Vol. 6 No. 1, 2025, pp. 47- 56

DOI: 10.23960/pepadun.v6i1.262

Jurnal Pepadun

Sentiment Analysis on Jobseekers Application in Google Play Store (KitaLulus)

[1]Muhamad Arrozi Irfan, *[2]Dewi Asiah Shofiana, [3]Rahman Taufik and [4]Admi Syarif

[1,2,3,4]Department of Computer Science, University of Lampung,
Jl. Prof. Dr. Ir. Sumatri Brojonegoro No.1, Bandar Lampung, Lampung Province, Indonesia, 35145.
e-mail:[1]oziirfan10@gmail.com, *[2]**dewi.asiah@fmipa.unila.ac.id**, [3]rahman.taufik@fmipa.unila.ac.id,
[4]admi.syarif@fmipa.unila.ac.id

Abstract *- Sentiment analysis on user reviews of the KitaLulus application in Google Play Store aims to assess user feedback and classify sentiment into positive and negative categories. This study applies and compares the performance of SVM and Naïve Bayes classifiers in sentiment classification. Data was collected using the Google-Play-Scraper API, yielding 16488 reviews, which underwent preprocessing, including case folding, tokenization, stopword removal, and lemmatization. The dataset was divided into training (80%), validation (5-fold cross-validation), and testing (20%) sets. During validation, the training data was further split, using 64% for training and 16% for validation in each iteration. The results indicate that SVM outperforms Naïve Bayes, achieving 93.99% accuracy, 97% precision, 92% recall, and an F1-score of 94%, while Naïve Bayes achieves 89.89% accuracy, 94% precision, 87% recall, and an F1-score of 90%. These findings demonstrate that SVM provides a more balanced classification performance, making it a more suitable approach for sentiment analysis in this context. This research contributes to a better understanding of user sentiment and provides valuable insights for improving the KitaLulus application.*

Keywords: *Google Play Store; Jobseekers Application; Sentiment Analysis; Support Vector Machine; Naïve Bayes.*

1. INTRODUCTION

Smartphone use in Indonesia has grown tremendously over the past few years. By 2023, around 190 million people in Indonesia own smartphones, which is about 68% of the population. This makes Indonesia the fourth largest smartphone market after China, India, and the United States [1][2]. Android is the most popular operating system in these regions since it supports a wide range of apps, including those that help people find jobs [3][4]. These developments have led to the rise of digital platforms that facilitate job seeking, improving accessibility and efficiency in the labor market.

Even though the unemployment rate in Indonesia has dropped from 6.26% in 2021 to 5.32% in 2023 (according to the Central Statistics Agency), many job seekers still struggle to find the right opportunities. Problems like unclear job information, slow application processes, and mixed user experiences often make it harder for users [5]. One popular platform trying to solve these issues is KitaLulus. Since launching in 2020, KitaLulus has helped millions of users apply for jobs and partnered with tens of thousands of companies, gaining over five million downloads across more than 60 cities in Indonesia [6]. Evaluating user sentiment through their reviews on platforms such as Google Play Store is a valuable approach to understanding user experiences and identifying areas for improvement.

Sentiment analysis has become a common method to capture user opinions and feedback from text data. Machine learning models such as Support Vector Machine (SVM) and Naïve Bayes have been widely applied due to their proven effectiveness in text classification tasks. For example, Naïve Bayes was used to analyze reviews of the KAI Access app and reached 89% accuracy [7]. Meanwhile, SVM combined with Term Frequency-Inverse Document Frequency (TF-IDF) features achieved around 85% accuracy when analyzing the by.U app reviews [8]. These successes show that machine learning can help make sense of large amounts of user feedback.

However, existing studies often focus on specific applications or employ limited datasets, which restricts the generalizability of their findings. Moreover, few have explored the comparative performance of SVM and Naïve Bayes algorithms specifically on large-scale datasets from job search applications like KitaLulus, which operate in a dynamic and diverse Indonesian labor market context. Also, challenges like imbalanced data, where positive reviews far outnumber negative ones, haven't been fully addressed in many studies. Techniques like Synthetic Minority Over-sampling Technique (SMOTE), which help balance the data, are often missing but are important for improving model accuracy. These gaps underline the need for a comprehensive sentiment analysis that rigorously compares classification algorithms on extensive real-world data from popular employment apps, thereby providing actionable insights for developers and stakeholders.

This study aims to close these gaps by analyzing a large collection of user reviews from the KitaLulus app using both SVM and Naïve Bayes classifiers. It employs both Naïve Bayes and SVM classifiers, enhanced by advanced preprocessing and data balancing techniques, to assess and compare their effectiveness in sentiment classification. What makes this research stand out is the scale of the dataset, the use of data balancing techniques, and the direct comparison of two popular algorithms in a real Indonesian job search environment. The findings are expected to contribute significant insights for improving user experience and guiding future developments in digital employment platforms.

2. RESEARCH METHODOLOGY

In this study, there are several main stages carried out to analyze the sentiment of KitaLulus application user reviews. Each stage plays an important role in ensuring the accuracy of the analysis results. Figure 1 shows the research workflow.

2.1. Literature Studies

This research begins with a literature study from journals, books, and related articles. Furthermore, relevant theories were developed, including definitions, formulations, and advantages and disadvantages of SVM and Naive Bayes [9].

2.2. Data Collection

Google-Play-Scraper is a Python API for retrieving app data and reviews from the Google Play Store without external dependencies. The data includes app information (title, developer, rating, description, etc.) as well as user reviews (name, score, version, date, and comments) [10]. The dataset contains 16,488 app reviews with variable name, score (1–5), date, comment, and sentiment, collected from December 20, 2020 to December 22, 2023.

2.3. Data Preprocessing

The third stage is data preprocessing to compile structured data to facilitate analysis [11]. The process includes Cleaning, Case Folding, Tokenizing, Stopword Removal, and Stemming.

2.4. Manual Labeling

After preprocessing, 12617 data were left manually labeled because Google Play reviews did not have built-in sentiment. The labeling process involved three people to ensure no ambiguity, in which resulted in 9862 positive reviews (78%) and 2755 negative reviews (22%).

Vol. 6 No. 1, 2025, pp. 47- 56 **Jurnal Pepadun**

DOI: 10.23960/pepadun.v6i1.262

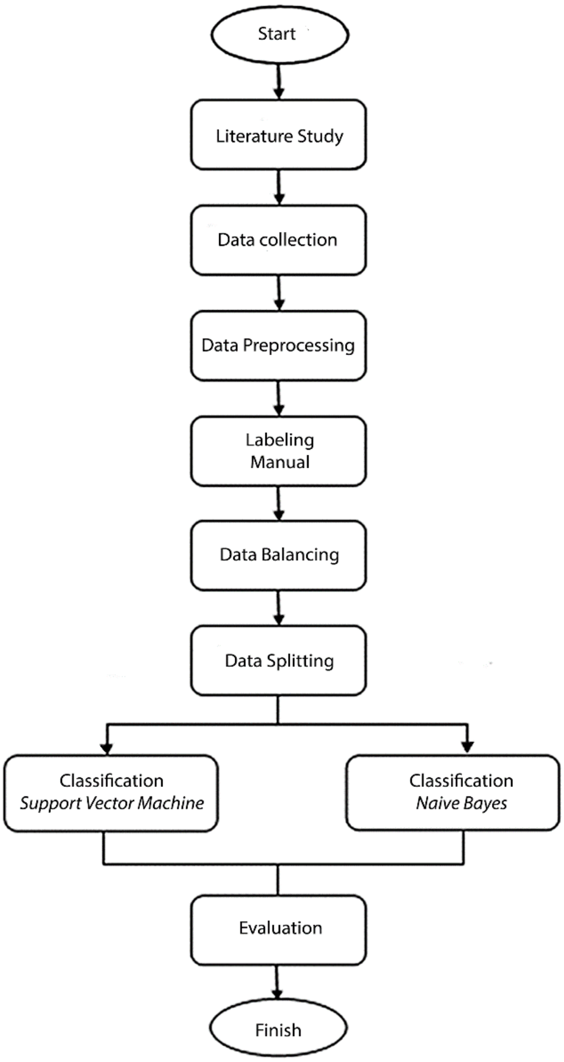

Figure 1. Research workflow.

2.5. Data Balancing (SMOTE)

SMOTE (Synthetic Minority Over-sampling Technique) addresses data imbalances by adding synthetic samples for minority groups. This method selects a minority sample, identifies the nearest neighbors, and then creates a new sample through interpolation between the original sample and its neighbors [12][13]. The dataset contained 2755 bad comments and 9862 good comments, causing a class imbalance that risked classification bias. To overcome this, SMOTE is used to increase the sample of minority classes.

Table 1. Comparison of data amount before and after SMOTE.

SMOTE	Sentiment	Total Data
Before	Negative	2755
	Positive	9862
After	Negative	7252
	Positive	8243

2.6. Data Sharing

Cross-validation is one of the methods resampling the most commonly used data in model selection and evaluation. This technique aims to assess the generalization ability of a predictive model and prevent overfitting [14].

Vol. 6 No. 1, 2025, pp. 47- 56 **Jurnal Pepadun**

DOI: 10.23960/pepadun.v6i1.262

The data was divided 80% for practice and 20% for testing. On the training data, a 5-Fold Cross-Validation is performed, where each iteration uses 64% for training and 16% for validation, ensuring the model is thoroughly tested before being applied to the test data.

2.7. Classification of SVM and Naïve Bayes

The sentiment analysis stage was carried out using Naïve Bayes and the SVM. Naïve Bayes formed a statistical model based on Bayes' theorem, with Naïve Bayes' Multinomial as the algorithm used [15][16]. SVM separates classes using hyperplanes, with linear kernels in this study [17]. Both methods follow the same stages: data preprocessing, the formation of models with certain parameters, and evaluation using classification metrics, including confusion matrix.

3. RESULT AND DISCUSSION

The data needed for this sentiment analysis is in the form of review text taken from reviews of the KitaLulus application on the Google Play Store. These reviews come from users who have installed the app and left comments regarding their experience. Comments given are generally related to the included rating, on a scale of 1 to 5, where rating 1 indicates negative sentiment and rating 5 indicates positive sentiment. In this study, the data collected included usernames, ratings, review dates, and comment content as the main variables in the analysis. The following is a review of the overall data entered with a value of 1 to 5 from the smallest to the largest presented in Figure 2.

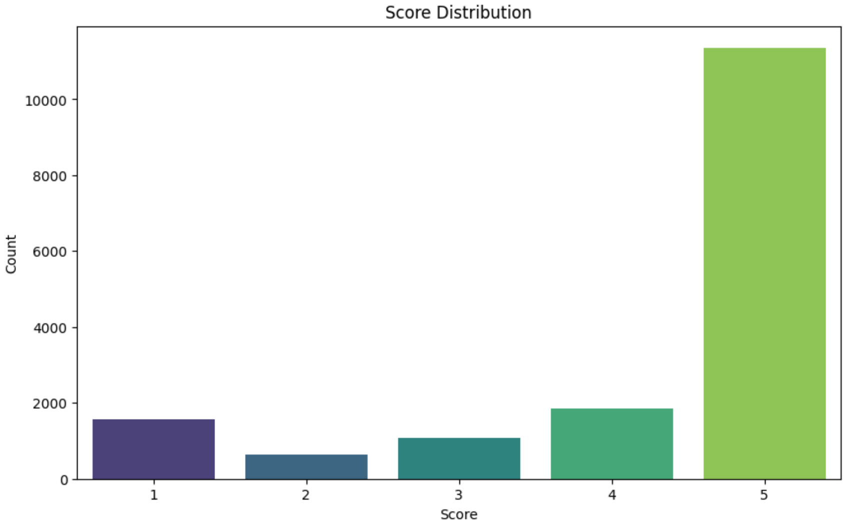

Figure 2. Score from the reviews given.

Wordcloud is a word visualization where the more frequency of data that appears, the greater the number of words that will be displayed in the visualization. Figure 3 is a word visualization that shows frequently appearing words in larger sizes. Based on the analysis of positive and negative sentiment in wordcloud, there are two main sides of the user experience towards this application or service.

On positive sentiment, the word "good" comes up most often, indicating general user satisfaction. Words like "searching", "job", "helping", "passing", and "hopefully" reflect that users find the app helpful in their job search and hope for success. In addition, the words "thank you" and "easy" show appreciation for the ease of use of the app. Users also find the app very useful, with words like "steady", "useful", and "update" that show the app's effectiveness in helping job searches. Words like "easy", " job vacancy ", "response", and "area" indicate that the app is easy to use and responsive, and helps with job searches in various locations.

Vol. 6 No. 1, 2025, pp. 47- 56

DOI: 10.23960/pepadun.v6i1.262

Figure 3. Wordcloud positive (left) and negative (right).

However, on the negative sentiment side, words such as "job vacancy" and "apply" often appear, indicating dissatisfaction related to inappropriate job openings and an application process that is considered inefficient. Additionally, words like "slow," "disconnected," and "cumbersome" reflect technical issues such as slow applications and access difficulties. Complaints about the lack of interview calls, slow responses from companies, as well as requests for app improvements are also seen in words such as "please", "fix", and "rebuild". Overall, this wordcloud gives the impression that while many users are satisfied with the app, there are also complaints related to technical issues and dissatisfaction with the job application process and the company's response.

3.1. Training Support Vector Machine

At this stage, classification training using SVM is carried out. For the SVM algorithm, it was done by comparing the tested SVM kernels, namely linear and Radial Basis Function using 5-fold. The parameters used for SVM are C and gamma with a specific range of values. The results of tests based on the SVM kernel can be seen in Table 2.

Table 2. The parameters used in SVM.

Parameter	Value
C	10
Kernel	RBF
Gamma	0.1

In Table 2, three parameters used in the SVM model are shown, where the parameters have been selected from the best parameters found through the hyperparameter tuning process. Here's an explanation of each parameter:

1. C: Parameter C in an SVM is a hyperparameter that controls the trade-off between maximizing the margin of the hyperplane and minimizing classification errors. C specifies how much penalty is imposed for classification errors in training data. In this case, parameter C has a value of 10, which indicates a fairly high penalty against classification errors. The model focuses more on minimizing classification errors than maintaining larger margins.

2. Kernel: looking for the best kernel that will be used as a kernel parameter obtained is the Radial Basis Function (RBF) kernel. This is because RBF can handle non-linear problems well by mapping data into higher-dimensional spaces.

3. Gamma: The gamma parameter controls how far a single training sample influences. Higher gamma values result in a greater influence and affect the separation distance between training samples. In this case, the gamma parameter has a value of 0.1, indicating a fairly significant influence.

The results of the classification training with SVM use the best parameters with a C value of 10 and a gamma of 1 which can be seen in the following Figure 4.

Vol. 6 No. 1, 2025, pp. 47- 56

DOI: 10.23960/pepadun.v6i1.262


```
Best Parameters: {'svc__C': 10, 'svc__gamma': 1, 'svc__kernel': 'rbf'}
Best Score: 0.9297349022108289
```

Figure 4. Results of hyperparameter SVM training.

3.2. Training Naïve Bayes

At this stage, classification training is carried out. Naïve Bayes which is done with the tuning parameter 'multinomialnb__alpha' using 5-fold. To apply hyperparameter tuning to the Naive Bayes model, the next step is to determine the parameters to be used. Table 3 shows the parameters to be applied in the Naive Bayes model.

Table 3. The parameters used in Bayes.

Parameter	**Value**
multinomialnb__alpha	1.0

'Alpha' is a parameter in the Multinomial Naive Bayes (MultinomialNB) model for smoothing, which prevents zero probability on the training data. An alpha value of 1.0 adds a constant value to each probability, keeping the probability positive and reducing the variability of the estimate.

When alpha is set to 1.0, a constant value is added to each probability estimate, ensuring that no probability becomes zero. This adjustment helps maintain numerical stability and allows the model to handle previously unseen words more effectively. Additionally, smoothing reduces the variability of probability estimates, leading to a more generalized model that avoids overfitting to specific training data patterns. The results of the classification training with Naïve Bayes used the best parameter 'multinomialnb__alpha' with a value of 1.0 which can be seen in Figure 5.

```
Best Parameters: {'multinomialnb__alpha': 1.0}
Best Score: 0.8899643131335477
```

Figure 5. Results of hyperparameter SVM training.

3.3. Evaluation Support Vector Machine

Furthermore, classification tests were carried out to calculate accuracy, precision, recall and f1-score. In the SVM algorithm, tests are carried out using 5-fold. The results of the classification calculation with SVM can be seen in Table 4, with the value of the SVM classification can also be seen in the form of a confusion matrix in the following Table 5.

Table 4. SVM accuracy result (per fold).

Fold	**Accuracy Values**
1	0.951
2	0.947
3	0.954
4	0.932
5	0.908
Average	0.938
Test set	0.939

Table 5. SVM confusion matrix.

Facts	**Predictions**	
	Negative	**Positive**
Negative	1360	42
Positive	144	1553

Vol. 6 No. 1, 2025, pp. 47- 56 **Jurnal Pepadun**

DOI: 10.23960/pepadun.v6i1.262

In this case, there were 1553 correct positive predictions (TP), 42 positive predictions that should be negative (FP), 1360 correct negative predictions (TN), and 144 negative predictions that should be positive (FN).

1. True Positives (TP): 1553 (Correct Positive Prediction).
2. False Positives (FP): 42 (Positive Predictions when they should be Negative).
3. True Negatives (TN): 1360 (Correct Negative Prediction).
4. False Negatives (FN): 144 (Negative Predictions when they should be Positive).

The results of the SVM classification using the best parameters with the values of C: 10, gamma: 1 and linear kernel, can be seen in Table 6.

Table 6. SVM performance evaluation.

Results of SVM Evaluation with Hypertuning	
Accuracy	0.94
Precision	0.97
Recall	0.92
F1-Score	0.94

Based on Table 6, it can be concluded that the results of classification calculation using the SVM algorithm using 5-fold produce an average accuracy of 93%, precision of 97%, general recall of 92% and f1-score of 94%.

3.4. Evaluation of Naïve Bayes

Furthermore, classification tests were carried out to calculate accuracy, precision, recall and f1-score. In the Naïve Bayes algorithm, the test was carried out using 5-fold validation with the parameter 'multinomialnb__alpha'. The results of the classification calculation with Naive Bayes using MultinomialNB can be seen in Table 7, with the value of the Naïve Bayes classification can also be seen in the form of a confusion matrix in the following Table 8.

Table 7. Naïve Bayes accuracy result (per fold).

Fold	**Accuracy value**
1	0.912
2	0.928
3	0.905
4	0.875
5	0.843
Average	0.893
Test set	0.898

Table 8. Naïve Bayes confusion matrix.

Facts	**Predictions**	
	Negative	**Positive**
Negative	1308	94
Positive	219	1478

In this case, there were 1478 correct positive predictions (TP), 219 positive predictions that should be negative (FP), 1308 correct negative predictions (TN), and 94 negative predictions that should be positive (FN).

1. True Positives (TP): 1478 (Correct Positive Prediction).
2. False Positives (FP): 219 (Positive Predictions when they should be Negative).
3. True Negatives (TN): 1360 (Correct Negative Prediction).
4. False Negatives (FN): 94 (Negative Predictions when they should be Positive).

DOI: 10.23960/pepadun.v6i1.262

The use of 5-fold cross-validation offers a more reliable and comprehensive evaluation of the Naïve Bayes model's performance. By dividing the dataset into five subsets and iteratively training and testing the model on different combinations of these subsets, this method reduces the potential variability that may arise from a single test with a single data split. This ensures that the model's performance is not overly dependent on a specific train-test division, leading to a more stable and consistent assessment.

Table 9. Naïve Bayes performance evaluation.

Naïve Bayes Performance Results with 5-fold	
Accuracy	0.90
Precision	0.94
Recall	0.87
F1-Score	0.90

Based on Table 9, it can be concluded that the classification results using the Naïve Bayes algorithm with 5-fold cross-validation demonstrate strong performance. The model achieves an average accuracy of 89%, indicating that the majority of predictions align with the actual sentiment labels. Additionally, the precision score of 94% suggests that when the model classifies a sentiment as positive or negative, it is correct in most cases. The general recall of 87% reflects the model’s ability to correctly identify relevant instances, ensuring that a significant portion of actual sentiments is captured. Lastly, the f1-score of 90% balances precision and recall, reinforcing the model’s effectiveness in sentiment classification. These results highlight the reliability of Naïve Bayes in analyzing user reviews with a high degree of accuracy

3.5. Comparison of SVM and Naïve Bayes

Table 10 shows a comparison of the test results between the SVM and Naïve Bayes algorithms based on the 5-fold test.

Table 10. Comparison of SVM and Naïve Bayes 5-fold results.

Algorithm	Accuracy	Precision	Recall	F1-Score
SVM	93.99%	97%	92%	94%
Naïve Bayes	89.89%	94%	87%	90%

1. Accuracy: SVM consistently delivers higher accuracy than Naïve Bayes, with SVM accuracy of 93.99% and Naïve Bayes of 89.89%.
2. Precision: SVMs tend to provide higher precision values than Naïve Bayes, with SVM precision of 97% and Naïve Bayes of 94%.
3. Recall: Naïve Bayes showed lower recall than SVM, with SVM recall of 92% and Naïve Bayes of 87%.
4. F1-Score: SVM is higher at 94% compared to Naïve Bayes which has an F1-Score of 90%. This shows that SVMs tend to have a better balance between precision and recall in classification.

Overall, SVM can be considered superior in terms of accuracy and precision for classification on the dataset used, while Naïve Bayes shows good performance but slightly inferior in recall.

4. CONCLUSIONS

This study applied two algorithms, Naïve Bayes and Support Vector Machine (SVM), to analyze user sentiments from reviews of the KitaLulus app. The results showed that most users had positive opinions, with 9,862 positive reviews (78.16%) and 2,755 negative reviews (21.84%). The researchers used 5-fold cross-validation with 80% of the data for training and 20% for testing to evaluate the models. During training, SVM had better performance, reaching 92.97% accuracy compared to 88.99% for Naïve Bayes.

DOI: 10.23960/pepadun.v6i1.262

The SVM algorithm performed better than Naïve Bayes in all key areas in the testing data. It achieved an accuracy of 93.99%, meaning it correctly classified almost 94% of the reviews. Its precision was 97%, showing it rarely misclassified negative reviews as positive. The recall was 92%, meaning it found 92% of all actual positive reviews. The F1-score, which balances precision and recall, was 94%. Naïve Bayes also did well but with slightly lower scores: 89.89% accuracy, 94% precision, 87% recall, and 90% F1-score. In summary, SVM showed stronger and more reliable results in analysing user sentiment for the KitaLulus app, making it a better choice for this kind of task than Naïve Bayes.

LITERATURE

[1] M. R. Akbar, "Perkembangan yang Pesat dan Tantangan yang Dihadapi oleh Perbankan Digital di Indonesia," *Ecobankers*, vol. 4, pp. 95–111, 2023.

[2] R. R. Putra, "Pemanfaatan Blockchain Bagi Akademisi Dalam Menyambut Bonus Demografi," *Cross-border,* vol. 5, no. 1, pp. 1–11, 2022.

[3] Z. Zakiyuddin, F. Reynaldi, F. Luthfi, S. Sriwahyuni, and F. Ilhamsyah, "Dampak Gadget pada Anak Usia Remaja di SMP Negeri 02 Meureubo Kecamatan Meureubo Kabupaten Aceh Barat," *J. Pengabdi. Masy. Darma Bakti Teuku Umar*, vol. 2, no. 1, p. 161, 2020, doi: 10.35308/baktiku.v2i1.1978.

[4] N. K. C. Dewi, I. B. G. Anandita, K. J. Atmaja, and P. W. Aditama, "Rancang Bangun Aplikasi Mobile Siska Berbasis Android," *SINTECH (Science Inf. Technol. J.*, vol. 1, no. 2, pp. 100–107, 2018.

[5] P. Kaur and S. Sharma, "Google Android a Mobile Platform," *2013 Eighth Int. Conf. Broadband, Wirel. Comput. Commun. Appl.*, pp. 1–5, 2014.

[6] M. D. Pamungkas and H. Februariyanti, "Penerapan Algoritma K-Means Clustering Untuk Mengelompokan Data Review Barang Pada E-Commerce Lazada," *semanTIK*, vol. 8, no. 2, p. 99, 2022, doi: 10.55679/semantik.v8i2.29058.

[7] M. Izunnahdi, G. Aburrahman, and A. E. Wardoyo, "Sentimen Analisis Pada Data Ulasan Aplikasi KAI Access Di Google PlayStore Menggunakan Metode Multinomial Naive Bayes Sentiment Analysis on KAI Access Application Review Data on Google PlayStore Using Multinomial Naive Bayes Method," *J. Smart Teknol.*, vol. 4, no. 2, pp. 2774–1702, 2023

[8] S. Fransiska and A. I. Gufroni, "Sentiment Analysis Provider by.U on Google Play Store Reviews with TF-IDF and SVM Method," *Sci. J. Informatics*, vol. 7, no. 2, pp. 2407–7658, 2020.

[9] B. A. Habsy, "Seni Memehami Penelitian Kuliatatif Dalam Bimbingan Dan Konseling : Studi Literatur," *JURKAM J. Konseling Andi Matappa*, vol. 1, no. 2, p. 90, 2017.

[10] A. Nur, A. Zulkifli, and N. A. Shafie, "Review of the Lazada application on Google Play Store: Sentiment Analysis," *J. Comput. Res. Innov.*, vol. 9, no. 1, p. 2024, 2024.

[11] L. Hickman, S. Thapa, L. Tay, M. Cao, and P. Srinivasan, "Text Preprocessing for Text Mining in Organizational Research: Review and Recommendations," *Organ. Res. Methods*, vol. 25, no. 1, pp. 114–146, 2022, doi: 10.1177/1094428120971683.

[12] A. S. Hussein, T. Li, C. W. Yohannese, and K. Bashir, "A-SMOTE: A new preprocessing approach for highly imbalanced datasets by improving SMOTE," *Int. J. Comput. Intell. Syst.*, vol. 12, no. 2, pp. 1412–1422, 2019, doi: 10.2991/ijcis.d.191114.002.

[13] A. Fernández, S. García, F. Herrera, and N. V. Chawla, "SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary," *J. Artif. Intell. Res.*, vol. 61, pp. 863–905, 2018, doi: 10.1613/jair.1.11192.

[14] D. Berrar**,** "Cross-Validation," in *Reference Module in Life Sciences*, pp. 1–13, 2024. doi: 10.1016/B978-0-323-95502-7.00032-4.

[15] F. Akbar, "Penerapan Algoritma Naïve Bayes Untuk Mengetahui Pasien Penyakit Gagal Jantung," *J.*

Vol. 6 No. 1, 2025, pp. 47- 56 **Jurnal Pepadun**

DOI: 10.23960/pepadun.v6i1.262

Inform. Dan Rekayasa Komputer(JAKAKOM), vol. 2, no. 2, pp. 263–266, 2022.

[16] M. R. Handoko, "Sistem Pakar Diagnosa Penyakit Selama Kehamilan Menggunakan Metode Naive Bayes Berbasis Web," *J. Teknol. dan Sist. Inf.*, vol. 2, no. 1, pp. 50–58, 2021

[17] D. A. Pisner and D. M. Schnyer, *Support vector machine*. Elsevier Inc., 2019. doi: 10.1016/B978-0-12-815739-8.00006-7.