CENTIVE
Conference on Electrical Engineering, Informatics,
Industrial Technology, and Creative Media
ISBN: 978-602-53004-0-0

# Sentiment Classification of FatSecret Application Reviews with Machine Learning Models

Mayang Gumelar[1], Farid Fitriyadi[2*]
[1,2] Department of Informatics, Sahid University Surakarta
[1]Jl.Adi Sucipto No.154, Jajar, Surakarta 57144 Indonesia
*Email corresponding : farid@usahidsolo.ac.id

Abstract — In the current digital era, mobile applications have become an indispensable part of daily life, leading to a surge in user reviews as invaluable repositories of opinions. Health and fitness applications, such as FatSecret, generate millions of reviews rich with insights. However, specific sentiment analysis on FatSecret reviews using a structured Machine Learning (ML) approach remains limited. This study presents a comprehensive approach for sentiment classification of FatSecret application reviews using ML models. We collected Indonesian-language reviews from the Google Play Store, performed extensive data pre-processing (case folding, tokenization, filtering, normalization), and extracted features using Term Frequency-Inverse Document Frequency (TF-IDF) and Bag of Words (BoW). Subsequently, we trained and evaluated five distinct sentiment classification algorithms: Random Forest, Decision Tree, Logistic Regression, SVM, and XGBoost, utilizing the StratifiedKFold method for automatic splitting in training and validation. Evaluation metrics include accuracy, precision, recall, and F1-score. The results of this research are expected to provide deep insights into user perceptions of FatSecret, identify favored and criticized features, and offer a replicable methodological framework for sentiment analysis of other applications in the future.
Keywords – Sentiment Analysis, FatSecret, Machine Learning, Application Reviews, Text Classification.

## I. INTRODUCTION

In today's digital age, mobile applications have become an inseparable part of daily life, encompassing various fields from communication and entertainment to health and fitness. The increasing adoption of smartphones has driven a surge in the number of available applications, and concurrently, the volume of user reviews left on app stores (such as Google Play Store and Apple App Store) has also drastically increased [1]. These reviews constitute a remarkably rich repository of opinions, experiences, suggestions, and even complaints from genuine users. Collectively, these millions of reviews represent an authentic and invaluable "voice of the customer" for app developers, service providers, and researchers alike. Manually analyzing this large-scale textual data is an inefficient and error-prone task, thereby driving the need for automated and systematic approaches to extract insights.

One of the fastest-growing mobile application fields is the health and fitness category, including diet and nutrition tracking apps [2]. Such applications help users monitor food intake, calories, and physical activity to achieve personal health goals. FatSecret stands out as a leading application in this category, used by millions worldwide to track diet, calories, and seek nutritional information [3]. FatSecret's popularity is reflected in the abundance of reviews available on the Google Play Store, which highlight diverse user experiences ranging from satisfaction with accurate tracking features and ease of use, to criticisms regarding bugs or incomplete features [4]. In-depth analysis of these reviews can provide crucial understanding of what works well and what needs improvement from the user's perspective.

Despite the vast volume of user reviews and their potential to offer profound insights, most of these reviews remain largely untapped [5]. While sentiment analysis has been widely applied across various domains, specific research focusing on the sentiment classification of FatSecret application reviews, particularly with a structured Machine Learning approach to automatically identify positive and negative opinions, is still limited [6]. Many studies tend to focus on applications with a more general scope or employ methods that may not fully capture the nuances of language and specific terminology used by health and fitness app users [7]. This gap includes the lack of models specifically validated for the context of diet tracking app reviews, as well as the potential to extract more detailed sentiment patterns that can be interpreted into actionable recommendations for developers [8].

In this study, we present a comprehensive approach for the sentiment classification of FatSecret application reviews using Machine Learning models. We will collect user review data from the Google Play Store, perform extensive text pre-processing, and train and evaluate several Machine Learning algorithms to accurately categorize review sentiments into positive, negative, or neutral [9]. The expected outcomes of this research include: (1) providing a deeper understanding of FatSecret user perceptions and experiences from a large volume of review data, (2) identifying the features most favored or complained about by users, and (3) offering a methodological framework that can be replicated for sentiment analysis of other applications in the future.

## II. RESEARCH METHOD

This research proposes a structured methodology for sentiment classification of FatSecret application reviews using a Machine Learning approach, designed to ensure the validity of findings and ease of replication. The research process is divided into five main stages: review data acquisition, data pre-processing, feature extraction, Machine Learning model development and training, and model evaluation.


Fig1. Research Flow Diagram

The review data acquisition phase involved programmatically collecting FatSecret application reviews from the Google Play Store over a specific period. The data collected specifically included Indonesian-language reviews, encompassing review text, star ratings (1-5), date, and username.

To establish the ground truth for model training, star ratings were used as the basis for initial sentiment labeling—ratings 1-2 as negative, 4-5 as positive, and 3 as neutral.

Tabel 1. Initial Labeling (Rating)

| Rating | Sentimen | Presentase |
|---|---|---|
| 4-5 | Positive | 94% |
| 3 | Neutral | 3% |
| 1-2 | Negative | 3% |

The initial sentiment labeling results show percentages of 92% (5,138) for positive labels, 3% (154) for neutral, and 3% (168) for negative. The next stage is data pre-processing. Data pre-processing is carried out to clean and standardize the review text, which is a crucial step to improve the performance of Machine Learning models. This process includes text cleaning, case folding, slang handling, tokenization, and stopwords removal..

First, text cleaning aims to filter reviews from unwanted noise such as mentions, hashtags, links, numbers, characters other than letters and numbers, punctuation, emoticons, and excessive spaces. Regex (Regular Expression) is used in this stage as a function that defines character patterns to be searched for and filtered from the text. Regex has proven effective in reducing lexical noise without eliminating the main semantic context in the review text.

Second, case folding aims to standardize the letter format in reviews so that words are not treated differently due to variations in capitalization[10]. *The case folding method used is lowercase folding.*

$$f(x) = lower(x) \quad (1)$$

With:
$x$ = review string
$f(x)$ = resulting string with all letters converted to lowercase.

Third, handling slang aims to normalize non-standard (slang) words into their standard forms according to the Great Dictionary of the Indonesian Language (KBBI). Words such as "bgt" (very), "nggk" (no), "maks" (max) "mslh" (problem) are transformed into "banget" (very), "tidak" (no/not), "maksimal" (maximal), and "masalah" (problem). The implementation of slang handling is done by creating a Python dictionary that contains slang words and their standard synonyms. The use of a slang dictionary helps to reduce linguistic variations that might not be recognized by the model.

$$f(w_i) = \begin{cases} \text{dictionary}[w_i], & \text{if } w_i \in \text{dictionary slang} \\ w_i, & \text{others} \end{cases} \quad (2)$$

With :
$w_i$ = token-$i$
dictionary = (dict Python)
$f(w_i)$ = normalized result

Fourth, tokenization is used to break down the review text into individual word units (tokens) that represent the smallest semantic units. Tokenization will produce a list of words that are then relevant for processing in the feature extraction stage. The resulting text can then have its frequency counted, vectors formed for each word, or sentiment features extracted. Tokenization can be represented as a function:

$$T = tokenize(x) \quad (3)$$

With:
x = text of previous preprocessing results (string)
$T = [w_1, w_2, w_3, \ldots, w_n]$ = list of generated tokens.

Fifth, stopwords removal aims to eliminate common words (such as conjunctions) such as “dan”

ISBN : 978-602-53004-0-0


(and), “yang” (which), “di” (at), “ke” (to), “ini” (it), and “sih”. These kinds of words often appear in reviews but add little to the actual sentiment meaning. Therefore, filtering is needed to prevent stopwords from dominating the representation vector and carrying irrelevant semantic information for classification. The stopword removal process can be expressed through the following function:

$$f(w_i) = \begin{cases} w_i, & if\ w_i \notin S \\ delete, & if\ w_i \in S \end{cases} \tag{4}$$

With :

$w_i$ = tokenized words

$S$ = set of stopwords

$f(w_i)$ = set of words without stopwords

After the stopword removal process, the word lists are recombined to form a complete review sentence. This merging utilizes Python's .join function.

| content | text_clean | text_caseFolding | text_slangWords | text_tokenizing | text_stopwords | text_akhir |
|---|---|---|---|---|---|---|
| sangat membantu mengontrol makanan harian | sangat membantu mengontrol makanan harian | sangat membantu mengontrol makanan harian | sangat membantu mengontrol makanan harian | [sangat, membantu, mengontrol, makanan, harian] | [membantu, mengontrol, makanan, harian] | membantu mengontrol makanan harian |
| Sangat membantu...! | Sangat membantu | sangat membantu | sangat membantu | [sangat, membantu] | [membantu] | membantu |
| sangat membantu | sangat membantu | sangat membantu | sangat membantu | [sangat, membantu] | [membantu] | membantu |
| Terima kasih, sangat membantu | Terima kasih sangat membantu | terima kasih sangat membantu | terima kasih sangat membantu | [terima, kasih, sangat, membantu] | [terima, kasih, membantu] | terima kasih membantu |

Fig 2. Preprocessing Results

*Text labeling is the final stage of sentiment text pre-processing*[12]. This process serves to assign each word (string) a label that represents a positive, neutral, or negative value. The results of this text labeling will be the final reference for user sentiment towards the FatSecret application. This is done by creating a dictionary that contains collections of words for each positive and negative sentiment. For example, "membantu" (helpful), "keren" (cool), "bagus" (good), "cocok" (suitable), "cepat" (fast) for positive sentiment, and "bug", "gagal" (failed), "lama" (slow/long), "ribet" (complicated), "jelek" (bad) for negative sentiment.

$$Label(t_i) = \begin{cases} 2, & if\ t_i \cap L_{pos} \neq \emptyset \\ 0, & if\ t_i \cap L_{neg} \neq \emptyset \\ 1, & neutral \end{cases} \tag{5}$$

With :

$t_i$ = review text to -i

$L_{pos}$ = set of words positive

$L_{neg}$ = set of words negative.

Feature extraction transforms preprocessed review text into a numerical representation that can be interpreted by a machine learning model. In this study, we explored two main approaches: Term Frequency-Inverse Document Frequency (TF-IDF), which measures the importance of a word within the context of the entire corpus.

$$\text{TF-IDF}\ (t, d) = \text{TF}\ (t, d) \times \text{IDF}\ (t) \tag{6}$$

TF ($t$, $d$) is the relative frequency of word t in document d.

$$\text{TF}\ (t, d) = \frac{f_{t,d}}{\sum_k f_{k,d}} \tag{7}$$

IDF($t$) is the scarcity of the word t throughout the document.

$$\text{IDF}\ (t) = \log\left(\frac{N}{df_t}\right) \tag{8}$$

With :

$N$ = Number of documents

$df_t$ = Number of documents containing the word $t$

The second approach, Bag of Words (BoW), maps each word's vector based on the frequency of word occurrence (term frequency) without regard to word order.

$$v_{i,j} = f_{i,j} \tag{9}$$

With :

$v_{i,j}$ = the value of the jth word feature in the i-th document

$f_{i,j}$ = frequency of occurrence of word j in the document i

Theoretically, TF-IDF has an advantage in balancing common words and rarely appearing emotional keywords[11]. However, BoW (Bag of Words) excels in simpler models with very limited data. Therefore, these two approaches will be tested using a k-fold cross-validation scheme to assess the stability of the technique's performance and the models used across various random data split.

Before entering the training phase, the process of data splitting must be performed. This process divides the dataset into two parts: training data and testing data.

Instead of using the train_test_split method, we employed a more accurate and flexible method, namely K-Fold Cross-Validation[13]. Its primary goal is to evaluate model performance diversely, rather than based on a single split of training and testing data. The dataset was divided based on k = 5 folds.

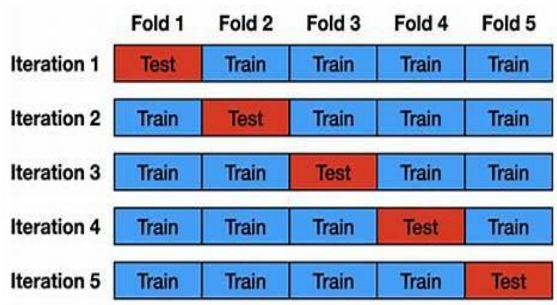


Fig 3. Split Data in 5-Fold

Machine Learning model development and training. In this stage, we utilized five Machine Learning models, each of which will be integrated with the two previously prepared feature extraction techniques. Thus, a total of 10 model training schemes will be conducted. The chosen models are Random Forest, Decision Tree, Logistic Regression, Support Vector Machine (SVM), and XGBoost.

ISBN : 978-602-53004-0-0


**Model evaluation** is performed on testing data that has not been used during training, to objectively assess sentiment classification performance. The evaluation metrics to be used include accuracy, precision, recall, and F1-score, which will be calculated for each sentiment class (positive, negative, neutral) as well as the overall model performance. In the evaluation function, the cross-validation parameter used is StratifiedKFold() to maximize handling of imbalanced classes. If not addressed, this imbalance would pose a significant risk of overfitting. Overfitting is a condition where a model's accuracy performance is too high during training but weak during testing [14].

## III. RESULT

```
print(df_cleaned['Sentiment'].value_counts())
```

```
Sentiment
2    3623
1    1721
0     116
Name: count, dtype: int64
```

Fig 4. Review Labeling Results

Based on the text labeling process, the class distribution results are as follows :

Table 2. Text Labeling Result

| Sentimen | Amount |
|---|---|
| 2 / Positive | 3623 |
| 1 / Neutral | 1721 |
| 0 / Negative | 116 |

To find out the distribution of each sentiment class, visualization was carried out using the following scatterplot :

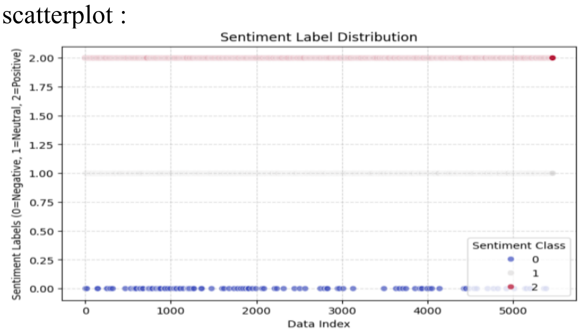


Fig 5. Label Distribution

With blue (0) = negative, gray (1) = neutral, and red (2) = positive. Red dots (class 2, positive) appear to dominate and are evenly distributed across the data index. Gray dots (class 1, neutral) appear randomly throughout the data index but are fewer than those in class 1. Blue dots (class 0, negative) appear least frequently throughout the data index and are also unevenly distributed. This indicates that the labeling process is not random and does not follow a specific sequence pattern, so the dataset does not contain bias in time or is fixated on index positions. indeks[15]. It can be concluded that positive reviews dominate the total sentiment in the FatSecret application.

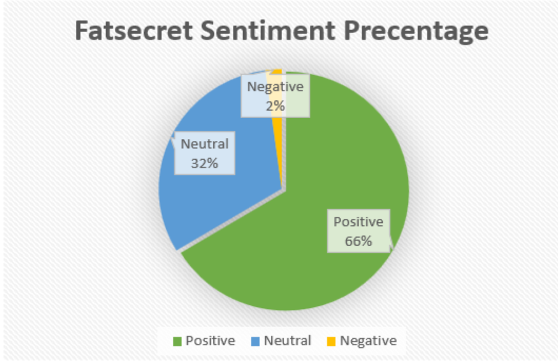


Fig 6. Percentage of Sentiment Classes

Looking at the results from text labeling, it is evident that the classes are unevenly distributed (66%:32%:2%). This is referred to as an **imbalanced dataset**. Therefore, in the splitting stage, the K-Fold Cross Validation method, specifically StratifiedKFold, is used. This also serves as one of the parameters for the evaluation function to handle the imbalanced data effectively. The divided data will then undergo feature extraction and proceed to the model training phase. Model training is performed using the 10 training schemes that have been established.

Table 3. Training Scheme

| Random Forest | Decision Tree | Logistic Regression | SVM | XGboost |
|---|---|---|---|---|
| TF-IDF | TF-IDF | TF-IDF | TF-IDF | TF-IDF |
| BoW | BoW | BoW | BoW | BoW |
| 2 | 2 | 2 | 2 | 2 |
| **agregat** | | | **10 Scheme** | |

From the training results of the 10 schemes, the following accuracy was produced:

Table 4. Model Scheme with TF-IDF

| | RF | DT | LR | SVM | XGB |
|---|---|---|---|---|---|
| **Accuracy** | 0.92 | 0.90 | 0.89 | 0.91 | 0.93 |
| ***Precision*** | 0.93 | 0.92 | 0.93 | 0.92 | 0.93 |
| ***Recall*** | 0.92 | 0.90 | 0.89 | 0.91 | 0.93 |
| **F1-Score** | 0.92 | 0.91 | 0.91 | 0.92 | 0.93 |

All algorithms demonstrated high performance with testing accuracy above 88%. The XGBoost model provided the best results with 94% accuracy, 94% precision indicating a good model in predicting classes, 94% recall indicating the model correctly guessed the class in 94% of all existing samples, and an f1 score of 94% indicating a balance between precision and recall. Meanwhile, Random Forest and SVM also demonstrated stable performance with consistent precision and recall values. Both models were able to detect three sentiment classes equally. Overall, the results prove the TF-IDF approach is effective in capturing language characteristics in FatSecret app reviews.

ISBN : 978-602-53004-0-0



Table 5. Model Scheme with BoW

| | RF | DT | LR | SVM | XGB |
|---|---|---|---|---|---|
| **Accuracy** | 0.95 | 0.97 | 0.96 | 0.93 | 0.97 |
| ***Precision*** | 0.95 | 0.97 | 0.96 | 0.93 | 0.97 |
| ***Recall*** | 0.95 | 0.97 | 0.96 | 0.93 | 0.97 |
| **F1-Score** | 0.95 | 0.97 | 0.96 | 0.93 | 0.97 |

After training and testing the data with the BoW approach, it was shown that the BoW approach produced higher accuracy (>92%) across all models. The Decision Tree and XGBoost models achieved the highest accuracy of 97%, followed by Logistic Regression at 96%, Random Forest at 95%, and finally SVM at 93%. BoW provided higher performance because the approach was applied to review data with sentences that tend to be short and use common, high-frequency words. BoW is more effective in representing word distribution patterns compared to the TF-IDF approach, which tends to downweight common words. Thus, the BoW approach proved to be more suitable for FatSecret app user review data, with the best models being Decision Tree and XGBoost.

WordCloud visualization to show a consistent visual representation of sentiment classification results:


Fig 7. Positive Sentiment WordCloud

The positive sentiment WordCloud shows dominant words such as “membantu” (helpful), “bagus” (good), “bermanfaat” (useful), “diet”, and “mantap” (great). [16] These are the words that users use most to give positive reviews.


Fig 8. Negatif Sentiment WordCloud

The negative sentiment WordCloud shows a predominance of negative words like "error," "loading," "gagal" (failed), "bug," and "bermasalah" (complicated). These are words users frequently use to express negative reviews.


Fig 9. Neutral Sentiment WordCloud

The neutral sentiment word cloud shows a predominance of neutral words such as "application," "diet," "calories," "food," "app," and other words representing neutral emotions.

Inference is performed after the model is trained for live testing to determine whether it performs well.

```
test = [
    "Jelek, loading lama. Burik",
    "Mantap, aplikasinya sangat membantu",
    "bug kebanyakan",
    "Saya nggak bisa login, padahal udah pake lama. Loading terus",
    "Aplikasi penurunan bb"
]

X_new = bow.transform(test)
prediction = dt_bow_model.predict(X_new)

for komentar, label in zip(test, prediction):
    print(f"Komentar: {komentar}\nSentimen: {label}\n")
```

```
Komentar: Jelek, loading lama. Burik
Sentimen: 0

Komentar: Mantap, aplikasinya sangat membantu
Sentimen: 2

Komentar: bug kebanyakan
Sentimen: 0

Komentar: Saya nggak bisa login, padahal udah pake lama. Loading terus
Sentimen: 0

Komentar: Aplikasi penurunan bb
Sentimen: 1
```

Fig 10. Inference Results

Pada uji coba inference menunjukkan bahwa model dapat mengklasifikasikan sentimen dengan sangat baik.

## IV. DISCUSSION.

### A. *Interpretation of Main Results*

The results of this study indicate that in the case of sentiment analysis and classification of FatSecret application reviews, the best performance was achieved using the Bag of Words (BoW) approach compared to the Term Frequency-Inverse Document Frequency (TF-IDF) approach. In the schemes utilizing the BoW approach, Decision Tree and XGBoost models achieved the highest accuracy of 97%, with consistent precision, recall, and F1-score values also at 97%. This performance improvement suggests that a simple frequency-based word representation is more effective at capturing sentiment patterns in review texts, which tend to be short. Meanwhile, the TF-IDF approach assigns low weights to high-frequency words, which actually weakens the sentiment signal because most important keywords are used across multiple reviews.

### *B, Effectiveness of Tree-Based Models*

From the results of testing five classification models (Random Forest, Decision Tree, Logistic Regression, SVM, and XGBoost), two ensemble-based (decision tree) models, Decision Tree and XGBoost, provided the best results for FatSecret app sentiment classification. Both proved to be the most adaptive to the BoW approach. This is due to several factors :

a) The BoW feature structure is discrete, making it easy to divide into nodes and split rules in the ensemble base.
b) XGBoost uses gradient boosting optimization, which has the advantage of iteratively minimizing classification errors and making it more sensitive to word patterns that contribute significantly to the review.
c) Decision Trees offer high interpretability, allowing them to explore key words that are key to the model's decision regarding a particular label.
d) Although Decision Tree is an ensemble base algorithm, it is the lightest (basic) tree model compared to Random Forest or XGBoost, making it compatible with various devices with varying specifications.

### *C. Relationship Analysis*

The results of this study show that the BoW approach, integrated with Decision Tree and XGBoost models, is capable of delivering high performance with accuracy, precision, recall, and F1-score all at 97%. When compared to the research by Putra, Kurniawan Tri, et al. (2023) in the Jurnal Cahaya Mandalika (JCM) titled "Comparison of TF-IDF and BoW Feature Extraction for SVM-based Sentiment Analysis," the combination of TF-IDF with SVM achieved an accuracy of 86%, while BoW with SVM reached 85%, excelling in recall (89%). Those researchers stated that TF-IDF was superior overall. However, the results of this study show a significant difference. BoW actually yielded higher performance, especially when integrated with ensemble-based models such as Decision Tree and XGBoost.

This difference is likely influenced by factors such as variations in classification algorithms, the specific characteristics of FatSecret application review data, and more optimal text pre-processing stages. Thus, this study confirms that the effectiveness of Bag of Words (BoW) is not static and heavily depends on the type of model used, and can even surpass the performance provided by TF-IDF.

### *D. Practical Implications*

The findings of this study have important practical implications for both FatSecret application developers and researchers in the field of Natural Language Processing (NLP), namely:

a) For application developers, BoW-based models integrated with Decision Tree can be applied for real-time sentiment monitoring due to their lightweight nature and high accuracy.
b) For researchers, these findings demonstrate that simple text representations are still highly relevant for applications with short Indonesian-language sentiment reviews, without requiring more resource-intensive embedding approaches.

In addition to the two points above, these findings also open up opportunities for developing sentiment-based recommendation systems. The classification results can be used to enhance user experience.

### *E. Research Limitations and Recommendations*

Although this study demonstrates a relatively high model performance, it still has several limitations:

a) An imbalanced sentiment class distribution, with positive sentiment having the highest percentage (66%), raising concerns that the model might be biased towards positively labeled sentiments.
b) Semantic context between words is not accounted for, as BoW relies solely on the frequency of each word's appearance.
c) The dataset is limited to Indonesian-language sentiment with many slang words, which could affect the model's generalization process.

Based on the three points above, future research could consider integrating a hybrid approach between BoW and embeddings like Word2Vec or FastText to capture semantic relationship patterns between words..

## V. CONCLUSSION

This study successfully presents a comprehensive framework for sentiment classification of FatSecret application reviews using a Machine Learning approach. Based on the analysis of review data collected from the Google Play Store, it was found that the majority of users consistently gave the highest review score (Score 5) from 2020 to 2025, indicating a very high level of satisfaction with the application. This pattern is clearly visible from the frequency distribution of review scores, the boxplot of review spread, and the heatmap of average monthly scores, which are dominated by high values. Systematic data pre-processing stages and feature extraction using TF-IDF and BoW have been prepared to support the development of accurate classification models. By evaluating Random Forest, Decision Tree, Logistic Regression, SVM, and XGBoost algorithms, this research aims not only to identify the most effective model in categorizing positive, negative, and neutral sentiments but also to provide deep insights into user perceptions. It is hoped that the results of this study can serve as a valuable foundation for FatSecret developers to improve features and services, and offer a replicable methodology for sentiment analysis of application reviews in various domains.

## REFERENCES

[1] A. Okta, K. Adi, F. Prayoganing Gusti, and F. Wijaya, "Analisis Sentimen Ulasan Pengguna Aplikasi Mobile Legends Pada Google Playstore Menggunakan Naïve Bayes" 2025.

[2] K. Muhammad Ibrahim *et al.*, "Artikel Nusantara Computer and Design Review," *NCDR*, vol. 1, no. 1, pp. 31–39, 2023, [Online]. Available: https://journal.unusida.ac.id/index.php/ncdr/

[3] Z. D. Ulfa, J. A. Perdana, and A. T. Abeng, "Program ZITASI Generasi Z melalui Aplikasi Fatsecret pada Siswa SMA Sebagai Agen Perubahan di Daerah Aliran Sungai Sabangau," *Jurnal Pengabdian Al-Ikhlas*, vol. 9, no. 2, Dec. 2023, doi: 10.31602/jpaiuniska.v9i2.9003.

[4] A. F. Idham, A. Putrawan, F. Athaya, F. Ramadhan, and A. P. Rasyid, "Evaluation of the Use of Structural Gamification-Based Applications by Users in Makassar City, Indonesia," *Jurnal Keperawatan Indonesia*, vol. 28, no. 1, pp. 12–21, Mar. 2025, doi: 10.7454/jki.v28i1.1266.

[5] M. Arya Java, M. Syafrullah, and F. Teknologi, "Analisis Sentimen Ulasan Pengguna Aplikasi Threads pada Google Play Store Menggunakan Multinomial Naive Bayes dan Support Vector," *Jurnal TICOM: Technology of Information and Communication*, vol. 12, no. 2, 2024, [Online]. Available: https://github.com/nasalsabila/kamus-alay

[6] J. Khatib Sulaiman, F. Junita Fauzan, M. Afdal, R. Novita, and U. Islam Negeri Sultan Syarif Kasim Riau, "Penerapan Machine Learning Pada Analisis Sentimen Aplikasi MyTelkomsel Menggunakan Data Ulasan Google," *Indonesian Journal of Computer Science*.

[7] B. Z. Ramadhan, I. Riza, and I. Maulana, "Analisis Sentimen Ulasan Pada Aplikasi E-Commerce Dengan Menggunakan Algoritma Naïve Bayes," 2022. [Online]. Available: http://jurnal.polibatam.ac.id/index.php/JAIC

[8] M. Lukman Hakim and A. Hermawan, "Pengembangan Aplikasi e-Diet Berbasis Android untuk Meningkatkan Pemahaman Nutrisi Bagi Masyarakat".

[9] A. C. T Angel and V. H. Pranatawijaya, "Analisis Sentimen dan Emosi dari Ulasan Google Maps untuk Layanan Rumah Sakit di Palangka Raya Menggunakan Machine Learning," 2024.

[10] R. Rahmadani, A. Rahim, and R. Rudiman, "Analisis Sentimen Ulasan 'Ojol The Game' Di Google Play Store Menggunakan Algoritma Naive Bayes Dan Model Ekstraksi Fitur Tf-Idf Untuk Meningkatkan Kualitas Game," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3, Aug. 2024, doi: 10.23960/jitet.v12i3.4988.

[11] D. Septiani and I. Isabela, "SINTESIA: Jurnal Sistem dan Teknologi Informasi Indonesia Analisis Term Frequency Inverse Document Frequency (Tf-Idf) Dalam Temu Kembali Informasi Pada Dokumen Teks".

[12] M. Fernanda and N. Fathoni, "Perbandingan Performa Labeling Lexicon InSet dan VADER pada Analisa Sentimen Rohingya di Aplikasi X dengan," *Jurnal Informatika dan Sains Teknologi*, vol. 1, no. 3, pp. 62–76, 2024, doi: 10.62951/modem.v1i3.112.

[13] B. U. Ca and Y. G. Fr, "No Unbiased Estimator of the Variance of K-Fold Cross-Validation Yoshua Bengio Yves Grandvalet," 2004.

[14] P. L. Bartlett, P. M. Long, G. ´ Abor Lugosi, and A. Tsigler, "Benign overfitting in linear regression," *PNAS*, vol. 117, 2020, doi: 10.1073/pnas.1907378117/-/DCSupplemental.y.

[15] Y. Primasanti, F. Fitriyadi, and E. H. Lukitasari, "The Role of the Internet of Things (IoT) in Improving the Efficiency and Quality of Bus Transportation Services," 2025. [Online]. Available: https://prosiding.utp.ac.id/index.php/ICEETE

[16] F. Fitriyadi and A. Astikasari, "Analisis Sentimen Masyarakat Terhadap Kebijakan Kenaikan Umk 6,5% Menggunakan Metode Naive Bayes," *Jurnal Riset Sistem dan Teknologi Informasi (RESTIA*, vol. 3, no. 1, pp. 26–35, 2024.