ANALYSIS OF USER REVIEWS FOR THE MYTELKOMSEL APP USING NAÏVE BAYES AND RANDOM FOREST METHODS

M. Rudi Sanjaya[1*], Annisa Khoiriah[2], Rahmat Izwan Heroza[3], Bayu Wijaya Putra[4]

Computer Science Faculty, Universitas Sriwijaya, Indonesia[14]
STIK Khadijah Palembang, Indonesia[2]
University of Essex, United Kingdom[3]
m.rudi.sjy@ilkom.unsri.ac.id[1*], annisakhsrsjy@gmail.com[2], rh22078@essex.ac.uk[3], bayuwisata@gmail.com[4]

Received: 08 April 2026, Revised: 17 May 2026, Accepted: 23 May 2026
**Corresponding Author*

ABSTRACT
While sentiment analysis of local application reviews predominantly utilizes native Indonesian data, these datasets frequently suffer from colloquial ambiguities and informal structures that degrade classifier performance. This study addresses this gap by implementing a language-filtering mechanism to separate and analyze English and Indonesian user opinions from the MyTelkomsel application, specifically justifying the inclusion of English reviews due to their superior grammatical structure and syntactic consistency, which inherently enhances feature extraction. A systematic methodology was employed, encompassing data collection from the Google Play Store, comprehensive pre-processing (case folding, tokenization, stopword removal, and stemming), and Term Frequency-Inverse Document Frequency (TF-IDF) vectorization. Evaluated using Naïve Bayes and Random Forest algorithms on 25,000 customer feedbacks, the models were compared across accuracy, precision, recall, and F1-score. The empirical results demonstrated that Random Forest outperformed Naïve Bayes, achieving a higher accuracy of 86.85% compared to 86.36%. This superiority stems from Random Forest's robust capability to mitigate class imbalance and minimize error distribution across sentiment categories. Ultimately, this approach provides precise, actionable insights into service quality, enabling Telkomsel to effectively distinguish user satisfaction, target operational improvements, and mitigate customer churn.
Keywords : *Sentiment Analysis, Random Forest, Naïve Bayes, Text Classification, Machine Learning*

1. Introduction

The rapid expansion of digital telecommunication services in Indonesia has positioned mobile applications as the primary channel for customer engagement. Among these platforms, the MyTelkomsel application serves a critical role, facilitating essential transactions such as data package purchases, quota monitoring, and customer service interactions. As the user base expands, the volume of daily user reviews on platforms like the Google Play Store has surged exponentially. These reviews constitute a valuable repository of consumer sentiment, reflecting real-time user experiences, grievances, and commendations. However, the sheer velocity and volume of this unstructured textual data render manual analysis inefficient and impractical for timely corporate decision-making. Consequently, automated computational approach leveraging text mining and machine learning is imperative to systematically extract actionable insights (Alanne & Sierla, 2022). In the domain of sentiment analysis, traditional machine learning classifiers, notably Naïve Bayes (NB) and Random Forest (RF), have been widely adopted due to their distinct architectural advantages. Naïve Bayes is highly regarded for its computational efficiency and robust performance in learning from text data with minimal processing overhead. Conversely, Random Forest, an ensemble learning method, enhances classification stability by aggregating decision trees, thereby effectively reducing variance and mitigating overfitting tendencies. Prior studies have extensively applied these algorithms to evaluate user feedback across various applications, such as MyPertamina, CapCut, and Minecraft, consistently reporting competitive baseline accuracies (Amelia et al., 2024; Arisula & Parjito, 2024; Irawan et al., 2024).

Despite these contributions, existing literature exhibits two critical gaps. First, sentiment analysis on local Indonesian applications predominantly focuses on native-language text, frequently discarding or misclassifying English reviews. In a multilingual market like Indonesia, English reviews written by tech-savvy users often exhibit superior grammatical structure and syntactic consistency compared to highly colloquial, slang-heavy Indonesian reviews. Separating and focusing on English-language opinions through structured language filtering remains under-explored, despite its potential to enhance feature extraction and boost model classification accuracy. Second, while recent advancements in deep learning have introduced powerful transformer-based architectures like BERT and IndoBERT—which excel at capturing deep contextual relationships and handling ambiguous semantic classes (Sanjaya et al., 2026)—their deployment introduces massive computational overhead, prolonged training periods, and a strict dependency on high-end GPU infrastructure. For enterprise-level monitoring where real-time deployment, low resource consumption, and model interpretability (such as identifying *feature importance*) are critical, conventional machine learning models like NB and RF remain practically superior and highly viable, particularly for moderately sized datasets. Furthermore, while limited studies have touched upon the MyTelkomsel application using Support Vector Machines or lexicon-based approaches, there is a notable scarcity of research evaluating these models on a large-scale, heterogeneous dataset. To bridge these empirical and practical gaps, this study utilizes a comprehensive dataset of 25,000 scraped Google Play Store reviews for the MyTelkomsel app. By applying a specialized language-filtering mechanism and conducting an empirical comparison between Naïve Bayes and Random Forest, this research aims to optimize sentiment classification into positive, negative, and neutral categories. Ultimately, this study provides a high-accuracy, computationally efficient framework to aid telecommunication providers in distinguishing customer satisfaction, thereby facilitating targeted service enhancements and reducing customer turnover.

2. Literature Review

2.1 Sentiment Analysis

Sentiment analysis, as part of NLP, is concerned with detecting and analyzing subjective information conveyed through text. It is also recognized as a technique for extracting information from text to classify sentiments into categories like positive, neutral, or negative(Maulidah et al., 2026). This approach aims to identify and interpret the emotions present in textual reviews to automatically predict and analyze public sentiment, mood, and emotional expressions regarding a specific topic. These emotional expressions are subsequently sorted into sentiment classes, usually reflecting supportive or opposing viewpoints (Haerani, 2026) This technique is extensively employed to comprehend public perceptions of products, services, and policies. In the case of user feedback on the Google Play Store, sentiment analysis serves as a crucial tool for app developers to systematically understand users' emotions and experiences. However, processing real-world app reviews introduces unique linguistic barriers, particularly multilingual structures where users frequently alternate between Indonesian and English or utilize distinct English language expressions to describe technical issues. Standard NLP pipelines often strip away non-native text, yet maintaining and properly isolating English reviews can significantly minimize feature sparsity and preserve high-quality, syntactically consistent feedback from tech savvy demographics. The insights derived from this analysis can be used to improve service quality and support data-driven decision-making. This research categorized user reviews into three sentiment types: positive, neutral, and negative, to differentiate between various levels of user satisfaction and the distinct technical issues encountered(Ridho et al., 2026). Sentiment analysis is generally divided into two classification types: binary and multi-class. Binary classification involves only two sentiment classes, whereas multi-class classification includes more than two. In this study, a multi-class classification method was employed because it encompassed three sentiment categories. Sentiment analysis commonly relies on machine learning methods, given their strength in managing vast datasets and identifying nuanced linguistic features. Despite its comprehensive nature, multi-class sentiment analysis faces a persistent bottleneck in classifying the neutral sentiment category. Unlike explicitly polarized positive or negative classes, neutral reviews typically contain objective system descriptions,

technical bug logs, or mixed feature requests that lack clear emotional markers. This absence of affective vocabulary creates heavily overlapping feature spaces, making the decision boundaries highly ambiguous for standard classifiers and requiring robust feature representations to avoid misclassification into polarized groups.

2.2 Text Mining and Natural Language Processing

Text mining is the process of deriving meaningful insights and patterns from unstructured text data. One of its primary objectives is to identify and understand key topics or issues within large-scale textual data (Mardiah et al., 2026). Such data is typically obtained from various sources, including news articles, emails, documents, online forums, and social media platforms. Through a series of preprocessing and feature extraction stages, text mining transforms unstructured text into structured and meaningful information. Text mining covers a range of methods, such as topic modeling, sentiment analysis, named entity recognition (NER), and document classification.. These techniques enable the identification of prevalent topics within large document collections, the analysis of sentiments expressed in text, and the extraction of important entities such as individuals, locations, and organizations (Pantouw & Tangkawarow, 2026). The process of text mining typically involves multiple stages, such as gathering data, preprocessing the text, transforming the data, extracting features, building models (like classification, clustering, or sentiment analysis), and interpreting the outcomes(Daely et al., 2026). Since the early 1950s, scholars have investigated Natural Language Processing, a vital method for assessing a machine's capacity to comprehend human language(Puspita, 2026). It combines computational methods, artificial intelligence, and linguistic principles to allow machines to automatically interpret and handle human language. This method enables computers to identify patterns in unstructured text, including informal language, slang, and the linguistic variations often present in user-generated reviews(Dinata et al., 2025). The implementation of NLP enables analytical results that more closely approximate human understanding. Its development has accelerated due to the rapid growth of textual data from digital platforms. In practical applications, NLP is widely used in various domains, including customer service chatbots, information retrieval systems, recommendation systems, and public opinion analysis. For example, search engines utilize NLP techniques to interpret user queries and deliver relevant results, while organizations use it to analyze user-generated content and gain insights into consumer perceptions (Pantouw & Tangkawarow, 2026). Furthermore, NLP’s ability to process large-scale textual data in real time provides a significant advantage over conventional methods that rely on manual and self-reported approaches (Rusdiansyah et al., 2026). Due to these capabilities, NLP is considered highly effective for sentiment analysis, as it is capable of understanding the contextual and semantic connections between words, enhancing the accuracy of classification, and be integrated with machine learning and deep learning models. Its scalability and automation capabilities also enable efficient analysis of large datasets, making it suitable for generating insights that support data-driven decision-making (Handayani et al., 2026).

2.3 Feature Extraction (TF-IDF)

In this research, Term Frequency–Inverse Document Frequency (TF-IDF) is utilized for feature extraction, a prevalent term-weighting method in Natural Language Processing (NLP) that converts text data into numerical vectors. TF-IDF assesses a word's significance within a document by taking into account both its frequency and its rarity throughout the entire corpus(Meyda et al., 2026; Pebrian et al., 2026). It highlights pertinent terms while diminishing the impact of overly frequent words that have limited semantic value(Patimah et al., 2026). The TF-IDF procedure involves several steps, such as tokenization to break down review text into individual tokens, calculating Term Frequency (TF) to ascertain how often a word appears in a document, and Inverse Document Frequency (IDF) to assess a word's rarity across the entire corpus. Consequently, words that appear frequently in multiple documents receive lower weights. Each review is then converted into a numerical vector representation based on the top 1,000 features chosen according to their TF-IDF scores. This approach effectively minimizes the impact of common terms, enabling the model to focus on more significant features. In this context, each

document is represented as a vector composed of TF-IDF weights derived from the corpus's vocabulary. Words that are frequent in a specific document but rare across the entire dataset are given higher weights, thereby enhancing their ability to differentiate during the classification process.

2.4 Machine Learning Algorithms

Naive Bayes is a classification technique based on probability, derived from Bayes' Theorem, and is noted for its "naïve" presumption that features are independent (Ridho et al., 2026). Despite this assumption often being unrealistic in natural language contexts, the algorithm remains popular for text classification due to its computational efficiency and robust performance with extensive datasets. It operates by calculating the posterior probability for each class, integrating prior probabilities with the likelihood of the observed features. This approach is particularly adept at managing high-dimensional text data, such as TF-IDF representations, consistently yielding dependable results despite its theoretical simplicity (Nyoto et al., 2026). In classification analysis, the Naïve Bayes algorithm is one of the most commonly used techniques. The approach utilizes probability and is grounded in Bayes' theorem, which calculates the posterior probability of a class based on the evidence from observed features(Al Sarwoto et al., 2026). The Naïve Bayes algorithm's mathematical foundation stems from Bayes' theorem, which can be expressed as follows (Manoppo et al., 2026).

$$P(C|X)=\frac{P\,(X|C)P\,(C)}{P(X)}$$

Description:

$P(C \mid X)$is the posterior probability, indicating the likelihood that a particular data point is part of a specific class.

$P(X \mid C)$is The likelihood, or conditional probability, indicates the chance of observing data assuming it is part of a specific class.

$P(C)$represents the initial likelihood of class, reflecting the probability of the class occurring prior to data observation.

$P(X)$is the probability of observing the data X.

In contrast, Random Forest is an ensemble learning technique that combines several independently built decision trees(Puspa & Indrati, 2026). Each tree produces a prediction, and the ultimate classification is determined by a majority vote(Nyoto et al., 2026). This ensemble approach significantly enhances the model's stability and minimizes the risk of overfitting, a frequent problem with individual decision trees. A key benefit of Random Forest is its capability to handle high-dimensional features and detect complex, non-linear relationships. By employing the bagging (bootstrap aggregating) technique, Random Forest enhances generalization and ensures more reliable predictions. In this research, both algorithms are trained using TF-IDF numerical vectors to discern essential patterns for sentiment classification tasks. The Random Forest mechanism in this study is explained through several key stages. Initially, the algorithm employs bootstrap sampling, also known as bagging, by creating multiple subsets of the training data through the method of sampling with replacement. Each decision tree is trained on a distinct bootstrap sample, introducing variation in the training data and helping to decrease model variance. Next, at each tree node, Random Forest performs random feature selection by considering only a subset of features, often called max_features, instead of evaluating all available features. This method promotes a diverse set of trees (decorrelation) and improves the model's ability to generalize. Additionally, each decision tree is built by continuously splitting nodes based on an impurity metric, such as Gini impurity or entropy. This splitting process persists until certain stopping conditions are met, such as achieving the maximum depth (max_depth) or when the number of samples in a node is less than min_samples_split or min_samples_leaf. In this study, which addresses a binary classification problem, each tree generates a class prediction, and the final outcome of the Random Forest model is determined through majority voting, where the class most frequently predicted by all trees is chosen. By combining bagging and random feature selection, Random Forest is generally more robust than a single decision tree and better at

capturing nonlinear relationships among features(Rahajoe et al., 2026). Mathematically, the majority voting process in Random Forest can be expressed as follows(Tsou, 2025).

$$\hat{y} = mode\ \{T1(x), T2(x), \dots, Tn(x)\}$$

Description:

$\hat{y}$: The final predicted output of the Random Forest algorithm.
$T_i(x)$: The output generated by the -th decision tree for the given input data .
n: The total number of decision trees used in the Random Forest model.

Despite its high computational efficiency and strong baseline performance in text analytics, the Naïve Bayes classifier possesses distinct architectural limitations when applied to real-world application reviews. The algorithm relies strictly on the conditional independence assumption, which presumes that the occurrence of a specific word is completely unrelated to the presence of any other word within the review. In natural language processing, this assumption is inherently flawed as context, multi-word phrases, and syntactic dependencies heavily dictate the true sentiment of a sentence. Furthermore, Naïve Bayes is highly sensitive to severe class imbalance challenges. In a heterogeneous dataset where certain sentiment classes (such as extreme positive ratings or negative complaints) vastly outnumber minority classes, Naïve Bayes tends to become heavily biased toward the majority class, leading to a high overall accuracy but a critically compromised recall rate for the underrepresented sentiments. These limitations necessitate an empirical comparison with robust ensemble-based techniques, such as Random Forest, which can inherently capture feature interactions and mitigate class imbalances more effectively.

2.5 Previous Studies

Previous studies have employed various machine learning methods to perform sentiment analysis on mobile app reviews. One study specifically assessed the performance of Support Vector Machine (SVM) and Logistic Regression in analyzing sentiments from MyTelkomsel reviews collected from the Google Play Store. This research analyzed 7,000 review entries and included several preprocessing steps, such as data cleaning, case folding, normalization, tokenization, stopword removal, stemming, and TF-IDF feature extraction. The findings indicated that SVM slightly surpassed Logistic Regression, achieving an accuracy of 93.36%, an AUC value of 0.9680, and a positive-class recall of 82%. In contrast, Logistic Regression achieved an accuracy of 93.12% with a positive-class recall of 79%. Although both models achieved similar weighted-average precision, recall, and F1-score values, SVM produced fewer misclassification errors than Logistic Regression. The study also revealed that Logistic Regression had a higher precision for the positive class, reaching 99%, while SVM achieved 96%. However, SVM showed a more balanced error distribution and better ability to recognize positive reviews. The findings suggest that SVM is more suitable for sentiment analysis of MyTelkomsel reviews because it provides better generalization performance and lower classification errors.

However, the previous study only focused on binary classification by dividing sentiments into positive and negative classes, while neutral reviews were excluded from the dataset. In addition, the study only compared SVM and Logistic Regression. A critical synthesis of previous literature reveals three persistent methodological bottlenecks in application review analysis that require deeper empirical resolution. First, existing studies often rely on binary classification, thereby omitting neutral sentiments which frequently contain objective feature requests or technical bug logs; excluding this class creates a significant gap in customer feedback evaluation, while its inclusion complicates decision boundaries due to heavily overlapping feature spaces. Second, large-scale datasets are inherently vulnerable to class imbalance, causing conventional algorithms like Logistic Regression and Naïve Bayes to become biased toward majority classes at the expense of minority-class recall—a challenge that justifies the use of ensemble methods like Random Forest to balance error distributions through bagging mechanisms. Lastly, the structural complexity of multilingual text is often ignored by native-language pipelines that discard English reviews, despite the fact that in multilingual markets like Indonesia, English and code-switched feedback often provide more syntactically consistent technical critiques than

colloquial native text. Consequently, integrating a structured language-filtering framework remains an under-explored yet vital approach to reduce feature sparsity and optimize multi-class sentiment detection.

2.6 Research Gap

While earlier research has shown that machine learning methods like Support Vector Machine (SVM) and Logistic Regression are effective for sentiment analysis of MyTelkomsel app reviews, they encounter several obstacles. Most studies have focused on binary classification, splitting sentiments into positive and negative, and have neglected the neutral category, which could offer valuable insights into user opinions. Additionally, previous research has primarily compared a narrow set of algorithms, such as SVM and Logistic Regression, without considering other widely used models like Naïve Bayes and Random Forest. This limits the understanding of how various algorithms perform, especially in managing high-dimensional text data and class imbalance issues. Moreover, some studies heavily depend on rating-based labeling, which might not accurately reflect the sentiment expressed in the text. This method can decrease the accuracy of sentiment representation, particularly when there is a mismatch between the text and the rating. Another challenge in earlier research is the small dataset size, which could affect the model's generalization capability. Furthermore, standard pipelines frequently overlook the linguistic diversity of user feedback by systematically purging or misclassifying English reviews, thereby ignoring the fact that English and code-switched opinions from local users often provide highly structured, syntactically consistent technical critiques. To address these challenges, this study focuses on multi-class sentiment classification—categorizing sentiments as positive, neutral, or negative while implementing a dedicated language-filtering mechanism to isolate English opinions, and assesses the effectiveness of Naïve Bayes and Random Forest algorithms. This is accomplished by using TF-IDF feature extraction on a larger dataset of 25,000 MyTelkomsel user reviews from the Google Play Store. The research aims to provide a more comprehensive evaluation of sentiment classification performance and offer valuable insights for improving MyTelkomsel application services.

3 Research Method

Earlier studies have encountered difficulties due to the small size of datasets, which may affect the model's generalization capabilities. To address these challenges, this research emphasizes multi-class sentiment classification, sorting sentiments into positive, neutral, or negative categories, and assesses the performance of Naïve Bayes and Random Forest algorithms. This is accomplished by applying TF-IDF feature extraction to an expanded dataset comprising 25,000 MyTelkomsel user reviews obtained from the Google Play Store. The study aims to deliver a more comprehensive evaluation of sentiment classification performance and provide valuable insights for improving MyTelkomsel application services. Sentiment analysis of user reviews employs natural language processing and machine learning techniques to classify textual feedback. Standard procedures include text preprocessing steps such as tokenization, converting text to lowercase, removing punctuation, stemming, and eliminating stopwords. Feature extraction is performed using TF-IDF weighting (Martanto & Istiono, 2024; Adinata et al., n.d.; Alhejaili et al., 2021). Comparative studies suggest that ensemble methods, such as Random Forest, generally outperform Naïve Bayes in terms of accuracy and F1-score for classifying application reviews (Adinata et al., n.d.; Kumar et al., 2025; Sagala & Samuel, 2024). To systematically evaluate these algorithms on MyTelkomsel reviews, the proposed methodology is divided into five stages, as illustrated in Figures 1 and 2.

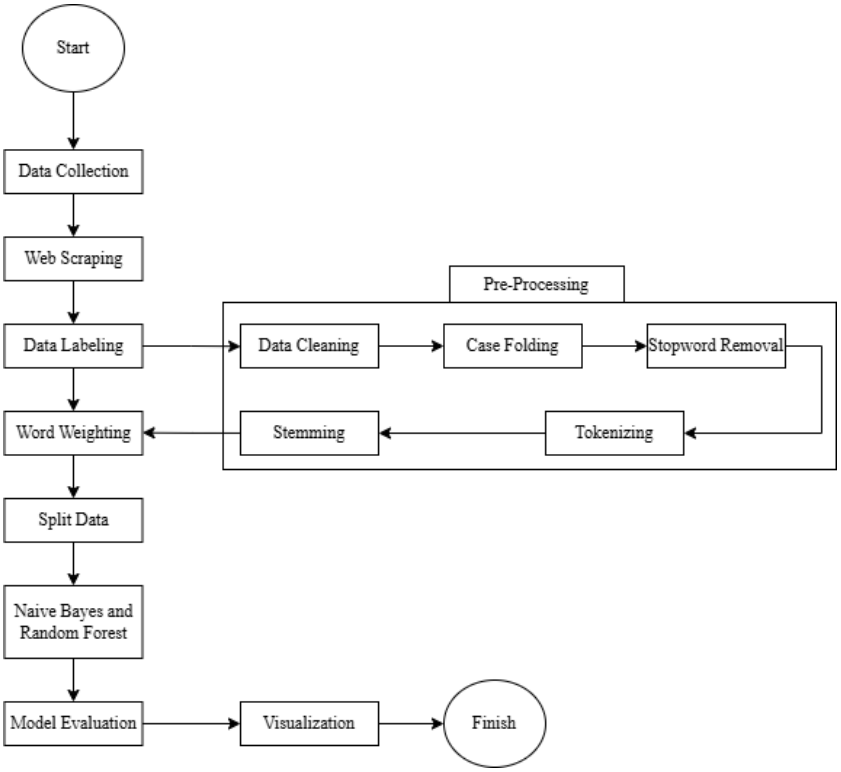

Figure 1. Methodology Pipeline Diagram

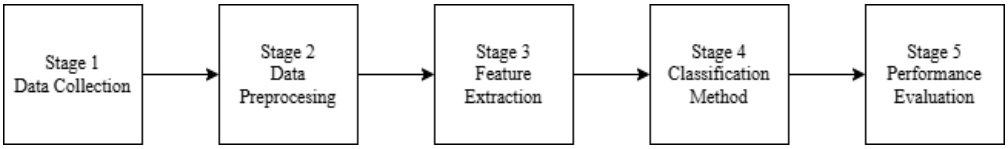

Figure 2. Methodology Pipeline Diagram

3.1 Data Collection

User reviews for the MyTelkomsel app were collected from the Google Play Store via an automated web-scraping tool leveraging the Google Play Scraper API. Unlike restricted baseline datasets in prior literature, this study collected a comprehensive raw dataset comprising 25,000 user review entries to ensure high model generalization and minimize out-of-vocabulary constraints. Each scraped data entry inherently consists of a textual customer comment, metadata timestamp, and a corresponding user rating ranging from 1 to 5 stars. To resolve the gaps identified in binary sentiment analysis pipelines, this research employs a multi-class sentiment labeling methodology. The ground-truth labeling was constructed systematically based on the star ratings: reviews with 4-5 stars were designated as positive sentiment, 3-star reviews were mapped as neutral sentiment to preserve non-polarized objective feedback and technical feature requests, and 1–2 stars were classified as negative sentiment. To establish experimental reproducibility, the final labeled multi-class dataset was split into a training set of 80% and a testing set of 20% using a stratified sampling approach to preserve the exact class distribution across both partitions during the Naïve Bayes and Random Forest evaluation phases (Adinata et al., n.d.).

3.2. Web Scraping

The initial phase of the data procurement pipeline involved automated web scraping to extract user reviews of the MyTelkomsel application directly from the Google Play Store. This process was programmatically executed in a Python environment utilizing the open-source google-play-scraper library, which allows direct API communication with the application distribution platform without encountering computational request blocks. The automated scraping script was configured targeting the unique application identifier package

(com.telkomsel.telkomselcx). To ensure a robust sample size for multi-class classification evaluation, the scraper parameters were set to retrieve a maximum volume of 25,000 of the most recent historical user reviews, specifically filtered from the Indonesian market region (lang='id', country='co.id'). The raw JSON payloads fetched by the API were systematically structured and converted into a tabular layout (DataFrame) utilizing the pandas library. To optimize computational storage and streamline subsequent pre-processing stages, a programmatic feature selection was applied to isolate core data attributes. From the comprehensive raw metadata, only four critical columns were preserved for the analytical dataset: user identifier (username), numerical evaluation score (rating), temporal stamp (date), and the unstructured textual opinion (review content). This systematic extraction framework establishes the reproducibility of the dataset collection phase, providing the foundational ground-truth text required for the subsequent language filtering and machine learning classification tasks.

```
!pip install google-play-scraper # Ensuring the library is available
import pandas as pd
from google_play_scraper import reviews, Sort

app_id = "com.telkomsel.telkomselcm"

result, _ = reviews(
    app_id,
    lang='id',
    country='id',
    sort=Sort.NEWEST,
    count=25000
)

df = pd.DataFrame(result)
# Ambil kolom yang penting saja
df = df[['userName', 'score', 'at', 'content']]
print(f"Berhasil mengambil {len(df)} ulasan asli.")
```

```
Requirement already satisfied: google-play-scraper in /usr/local/lib/python3.12/dist-packages (1.2.7)
Berhasil mengambil 25000 ulasan asli.
```

Figure 3. Methodology Pipeline Diagram

3.3. Data Labeling

Data labeling is a critical process in supervised machine learning designed to assign ground-truth multi-class targets—positive, neutral, and negative—to unstructured MyTelkomsel reviews. To eliminate human subjectivity and ensure deterministic consistency, a programmatic, rules-based mapping function was applied directly to the numeric star ratings. Reviews with 4–5 stars were labeled as **Positive** (high satisfaction), while 1–2 stars were classified as **Negative** (system bugs or user dissatisfaction). Crucially, 3-star reviews were mapped as **Neutral** to preserve objective user comments, dual-polarized feedback, or technical feature requests that lack definitive emotional extremes, thereby resolving decision boundary ambiguities. To verify the alignment between the assigned categorical labels and the actual textual semantics, a stratified validation sampling protocol was executed. Using the pandas library via the pd.concat() function, the top two descriptive rows from each star rating tier (1 to 5) were isolated and consolidated into a verification sub-table containing userName, score, sentiment, and content. This rigorous verification procedure confirms the structural validity of the mapping function, ensuring that the ground-truth classes successfully encapsulate customer sentiment characteristics prior to the feature extraction stage.

```python
import pandas as pd

def pelabelan_sentimen(score):
    if score >= 4:
        return 'Positive'
    elif score == 3:
        return 'Neutral'
    else:
        return 'Negative'

df['sentiment'] = df['score'].apply(pelabelan_sentimen)

bintang_5 = df[df['score'] == 5].head(2)
bintang_4 = df[df['score'] == 4].head(2)
bintang_3 = df[df['score'] == 3].head(2)
bintang_2 = df[df['score'] == 2].head(2)
bintang_1 = df[df['score'] == 1].head(2)

tabel_verifikasi = pd.concat([bintang_5, bintang_4, bintang_3, bintang_2, bintang_1]).reset_index(drop=True)

print("Label Verification Table (10 Samples):")
print("-" * 50)
display(tabel_verifikasi[['userName', 'score', 'sentiment', 'content']])
```

```
Label Verification Table (10 Samples):
--------------------------------------------------
```

	userName	score	sentiment	content
0	Pengguna Google	5	Positive	Telkomsel hebat
1	Pengguna Google	5	Positive	terbaik 👍
2	Pengguna Google	4	Positive	Tingkatkan pelayanan terbaik bagi pelanggan
3	Pengguna Google	4	Positive	sangat menyenangkan
4	Pengguna Google	3	Neutral	bagus
5	Pengguna Google	3	Neutral	ga bisa copot nomer orang lain, aku pengin hap...
6	Pengguna Google	2	Negative	aplikasi berat ,dr dlu smpai sekarang tidak ad...
7	Pengguna Google	2	Negative	ok,tpi sring lag
8	Pengguna Google	1	Negative	apl lemot jelek sinyal bad
9	Pengguna Google	1	Negative	Lambat loading aplikasi suka berhenti tiba-tiba

Figure 4. Data Labeling

3.4. Stopword removal

Stopword removal is a pivotal refinement phase within the text preprocessing pipeline designed to eliminate high-frequency, non-semantic words that do not contribute to the emotional valence of a review. Executed immediately following the cleaning and normalization stages, this process systematically filters out linguistic noise by retaining only highly informative sentiment carriers such as "slow," "good," "error," or "fast." This targeted reduction in feature dimensionality directly optimizes downstream classifiers, enabling Naïve Bayes to calculate conditional token probabilities without distortion from irrelevant terms and empowering Random Forest to construct more efficient decision trees, thereby accelerating computational throughput and enhancing multi-class classification accuracy.

The empirical outcome of this stage is reflected in the structural transformation of the isolated English review corpus within the review_english data attribute. During this procedure, standard syntactic particles lacking affective weight—including "the," "is," "it's," "to," and "has"—are programmatically purged. For instance, the raw expression "the packet actually has a bad signal" is reduced to "packet actually bad signal", while "Telkomsel is great" is streamlined into "Telkomsel great". This filtering mechanism isolates dense, high-impact descriptors (e.g., "bad," "slow," "great," "expensive"), effectively resolving feature sparsity bottlenecks and allowing the subsequent layer to precisely map the distinctive patterns separating positive, negative, and neutral customer feedback.

```
... Stopword removal completed.
```

	review_english	Stopword_Removal
0	slow app, bad signal	slow app, bad signal
1	slow loading application like stopping suddenly	slow loading application like stopping suddenly
2	the packet actually has a bad signal	packet actually bad signal
3	It's really hard to open the application to ge...	really hard open application get people buy qu...
4	really bad network, exorbitant quota prices re...	really bad network, exorbitant quota prices re...
5	Telkomsel is great	Telkomsel great
6	It's weird, it's a weird network, I can't open...	weird, weird network, can't open apk, works ma...
7	Why is this signal, wow, it's already expensiv...	signal, wow, already expensive? signal keeps d...
8	Stupid service at night is really bad, the net...	Stupid service night really bad, network expen...
9	the best	best

Figure 5. Stopword removal

3.5. Tokenization

Tokenization is a foundational preprocessing step deployed to segment unstructured review sentences into discrete lexical tokens. Utilizing the word_tokenize function from Python's Natural Language Toolkit (nltk) library, continuous text sequences within the isolated English corpus are programmatically broken down at word boundaries into independent character strings (e.g., separating a sentence into unique components like *'analysis'*, *'user'*, *'reviews'*, and *'mytelkomsel'*). Isolating text into a granular token array prevents semantic distortion and directly optimizes downstream multi-class classification tasks. This structured representation allows the Naïve Bayes model to build an uncompromised vocabulary dictionary for calculating independent conditional word probabilities, while simultaneously enabling the Random Forest model to isolate high-impact keyword nodes when constructing decision trees across the positive, negative, and neutral sentiment spectrum.

Tokenization completed successfully.

	Stopword_Removal	Tokenizing
0	slow app, bad signal	[slow, app, ,, bad, signal]
1	slow loading application like stopping suddenly	[slow, loading, application, like, stopping, s...
2	packet actually bad signal	[packet, actually, bad, signal]
3	really hard open application get people buy qu...	[really, hard, open, application, get, people,...
4	really bad network, exorbitant quota prices re...	[really, bad, network, ,, exorbitant, quota, p...
5	Telkomsel great	[Telkomsel, great]
6	weird, weird network, can't open apk, works ma...	[weird, ,, weird, network, ,, ca, n't, open, a...
7	signal, wow, already expensive? signal keeps d...	[signal, ,, wow, ,, already, expensive, ?, sig...
8	Stupid service night really bad, network expen...	[Stupid, service, night, really, bad, ,, netwo...
9	best	[best]

Figure 6. Tokenization

3.6. Term Frequency–Inverse TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) was implemented to vectorize tokenized reviews into a structured numeric matrix by calculating relative term importance. While TF, weights localized token densities within single reviews, IDF, penalizes ubiquitous words across the entire 25,000-entry dataset. This mathematical vectorization compressed the unstructured text into an optimized space consisting of 1,681 unique features. As validated in Figure 7, the pipeline successfully isolated dense, domain-specific technical critiques and affective indicators, where terms like "good", "expens" (stemmed from expensive), "packag", "network", and "slow" achieved the highest cumulative weight scores. This highly stratified feature distribution directly reduces semantic ambiguity across the multi-class spectrum, allowing Naïve Bayes to establish uncompromised conditional probability arrays and enabling Random Forest to construct highly discriminative decision trees for positive, negative, and neutral sentiment categories.

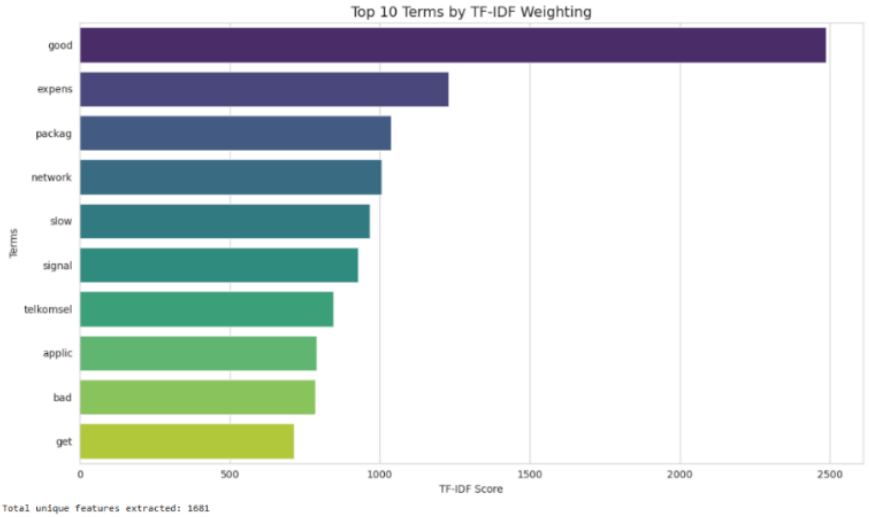

Figure 7. Top 10 Terms Distributions and Feature Weights by TF-IDF Vectorization

3.7 Preprocessing and Feature Extraction

Following the preprocessing and feature extraction stages, the classification performance of the Naïve Bayes (NB) and Random Forest (RF) models was systematically assessed. To evaluate the models' ability to predict each sentiment class accurately, the evaluation framework utilized standard performance metrics, including accuracy, precision, recall, and F1-score (Alhejaili et al., 2021). These metrics are highly critical for addressing real-world text classification challenges, particularly in cases of severe class imbalance, where the F1-score is heavily prioritized for its capacity to represent the performance of minority classes without majority-class distortion (Sagala & Samuel, 2024). In conclusion, this research outlines a structured, five-stage analytical process: data collection, text preprocessing, TF-IDF feature extraction, machine learning classification, and multi-class model evaluation. By focusing on these core phases, this study establishes a robust comparative foundation to evaluate how effectively the Naïve Bayes and Random Forest algorithms can capture complex sentiment patterns across the positive, negative, and neutral categories within the MyTelkomsel application reviews.

3.8 Classification Methods

This study evaluated two distinct supervised learning techniques to execute the multi-class sentiment classification task: Multinomial Naïve Bayes (NB) and Random Forest (RF). The Multinomial Naïve Bayes model calculates the posterior probability for each category based on the strong premise that text features are conditionally independent given the class target. This probabilistic approach operates efficiently on high-dimensional data, making it highly robust for calculating word occurrences across the positive, neutral, and negative sentiment spectrum (Alhejaili et al., 2021). Conversely, the Random Forest model leverages an ensemble architecture consisting of multiple individual decision trees. In this approach, each decision tree is constructed using a bootstrap sample of the training data combined with a randomly selected subset of TF-IDF features. The ensemble then utilizes a majority voting mechanism to determine the final multi-class sentiment assignment, which significantly achieves higher generalization accuracy by minimizing model variance (Adinata et al., n.d.). In this implementation, the Random Forest classifier was trained using a fixed ensemble of 100 decision trees (n_estimators = 100) with optimized depth boundaries fine-tuned via cross-validation, allowing the model to construct highly discriminative split nodes that mitigate class imbalance and neutral-zone ambiguities. For the probabilistic baseline, the multinomial variant of Naïve Bayes was deployed to directly process the fractional weights generated by the TF-IDF vectorizer. Both classifiers were mapped using the exact same labeled training vector space. This comparative setup builds directly upon prior literature in app-review sentiment environments, where ensemble structures have demonstrated strong mathematical capabilities in resolving dense textual feedback (Alhejaili et al., 2021; Sagala & Samuel, 2024).

3.9 Performance Evaluation

The trained models were evaluated on an independent test dataset using multi-class accuracy, precision, recall, and F1-score (Alhejaili et al., 2021). Accuracy reflects the overall percentage of correct predictions, while precision and recall quantify model validity per category. The F1-score was computed as the harmonic mean of precision and recall to provide a balanced performance metric across all three distinct classes: positive, neutral, and negative. Given the inherent class imbalance within the MyTelkomsel dataset, the F1-score was heavily prioritized over simple accuracy for algorithm comparison, as it accurately reflects the model's predictive power on minority categories and neutral-zone ambiguities (Sagala & Samuel, 2024). Lastly, to ensure experimental robustness and reproducibility, all performance statistics were averaged across multiple stratified random train-test splits.

4. Results and Discussions

4.1 Dataset Characteristics

The primary dataset distribution extracted from the MyTelkomsel application reviews is illustrated in Figure 8. Across the 25,000 total scraped entries, a severe class imbalance is highly visible. The ground-truth mapping yielded 14,912 Negative reviews (13,396 from 1-star and 1,516 from 2-star ratings), 8,931 Positive reviews (1,097 from 4-star and 7,834 from 5-star ratings), and a minority share of 1,157 Neutral reviews (from 3-star ratings). This skewed distribution demonstrates a definitive heavy-tailed pattern dominated by polarized extremes, specifically heavily leaning toward negative user experiences. In computational sentiment analysis, this class imbalance significantly threatens classification integrity. The majority class (Negative sentiment) provides nearly 13 times more data points than the minority class (Neutral sentiment). Without rigorous evaluation adjustments, standard classifiers mathematically default to predicting the majority distribution to optimize global accuracy. This data structure explains why prioritizing macro-averaged F1-scores rather than basic accuracy is an absolute necessity in this research, as it prevents the high volume of negative reviews from masking poor predictive performance within the critical, data-sparse neutral sentiment boundaries.

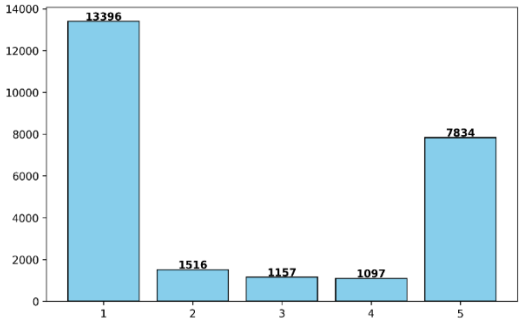

Figure 8. Dataset Charasterictics (N = 25,000)

4.2 Preprocessing Results

The task of text preprocessing, which involves transforming text to lowercase, eliminating punctuation, breaking it into tokens, removing stopwords, and applying stemming, effectively transformed raw English review data into a structured format suitable for machine learning models. The specific removal of stopwords and application of stemming for the English language played a crucial role in reducing noise and handling linguistic variations within the dataset. To visually evaluate the empirical outcomes of this pipeline, Figure 9-11 presents Word Clouds across all three categories. While the Positive and Negative Word Clouds isolate highly polarized affective terms (e.g., *"best"*, *"nice"* vs. *"slow"*, *"signal"*), the Neutral Word Cloud exposes severe semantic overlaps, dominated by generic tokens like *"network"*, *"get"*, and *"buy"*. This high-density linguistic crossover at the center of the vector space explains the severe decision-boundary confusion and the resulting weaker classification performance within the neutral class.

Figure 9. Word cloud visualization of positive sentiment keywords

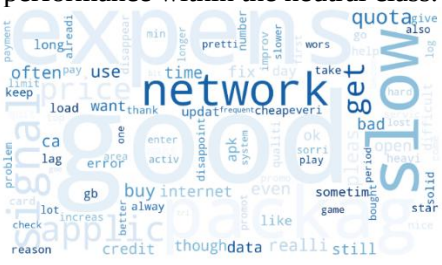

Figure 10. Word cloud visualization of neutral sentiment keywords

Figure 11. Word cloud visualization of negative sentiment keywords

4.3 Accuracy Comparison

As illustrated in Figure 12, both machine learning models demonstrated competitive predictive performance on the test dataset, with the Random Forest classifier achieving an overall accuracy of **86.85%**, slightly outperforming Multinomial Naïve Bayes which reached **86.36%**. This narrow margin suggests that both classifiers benefit significantly from the optimized text preprocessing and TF-IDF feature extraction pipeline, which effectively stabilized the high-dimensional feature space before model execution. The slight operational edge of Random Forest is driven by its ensemble architecture. By combining multiple decision trees via bootstrap aggregation, the algorithm can capture intricate, non-linear relationships and contextual word dependencies within the user reviews. This structure makes Random Forest highly robust against computational noise, informal syntax, and minor typos common in mobile application feedback. Conversely, Multinomial Naïve Bayes assumes strict conditional independence between words given the class label. While this assumption simplifies mathematical operations and allows the model to perform remarkably well despite a highly skewed data distribution, it occasionally misinterprets contextually linked expressions where the collective meaning of a token sequence contradicts individual word probabilities. Nevertheless, the marginal 0.49% accuracy difference indicates that both models are highly effective baseline classifiers for the MyTelkomsel review corpus.

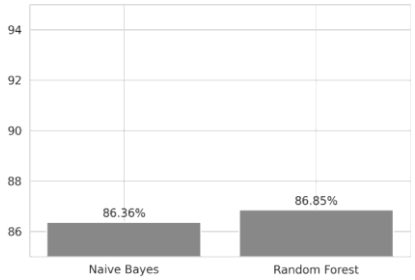

Figure 12. Comparison of model accuracy between Naïve Bayes and Random Forest

4.4 Precision, Recall, and F1-Score Analysis

The classification report in Table 1 reveals how each model navigates the severely skewed multi-class distribution. Driven by the massive data density in the majority Negative class (support: 2,980), both models achieved a dominant recall of 0.96. Random Forest showed a tighter decision boundary with a precision of 0.86 (F1-score: 0.91) compared to Naïve Bayes' precision of 0.85 (F1-score: 0.90). This indicates that while the high training volume enables near-perfect capture of user complaints, it mathematically biases both classifiers. For the polarized Positive class (support: 1,740), Naïve Bayes reached a precision of 0.90 but a lower recall of 0.81 (F1-score: 0.85), whereas Random Forest provided a better trade-off with a precision of 0.89 and a recall of 0.83, leading to an F1-score of 0.86.

Model	Class	Precision	Recall	F1-Score	Support
Naïve Bayes	Negative	0.85	0.96	0.90	2980
	Neutral	0.12	0.02	0.03	230
	Positive	0.90	0.81	0.85	1740
	Accuracy			0.86	4950
	Macro Avg	0.62	0.60	0.60	4950
	Weighted Avg	0.84	0.86	0.85	4950
Random Forest	Negative	0.86	0.96	0.91	2980
	Neutral	0.06	0.00	0.01	230
	Positive	0.89	0.83	0.86	1740
	Accuracy			0.87	4950
	Macro Avg	0.60	0.60	0.59	4950
	Weighted Avg	0.83	0.87	0.85	4950

Figure 13. Performance metrics comparison between Naive Bayes and Random Forest

Crucially, both models collapsed on the minority Neutral class (support: 230), with Naïve Bayes securing a dismal F1-score of 0.03 and Random Forest dropping to 0.01. This performance bottleneck directly demonstrates the severe joint impact of class imbalance and semantic overlap. Because the 3-star neutral reviews are mathematically sparse, their unique TF-IDF weights were completely overwhelmed by the prior class probabilities in Naïve Bayes and heavily diluted during tree node splits in Random Forest. Consequently, both algorithms systematically misclassified non-polarized, ambiguous technical feedback into the dominant categories, proving that baseline term-weighting cannot effectively resolve neutral-zone semantic crossovers.

4.5 Confusion Matrix Analysis

To examine the localized classification errors and behavioral biases of both models, a comparative analysis was performed on the test confusion matrices. The Multinomial Naïve Bayes matrix demonstrates strong true positive rates for the polarized classes, correctly identifying 2,866 negative and 1,405 positive instances. However, significant classification leakage occurred, with 322 actual positive reviews misclassified as negative, and 174 neutral reviews pulled into the negative zone. Random Forest exhibited a similar diagonal dominance, successfully predicting 2,852 negative and 1,446 positive instances, while suffering fewer false-negative errors in the positive tier (287 reviews predicted as negative).

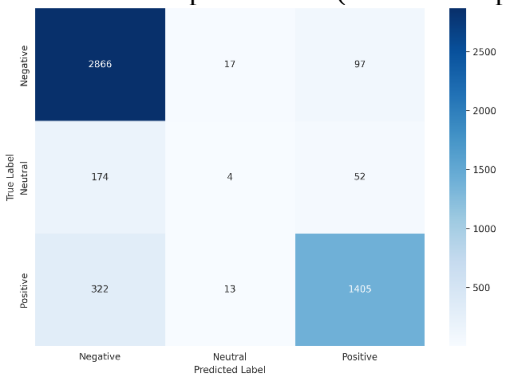

Figure 14. Confusion matrix for Naive Bayes sentiment classification model

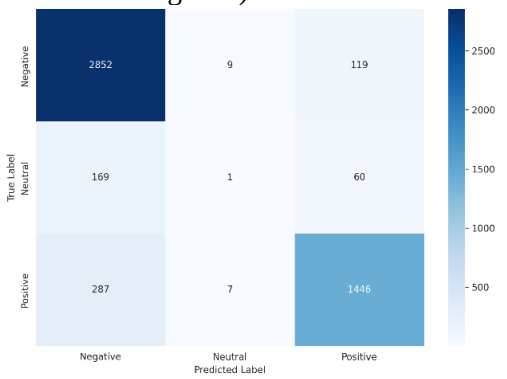

Figure 15. Confusion matrix for Random Forest sentiment classification model

The off-diagonal distributions provide definitive empirical proof of the neutral-class collapse caused by the severe class imbalance. Out of 230 actual neutral test samples, Naïve Bayes only correctly predicted 4 instances (misclassifying 174 as negative and 52 as positive), while Random Forest collapsed further, correctly identifying just 1 single instance (misclassifying 169 as negative and 60 as positive). This systemic bias toward the majority class occurs because the mathematical mass of the 14,912 negative training samples heavily skewed the prior probabilities in Naïve Bayes and overwhelmed the tree split criteria in Random Forest. As a result, both algorithms almost completely failed to resolve the ambiguous decision boundaries of non-polarized technical feedback, instead treating the sparse neutral boundaries as residual extensions of the dominant negative class.

5. Conclusion

This study shows that both Multinomial Naïve Bayes and Random Forest work well for classifying polarized sentiments in MyTelkomsel user reviews. Both models achieved a high recall of 0.96 in the majority negative class and stable F1-scores for positive feedback. Overall, Random Forest performed slightly better with 87% accuracy compared to Naïve Bayes at 86%, showing better stability in handling high-dimensional data and a more balanced error distribution. However, the minority neutral class remains a major challenge. Due to limited data and overlapping word usage, both models experienced a severe classification collapse in this zone. The confusion matrices confirm that both algorithms frequently misclassified non-polarized technical comments into the dominant positive or negative groups.

This work contributes to the NLP field by showing how traditional machine learning and standard term-weighting handle neutral data boundaries within a highly skewed multi-class dataset. From a practical standpoint, these findings help the MyTelkomsel development team automate complaint tracking and pinpoint specific technical issues, such as slow networks or packet errors, directly from customer feedback. To improve these multi-class decision boundaries further, future studies should focus on implementing data resampling methods like SMOTE or testing transformer-based deep learning models to better capture the meaning of non-polarized user comments.

References

Adinata, R. B., Supriyono, S., & Fithri, D. L. (n.d.). Sentiment Classification of MyTelkomsel Reviews Using SVM and Logistic Regression. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *20*(1).

Al Sarwoto, M. S. N., Firmansyah, H., & Asriyani, W. (2026). Application of the Naive Bayes Method for Classification of Food Nutritional Values. *Jejak Digital: Jurnal Ilmiah Multidisiplin*, *2*(1), 337–347.

Alanne, K., & Sierla, S. (2022). An overview of machine learning applications for smart buildings. In *Sustainable Cities and Society* (Vol. 76). https://doi.org/10.1016/j.scs.2021.103445

Arisula, J. P., & Parjito, P. (2024). COMPARISON OF NAIVE BAYES AND RANDOM FOREST METHODS IN SENTIMENT ANALYSIS ON THE GETCONTACT APPLICATION. *Jurnal Teknik Informatika (Jutif)*, *5*(5). https://doi.org/10.52436/1.jutif.2024.5.5.2004

Daely, S. A. G., Sanjaya, A. E., & Wijaya, A. (2026). Analysis of Visitor Satisfaction Patterns at Amanzi Waterpark Palembang Using the K-Means Clustering Algorithm. *Jurnal Ilmu Komputer Dan Informatika| E-ISSN: 3063-9026*, *2*(3), 52–60.

Dinata, R. M., Marhaeni, M., Atmadja, K., Rayhana, E., Hadi, V., & Al Kaf, U. (2025). Comprehensive Analysis of Sentiment Classification Model Performance: Cross-Metric Evaluation on Indonesian Movie Tweet Dataset – Sentiment Analytics Data from Twitter (X) About Indonesian-Language Films. *JURNAL REKAYASA INFORMASI*, *14*(1), 38–47.

Haerani, F. (2026). SENTIMENT ANALYSIS OF THE FREE NUTRITIOUS MEAL PROGRAM BASED ON COMMENTS ON SOCIAL MEDIA X USING GRU. *Jurnal Informatika Dan Teknik Elektro Terapan*, *14*(1).

Handayani, R., Berti, M. D., Japung, V. Y., Sudda, S. D., Maghfirah, N., & Rahmadani, F. (2026). Natural Language Processing (NLP) for Sentiment Analysis of Service Management Reviews on Kimia Farma. *Integrated Journal of Pharmacy Innovations*, *2*(1), 37–46.

irawan, I., Wardianto, W., Wathan, M. H., & Prayogi, M. B. (2024). Comparative Study: Random Forest, Naive Bayes, and Support Vector Machine Algorithms in Sentiment Analysis of

the Capcut Application on Google Play Store. *Jurnal Pengembangan Sistem Informasi Dan Informatika*, *5*(4). https://doi.org/10.47747/jpsii.v5i4.1959

Kumar, M., Khan, L., & Chang, H.-T. (2025). Evolving techniques in sentiment analysis: a comprehensive review. *PeerJ Computer Science*, *11*, e2592.

Liang, P. P., Zadeh, A., & Morency, L. P. (2024). Foundations & Trends in Multimodal Machine Learning: Principles, Challenges, and Open Questions. *ACM Computing Surveys*, *56*(10). https://doi.org/10.1145/3656580

Manoppo, A. R., Nurandani, D., Adrian, D., & Ayuda, M. I. (2026). Penerapan Naive Bayes untuk Memprediksi Status Keberhasilan Transaksi pada Sistem Top Up AY Pulsa Menggunakan Aplikasi Orange. *SINERGI*, *1*(1), 24–34.

Mardiah, M., Masruriyah, A. F. N., Tiana, A. H., Prakoso, B. S., Prasetyo, R. T., & Ardika, S. B. (2026). Pemodelan topik Dokumen Tesis menggunakan Metode Latent dirichlet allocation. *Technologica*, *5*(1), 125–135.

Martanto, M. L. F., & Istiono, W. (2024). Sentiment analysis of M-Paspor app reviews using multinomial Naive Bayes. *Journal of Logistics, Informatics and Service Science*, *11*(10), 311–326.

Maulidah, M., Suleman, S., Ardiansyah, A., Rahma, E., & Widodo, Q. E. A. (2026). Analisis Sentimen pada Ulasan Aplikasi Notion AI dengan Metode Support Vector Machine dan Random Forest. *SENTRI: Jurnal Riset Ilmiah*, *5*(2), 1147–1161.

Meyda, N., Akbar, M. T., Nurdin, M., Surya, S. H., Ibad, A. I., & Nursodiq, A. (2026). Analisis Sentimen untuk Deteksi Penipuan pada Twitter Menggunakan TF-IDF dan Naive Bayes. *RIGGS: Journal of Artificial Intelligence and Digital Business*, *5*(1), 7346–7354.

Nyoto, V. J., Riti, Y. F., & Tantokusumo, R. V. P. (2026). Analisis Perbandingan Algoritma Random Forest, Decision Tree Dan Naive Bayes Dalam Mendeteksi Spam SMS. *Jurnal Sains Informatika Terapan*, *5*(1), 279–286.

Pantouw, J. M., & Tangkawarow, I. R. H. T. (2026). Studi Literatur: Identifikasi Isu Logistik Pemilu 2024 di Indonesia Menggunakan Pendekatan Text Mining dan Topic Modeling Berbasis Latent Dirichlet Allocation (LDA) pada Data Google News. *Edutik: Jurnal Pendidikan Teknologi Informasi Dan Komunikasi*, *6*(1), 359–373.

Patimah, N. F., Kurniawan, R., Nurhakim, B., Putra, A. P., & Anwar, S. (2026). ANALISIS SENTIMEN ULASAN PENGGUNA APLIKASI QUORA MENGGUNAKAN METODE MULTINOMIAL NAÏVE BAYES DAN TF-IDF. *Journal of Computer Science and Artificial Intelligence (JCSAI)*, *3*(01).

Pebrian, H., Kusuma, A. A., & Pribadi, M. R. (2026). Analisis Sentimen Opini Publik terhadap Dedi Mulyadi di Twitter Menggunakan Ekstraksi Fitur TF-IDF dan Klasifikasi Naive Bayes. *Innovative: Journal Of Social Science Research*, *6*(2), 1–9.

Puspa, A., & Indrati, A. (2026). Implementasi Algoritma Random Forest Classifier Dalam Klasifikasi Kelayakan Air Minum. *RIGGS: Journal of Artificial Intelligence and Digital Business*, *5*(1), 3958–3965.

Puspita, R. (2026). IMPLEMENTASI NATURAL LANGUAGE PROCESSING PADA APLIKASI CHATBOT BERBASIS ARTIFICIAL INTELLIGENCE UNTUK LAYANAN INFORMASI PENAWARAN HARGA (STUDI KASUS: BRANZ BSD). *Jurnal Riset Multidisiplin Edukasi*, *3*(1), 346–364.

Rahajoe, A. D., Azaidane, D., Putra, B. A., Pahlevy, M. R., Bimantoro, B. S., & Akash, F. R. (2026). Heart Disease Analysis Using the Random Forest Method. *Digital Transformation Technology*, *6*(1), 150–157.

Ridho, M. R., Revindo, M. K., & Saputra, D. M. (2026). SENTIMENT ANALYSIS OF SATUSEHAT MOBILE APPLICATION USERS USING SVM AND NAÏVE BAYES

WITH SMOTE OPTIMIZATION. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *10*(2), 3480–3487.

Rusdiansyah, I., Pangestu, R., Azalia, D., Zhafran, M. F., Saputra, F., & Amsury, F. (2026). Integration of a Student Stress Level Classification Model Based on Natural Language Processing. *RIGGS: Journal of Artificial Intelligence and Digital Business*, *4*(4), 7823–7831.

Sagala, G. J., & Samuel, Y. T. (2024). Sentiment analysis on ChatGPT App reviews on google play store using random forest algorithm, support vector machine and naïve bayes. *International Journal of Engineering Business and Social Science*, *2*(04), 1194–1204.

Sanjaya, M. R. S., Indah, D. R., & Ruskan, E. L. (2026). HYBRID FINE-TUNING INDOBERT DAN HYBRID FINE-TUNING INDOBERT DAN ENSEMBLE TF-IDF LOGISTIC REGRESSION UNTUK ANALISIS SENTIMEN ULASAN APLIKASI ZALORA: HYBRID FINE-TUNING INDOBERT DAN ENSEMBLE TF-IDF LOGISTIC REGRESSION UNTUK ANALISIS SENTIMEN ULASAN APLIKASI ZALORA. *Rabit: Jurnal Teknologi Dan Sistem Informasi Univrab*, *11*(1), 324–336.

Tsou, M.-C. (2025). A machine learning-based model for predicting high deficiency risk ships in port state control: A case study of the port of singapore. *Journal of Marine Science and Engineering*, *13*(8), 1485.

Tuasamu, A., Gumilang, R. C., Fachrian, M. A., Krisnandi, D. R., & Indryani, A. W. (2026). PUBLIC SENTIMENT ANALYSIS OF THE ‘NEGLIGENT’ STATEMENT ON TRANS 7 PESANTREN BROADCAST USING SUPPORT VECTOR MACHINE. *Jurnal Informatika Dan Teknik Elektro Terapan*, *14*(1).