

# Perbandingan Kinerja XGBoost dan Naive Bayes dalam Analisis Sentimen Komentar TikTok Terhadap Ibu Kota Nusantara (IKN) pada Data Tidak Seimbang

Novi Purnamasari, Nirwana Hendrastuty\*

Fakultas Teknik dan Ilmu Komputer, Sistem Informasi, Universitas Teknokrat Indonesia, Bandar Lampung, Indonesia

Email: <sup>1</sup>novi\_purnamasari@teknokrat.ac.id, <sup>2,\*</sup>nirwanahendrastuty@teknokrat.ac.id

Email Penulis Korespondensi: novi\_purnamasari@teknokrat.ac.id

Submitted: 06/03/2026; Accepted: 31/03/2026; Published: 31/03/2026

**Abstrak**—Perkembangan media sosial memunculkan berbagai tanggapan masyarakat terhadap pembangunan Ibu Kota Nusantara (IKN), termasuk melalui platform TikTok yang memiliki tingkat interaksi pengguna yang tinggi. Penelitian ini bertujuan untuk membandingkan kinerja algoritma Naive Bayes dan XGBoost dalam analisis sentimen komentar TikTok terkait pembangunan IKN pada kondisi data yang tidak seimbang. Sebanyak 1.132 komentar dikumpulkan dan diproses melalui tahapan pembersihan teks, meliputi case folding, text cleaning, tokenisasi, normalisasi, dan stemming. Ekstraksi fitur dilakukan menggunakan metode TF-IDF yang menghasilkan 1.926 fitur untuk merepresentasikan bobot kata dalam dokumen. Proses klasifikasi dilakukan dengan pembagian data latih dan data uji dengan proporsi 80% dan 20%. Hasil pengujian menunjukkan bahwa Naive Bayes memperoleh akurasi sebesar 61,23%, sedangkan XGBoost mencapai akurasi sebesar 62,11%. XGBoost menunjukkan peningkatan nilai recall pada kelas negatif (0,21 menjadi 0,40) dan kelas netral (0,11 menjadi 0,26), meskipun peningkatan tersebut masih terbatas. Perbedaan akurasi antar model relatif kecil sehingga belum menunjukkan peningkatan yang signifikan secara keseluruhan. Penelitian ini memiliki keterbatasan pada ukuran dataset yang relatif kecil dan distribusi data yang tidak seimbang, yang berpotensi mempengaruhi representativitas data dan kemampuan generalisasi model. Oleh karena itu, hasil yang diperoleh belum sepenuhnya optimal untuk penerapan dalam skala yang lebih luas.

**Kata Kunci:** IKN; Naive Bayes; Analisis Sentimen; TF-IDF; TikTok; Word Cloud; XGBoost

**Abstract**—The growth of social media has generated diverse public responses regarding the development of Indonesia's new capital city, Ibu Kota Nusantara (IKN), particularly on TikTok, a platform with high user interaction. This study aims to compare the performance of Naive Bayes and eXtreme Gradient Boosting (XGBoost) algorithms in sentiment analysis of TikTok comments related to IKN development under imbalanced data conditions. The dataset consists of 1,132 comments that underwent preprocessing, including case folding, text cleaning, tokenization, normalization, and stemming. Feature extraction was performed using the Term Frequency–Inverse Document Frequency (TF-IDF) method, generating 1,926 features to represent word importance. The classification process used an 80:20 split for training and testing data. The results show that Naive Bayes achieved an accuracy of 61.23%, while XGBoost obtained a slightly higher accuracy of 62.11%. XGBoost improved recall in the negative class (from 0.21 to 0.40) and neutral class (from 0.11 to 0.26), although the improvement remains limited. The difference in accuracy between the models is relatively small and does not indicate a significant overall performance improvement. This study is limited by the relatively small dataset size and imbalanced class distribution, which may affect data representativeness and model generalization. Therefore, the results are not yet optimal for broader real-world applications.

**Keywords:** IKN; Naive Bayes; Sentiment Analysis; TF-IDF; TikTok; Word Cloud; XGBoost

## 1. PENDAHULUAN

Media sosial telah menjadi sarana penting bagi publik untuk mengekspresikan opini mereka terhadap berbagai isu sosial dan kebijakan secara *real-time*[1]. TikTok dan platform serupa memungkinkan pengguna menyampaikan pendapat tentang fenomena sosial dan kebijakan dalam konteks digital[2]. *Sentiment analysis* pada isu kebijakan nasional dapat memetakan persepsi masyarakat terhadap program pemerintah[3]. Analisis sentimen dari teks media sosial memberikan wawasan kuantitatif terkait persepsi positif, negatif, dan netral masyarakat tentang suatu isu[4]. Analisis manual komentar media sosial tidak praktis untuk *dataset* besar, sehingga diperlukan pendekatan otomatis seperti *text mining*[5]. *Text mining* dan *sentiment analysis* digunakan untuk mengekstraksi pola dan informasi penting dari data teks media sosial yang tidak terstruktur sehingga menjadi informasi bermakna[6]. *Sentiment analysis* dapat mengelompokkan opini menjadi sentimen positif, negatif, dan netral berdasarkan teks input[7]. Analisis opini publik melalui *sentiment analysis* membantu pembuat kebijakan mengevaluasi respons sosial terhadap program pemerintah[8]. Dalam beberapa waktu terakhir, pembangunan Ibu Kota Nusantara (IKN) menjadi topik yang menimbulkan berbagai respons di media sosial, khususnya di TikTok. Berbagai komentar menunjukkan adanya pro dan kontra dari masyarakat, mulai dari dukungan terhadap pemerataan pembangunan hingga kritik terkait anggaran dan dampak sosial. Tingginya interaksi pengguna pada platform ini menjadikan TikTok sebagai sumber data yang relevan untuk menganalisis sentimen publik secara *real-time*.

Data media sosial bersifat tidak terstruktur dan beragam, sehingga memerlukan tahapan *preprocessing* sebelum dianalisis[9]. Proses seperti pembersihan teks, *tokenisasi*, *normalisasi* kata, dan pembobotan fitur menjadi langkah penting dalam meningkatkan kualitas data sebelum masuk ke tahap klasifikasi. *Preprocessing* termasuk pembersihan, *tokenisasi*, dan *normalisasi* teks sangat penting untuk analisis sentimen yang akurat[9]. *Term Frequency–Inverse Document Frequency (TF-IDF)* adalah teknik pembobotan yang efektif dalam mengekstraksi fitur dari data teks[10]. Dalam penelitian klasifikasi teks, algoritma *machine learning* sering digunakan untuk menentukan kategori sentimen.

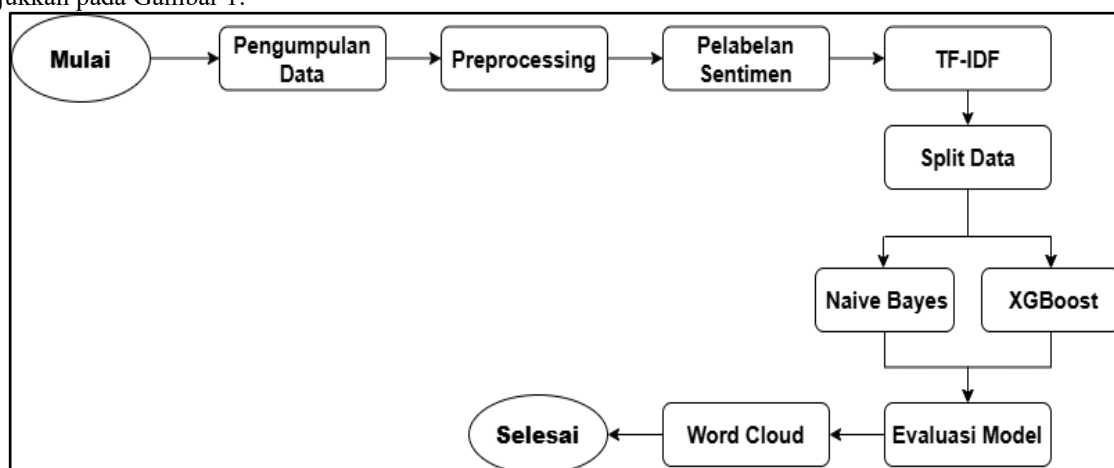
*Naive Bayes* adalah algoritma klasifikasi yang populer dalam analisis sentimen karena kesederhanaan dan performanya pada data teks[11]. *Naive Bayes* sering diaplikasikan dalam *sentiment analysis* untuk memetakan pola opini dalam data teks[12]. *XGBoost* merupakan metode klasifikasi *ensemble* yang sering memberikan performa tinggi dalam data teks[13]. Meskipun berbagai penelitian telah membandingkan performa algoritma klasifikasi seperti *Naive Bayes* dan *XGBoost*, sebagian besar penelitian tersebut dilakukan pada dataset umum dan belum secara spesifik mengkaji karakteristik data dari platform TikTok yang cenderung tidak terstruktur, informal, dan mengandung noise tinggi.

Selain itu, penelitian terdahulu umumnya hanya berfokus pada perbandingan performa algoritma tanpa mengaitkannya dengan konteks data tidak seimbang pada isu kebijakan publik tertentu. Selanjutnya, kajian-kajian terdahulu telah mengkaji secara komparatif. Kajian-kajian terdahulu telah mengkaji secara komparatif performa tiga pendekatan klasifikasi *Naive Bayes*, *Support Vector Machine*, serta *XGBoost* dalam konteks analisis sentimen, dengan tujuan mengidentifikasi metode yang paling optimal[13]. *Sentiment analysis* pada aplikasi TikTok telah digunakan untuk mengevaluasi respons publik terhadap platform digital[14]. Media berbasis video pendek terutama TikTok telah berevolusi menjadi repositori opini publik yang signifikan, khususnya dalam konteks evaluasi kebijakan negara. Viralitas konten yang berdurasi minimal ini, ditambah dengan penetrasi yang masif di kalangan pengguna Indonesia, menjadikannya medium yang representatif untuk menangkap dinamika sentimen sosial[15]. Dataset komentar dari platform sosial media menuntut serangkaian preparasi data yang komprehensif. Proses ini mencakup sanitasi karakter, segmentasi leksikal, dan standarisasi bentuk kata termasuk adaptasi terhadap variasi bahasa informal, akronim, serta simbol emosi guna memastikan kualitas input sebelum diproses oleh model pembelajaran mesin[16]. Perbandingan performa antara *Naive Bayes* dan *XGBoost* dalam klasifikasi sentimen menunjukkan bahwa *XGBoost* cenderung unggul dalam menangani *dataset* tidak seimbang dan fitur kompleks, sementara *Naive Bayes* memberikan keunggulan dalam efisiensi komputasional[17]. Seleksi fitur berbasis *Information Gain* terbukti efektif dalam meningkatkan akurasi klasifikasi dengan mengurangi dimensi data yang tidak relevan sebelum proses klasifikasi[18]. Metrik pembobotan *TF-IDF* diadopsi dalam penelitian ini berdasarkan kapasitasnya dalam mengkuantifikasi signifikansi terminologi unik dalam koleksi dokumen berbahasa Indonesia yang bersumber dari interaksi sosial media[19]. Integrasi antara teknik *text mining* dan framework *Decision Support Systems (DSS)* memberikan landasan sistematis bagi pembuat kebijakan untuk mengevaluasi respons publik terhadap pembangunan nasional berdasarkan data yang terstruktur[20]. Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk membandingkan kinerja algoritma *Naive Bayes* dan *XGBoost* dalam analisis sentimen komentar TikTok terkait pembangunan IKN, serta mengevaluasi pengaruh ketidakseimbangan data terhadap performa kedua algoritma tersebut.

## 2. METODOLOGI PENELITIAN

### 2.1 Tahapan Penelitian

Penelitian ini dilakukan melalui tahapan yang disusun secara sistematis untuk mengklasifikasikan sentimen komentar pengguna terhadap pembangunan Ibu Kota Nusantara (IKN) pada media sosial TikTok menggunakan algoritma *Naive Bayes* dan *XGBoost*. Tahapan penelitian diawali dengan pengumpulan data berupa komentar TikTok yang relevan dengan isu IKN, kemudian dilanjutkan dengan preprocessing data untuk membersihkan dan menyiapkan teks. Data yang telah diproses selanjutnya diberikan label sentimen ke dalam tiga kategori, yaitu positif, negatif, dan netral, sebelum diekstraksi menggunakan metode *Term Frequency–Inverse Document Frequency (TF-IDF)*. Dataset kemudian dibagi menjadi data latih dan data uji, lalu diproses menggunakan algoritma *Naive Bayes* dan *XGBoost* untuk melakukan klasifikasi sentimen. Kinerja kedua model dievaluasi untuk mengetahui tingkat akurasi dan performa klasifikasi, kemudian hasil analisis divisualisasikan menggunakan *Word Cloud*. Alur penelitian secara keseluruhan ditunjukkan pada Gambar 1.



Gambar 1. Metodologi Penelitian

Gambar 1 menunjukkan alur penelitian yang dimulai dari pengumpulan data, preprocessing, pelabelan sentimen, ekstraksi fitur, klasifikasi, hingga evaluasi model dan visualisasi hasil.

## 2.2 Pengumpulan Data

Pengumpulan data pada penelitian ini dilakukan dengan menghimpun komentar pengguna dari platform media sosial TikTok yang membahas isu pembangunan Ibu Kota Nusantara (IKN). Data komentar dikumpulkan dari beberapa konten yang relevan dengan topik penelitian dan disimpan dalam format CSV agar dapat diolah menggunakan bahasa pemrograman *Python* melalui *Google Colab*. Jumlah data yang digunakan sebanyak 1.132 komentar, yang seluruhnya berupa teks tanpa menyertakan identitas pengguna. Pemilihan komentar TikTok sebagai sumber data didasarkan pada tingginya tingkat interaksi pengguna, sehingga data yang diperoleh dapat mencerminkan opini publik terhadap pembangunan IKN dan selanjutnya digunakan dalam proses analisis sentimen menggunakan algoritma *Naive Bayes* dan *XGBoost*. Meskipun jumlah dataset yang digunakan relatif terbatas, data yang diperoleh tetap mencerminkan variasi opini pengguna TikTok terhadap isu IKN. Namun demikian, ukuran dataset yang kecil berpotensi mempengaruhi representativitas data dan menjadi salah satu keterbatasan dalam penelitian ini.

## 2.3 Preprocessing Data

Tahap awal pengolahan data dilakukan dengan membersihkan serta menyederhanakan teks komentar agar struktur katanya lebih konsisten sebelum diproses ke tahap pembobotan dan klasifikasi. Tahap ini penting untuk mengurangi noise sehingga model dapat mengenali pola kata dengan lebih baik. Pada penelitian ini, tahapan *preprocessing* yang diterapkan adalah empat tahap sebagai berikut:

### a. Case Folding

Seluruh karakter dalam komentar diseragamkan ke bentuk huruf kecil guna memastikan tidak terjadi perbedaan pengenalan kata akibat variasi penggunaan kapitalisasi[21].

### b. Text Cleaning

Proses pembersihan teks dilakukan dengan menghapus karakter khusus, angka, tanda baca, simbol, serta elemen lain yang tidak relevan menggunakan teknik regular expression (*regex*)[22]. Tahap ini bertujuan agar data teks yang dianalisis hanya terdiri dari kata-kata yang memiliki makna.

### c. Tokenisasi dan Normalisasi

Teks yang telah dibersihkan dipecah menjadi unit kata (token) berdasarkan spasi. Selanjutnya dilakukan *normalisasi* singkatan dan bahasa gaul media sosial ke bentuk baku menggunakan kamus *normalisasi*. Contoh: "blm" menjadi "belum", "tdk" menjadi "tidak", "jokowidodo" menjadi "jokowi". *Normalisasi* penting untuk mengatasi penggunaan bahasa gaul, singkatan, dan variasi penulisan sebelum dianalisis menggunakan algoritma *machine learning*.

### d. Stemming

Setiap token dikonversi ke bentuk kata dasar (root word) menggunakan algoritma Nazief-Adriani. Contoh: "membangun", "pembangunan", dan "bangunan" menjadi "bangun"; "pengusaha" menjadi "usaha". Proses *stemming* dilakukan agar variasi bentuk kata yang memiliki makna serupa tidak diperlakukan sebagai fitur yang berbeda, sehingga representasi teks menjadi lebih ringkas dan efisien, sehingga variasi morfologi kata yang memiliki makna sama tidak dihitung sebagai fitur berbeda.

## 2.4 Pelabelan Sentimen

Pelabelan sentimen merupakan tahap pemberian kelas pada setiap data komentar berdasarkan kecenderungan opini yang terkandung di dalamnya. Pada penelitian ini, proses pelabelan dilakukan secara manual dengan mengelompokkan komentar TikTok ke dalam tiga kategori sentimen, yaitu positif, negatif, dan netral. Proses pelabelan dilakukan oleh peneliti dengan mengacu pada pedoman klasifikasi sentimen yang telah ditentukan berdasarkan konteks pembahasan. Untuk menjaga konsistensi, peneliti melakukan pengecekan ulang terhadap hasil pelabelan secara menyeluruh. Penentuan label dilakukan secara manual agar klasifikasi sentimen tetap mempertimbangkan konteks pembahasan serta makna yang terkandung dalam setiap komentar[23]. Komentar dengan sentimen positif mencerminkan dukungan, apresiasi, atau pandangan optimis terhadap pembangunan IKN. Sentimen negatif menunjukkan ketidaksetujuan, kritik, atau pandangan pesimistis terhadap kebijakan tersebut, sedangkan sentimen netral berisi komentar yang bersifat informatif atau tidak menunjukkan kecenderungan opini yang jelas[24]. Meskipun demikian, proses pelabelan manual berpotensi mengandung subjektivitas karena tidak melibatkan lebih dari satu anotator maupun pengujian inter-annotator agreement seperti Cohen's Kappa, sehingga hal ini menjadi salah satu keterbatasan dalam penelitian ini. Hasil pelabelan sentimen ini selanjutnya digunakan sebagai data target (label) dalam proses klasifikasi menggunakan algoritma *Naive Bayes* dan *XGBoost*.

## 2.5 Ekstraksi Fitur menggunakan TF-IDF

Ekstraksi fitur merupakan tahap mengubah data teks menjadi bentuk numerik agar dapat diproses oleh algoritma klasifikasi. Pada penelitian ini digunakan metode *Term Frequency-Inverse Document Frequency (TF-IDF)* untuk merepresentasikan tingkat kepentingan suatu kata dalam dokumen terhadap keseluruhan *dataset*[22].

*Term Frequency* digunakan untuk menghitung intensitas kemunculan suatu kata dalam suatu dokumen tertentu dibandingkan dengan keseluruhan kata yang ada pada dokumen tersebut[25]. Rumus TF dirumuskan sebagai berikut:

$$TF(t, d) = \frac{f_{t,d}}{\sum_{k=1}^n f_{k,d}} \quad (1)$$

di mana  $f(t,d)$  merupakan jumlah kemunculan token  $t$  dalam dokumen  $d$ , sedangkan  $\sum f(k,d)$  menunjukkan total seluruh token dalam dokumen tersebut.

*Inverse Document Frequency* (IDF) berfungsi menekan kontribusi kata yang terlalu umum diseluruh dokumen, sehingga istilah yang lebih khas pada dokumen tertentu memperoleh bobot yang lebih besar[25]. Rumus IDF dirumuskan sebagai berikut:

$$IDF(t) = \log_{10} \left( \frac{N}{df_t} \right) \quad (2)$$

di mana  $N$  merupakan jumlah dokumen pada data latih, yaitu sebanyak 905 dokumen, sedangkan  $df(t)$  adalah jumlah dokumen yang mengandung token  $t$ .

Bobot akhir *TF-IDF* diperoleh dari perkalian TF dan IDF:

$$TF-IDF(t, d) = TF(t, d) \times IDF(t) \quad (3)$$

Perlu dicatat bahwa seluruh perhitungan *TF-IDF* dalam penelitian ini dilakukan pada token yang telah melalui proses *Stemming*, sehingga frekuensi kemunculan dihitung per kata dasar. Implementasi pada penelitian ini menggunakan *TfidfVectorizer* dari *library scikit-learn* dengan parameter *max\_features* = 5000 dan *normalisasi* L2 (*Euclidean norm*) secara default.[22] Parameter *max\_features* membatasi jumlah fitur menjadi 5000 token unik dengan bobot *TF-IDF* tertinggi. *Normalisasi* L2. *Normalisasi* L2 digunakan untuk menstandarisasi panjang vektor dokumen agar memiliki skala yang seragam, sehingga dapat meningkatkan performa model dalam proses klasifikasi. diterapkan untuk menstandarisasi panjang vektor dokumen, dirumuskan sebagai:

$$v_{normalized} = \frac{v}{\sqrt{\sum_{i=1}^n v_i^2}} \quad (4)$$

di mana  $v$  adalah vektor *TF-IDF* hasil perkalian TF dan IDF. Karena *normalisasi* ini, nilai TF pada Tabel 3 merupakan hasil rekonstruksi dari bobot *TF-IDF* ter-*normalisasi* dibagi dengan nilai IDF. IDF dihitung berdasarkan frekuensi dokumen dari data latih ( $N = 905$ ). Hasil ekstraksi berupa matriks sparse yang digunakan sebagai input untuk klasifikasi *Naive Bayes* dan *XGBoost*.

## 2.6 Pembagian Data (Split Data)

Tahap pembagian data dilakukan untuk memisahkan *dataset* menjadi data latih dan data uji sebelum proses klasifikasi dilakukan. Pada penelitian ini, data hasil ekstraksi fitur *TF-IDF* dibagi menggunakan perbandingan 80% sebagai data latih dan 20% sebagai data uji[26]. Proses ini dilakukan menggunakan fungsi *train\_test\_split* dari *library scikit-learn* dengan parameter *test\_size* = 0.2 dan *random\_state* = 42 untuk menjaga konsistensi pembagian data[27]. Dataset dibagi menjadi dua bagian, dimana sebagian digunakan untuk melatih model agar mengenali pola sentimen, sementara sisanya dimanfaatkan untuk menguji sejauh mana model mampu melakukan prediksi terhadap data yang belum pernah dianalisis sebelumnya. Dengan pembagian ini, evaluasi model dapat dilakukan secara lebih objektif.

## 2.7 Klasifikasi Sentimen menggunakan *Naive Bayes*

Algoritma *Multinomial Naive Bayes* digunakan untuk mengklasifikasikan teks komentar TikTok ke dalam tiga kategori sentimen. Algoritma ini bekerja berdasarkan Teorema Bayes dengan asumsi bahwa setiap fitur kata bersifat independen satu sama lain[28]. Teorema Bayes dirumuskan sebagai berikut:

$$P(C_k | X) = \frac{P(X|C_k)P(C_k)}{P(X)} \quad (5)$$

di mana  $P(C_k | X)$  merupakan probabilitas suatu kelas  $C_k$  diberikan fitur  $X$  (posterior),  $P(X | C_k)$  merupakan probabilitas fitur  $X$  pada kelas  $C_k$  (likelihood),  $P(C_k)$  adalah probabilitas awal dari kelas  $C_k$  (prior), dan  $P(X)$  adalah probabilitas keseluruhan dari fitur  $X$ .

Karena data teks memiliki sifat frekuensi kemunculan kata, digunakan varian *Multinomial Naive Bayes*[29]. Probabilitas *likelihood* dihitung dengan rumus:

$$P(x_i | C_k) = \frac{\text{count}(x_i, C_k) + \alpha}{\text{count}(C_k) + \alpha|V|} \quad (6)$$

di mana  $\text{count}(x_i, C_k)$  menunjukkan jumlah kemunculan fitur  $x_i$  pada kelas  $C_k$ ,  $\text{count}(C_k)$  merupakan total seluruh kata pada kelas  $C_k$ ,  $\alpha$  adalah parameter smoothing dengan nilai 1, dan  $|V|$  adalah jumlah seluruh fitur unik (vocabulary).

Dalam praktik pengembangan model, peneliti memanfaatkan modul *MultinomialNB* yang tersedia pada pustaka *scikit-learn*, dengan konfigurasi hyperparameter mengikuti nilai bawaan (*default*) dari *framework* tersebut.

Proses training dilakukan menggunakan dataset latihan hasil transformasi *TF-IDF*, yang selanjutnya dievaluasi untuk prediksi kategori sentimen pada data testing.

## 2.8 Klasifikasi Sentimen menggunakan *XGBoost*

*XGBoost* merujuk pada *eXtreme Gradient Boosting* mengimplementasikan strategi ensemble learning dengan cara mengagregasi multiple *decision trees* secara sekuensial dan aditif. Kemampuan algoritma ini dalam menangani *feature space* dengan dimensionalitas tinggi seperti representasi vektorial hasil transformasi *TF-IDF* telah terdokumentasi secara ekstensif dalam literatur *machine learning*. [30]. Fungsi objektif *XGBoost* dirumuskan sebagai berikut:

$$\mathcal{L}(f) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (7)$$

di mana  $L(f)$  merupakan fungsi loss total,  $l(y_i, \hat{y}_i)$  adalah fungsi loss pada data ke- $i$  (menggunakan log loss untuk klasifikasi),  $y_i$  adalah nilai aktual,  $\hat{y}_i$  adalah nilai prediksi model, dan  $\Omega(f_k)$  merupakan fungsi regularisasi pada pohon ke- $k$ . Prediksi akhir diperoleh dari:

$$\hat{y}_i = \arg \max_k F_k(x_i) \quad (8)$$

di mana  $\arg\max$  digunakan untuk menentukan kelas dengan nilai prediksi tertinggi, dan  $F_k(x_i)$  merupakan skor prediksi dari model untuk setiap kelas sentimen. *Parameter* yang digunakan dalam implementasi algoritma *XGBoost* pada penelitian ini ditunjukkan pada Tabel 1.

**Tabel 1.** *Parameter XGBoost*

Parameter	Nilai	Keterangan
Objective	multi:softmax	Klasifikasi multi-kelas
num_class	3	Jumlah kelas sentimen
eval_metric	mlogloss	Evaluasi multi-class logloss

Pada Tabel 1 ditunjukkan parameter yang digunakan dalam implementasi algoritma *XGBoost*, yang berperan dalam mengatur proses klasifikasi sentimen ke dalam tiga kategori, yaitu positif, negatif, dan netral. Model dilatih menggunakan data latihan hasil ekstraksi *TF-IDF*, kemudian diuji pada data uji untuk memprediksi sentimen komentar TikTok.

## 2.9 Evaluasi Model

Evaluasi model dilakukan untuk mengukur kinerja algoritma *Naive Bayes* dan *XGBoost* dalam mengklasifikasikan sentimen opini publik terhadap pembangunan Ibu Kota Nusantara (IKN). Evaluasi dilakukan dengan membandingkan hasil prediksi model terhadap data uji menggunakan beberapa metrik pengukuran kinerja klasifikasi. Metrik evaluasi yang digunakan dalam penelitian ini meliputi akurasi, presisi, *recall*, dan *F1-score* [31]. Hasil evaluasi dari kedua algoritma selanjutnya dianalisis untuk mengetahui tingkat efektivitas masing-masing metode serta menentukan algoritma yang memiliki performa terbaik dalam analisis sentimen terhadap data komentar TikTok.

## 2.10 Word Cloud

Visualisasi *Word Cloud* digunakan untuk menampilkan kata-kata yang paling sering muncul dalam data komentar berdasarkan kategori sentimen [32]. Pada penelitian ini, *Word Cloud* dibuat dari data komentar TikTok yang telah melalui tahap *preprocessing* dan pelabelan sentimen. Ukuran kata pada *Word Cloud* menunjukkan tingkat frekuensi kemunculan kata, sehingga kata yang lebih sering muncul akan ditampilkan dengan ukuran yang lebih besar [33]. Visualisasi *Word Cloud* dikembangkan guna menampilkan representasi grafis dari terminologi-terminologi frekuensi tinggi yang mencerminkan opini masyarakat mengenai agenda pembangunan Ibu Kota Nusantara. Visualisasi ini digunakan sebagai pelengkap hasil analisis klasifikasi sentimen dan membantu dalam memahami kecenderungan topik pembahasan pada masing-masing kategori sentimen.

# 3. HASIL DAN PEMBAHASAN

## 3.1 Hasil Pengumpulan Data

Data yang digunakan dalam penelitian ini berupa komentar pengguna media sosial TikTok yang membahas isu pembangunan Ibu Kota Nusantara (IKN). Proses pengumpulan data menghasilkan sebanyak 1.132 komentar yang kemudian disimpan dalam format *Comma Separated Values (CSV)* untuk memudahkan proses pengolahan menggunakan bahasa pemrograman *Python* pada platform *Google Colab*. Data yang dikumpulkan hanya berupa teks komentar tanpa menyertakan identitas pengguna sehingga tidak melibatkan informasi pribadi.

Berdasarkan hasil pelabelan manual, data komentar diklasifikasikan ke dalam tiga kategori sentimen, yaitu positif, negatif, dan netral. Distribusi data menunjukkan bahwa sentimen positif memiliki jumlah yang lebih dominan dibandingkan dua kategori lainnya. Kondisi ini mengindikasikan bahwa sebagian besar pengguna TikTok dalam

*dataset* penelitian menunjukkan kecenderungan opini yang mendukung atau bersikap optimis terhadap pembangunan IKN.

**Tabel 2.** Distribusi Data Sentimen Komentar TikTok

Sentimen	Jumlah Data
Positif	619
Negatif	288
Netral	225
Total	1.132

Distribusi pada Tabel 2 menunjukkan adanya ketidakseimbangan data (*class imbalance*), di mana kelas positif mendominasi sebesar 54,7%, dibandingkan kelas negatif sebesar 25,4% dan netral sebesar 19,9%. Kondisi ini berpotensi menyebabkan model klasifikasi cenderung bias terhadap kelas mayoritas.

Oleh karena itu, dalam penelitian ini metrik evaluasi seperti *precision*, *recall*, dan *F1-score* menjadi lebih penting untuk dianalisis dibandingkan hanya menggunakan akurasi, karena mampu memberikan gambaran performa model secara lebih menyeluruh pada masing-masing kelas, terutama pada kelas minoritas.

### 3.2 Hasil *Preprocessing*

Tahap *preprocessing* dilakukan untuk meningkatkan kualitas data teks komentar TikTok mengenai Ibu Kota Nusantara (IKN) sebelum dilakukan ekstraksi fitur. Proses *preprocessing* dalam penelitian ini terdiri dari empat tahap: *Case Folding*, *Text Cleaning*, *Tokenisasi* dan *Normalisasi*, serta *Stemming*.

Berdasarkan hasil *preprocessing*, data komentar TikTok mengalami transformasi signifikan. Proses *Case Folding* mengubah seluruh teks menjadi huruf kecil. *Text Cleaning* berhasil menghilangkan URL, mention, hashtag, angka, dan karakter non-alfabet menggunakan *regex*. *Tokenisasi* memecah teks menjadi unit kata, dilanjutkan *normalisasi* yang mengubah singkatan seperti "knp" menjadi "kenapa", "klo" menjadi "kalau", "blm" menjadi "belum", dan "yg" menjadi "yang". Tahap akhir *Stemming* menggunakan algoritma Nazief-Adriani mengkonversi kata ke bentuk dasarnya seperti "ternaknya" menjadi "ternak", "menanam" menjadi "tanam", dan "modalnya" menjadi "modal". Tabel 3 menampilkan contoh hasil *preprocessing* data untuk dua sampel dokumen.

**Tabel 3.** Contoh Hasil *Preprocessing* Data

No	Data Asli	Case Folding	Text Cleaning	Tokenisasi + Normalisasi	Stemming
1	knp mulyono dan ternaknya blm pada pindah di ikn	knp mulyono dan ternaknya blm pada pindah di ikn	knp mulyono dan ternaknya blm pada pindah di ikn	kenapa mulyono dan ternaknya belum pada pindah di ikn	kenapa mulyono dan ternak belum pada pindah ikn
2	klo boleh jujur sampai detik ini blm ada investor yg mau menanam modalnya di ikn	klo boleh jujur sampai detik ini blm ada investor yg mau menanam modalnya di ikn	klo boleh jujur sampai detik ini blm ada investor yg mau menanam modalnya di ikn	kalau boleh jujur sampai detik ini belum ada investor yang mau menanam modalnya di ikn	kalau boleh jujur sampai detik ini belum ada investor yang mau tanam modal ikn

Hasil pada Tabel 3 menunjukkan bahwa proses *preprocessing* berhasil mengurangi noise pada data teks serta menstandarkan bentuk kata, khususnya pada penggunaan bahasa informal yang umum ditemukan pada platform TikTok. Data asli sebagian besar sudah dalam format huruf kecil, sehingga kolom "Data Asli" dan "Case Folding" menampilkan hasil yang identik pada beberapa dokumen. Proses *Text Cleaning* efektif menghilangkan karakter non-alfabet yang tidak relevan. *Tokenisasi* dan *normalisasi* berhasil mengubah singkatan menjadi bentuk baku, sementara *stemming* mengkonversi kata ke bentuk dasarnya untuk menghindari redundansi fitur. *Normalisasi* dan *stemming* berperan penting dalam menyederhanakan variasi kata sehingga kata-kata dengan makna serupa tidak dianggap sebagai fitur yang berbeda, yang pada akhirnya dapat meningkatkan kualitas representasi fitur pada tahap ekstraksi TF-IDF. Selain itu, karakteristik data TikTok yang cenderung tidak baku, mengandung singkatan, serta variasi bahasa gaul menjadikan tahap *preprocessing* sebagai komponen krusial dalam meningkatkan performa model klasifikasi. Data hasil *preprocessing* pada kolom "Stemming" selanjutnya digunakan sebagai input pada tahap ekstraksi fitur menggunakan metode TF-IDF.

### 3.3 Ekstraksi Fitur

Tahap ekstraksi fitur dilakukan setelah data melalui proses *preprocessing*. Metode *Term Frequency-Inverse Document Frequency (TF-IDF)* digunakan untuk mengubah data teks menjadi representasi numerik berupa vektor fitur. Implementasi menggunakan *TfidfVectorizer* dari *library scikit-learn* dengan parameter *max\_features=5000*. Berdasarkan hasil ekstraksi, jumlah fitur (kata unik) yang terbentuk adalah 1.926 fitur, karena *vocabulary* unik dalam *dataset* kurang dari batas maksimum 5000. Hasil ekstraksi berupa matriks sparse dengan dimensi (905, 1926) untuk

data latih dan (227, 1926) untuk data uji, di mana 905/227 merepresentasikan jumlah dokumen (komentar TikTok) dan 1926 merepresentasikan jumlah fitur (kata unik) terpilih.

Setiap nilai dalam matriks merepresentasikan bobot  $TF-IDF$  yang diperoleh dari perkalian *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF). Semakin tinggi nilai bobot, semakin penting kata tersebut dalam dokumen tersebut dibandingkan dengan dokumen lain dalam korpus. Tabel 4 menampilkan contoh hasil ekstraksi fitur  $TF-IDF$  untuk lima kata dengan bobot tertinggi dari dua sampel dokumen.

**Tabel 4(a).** Contoh Hasil Ekstraksi Fitur  $TF-IDF$  (Dokumen 1)

Kata	TF	DF	IDF	TF-IDF
mulyono	0,06988	3	2,479	0,1732
kenapa	0,06988	8	2,054	0,1435
ternak	0,06988	2	2,656	0,1856
pada	0,06988	45	1,303	0,0910
belum	0,06988	25	1,559	0,1089

Tabel 4(a) menunjukkan bahwa kata dengan nilai IDF tinggi seperti "ternak" memiliki bobot TF-IDF yang lebih besar karena kemunculannya relatif jarang di seluruh dokumen, sehingga kata tersebut menjadi lebih representatif dalam membedakan karakteristik suatu dokumen.

**Tabel 4(b).** Contoh Hasil Ekstraksi Fitur  $TF-IDF$  (Dokumen 2)

Kata	TF	DF	IDF	TF-IDF
detik	0,05263	5	2,258	0,1189
tanam	0,05263	4	2,355	0,1240
jujur	0,05263	3	2,479	0,1305
Investor	0,05263	2	2,656	0,1398
modal	0,05263	8	2,054	0,1081

Pada Tabel 4(b) terlihat bahwa kata "investor" memiliki bobot TF-IDF tertinggi karena frekuensi kemunculannya rendah di seluruh dokumen, namun cukup dominan dalam dokumen tertentu. Hal ini menunjukkan bahwa kata tersebut memiliki kontribusi penting dalam proses klasifikasi. Perhitungan TF-IDF pada penelitian ini mengacu pada rumus [1], [2], dan [3] yang telah dijelaskan pada subbab metodologi. Implementasi dilakukan menggunakan `TfidfVectorizer` dari library `scikit-learn` dengan parameter `max_features = 5000` dan normalisasi L2. Normalisasi ini bertujuan untuk menstandarisasi panjang vektor dokumen sehingga dapat dibandingkan secara adil antar dokumen dengan panjang yang berbeda. Nilai TF yang ditampilkan pada tabel merupakan hasil dari bobot yang telah dinormalisasi, sehingga tidak sepenuhnya merepresentasikan nilai frekuensi asli sesuai rumus dasar TF. Oleh karena itu, interpretasi nilai TF dalam tabel perlu dipahami sebagai bagian dari representasi vektor hasil normalisasi, bukan nilai frekuensi mentah.

### 3.4 Hasil Klasifikasi Menggunakan Algoritma *Naive Bayes*

Proses klasifikasi menggunakan algoritma Multinomial Naive Bayes dilakukan setelah dataset melalui tahapan persiapan yang komprehensif. Dataset sebanyak 1.132 komentar TikTok terkait Ibu Kota Nusantara dibagi menjadi data latih dan data uji dengan perbandingan 80:20 menggunakan fungsi `train_test_split` dari library `scikit-learn` dengan parameter `random_state=42` untuk memastikan reproducibilitas hasil. Dengan pembagian ini, diperoleh 905 komentar sebagai data latih dan 227 komentar sebagai data uji yang digunakan untuk evaluasi model.

Hasil pengujian model Multinomial Naive Bayes menunjukkan nilai akurasi sebesar 61,23% terhadap data uji. Pada tingkat akurasi keseluruhan, angka ini mengindikasikan bahwa model berhasil mengklasifikasikan lebih dari tiga perlima komentar dengan benar. Namun demikian, nilai akurasi sebesar 61,23% masih tergolong rendah untuk tugas klasifikasi sentimen, sehingga menunjukkan bahwa model belum mampu menangkap pola sentimen secara optimal. Oleh karena itu, evaluasi tidak dapat hanya bergantung pada akurasi, melainkan perlu memperhatikan metrik lain seperti precision, recall, dan F1-score, terutama pada kondisi data yang tidak seimbang.

Model menunjukkan kemampuan klasifikasi yang sangat baik pada kelas positif dengan nilai precision 0,61, recall 0,98, dan F1-score 0,76 dari 124 data uji. Nilai recall 0,98 yang mendekati sempurna menunjukkan bahwa hampir seluruh komentar positif berhasil terdeteksi oleh model dengan benar. Hanya 2 dari 124 komentar positif yang terlewat. Namun, precision 0,61 mengindikasikan bahwa sebagian prediksi positif ternyata adalah false positive, yang sebagian besar berasal dari kelas netral dan negatif yang salah diklasifikasikan. Meskipun nilai recall tinggi, kondisi ini juga mengindikasikan adanya kecenderungan model untuk mengklasifikasikan sebagian besar data ke dalam kelas positif, yang berpotensi menyebabkan bias terhadap kelas mayoritas.

Pada kelas negatif yang terdiri dari 58 data uji, model mencapai precision 0,63, recall 0,21, dan F1-score 0,31. Precision yang relatif tinggi menunjukkan ketika model memprediksi negatif, kebenarannya cukup tinggi. Akan tetapi, recall 0,21 mengindikasikan bahwa 79% komentar negatif tidak berhasil terdeteksi dan salah klasifikasi ke kelas lain, terutama ke kelas positif. Kondisi ini menunjukkan model cenderung melewatkan komentar-komentar kritis terhadap pembangunan IKN. Nilai recall yang rendah ini menunjukkan bahwa model kurang efektif dalam mendeteksi opini

yang bersifat kritis, sehingga berpotensi mengabaikan informasi penting yang berkaitan dengan sentimen negatif terhadap kebijakan publik.

Fenomena yang paling signifikan terjadi pada kelas netral yang terdiri dari 45 data uji. Model mencatat precision 0,56, namun dengan recall yang sangat rendah yaitu 0,11, menghasilkan F1-score 0,19. Precision 0,56 menunjukkan ketika model memprediksi netral, kebenarannya moderat. Namun, recall 0,11 menunjukkan hanya 5 dari 45 komentar netral yang berhasil teridentifikasi dengan benar, sementara 40 komentar netral lainnya (88,9%) gagal terdeteksi. Rendahnya kemampuan deteksi pada kelas netral menunjukkan bahwa model belum mampu membedakan secara jelas antara opini netral dan opini positif, yang umumnya memiliki karakteristik kata yang mirip dalam representasi TF-IDF.

Analisis Confusion Matrix yang divisualisasikan pada Gambar 2 memberikan pemahaman lebih dalam mengenai pola kesalahan klasifikasi. Dari 45 komentar netral, sebanyak 34 komentar (75,6%) salah diklasifikasikan sebagai positif dan 6 komentar (13,3%) salah sebagai negatif. Hanya 5 komentar netral yang terklasifikasi dengan benar. Dari 58 komentar negatif, 43 komentar (74,1%) salah diklasifikasikan sebagai positif, 3 komentar (5,2%) sebagai netral, dan hanya 12 komentar (20,7%) yang teridentifikasi dengan tepat. Kondisi ini menunjukkan bahwa model hampir seluruhnya gagal mengenali karakteristik khas kelas netral dan negatif. Temuan ini memperkuat indikasi bahwa model memiliki bias yang tinggi terhadap kelas positif dan kurang mampu menangkap pola dari kelas minoritas.

Kegagalan deteksi kelas minoritas disebabkan oleh beberapa faktor. Pertama, ketidakseimbangan distribusi data latih dimana kelas positif mendominasi dengan 619 komentar (54,7%), sementara kelas netral hanya 225 komentar (19,9%) dan negatif 288 komentar (25,4%). Model cenderung mempelajari pola dari kelas mayoritas dan mengabaikan pola kelas minoritas. Kedua, komentar netral umumnya mengandung kata-kata umum tanpa muatan emosional kuat yang membedakannya dari kelas lain, sehingga fitur TF-IDF kurang mampu menangkap karakteristik diskriminatifnya. Ketiga, asumsi independensi fitur pada Naive Bayes tidak selalu terpenuhi dalam data teks yang memiliki keterkaitan antar kata. Selain itu, ukuran dataset yang relatif kecil juga berkontribusi terhadap keterbatasan model dalam mempelajari pola yang representatif dari masing-masing kelas sentimen.

Dengan demikian, meskipun model menghasilkan akurasi di atas 60%, performa secara keseluruhan masih belum memadai untuk digunakan dalam pengambilan keputusan yang membutuhkan tingkat keandalan tinggi, terutama dalam konteks analisis kebijakan publik. Perbandingan mendalam antar kelas dan metrik evaluasi lengkap disajikan pada Tabel 5.

**Tabel 5.** Hasil Klasifikasi *Naive Bayes* per Kelas

Sentimen	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>	<i>Support</i>
Negatif	0,62	0,21	0,31	58
Netral	0,56	0,11	0,19	45
Positif	0,61	0,98	0,76	124
Akurasi			0,61 (61,23%)	227
Macro Avg	0,60	0,43	0,42	227
Weighted Avg	0,61	0,61	0,53	227

Hasil pada Tabel 5 menunjukkan bahwa model *Naive Bayes* mengalami bias signifikan terhadap kelas positif dengan recall 0,98 yang mendekati sempurna, namun gagal mengenali kelas netral (recall 0,11) dan negatif (recall 0,21). *Confusion Matrix* memperkuat temuan ini, di mana 75,6% komentar netral dan 74,1% komentar negatif salah diklasifikasikan sebagai positif. Kondisi ini disebabkan oleh ketidakseimbangan distribusi data latih dan karakteristik komentar netral yang sulit dibedakan dari positif oleh fitur TF-IDF. Hal ini menunjukkan bahwa model lebih cenderung mengikuti distribusi data latih daripada benar-benar memahami pola sentimen yang beragam.

### 3.5 Hasil Klasifikasi Menggunakan Algoritma *XGBoost*

Selain Multinomial *Naive Bayes*, penelitian ini menerapkan algoritma *XGBoost* untuk klasifikasi sentimen multi-kelas dengan tiga kategori yang sama, yaitu positif, negatif, dan netral. Proses pelatihan dan pengujian dilakukan menggunakan pembagian data yang identik dengan *Naive Bayes*, yaitu 80% data latih dan 20% data uji, untuk memastikan perbandingan yang fair antar kedua algoritma.

Berdasarkan hasil evaluasi terhadap 227 data uji, model *XGBoost* memperoleh nilai akurasi sebesar 62,11%, yang menunjukkan peningkatan 0,88% dibandingkan dengan *Naive Bayes*. Namun demikian, selisih peningkatan akurasi yang relatif kecil ini menunjukkan bahwa keunggulan *XGBoost* terhadap *Naive Bayes* tidak signifikan secara praktis, sehingga perlu dianalisis lebih lanjut melalui metrik evaluasi lainnya.

Analisis per kelas menunjukkan adanya perbaikan pada kelas minoritas. Pada kelas negatif yang terdiri dari 58 data uji, *XGBoost* mencatat precision 0,53, recall 0,40, dan F1-score 0,47. Dibandingkan dengan *Naive Bayes* yang mencapai recall 0,21, nilai recall meningkat menjadi 0,40. Peningkatan ini merupakan peningkatan relatif (dari 0,21 menjadi 0,40), namun secara absolut nilai recall tersebut masih tergolong rendah, yang berarti sebagian besar komentar negatif masih belum terdeteksi dengan baik oleh model.

Pada kelas netral yang terdiri dari 45 data uji, *XGBoost* mencapai precision 0,40, recall 0,26, dan F1-score 0,32. Dibandingkan dengan *Naive Bayes* yang memiliki recall 0,11, nilai ini meningkat menjadi 0,26.

Meskipun terjadi peningkatan relatif yang cukup besar, secara absolut kemampuan model dalam mendeteksi kelas netral masih terbatas, karena hanya sekitar seperempat dari total data netral yang berhasil diklasifikasikan dengan benar.

Pada kelas positif yang terdiri dari 124 data uji, XGBoost mencatat precision 0,71, recall 0,91, dan F1-score 0,80. Terjadi penurunan recall dari 0,98 (Naive Bayes) menjadi 0,91, yang mengindikasikan XGBoost lebih selektif dalam mengklasifikasikan komentar sebagai positif. Penurunan ini menunjukkan adanya trade-off antara peningkatan performa pada kelas minoritas dengan penurunan performa pada kelas mayoritas, yang merupakan karakteristik umum pada model klasifikasi dengan data tidak seimbang.

Analisis Confusion Matrix pada Gambar 3 menunjukkan distribusi prediksi yang lebih merata antar kelas dibandingkan dengan Naive Bayes. Dari 58 komentar negatif, 23 terklasifikasi benar, 14 salah ke netral, dan 21 salah ke positif. Dari 45 komentar netral, 12 terklasifikasi benar, 8 salah ke negatif, dan 25 salah ke positif. Meskipun distribusi kesalahan lebih seimbang, jumlah kesalahan klasifikasi pada kelas netral dan negatif masih cukup tinggi, yang menunjukkan bahwa model belum mampu mempelajari pola sentimen secara optimal.

Mekanisme gradient boosting pada XGBoost memungkinkan model memperbaiki kesalahan prediksi secara bertahap melalui pembentukan pohon keputusan secara iteratif. Selain itu, penerapan regularisasi membantu mengurangi overfitting terhadap kelas mayoritas. Namun demikian, tanpa proses tuning hyperparameter dan validasi silang, performa model yang dihasilkan masih berpotensi belum optimal dan sangat bergantung pada pembagian data yang digunakan.

Secara keseluruhan, XGBoost menunjukkan performa yang relatif lebih seimbang dibandingkan Naive Bayes, dengan nilai macro average F1-score sebesar 0,52 dibandingkan 0,42 pada Naive Bayes. Meskipun demikian, nilai tersebut masih tergolong rendah untuk aplikasi analisis sentimen yang membutuhkan tingkat akurasi dan keandalan tinggi, sehingga hasil penelitian ini lebih tepat dipandang sebagai eksplorasi awal daripada solusi yang siap diterapkan secara langsung. Rincian metrik evaluasi disajikan pada Tabel 6.

**Tabel 6.** Hasil Klasifikasi XGBoost per Kelas

Sentimen	Precision	Recall	F1-score	Support
Negatif	0,53	0,40	0,47	58
Netral	0,40	0,26	0,32	45
Positif	0,71	0,91	0,80	124
Akurasi			0,62 (62,11%)	227
Macro Avg	0,55	0,52	0,52	227
Weighted Avg	0,62	0,62	0,61	227

Hasil pada Tabel 6 menunjukkan bahwa model XGBoost menghasilkan distribusi prediksi yang lebih merata dibandingkan Naive Bayes. Nilai recall pada kelas negatif meningkat dari 0,21 menjadi 0,40, dan pada kelas netral meningkat dari 0,11 menjadi 0,26. Peningkatan tersebut merupakan peningkatan relatif, bukan absolut, sehingga perlu ditafsirkan secara hati-hati karena nilai recall akhir masih berada pada tingkat yang rendah.

### 3.6 Evaluasi dan Perbandingan Kinerja Model

Evaluasi model menggunakan Confusion Matrix mengungkapkan perbedaan antara Naive Bayes dan XGBoost dalam mengklasifikasikan sentimen komentar TikTok. Naive Bayes menunjukkan dominasi prediksi pada kelas positif, dengan 122 dari 124 komentar positif terdeteksi benar, namun hanya 5 dari 45 komentar netral dan 12 dari 58 komentar negatif yang teridentifikasi dengan tepat. Sebaliknya, XGBoost menghasilkan distribusi prediksi yang lebih merata dengan 113 komentar positif, 12 komentar netral, dan 23 komentar negatif yang terklasifikasi dengan benar.

Secara keseluruhan, XGBoost mencapai akurasi sebesar 62,11%, meningkat sebesar 0,88% dibandingkan dengan Naive Bayes yang memperoleh akurasi 61,23%. Namun demikian, peningkatan akurasi yang relatif kecil ini menunjukkan bahwa perbedaan performa kedua model tidak signifikan secara praktis, sehingga tidak dapat dijadikan satu-satunya dasar dalam menentukan model terbaik.

Peningkatan performa lebih terlihat pada kelas minoritas. Nilai recall pada kelas netral meningkat dari 0,11 menjadi 0,26, sedangkan pada kelas negatif meningkat dari 0,21 menjadi 0,40. Peningkatan ini merupakan peningkatan relatif (bukan absolut), sehingga meskipun terlihat besar secara persentase, nilai recall akhir masih tergolong rendah dan menunjukkan bahwa sebagian besar data pada kelas minoritas masih belum berhasil diklasifikasikan dengan baik.

Di sisi lain, recall pada kelas positif mengalami penurunan dari 0,98 menjadi 0,91. Hal ini menunjukkan adanya trade-off antara peningkatan performa pada kelas minoritas dengan penurunan performa pada kelas mayoritas. Kondisi ini mengindikasikan bahwa XGBoost mencoba mengurangi bias terhadap kelas positif, namun belum sepenuhnya mampu meningkatkan akurasi klasifikasi pada kelas lain secara optimal.

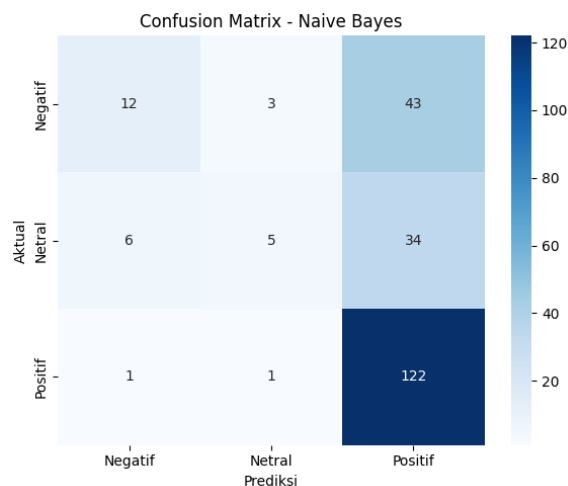
**Tabel 7.** Perbandingan Kinerja Naive Bayes dan XGBoost

Metrik	Naive Bayes	XGBoost	Selisih
Akurasi	61,23%	62,11%	+0,88%
Macro Avg F1-score	0,42	0,52	+0,10

Metrik	Naive Bayes	XGBoost	Selisih
Recall Negatif	0,21	0,40	↑
Recall Netral	0,11	0,26	↑
Recall Positif	0,98	0,91	↓

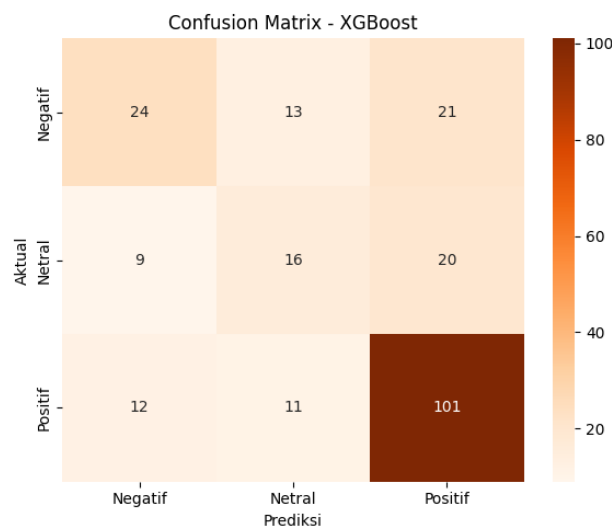
Hasil pada Tabel 7 menunjukkan bahwa XGBoost memiliki nilai macro average F1-score yang lebih tinggi dibandingkan Naive Bayes, yaitu 0,52 dibandingkan 0,42, yang mengindikasikan performa yang lebih seimbang antar kelas. Namun demikian, nilai tersebut masih tergolong rendah. Hal ini menunjukkan bahwa kedua model masih memiliki keterbatasan dalam mengklasifikasikan data sentimen secara akurat, khususnya pada kelas netral yang memiliki tingkat kesalahan yang tinggi.

Dengan demikian, meskipun XGBoost menunjukkan performa yang relatif lebih baik dibandingkan Naive Bayes, hasil yang diperoleh belum cukup kuat untuk mendukung penerapan langsung dalam konteks pengambilan keputusan kebijakan publik tanpa adanya perbaikan lebih lanjut pada metode maupun data yang digunakan.



**Gambar 2.** *Confusion Matrix Naive Bayes*

Gambar 2 menunjukkan bahwa model Naive Bayes memiliki kecenderungan yang sangat kuat dalam mengklasifikasikan data ke dalam kelas positif, sehingga banyak data dari kelas lain yang salah diklasifikasikan.



**Gambar 3.** *Confusion Matrix XGBoost*

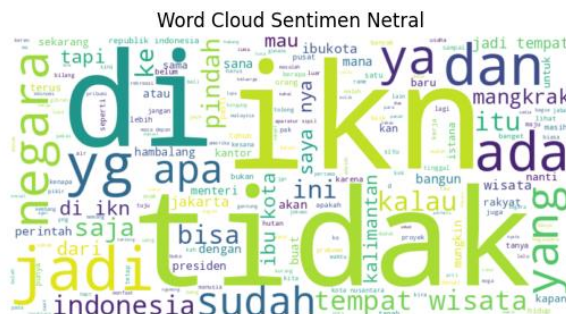
Gambar 3 menunjukkan bahwa XGBoost memiliki distribusi prediksi yang lebih seimbang, meskipun kesalahan klasifikasi pada kelas netral dan negatif masih cukup tinggi.

### 3.7 Visualisasi Word Cloud

Visualisasi *Word Cloud* digunakan untuk menggambarkan kata-kata yang paling sering muncul dalam komentar TikTok terkait pembangunan IKN. Ukuran kata dalam visualisasi menunjukkan frekuensi kemunculan kata tersebut dalam *dataset*.



memberikan insight bahwa sentimen negatif tidak hanya bersifat umum, tetapi juga mengarah pada isu kepercayaan dan manfaat pembangunan, yang penting untuk menjadi perhatian dalam pengambilan kebijakan.



**Gambar 7.** Word Cloud Sentimen Netral

Pada Gambar 7 ditampilkan visualisasi Word Cloud untuk sentimen netral, di mana ukuran kata merepresentasikan frekuensi kemunculan dalam dataset, dengan kata-kata dominan seperti “IKN”, “ada”, “jadi”, “yang”, dan “sudah”. Dominasi kata-kata umum tersebut menunjukkan bahwa komentar netral cenderung bersifat informatif dan deskriptif tanpa kecenderungan opini yang kuat. Karakteristik ini juga menjadi salah satu faktor yang menyebabkan model kesulitan dalam mengidentifikasi sentimen netral secara akurat.

### 3.8 Pembahasan

Penelitian ini menunjukkan bahwa kedua algoritma yang digunakan, yaitu *Naive Bayes* dan *XGBoost*, masih memiliki keterbatasan dalam mengklasifikasikan sentimen secara optimal, terutama pada kelas minoritas. Nilai akurasi yang diperoleh (61,23% dan 62,1%) menunjukkan bahwa performa model masih tergolong moderat dan belum cukup kuat untuk digunakan sebagai dasar pengambilan keputusan yang membutuhkan tingkat akurasi tinggi.

Hasil ini sejalan dengan penelitian terdahulu yang menyatakan bahwa metode berbasis *Naive Bayes* cenderung mengalami penurunan performa pada data teks yang tidak seimbang karena asumsi independensi fitur yang sederhana. Sementara itu, *XGBoost* mampu memberikan peningkatan performa karena memanfaatkan pendekatan ensemble learning yang dapat menangkap hubungan kompleks antar fitur.

Meskipun *XGBoost* menunjukkan peningkatan performa dibandingkan *Naive Bayes*, peningkatan tersebut masih relatif kecil secara absolut, sehingga belum dapat dikatakan signifikan dalam konteks performa model klasifikasi. Hal ini terlihat dari nilai *recall* pada kelas netral (0,26) dan negatif (0,40) yang masih rendah, sehingga model belum mampu mengenali sebagian besar data pada kelas tersebut.

Kondisi ini juga dipengaruhi oleh ketidakseimbangan distribusi data, di mana kelas positif mendominasi dataset, sehingga model cenderung bias terhadap kelas mayoritas. Hal serupa juga ditemukan pada penelitian sebelumnya yang menyatakan bahwa data imbalance dapat menyebabkan model lebih sering memprediksi kelas dominan.

Selain itu, penggunaan *Word Cloud* dalam penelitian ini hanya memberikan gambaran umum terkait kata yang sering muncul, namun belum mampu memberikan analisis yang mendalam terhadap pola linguistik yang membedakan masing-masing kelas sentimen.

## 4. KESIMPULAN

Penelitian ini menunjukkan bahwa analisis sentimen terhadap komentar TikTok terkait pembangunan Ibu Kota Nusantara dapat dilakukan menggunakan pendekatan machine learning, dengan *XGBoost* menghasilkan akurasi sebesar 62,1% yang sedikit lebih tinggi dibandingkan *Naive Bayes* sebesar 61,23%. Meskipun demikian, kedua model masih memiliki keterbatasan dalam mengklasifikasikan sentimen secara akurat dan seimbang, terutama pada kelas netral dan negatif yang memiliki nilai *recall* relatif rendah. *XGBoost* menunjukkan performa yang lebih baik dibandingkan *Naive*

## REFERENCES

- [1] S. M. Anugerah, R. Wijaya, and M. A. Bijaksana, “Sentimen Analysis Social Media for Disaster using Naïve Bayes and IndoBERT,” *INTEK J. Penelit.*, vol. 11, no. 1, pp. 51–58, 2024, doi: 10.31963/intek.v11i1.4771.
- [2] Pius Deski Manalu, Mutiara Simanjuntak, and Chairil Umri, “Implementasi Algoritma Klasifikasi untuk Analisis Sentimen Media Sosial Tiktok Tahun 2025,” *J. Tek. Inform. dan Teknol. Inf.*, vol. 5, no. 1, pp. 488–504, 2025, doi: 10.55606/jutiti.v5i2.5644.
- [3] J. Homepage, A. T. Chidayat, F. Shiddieq, and D. Nurhayati, “Analisis Sentimen Publik di platform X Pasca Skandal Bahan Bakar Minyak Oplosan Menggunakan Algoritma Naïve Bayes: Public Sentiment analysis on platform X,” *Journal.Irpi.or.Id*, vol. 5, no. October, pp. 1220–1230, 2025, [Online]. Available: <https://www.journal.irpi.or.id/index.php/malcom/article/view/2220>
- [4] A. S. Muliana, D. Lestarini, and S. P. Raflesia, “Analysis of Public Sentiment on Election Results using Naïve Bayes in



- Social Media X,” *Sistemasi*, vol. 13, no. 6, p. 2467, 2024, doi: 10.32520/stmsi.v13i6.4592.
- [5] R. N. Mauliza and Y. R. Sipayung, “Penerapan *Text mining* Dalam Menganalisis Pendapat Masyarakat Terhadap Pemilu 2024 Pada Media Sosial X Menggunakan Metode *Naive Bayes*,” *Technomedia J.*, vol. 9, no. 1, pp. 1–16, 2024, doi: 10.33050/tmj.v9i1.2212.
  - [6] A. Hermawan, I. Jowensen, J. Junaedi, and Edy, “Implementasi Text-Mining untuk Analisis Sentimen pada Twitter dengan Algoritma *Support Vector Machine*,” *JST (Jurnal Sains dan Teknol.)*, vol. 12, no. 1, pp. 129–137, 2023, doi: 10.23887/jstundiksha.v12i1.52358.
  - [7] A. Halim, F. Zidan, I. Handayani, and A. Anggara, “*Sentiment analysis* of the 2024 election using the naïve bayes method using data x,” *Jurnal Mandiri IT*, vol. 14, no. 2, pp. 225–234, 2025, doi: 10.35335/mandiri.v14i2.471
  - [8] L. H. Sarumpaet and R. R. Suryono, “Analisis Sentimen Publik Program PPPK di Media Sosial X menggunakan Naïve Bayes dan SVM,” *Edumatic J. Pendidik. Inform.*, vol. 9, no. 2, pp. 362–371, 2025, doi: 10.29408/edumatic.v9i2.30065.
  - [9] G. S. Rasyad and W. Maharani, “Logistic Regression and Naïve Bayes Comparison in Classifying Emotions on Indonesian X Social Media,” *Edumatic J. Pendidik. Inform.*, vol. 9, no. 1, pp. 31–40, 2025, doi: 10.29408/edumatic.v9i1.29120.
  - [10] R. Wati, S. Ernawati, and H. Rachmi, “Pembobotan *TF-IDF* Menggunakan Naïve Bayes pada Sentimen Masyarakat Mengenai Isu Kenaikan BIPIH,” *J. Manaj. Inform.*, vol. 13, no. 1, pp. 84–93, 2023, doi: 10.34010/jamika.v13i1.9424.
  - [11] Y. H. Agustin, N. Cici Mulyani, and W. Sindu Prasetya, “Analisis Sentimen Opini Publik Menggunakan Algoritma *Naive Bayes* dan *TF-IDF*,” *J. Algoritm.*, vol. 22, no. 2, pp. 1373–1384, 2025, doi: 10.33364/algoritma/v.22-2.2671.
  - [12] Riangga and N. Ruhyana, “Analysis of Public Sentiment Towards 2024 Presidential Candidacy Using Naïve Bayes Algorithm,” *J. Ris. Inform.*, vol. 7, no. 1, pp. 21–30, 2024, doi: 10.34288/jri.v7i1.356.
  - [13] I. G. B. A. Budaya and I. K. P. Suniantara, “Comparison of *Sentiment analysis* Algorithms with *SMOTE* Oversampling and *TF-IDF* Implementation on Google Reviews for Public Health Centers,” *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 3, pp. 1077–1086, 2024, doi: 10.57152/malcom.v4i3.1459.
  - [14] I. A. Hidayah, R. Kusumawati, Z. Abidin, and M. Imamuddin, “Analysis of Public Sentiment Towards the TikTok Application Using the Naive Bayes Algorithm and Support Vector Machine,” *Journal of Computer Networks, Architecture and High Performance Computing*, vol. 6, no. 2, pp. 881–891, 2024
  - [15] M. F. Ramadhan, F. Panjaitan, and H. Oktafiandi, “Analisis Sentimen Kutipan Media Sosial Berbahasa Indonesia Menggunakan Convolutional Neural Network,” *Jurnal Komputer, Informasi dan Teknologi*, vol. 5, no. 2, pp. 1-17, 2025, doi: 10.53697/jkomitek.v6i1.3627
  - [16] R. S. Putra, W. Agustin, M. K. Anam, L. Lusiana, and S. Yaakub, “The Application of Naive Bayes Classifier Based Feature Selection on Analysis of Online Learning Sentiment in Online Media,” *J. Transform.*, vol. 20, no. 1, pp. 44–56, 2022, doi: 10.26623/transformatika.v20i1.5144.
  - [17] F. Riza, D. F. Hendrakusuma, B. Wibowo, D. Y. Al Afghani, and A. Abdurrahman, “Perbandingan Kinerja Algoritma Klasifikasi Machine Learning dalam Analisis Sentimen Ulasan Mobile Banking WONDR BY BNI,” *INTECOMS J. Inf. Technol. Comput. Sci.*, vol. 8, no. 2, pp. 425–436, 2025, doi: 10.31539/intecomsv8i2.14826.
  - [18] K. N. Asniyah, A. Prahutama, and I. T. Utami, “Sentiment Analysis with TF-IDF Weighting Based on N-Gram for Support Vector Machine Model (Case Study: The 2024 General Election in Indonesia),” *International Journal of Research Publication and Reviews*, no. 6, pp. 4049–4055, 2025.
  - [19] M. R. Manoppo *et al.*, “Analisis Sentimen Publik Di Media Sosial Terhadap Kenaikan Ppn 12% Di Indonesia Menggunakan *IndoBERT*,” *J. Kecerdasan Buatan dan Teknol. Inf.*, vol. 4, no. 2, pp. 152–163, 2025, doi: 10.69916/jkbt.v4i2.322.
  - [20] N. Hendrastuty, S. Setiawansyah, M. G. An’ars, F. A. Rahmadiani, V. H. Saputra, and M. Rahman, “G2M weighting: a new approach based on multi-objective assessment data (case study of MOORA method in determining supplier performance evaluation),” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 38, no. 1, p. 403, 2025, doi: 10.11591/ijeecs.v38.i1.pp403-416.
  - [21] R. Ahuja, A. Chug, S. Kohli, S. Gupta, and P. Ahuja, “The impact of features extraction on the *sentiment analysis*,” *Procedia Comput. Sci.*, vol. 152, pp. 341–348, 2019, doi: 10.1016/j.procs.2019.05.008.
  - [22] K. Alemerien, A. Al-Ghareeb, and M. Z. Alksasbeh, “*Sentiment analysis* of Online Reviews: A Machine Learning–Based Approach with *TF-IDF* Vectorization,” *J. Mob. Multimed.*, vol. 20, no. 5, pp. 1089–1116, 2024, doi: 10.13052/jmm1550-4646.2055.
  - [23] L. E. M. Äyräväinen, J. Hinds, and B. I. Davidson, “Disambiguating sentiment annotation: A mixed methods investigation of annotator experience and impact of instructions on annotator agreement,” *PLoS One*, vol. 20, no. 12 December, pp. 1–47, 2025, doi: 10.1371/journal.pone.0336269.
  - [24] P. Mæhlum *et al.*, “It’s Difficult to Be Neutral: Human and LLM-Based Sentiment Annotation of Patient Comments,” in *Proceedings of the 1st Workshop on Patient-Oriented Language Processing (CL4Health)*, 2024, pp. 8–19.
  - [25] M. Kamruzzaman and G. Kim, “Efficient Sentiment Analysis: A Resource-Aware Evaluation of Feature Extraction Techniques, Ensembling, and Deep Learning Models,” *Proceedings of the 11th International Workshop on Natural Language Processing for Social Media*, November 2023, pp. 9-20, 2024, doi: 10.18653/v1/2023.socialnlp-1.2.
  - [26] M. Sivakumar, S. Parthasarathy, and T. Padmapriya, “Trade-off between training and testing ratio in *accuracy machine learning* for medical image processing,” *PeerJ Comput. Sci.*, vol. 10, pp. 1–17, 2024, doi: 10.7717/PEERJ-CS.2245.
  - [27] A. Wijaya and W. Bismi, “Penerapan Algoritma *Accuracy machine learning* Dalam Mengklasifikasi Data Masa Studi di Indonesia Berdasarkan Jenis Kelamin,” *J. Inf. Eng. Educ. Technol.*, vol. 8, no. 2, pp. 62–74, 2025, doi: 10.26740/jieet.v8n2.p62-74.
  - [28] D. Yulawati and M. Faang Ogya Widi, “Application of Multinomial Naïve Bayes for Sentiment Classification on Bukalapak Reviews,” *J. Appl. Informatics Comput.*, vol. 9, no. 6, pp. 3883–3891, 2025, doi: 10.30871/jaic.v9i6.11671.
  - [29] M. Cherradi and A. El Haddadi, “Comparative Analysis of *Machine Learning Algorithms Accuracy machine learning* for *Sentiment analysis* in Film Reviews,” *Acadlore Trans. AI Mach. Learn.*, vol. 3, no. 3, pp. 137–147, 2024, doi: 10.56578/ataiml030301.
  - [30] Z. Rais, S. Muhammad Fahmuddin, Saida, and A. T. Utomo, “Implementation of *Accuracy machine learning* Algorithm with *eXtreme Gradient Boosting (XGBoost)* Method in Hypertension Level Classification,” *J. Appl. Sci. Eng. Technol. Educ.*, vol. 7, no. 1, pp. 126–136, 2025, doi: 10.35877/454RI.asci4191.
  - [31] K. M. Sujon, R. Hassan, K. Choi, and M. A. Samad, “Accuracy, Precision, Recall, F1-Score, or MCC? Empirical Evidence



- from Advanced Statistics, ML, and XAI for Evaluating Business Predictive Models,” *Journal of Big Data*, vol. 12, no. 1, 2025, doi: 10.1186/s40537-025-01313-4.
- [32] J. Kim and M. Kim, “From Comparison to Confidence: The Dove Self-Esteem Project and the Transformation of Beauty Perceptions on Social Media,” *Behav. Sci. (Basel)*, vol. 15, no. 4, pp. 1–24, 2025, doi: 10.3390/bs15040414.
- [33] R. Atenstaedt, “*Word Cloud* analysis of the BJGP,” *Br. J. Gen. Pract.*, vol. 62, no. 596, p. 148, 2012, doi: 10.3399/bjgp12X630142.