, proses pelatihan dihentikan lebih awal. Saat terjadi perbaikan, model disimpan dalam file `'best_model.pt'`. Strategi ini berguna untuk mencegah overfitting dan menghemat waktu pelatihan jika performa model sudah stagnan.

Modelling

Topic Detection

```
[ ] df["tweet_bersih"] = df["tweet"].apply(preprocess_text_bert)
docs = df["tweet_bersih"].tolist()
sentiments = df["sentiment"].tolist()

[ ] topic_model = BERTopic()

[ ] topics, probs = topic_model.fit_transform(docs)

[ ] topic_model.get_topic_info()
```

Topic	Count	Name	Representation	Representative_Docs	
0	-1	438	-1_ibu_pindah_kota_setuju	[ibu, pindah, kota, setuju, ikn, dan, yang, pe...]	[Saya sebagai orang kalimantan timur tidak set...
1	0	74	0_perekonomian_besar_manfaat_indonesia	[perekonomian, besar, manfaat, indonesia, tiga...	[Tiga manfaat besar IKN untuk Perekonomian Ind...
2	1	49	1_investasi_hastagbhastag_investor_hastagarsja...	[investasi, hastagbhastag, investor, hastagars...	[Ajakan Kadin kepada pengusaha global untuk In...
3	2	46	2_tolak_hastagiknproyekoligarkihastag_pemindah...	[tolak, hastagiknproyekoligarkihastag, peminda...	[hastagIKNProyekOligarkihastag Pemindahan Ibu ...]
4	3	45	3_timur_jawa_dukung_iknxexxa	[timur, jawa, dukung, iknxexxa, masyarakat, pe...	[Masyarakat Jawa Timur bangga dan terus dukung...
5	4	45	4_nhastagkotaduniauntuksemuahastagnhastagiknse...	[nhastagkotaduniauntuksemuahastagnhastagiknsej...	[Dukung dan Kawal Pemindahan IKN Nusantara nha...

14. Model BERTopic

- Menggunakan metode topik modeling BERTopic untuk mengidentifikasi tema utama dalam kumpulan dokumen teks.
- Dimulai dengan mengubah dokumen menjadi vektor menggunakan model BERT, kemudian dilakukan clustering untuk mengelompokkan dokumen ke dalam topik-topik tertentu.
- BERTopic mengekstraksi kata-kata kunci dan dokumen perwakilan dari setiap topik untuk menggambarkan isi utama masing-masing kelompok.
- BERTopic berhasil mengelompokkan keseluruhan data menjadi 43 topik utama, termasuk satu topik outlier yang diklasifikasikan sebagai topik -1 yang terdiri atas dokumen-dokumen yang tidak memiliki kemiripan cukup signifikan dengan kelompok topik lainnya.

Modelling

Topic Detection

```
[ ] topic_model.get_topic(0)
[+] [('perekonomian', np.float64(0.0722124904161524)),
     ('besar', np.float64(0.06243668725140407)),
     ('manfaat', np.float64(0.061179180907237285)),
     ('indonesia', np.float64(0.06093732079084105)),
     ('tiga', np.float64(0.06028878984468611)),
     ('untuk', np.float64(0.04819994078326227)),
     ('hashtagiknsejahteraikanindonesiahastagxexxa',
      np.float64(0.040017200361755934)),
     ('nndki', np.float64(0.03937079856047442)),
     ('jakarta', np.float64(0.03826939997980872)),
     ('nhastagkotaduniauntuksemuahastag', np.float64(0.037711432135372695))]

[ ] df = pd.DataFrame({"topic": topics, "document": docs, "sentiment": sentiments})
df

[+] Show hidden output

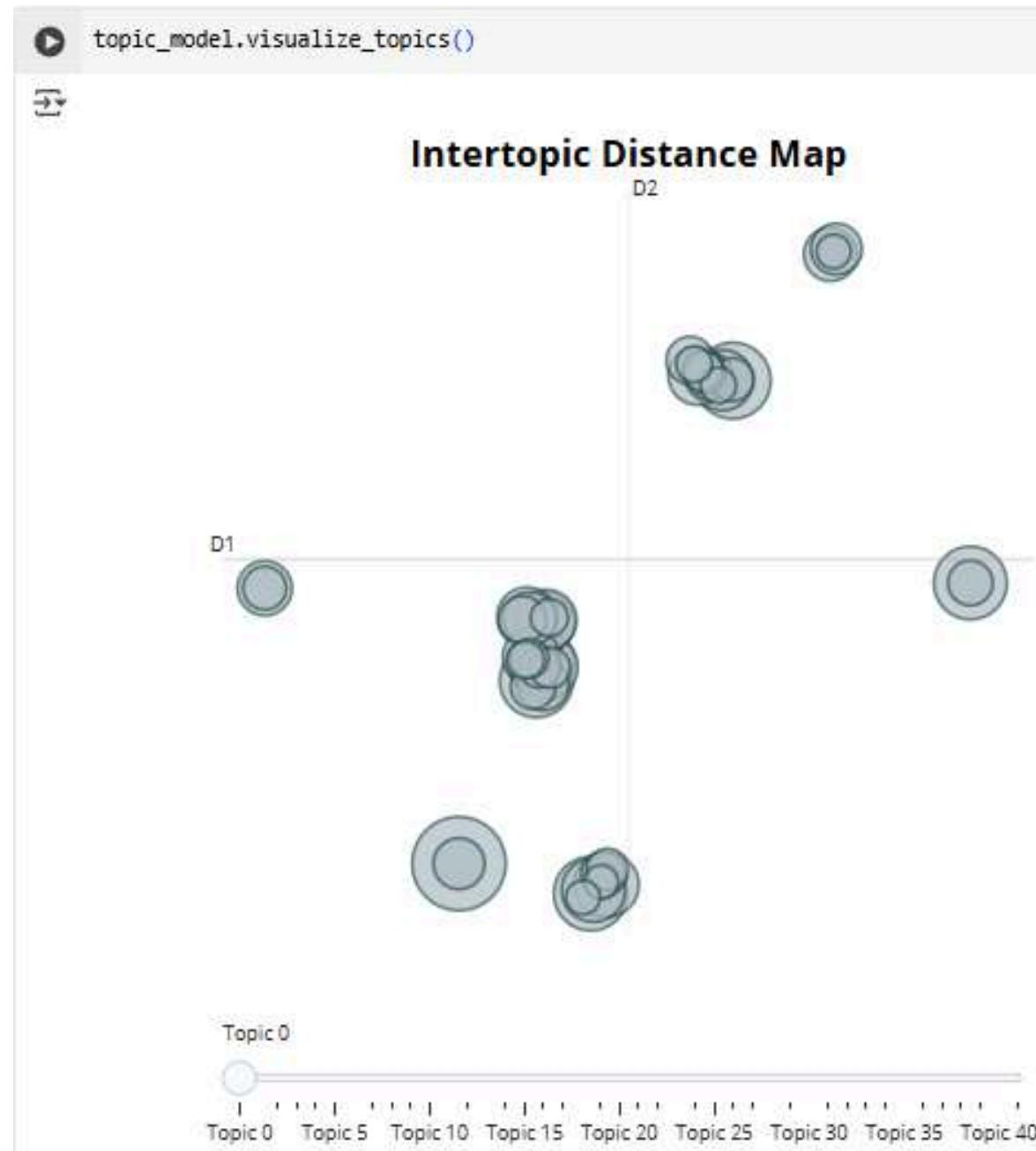
[ ] df.to_csv("output_with_topics_and_sentiment.csv", index=False)
```

15. Ekstraksi Kata Kunci, Dokumen, dan Penyimpanan Hasil Klasifikasi Topik

- Kata kunci lengkap yang merepresentasikan setiap topik dapat dilihat menggunakan `topic_model.get_topic(n)`, di mana `n` adalah ID topik yang ingin dilihat.
- Untuk melihat dokumen-dokumen yang paling mewakili suatu topik tertentu, dapat menggunakan `topic_model.get_representative_docs(n)`.
- Hasil klasifikasi setiap dokumen—berisi informasi topik, isi dokumen, dan sentimen—dapat dimasukkan ke dalam sebuah DataFrame menggunakan `pd.DataFrame(...)`. Data ini kemudian disimpan ke dalam file CSV.

Modelling

Hasil Visualisasi

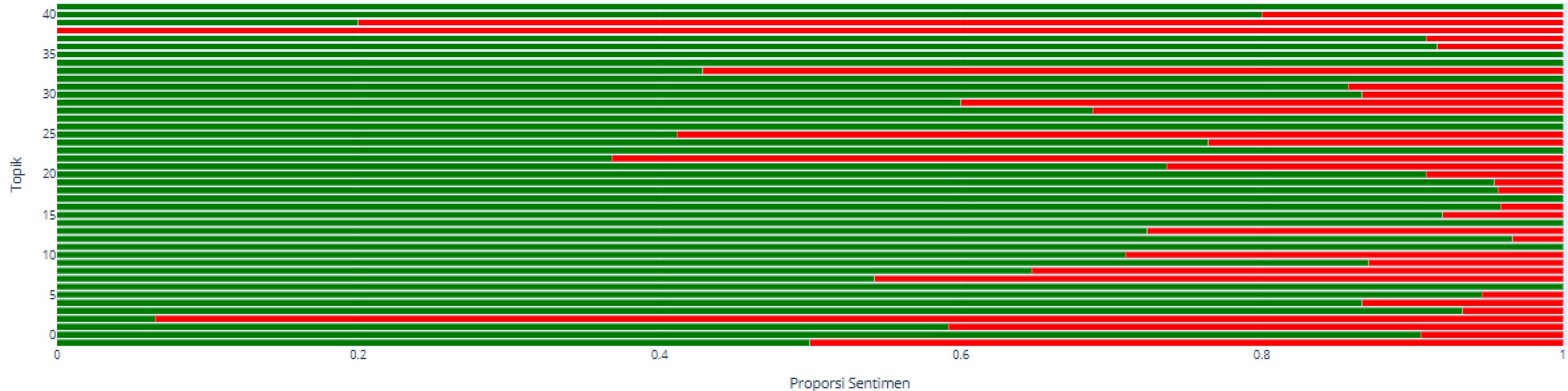


16. Intertopic Distance Map

- Terdapat beberapa kelompok topik yang saling berdekatan, menunjukkan kemiripan tema dalam klaster tersebut.
- Salah satu topik di kiri bawah memiliki lingkaran besar, menandakan topik ini sangat dominan dalam kumpulan data.

Modelling

Hasil Visualisasi



17. Distribusi Sentimen per Topik

- Visualisasi ini menampilkan proporsi sentimen positif dan negatif pada setiap topik yang dihasilkan dari model BERTopic, dengan rincian sebagai berikut:
- Sumbu y** menampilkan daftar topik, **sumbu x** menunjukkan proporsi masing-masing sentimen (positif dan negatif) (%)
- Hijau = Sentimen positif, Merah = Sentimen negatif
- Beberapa topik seperti Topik 2, 22, 38, dan 39 memiliki proporsi negatif yang tinggi, lainnya menunjukkan dominasi sentimen positif.

Preprocessing

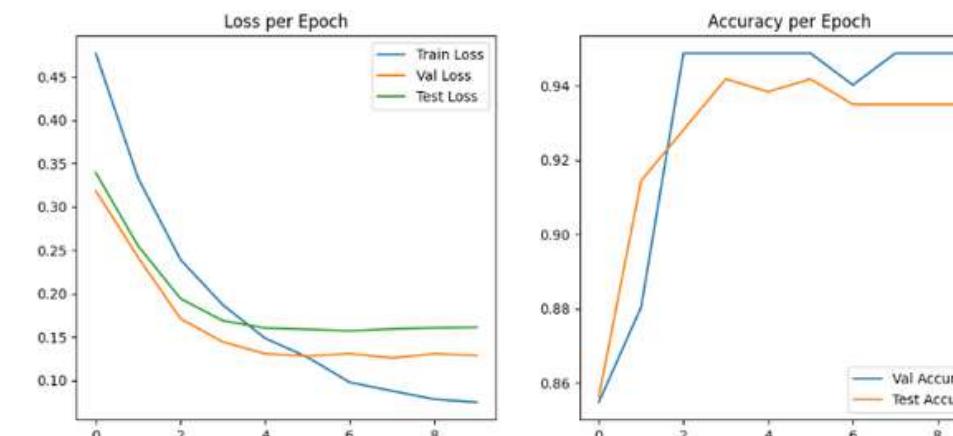
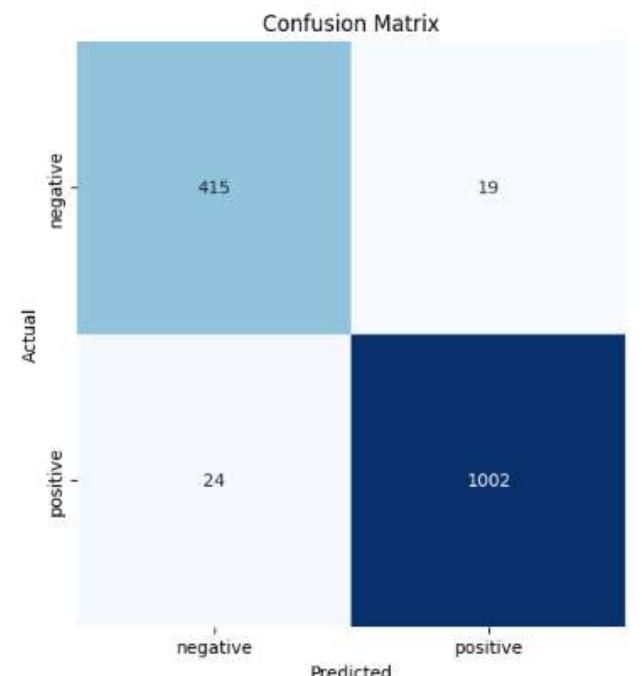
Tahap	Nama Proses	Output
1	Data	@TheArieAir Kita sepakat aja prof... Semua kita dukung rencana pemindahan Ibu Kota...
2	Data Cleaning	Kita sepakat aja prof Semua kita dukung rencana pemindahan Ibu Kota
3	Text Cleaning	kita sepakat saja prof semua kita dukung rencana pemindahan ibu kota
4	Stripping	kita sepakat saja prof semua kita dukung rencana pemindahan ibu kota
5	Tokenizing (BERT Tokenizer)	[CLS], kita, sepakat, saja, prof, semua, kita, dukung, rencana, pemindahan, ibu, kota, [SEP]

Hasil Model IndoBERTweet

```

Epoch 1/10 | Train Loss: 0.4768 | Val Loss: 0.3183 | Test Loss: 0.3393
Val Acc: 0.8547 | Test Acc: 0.8562
Epoch 2/10 | Train Loss: 0.3330 | Val Loss: 0.2417 | Test Loss: 0.2549
Val Acc: 0.8803 | Test Acc: 0.9144
Epoch 3/10 | Train Loss: 0.2390 | Val Loss: 0.1709 | Test Loss: 0.1942
Val Acc: 0.9487 | Test Acc: 0.9281
Epoch 4/10 | Train Loss: 0.1869 | Val Loss: 0.1442 | Test Loss: 0.1686
Val Acc: 0.9487 | Test Acc: 0.9418
Epoch 5/10 | Train Loss: 0.1484 | Val Loss: 0.1306 | Test Loss: 0.1602
Val Acc: 0.9487 | Test Acc: 0.9384
Epoch 6/10 | Train Loss: 0.1266 | Val Loss: 0.1279 | Test Loss: 0.1587
Val Acc: 0.9487 | Test Acc: 0.9418
Epoch 7/10 | Train Loss: 0.0976 | Val Loss: 0.1307 | Test Loss: 0.1567
Val Acc: 0.9402 | Test Acc: 0.9349
Epoch 8/10 | Train Loss: 0.0876 | Val Loss: 0.1257 | Test Loss: 0.1592
Val Acc: 0.9487 | Test Acc: 0.9349
Epoch 9/10 | Train Loss: 0.0781 | Val Loss: 0.1305 | Test Loss: 0.1604
Val Acc: 0.9487 | Test Acc: 0.9349
Epoch 10/10 | Train Loss: 0.0746 | Val Loss: 0.1286 | Test Loss: 0.1610
Val Acc: 0.9487 | Test Acc: 0.9349
Early stopping triggered!

```



Classification Report:				
	precision	recall	f1-score	support
negative	0.9453	0.9562	0.9507	434
positive	0.9814	0.9766	0.9790	1026
accuracy			0.9705	1460
macro avg	0.9634	0.9664	0.9649	1460
weighted avg	0.9707	0.9705	0.9706	1460

- Gambar di samping menampilkan hasil evaluasi model setelah proses pelatihan, berupa log pelatihan, grafik metrik, classification report, dan confusion matrix.
- Grafik kiri menunjukkan tren penurunan pada train, validation, dan test loss. Ketiganya stabil tanpa perbedaan mencolok, mengindikasikan bahwa model tidak mengalami overfitting. Grafik kanan menunjukkan akurasi validation dan test meningkat tajam pada awal pelatihan, lalu stabil pada nilai tinggi (val > 0.94, test ~0.93). Early stopping aktif pada epoch ke-10 saat tidak ada lagi peningkatan signifikan pada akurasi validasi, menandakan model telah mencapai performa optimal.
- Di bagian bawah, terdapat classification report yang menunjukkan metrik evaluasi seperti precision, recall, dan f1-score untuk kedua kelas: “negative” dan “positive”. Nilai-nilai tersebut sangat tinggi (di atas 0.94), menandakan model bekerja dengan sangat baik.
- Confusion matrix memperlihatkan prediksi model terhadap data test: hanya sedikit kesalahan klasifikasi (19 dan 24 kesalahan dari total 1460 data), yang memperkuat bahwa model memiliki akurasi tinggi dan seimbang.

Automasi Interpretasi topik

Prompt Automasi Interpretasi Topik dengan GPT:

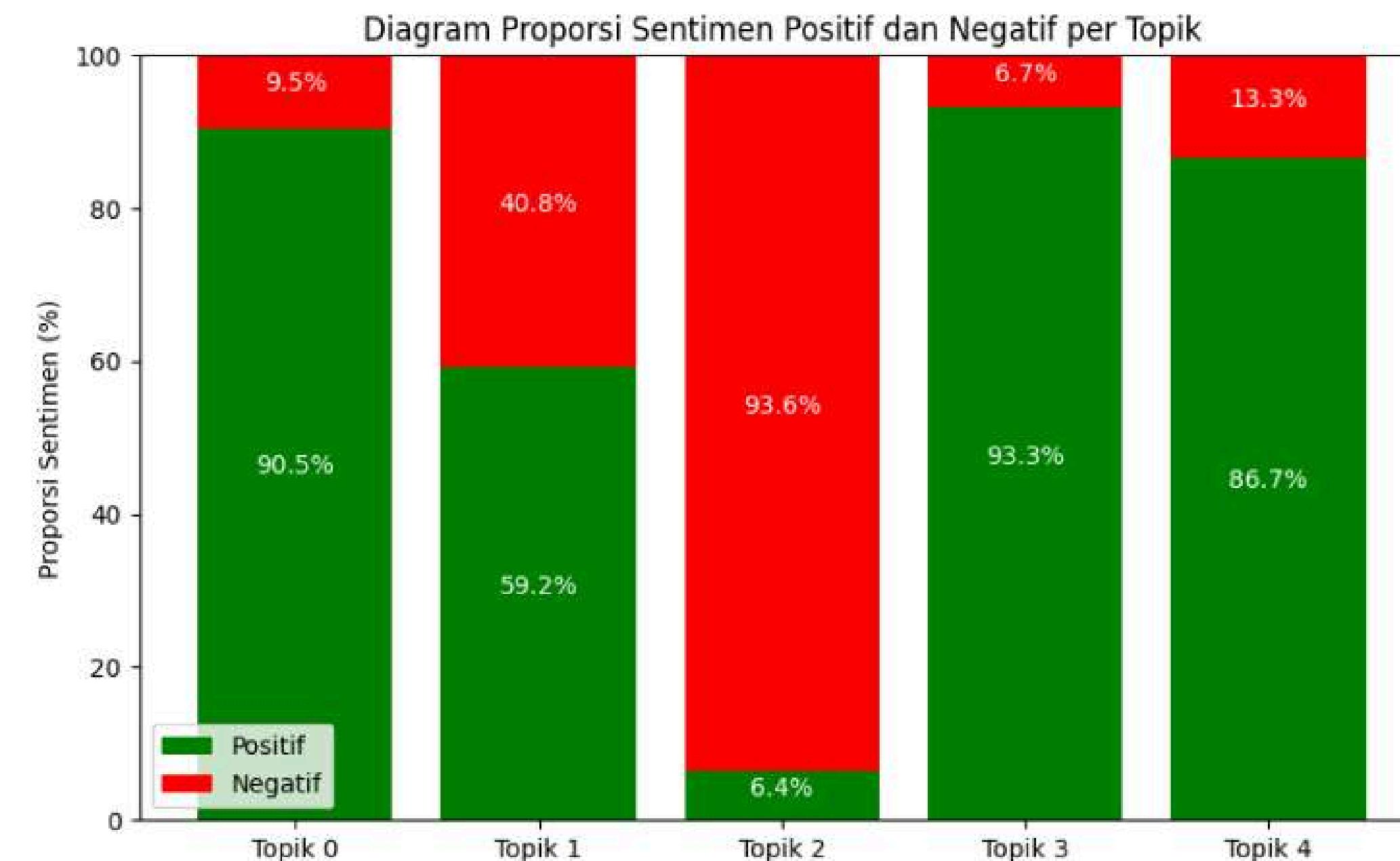
Tentukan topik utama dari [KEYWORD] berdasarkan dokumen berikut: [DOKUMEN] respon hanya dengan satu kalimat, gunakan bahasa indonesia yang formal.

Model: GPT 4.0 (limited)

No	TOPIK	KEYWORD
0	Dukungan terhadap pembangunan Ibu Kota Negara (IKN) Nusantara sebagai upaya strategis untuk memberikan manfaat besar bagi perekonomian Indonesia.	Perekonomian, besar, manfaat, indonesia, tiga, untuk, #iknsejahterakanindonesia, DKI, Jakarta, #kotaduniauntuksemua
1	Ajakan dan upaya KADIN serta pemerintah untuk menarik minat pengusaha dan investor global dalam menanamkan modal pada proyek pembangunan Ibu Kota Negara (IKN) Nusantara.	Investasi, investor, pengusaha, #iknnusantara, global, mengajak, KADIN, invest
2	Penolakan terhadap proyek pemindahan Ibu Kota Negara (IKN) yang dianggap sebagai proyek oligarki yang merugikan rakyat dan lingkungan, terutama hutan.	Tolak, #iknproyekoligarki, pemindahan, kota, bu, #ayotolakuanikn, proyek, oligarki, hutan, berjalan
3	Dukungan masyarakat Jawa Timur terhadap pembangunan Ibu Kota Negara (IKN) Nusantara yang dinilai telah dipikirkan matang dan diharapkan membawa kelancaran serta kemajuan.	Timur, jawa, dukung, ikn, masyarakat, pembangunan, matang, kelancaran, semoga, warga
4	Dukungan terhadap IKN Nusantara sebagai kota dunia yang diharapkan dapat menyejahterakan Indonesia dan mendorong pemerataan ekonomi nasional.	#kotaduniauntuksemua, #iknsejahterakanindonesia, #iknpemerataanekonomi, #kotaduniauntuksemua, sangat, nusantara, ikn, dunia, dukung

Topic Modelling

Gambar berikut menunjukkan diagram proporsi sentimen positif dan negatif untuk lima topik teratas pada dataset.



05

Kesimpulan

51



From Sawyer Merritt

OUTPUT

Kesimpulan

- Penelitian ini berhasil mengevaluasi kinerja model IndoBERTweet dalam mengklasifikasikan sentimen publik terhadap IKN dengan hasil yang memuaskan. Model mencapai akurasi tinggi sebesar 93,49%, dengan nilai precision, recall, dan F1-score yang seimbang pada kedua kelas sentimen, menunjukkan kemampuan generalisasi yang baik terhadap data baru. Proses pelatihan juga berlangsung stabil tanpa overfitting, didukung oleh penerapan early stopping yang efektif.
- Berdasarkan hasil topik modeling menggunakan BERTopic, ditemukan lima topik utama yang sering dibahas masyarakat terkait IKN. Analisis sentimen terhadap masing-masing topik menunjukkan bahwa mayoritas topik didominasi oleh sentimen positif, terutama topik 0, 3, dan 4 dengan proporsi positif di atas 86%. Di sisi lain, topik 2 didominasi oleh sentimen negatif (93,6%), menunjukkan adanya kritik publik terhadap isu-isu seperti oligarki dan kerusakan lingkungan. Hasil ini mengindikasikan bahwa percakapan publik di Twitter mengenai IKN secara umum bernuansa positif, namun tetap menyimpan kekhawatiran terhadap aspek-aspek tertentu dari proyek tersebut. Temuan ini memberikan gambaran yang lebih dalam mengenai opini publik dan dapat menjadi masukan penting bagi pengambil kebijakan.

06

Daftar Pustaka



DAFTAR PUSTAKA

54

- Kurniawan, M. R., Wicaksono, R. A., Munthe, J. A., Hidayat, V. Y., & Arifin, A. (2024). Membangun Ibu Kota Negara Nusantara (IKN) baru yang Berlandaskan Pancasila: Menuju Indonesia Emas 2045. *Nusantara: Jurnal Pendidikan, Seni, Sains dan Sosial Humaniora*, 2(01).
- Yusuf, A., Rizani, A., Fitri, R., Pamungkas, K. N. P., Saputra, W. A., & Shaddiq, S. (2024). SENTIMEN POSITIF ATAU NEGATIF: PERSPEKTIF MASYARAKAT TERHADAP PEMINDAHAN IBU KOTA NUSANTARA. *Masyarakat Indonesia*, 50(2), 277-300.
- Bhuvaneswari, A., & Kumudha, M. (2024, April). Topic Modeling Based Clustering of Disaster Tweets Using BERTopic. In 2024 MIT Art, Design and Technology School of Computing International Conference (MITADTSoCiCon) (pp. 1-6). IEEE.
- Grootendorst, M. (2022). BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794*.
- Palani, S., Rajagopal, P., & Pancholi, S. (2021). T-BERT: Model for sentiment analysis of micro-blogs integrating topic model and BERT. *arXiv Preprint, arXiv:2106.01097*.



THANK YOU



From Sawyer Merritt