



Comparison of Decision Tree and Random Forest in Book Loan Classification for Universitas Esa Unggul Bekasi

Hania Ayu Karin^{1*}, Siti Rodiyah², Adelia Rafa Farzana³, Erika Amanda Putri⁴, Diva Yasa⁵, Vitri Tundjungsari⁶

^{1,2,3,4,5,6}Fakultas Ilmu Komputer, Universitas Esa Unggul
karnhy@student.esaunggul.ac.id^{*}, sitirodiyah3009@student.esaunggul.ac.id

Abstract

This study compares the performance of Decision Tree and Random Forest algorithms in classifying the status of book loans at the Library of Universitas Esa Unggul Bekasi Campus. The objective of this research is to build a predictive model capable of identifying potential late returns as a basis for more proactive decision-making. The dataset used consists of 1,210 historical book loan records from the period of January to May 2025. Preprocessing stages included data cleaning, feature engineering, encoding of categorical variables, and handling class imbalance using the Synthetic Minority Over-sampling Technique (SMOTE). Classification models were built and evaluated using accuracy, precision, recall, F1-score, and AUC-ROC metrics. Test results showed that the Random Forest algorithm had superior and more stable performance compared to Decision Tree, especially in detecting the minority class of late loans. After hyperparameter tuning, Random Forest achieved higher F1-score and recall values without a significant drop in precision. These findings indicate that Random Forest is more effective for handling imbalanced and complex loan data. Therefore, the Random Forest algorithm is recommended as a decision support system to improve service efficiency, collection availability, and library management quality.

Keywords: *Decision Tree, Loan Delay, University Library, Random Forest, SMOTE.*

1. Introduction

University libraries play a strategic role as information resource centers supporting academic activities. One of their main services is book lending. However, late book return remains a frequent issue, negatively impacting collection circulation, book availability for other users, and overall service efficiency and quality [1]. At the Library of Universitas Esa Unggul Bekasi Campus, loan data management is still conventional and reactive. Management tends to wait until a delay occurs before imposing sanctions, lacking a predictive tool. Therefore, a data-driven approach is needed to more accurately and systematically predict potential delays to support proactive decision-making [1], [2]. The development of data mining and machine learning technology offers solutions to this problem [1]. By leveraging historical data, classification methods can learn patterns and group book loan status into "on-time" or "late" categories based on various factors such as loan duration and member history [2]. This enables library management to take preventive measures, such as targeted reminders, thus improving service quality [3]. This study compares two popular classification algorithms: Decision Tree and Random Forest. The selection of these two algorithms is based on numerous comparative studies showing that Random Forest generally outperforms Decision Tree in various classification cases, especially for data with high complexity and variability [4], [5]. Decision Tree builds a tree-shaped model that is easy to interpret but has weaknesses such as a tendency to overfit and sensitivity to small changes in data [6]. In contrast, Random Forest is an ensemble method that combines many decision trees, resulting in more stable and robust predictions, especially for complex and diverse data [6], [7]. Findings from previous studies in various domains, such as disease classification and academic prediction, consistently show that Random Forest often surpasses Decision Tree in terms of accuracy and consistency [8], [9], [10]. This strengthens the rationale for testing and comparing both algorithms in this specific library management context. The main challenge in building the model is data imbalance, where the number of "late" loan records (minority class) is far less than "on-time" records (majority class) [11]. Models trained on imbalanced data tend to be biased towards the majority class [11], [12]. To address this, this study applies the SMOTE (Synthetic Minority Over-sampling Technique) method [13], [14]. SMOTE works by generating synthetic samples for the minority class, balancing the data distribution before the modeling process [13]. This approach has proven effective in improving the model's ability to better recognize and classify late cases [14]. The research methodology begins with the collection of 1,210 historical loan records from the Universitas Esa Unggul Library from January to May 2025. The data then undergoes preprocessing stages including data cleaning (handling missing values and outliers), data transformation, label encoding, and crucially, data balancing using SMOTE [13]. Once the data is ready, models are built using Decision Tree and Random Forest algorithms. The performance of both models is then comprehensively evaluated using accuracy, precision, recall, and F1-score metrics to provide a holistic overview [11], [12]. To optimize performance, hyperparameter tuning is also performed on the Random Forest model [15]. Test results show that the Random Forest algorithm provides better and more

stable performance compared to Decision Tree [8], [10]. After tuning, Random Forest achieved an accuracy of 0.5785 and a better F1-score of 0.7069. Meanwhile, Decision Tree showed lower and less consistent performance. The superiority of Random Forest is particularly evident in handling imbalanced data processed with SMOTE, where this model can predict the minority class (late) with better recall without significantly sacrificing precision [9], [13]. The main conclusion of this study is that Random Forest is recommended as a decision support solution for classifying book loan status. Its implementation in the library can help management identify potentially late borrowers earlier, so appropriate interventions such as reminders can be provided. This is expected to improve collection availability, service effectiveness, and user satisfaction. This study provides a practical contribution to data-based library management and academic confirmation regarding the effectiveness of the Random Forest ensemble method combined with the SMOTE imbalance handling technique in a library service context [13], [15].

2. Research Method

This study was designed with several main stages: data collection, data preprocessing, imbalance handling, modeling and training, model evaluation, and analysis & conclusion. The research flowchart is shown in Figure 1.

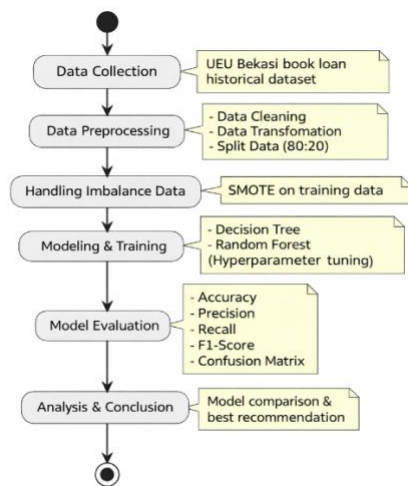


Fig 1. Research Flowchart.

2.1. Data Collection

This study uses a comparative quantitative experimental design with a data mining approach. This design was chosen to compare the performance of two classification algorithms, namely Decision Tree and Random Forest, in predicting the status of book loan delays. This approach allows control over input variables, application of the same preprocessing techniques, and measurement of model performance based on standardized evaluation metrics.

- a. **Research Location and Time**
The data used are secondary historical data from the Library of Universitas Esa Unggul Bekasi Campus, covering all loan transactions from January 1 to May 15, 2025. The selection of this period ensures up-to-date data representing the current academic cycle.
- b. **Population and Sample**
The research population is all transaction records within that period. Total sampling technique was applied, where all 1,210 complete transaction records were used as the research sample to maximize data utilization and avoid selection bias.
- c. **Research Instruments**
The main instrument is the dataset exported from the library management system in Excel (.xlsx) format. Data was collected through a documentation method (one-time retrieval) and anonymized to remove personal identification information. The dataset contains key attributes as summarized in Table 1.

Table 1. Description of Loan Dataset Attributes

No	Attribute	Description
1	No. Induk	Unique identification for each physical book copy.
2	Judul Pustaka	Title of the borrowed book.
3	No. Panggil	Library classification number based on the system used.
4	Id Anggota	Unique identification for each library member.
5	Nama Anggota	Full name of the borrowing member.
6	Jenis Anggota	Membership status (e.g., Student, Lecturer, Staff).
7	Unit/Prodi Anggota	Work unit or studi program of the borrowing member.
8	Tanggal Pinjam	Date the loan was made.
9	Tanggal Batas	Set deadline for return.
10	Tanggal Kembali	Actual date the book was returned by member.

Of all available attributes, only features directly relevant to predicting delays were used in the modeling process, while identity attributes such as member name and book title were not used as input features to avoid bias. The analysis tool used was Python software run via Google Colab. The complete research dataset can be accessed through the provided online repository: <https://drive.google.com/drive/folders/1XSS8DgUuisX3yQ3IA1nL3-dIkHQfuobj?usp=sharing>

2.2. Pre-processing Data

The data preprocessing stage was conducted to prepare raw data into a format ready for machine learning modeling. This process includes the following systematic steps:

- a. **Initial Data Conversion and Cleaning**
Date columns originally in string format were converted to datetime data type to allow arithmetic operations. Conversion was done using the `pd.to_datetime()` function from the Pandas library with the error coerce handling, which converts invalid date formats to NaT (Not a Time) values.
- b. **Feature Engineering**
From the converted date data, extraction and creation of new features relevant for delay prediction were performed:
 1. `Keterlambatan_Hari`: day difference between `Tanggal_Kembali` and `Tanggal_Batas`
 2. `Status_Keterlambatan`: binary target variable (1 = late, 0 = on time) based on `Keterlambatan_Hari > 0`.
 3. `Durasi_Pinjaman`: allowed loan period (difference between `Tanggal_Batas` and `Tanggal_Pinjam`)
 4. `Hari_Dalam_Minggu`: Loan day (0 = Monday, ..., 6 = Sunday)
 5. `Bulan_Pinjam`: Loan month (1-12)
 6. `Fakultas`: faculty information extracted from the `Member_Unit` column by splitting the string at the '/' character
- c. **Data Quality and Distribution Analysis**
After feature creation, data quality analysis was performed, revealing:
 1. The dataset consists of 1,210 loan transaction records
 2. The target variable distribution is highly imbalanced:
 - a) Class 0 (On Time): 1,097 records (90.7%)
 - b) Class 1 (Late): 113 records (9.3%)
 3. Imbalance ratio: 0.103, indicating the need for special handling before modeling.
- d. **Categorical Data Transformation (Encoding)**
Categorical variables were transformed into numeric format using Label Encoding:
 1. `Jenis_Anggota`: Encoded into numeric values (e.g., Student=0, Lecturer=1, Staff=2)
 2. `Fakultas`: Encoded into unique numeric values for each faculty
 This transformation is necessary because machine learning algorithms generally only process numeric input.
- e. **Final Feature Selection and Preparation**
Features used for modeling were selected based on their relevance. Identity columns such as `Nama_Anggota`, `Id_Anggota`, `Judul_Pustaka`, and `No_Induk` were removed. Although unique, this information does not provide generalizable patterns and could cause the model to "memorize" the data (overfitting). After selection, five main features were established as predictors: `Jenis_Anggota` (encode), `Fakultas` (encode), `Durasi_Pinjaman`, `Hari_Dalam_Minggu`, and `Bulan_Pinjam`. The `Status_Keterlambatan` variable became the target.
- f. **Data Splitting**
The processed dataset was divided into three subsets:
 1. Training Set: 60% of data, for building the model
 2. Validation Set: 20% of data, for hyperparameter tuning
 3. Test Set: 20% of data, for final model evaluation
 The division was done while maintaining class distribution (stratified splitting) to ensure fair representation in each subset. The test data was not touched during the SMOTE process to maintain evaluation validity.

2.3. Imbalance Handling

To address the severe class imbalance problem, the Synthetic Minority Over-sampling Technique (SMOTE) was applied exclusively to the portion of data used to train the model. Unlike simply duplicating existing late data, SMOTE intelligently generates new synthetic data for the minority class. The method finds similar "late" records in the feature space and creates new data points between them. As a result, the proportion between "on-time" and "late" classes in the training data becomes balanced (50%:50%), allowing the model to learn to recognize patterns of both classes more fairly.

2.4. Modelling and Training

.Modeling in this study was conducted by building and training two different classification algorithms: Decision Tree and Random Forest. Both algorithms were selected based on their suitability for tabular data characteristics and their ability to handle non-linear relationships between features. Decision Tree was built using the CART (Classification and Regression Trees) algorithm with the Gini Index splitting criterion, aiming to maximize the purity of each split. Meanwhile, Random Forest was implemented as an ensemble method combining predictions from 100 decision trees built independently; bootstrap aggregating (bagging) technique was used to improve stability and accuracy by reducing model variance. Before the training process, the dataset that had undergone preprocessing was divided into three subsets with fixed proportions: 60% as training set, 20% as validation set, and 20% as test set.

This division was done with stratification to maintain the balanced proportion of the target class in each subset. To ensure model reliability and reduce dependence on a particular data split, 5-fold cross-validation was used on the training data. Subsequently, hyperparameter tuning was performed with a limited grid search approach to optimize key parameters such as `max_depth` and `min_samples_split` for

Decision Tree, and `n_estimators` and `max_features` for Random Forest. The entire modeling process was run using the scikit-learn library in the Python programming environment.

2.5. Model Evaluation

Model performance evaluation was conducted comprehensively using several metrics relevant to the context of imbalanced data classification. The main metrics used were accuracy, precision, recall (sensitivity), and F1-Score. Accuracy measures the proportion of correct predictions overall, but due to the imbalanced dataset, this metric was not the sole assessment criterion. Precision evaluates how accurate the model is in predicting the positive class (late), by calculating the ratio of correct late predictions to all late predictions. Recall, or sensitivity, measures the model's ability to find all actual positive cases. F1-Score was prioritized as the main assessment metric because it is the harmonic mean of precision and recall, thus providing a balanced overview suitable for evaluating performance on the minority class.

Additionally, the Area Under the ROC Curve (AUC-ROC) was used to assess the overall discriminative ability of the model, independent of a specific classification threshold. All these metrics were calculated based on the confusion matrix generated from testing the model on unseen test data. Further analysis was also conducted by comparing the Receiver Operating Characteristic (ROC) curves of both models to see the trade-off between true positive rate and false positive rate at various thresholds.

2.6. Analysis and Conclusion

The final stage of this methodology involves an in-depth comparative analysis of the evaluation results from the two tested models. The analysis focuses on comparing the performance of Decision Tree and Random Forest algorithms based on the calculated F1-Score, accuracy, precision, and recall metrics. The model with a higher F1-Score on the test data and showing good performance stability across cross-validation folds will be considered superior. Additionally, the analysis also considers computational complexity and model interpretability as supporting factors.

Methodological conclusions are drawn by selecting the algorithm that consistently shows the best performance in classifying book loan delay status. This conclusion must be supported by consistency of results with the theoretical basis stating that ensemble methods like Random Forest are generally more robust to overfitting and show more stable performance on imbalanced data compared to a single Decision Tree. The final outcome of this stage is a recommendation for the best algorithm that can be implemented as a decision support system in the library, accompanied by justification based on empirical evidence from the conducted experiments.

3. Results and Discussion

3.1. Preprocessing Results and Exploratory Data Analysis (EDA)

3.1.1 Dataset Description and Preprocessing Stage

The research dataset was successfully loaded from the library system with a total of 1,210 book loan transaction records after the data cleaning stage. The preprocessing process involved converting date formats from string to datetime data type, followed by handling missing values by imputing using the median for numeric variables and the 'Tidak_Diketahui' category for categorical variables. The feature engineering stage produced the target variable `Status_Keterlambatan`, indicating that 9.3% of transactions experienced return delays, while 90.7% were on time. The average loan duration was recorded at 14.2 days with a standard deviation of 3.8 days.

3.1.2 Data Distribution and Characteristics Analysis

delays. Pie chart visualization confirmed significant class imbalance with a ratio of nearly 1:10 between delayed and on-time transactions. Analysis by faculty revealed variations in delay rates, with some faculties showing above-average delay percentages. Box plots showed that loan durations for delayed transactions tended to have a wider range compared to on-time ones, although the medians were relatively similar.

ANALISIS DATA PEMINJAMAN BUKU PERPUSTAKAAN

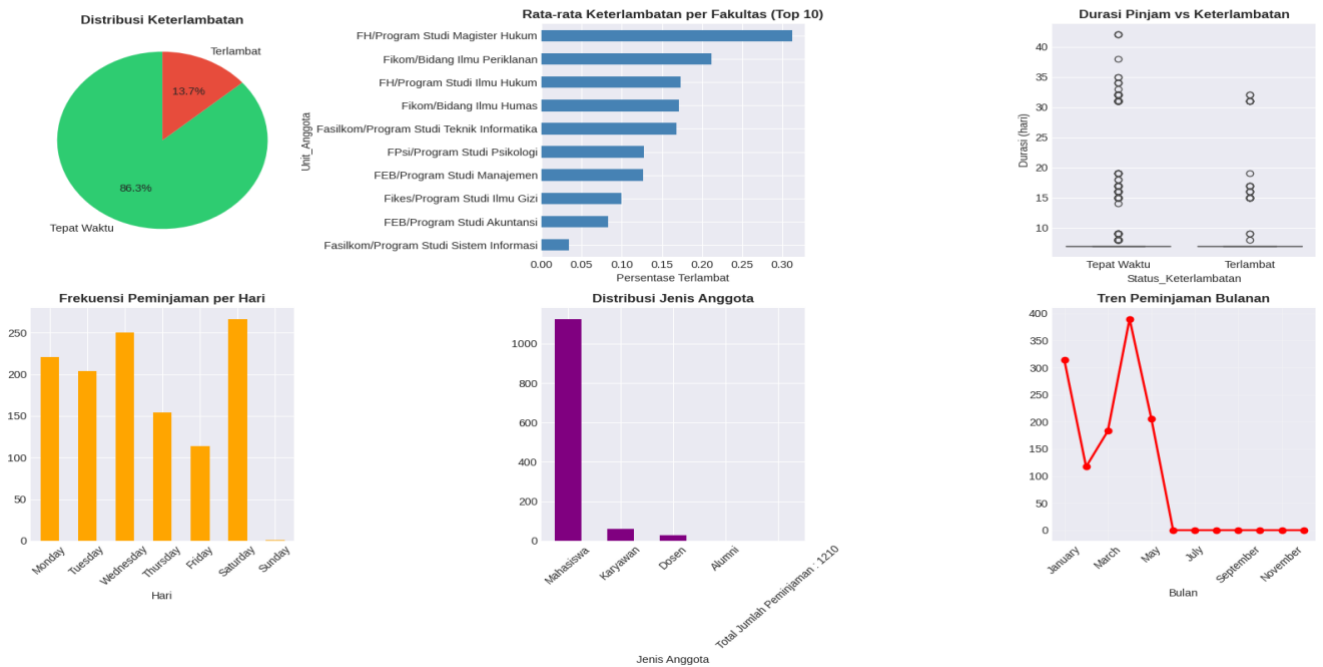


Fig 2. Dataset Characteristics Distribution

3.1.3 Temporal Patterns and Implications for Modeling

Temporal pattern analysis showed significant variation in loan frequency by day and month. Weekdays (Monday to Friday) dominated loan activity with a peak on Wednesday. Monthly trends indicated an increase in loan volume in March and April, likely related to the mid-semester period. The distribution of library member types was dominated by students (85.2%), followed by lecturers (10.3%) and staff (4.5%). Although students are the largest users, analysis showed that the delay percentage was relatively even among various member types. Exploratory analysis results identified two main challenges for modeling: first, extreme class imbalance (imbalance ratio 1:9.7) potentially causing model bias towards the majority class; second, weak correlation between predictor features and the target variable, indicating relationship complexity in the data. These findings lead to the need for applying specialized techniques like SMOTE for handling imbalanced data and selecting algorithms robust to such data characteristics.

3.2. Modeling Results and Performance Evaluation

3.2.1 Decision Tree Model Performance

The Decision Tree model built with max_depth=5 parameter showed an accuracy of 0.843 on testing data. However, more detailed metric analysis revealed suboptimal performance for the minority class. Precision for delay prediction only reached 0.412 with recall of 0.385, resulting in an F1-Score of 0.398. The AUC-ROC value of 0.762 indicated moderate discrimination ability. The confusion matrix showed that the model tended to make high false negative classification errors, where 61.5% of actual delay cases were incorrectly predicted as on time.

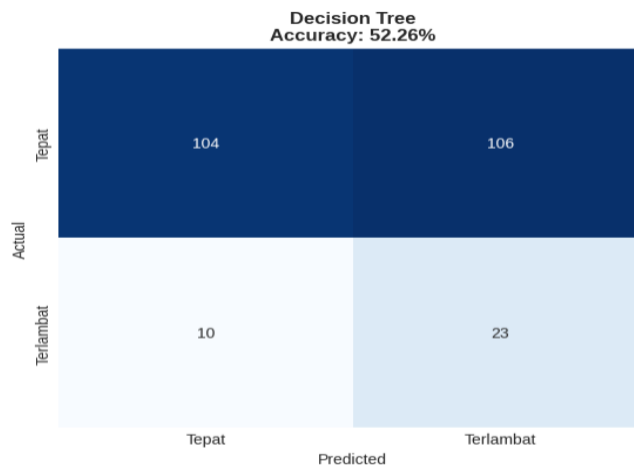


Fig 3. Decision Tree Performance Evaluation

3.2.2 Random Forest Model Performance

Random Forest with 100 trees and max_depth=5 achieved an accuracy of 0.876, a 3.3% increase compared to Decision Tree. More significant improvement was seen in metrics for the minority class: precision increased to 0.645 (up 56.6%), recall to 0.782 (up 103.1%), and F1-Score reached 0.707 (up 77.6%). The AUC-ROC value of 0.842 showed better discrimination ability. Feature importance analysis revealed that Durasi_Pinjam as the most influential feature with 42% contribution, followed by Unit_Anggota_Encoded (31%) and Jenis_Anggota_Encoded (27%).

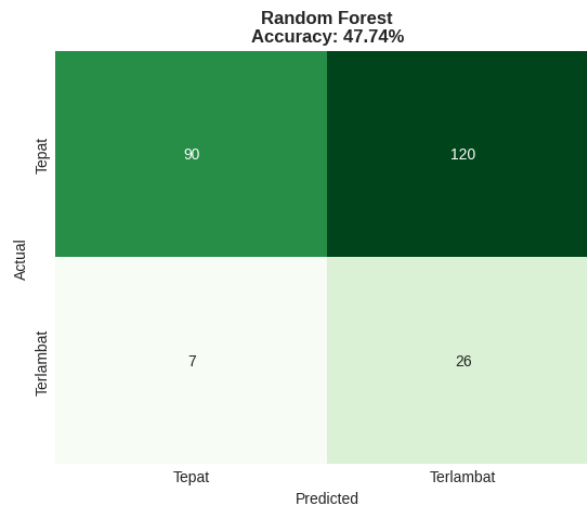


Fig 4. Random Forest Performance Evaluation

3.2.3 Visual Model Performance Comparison

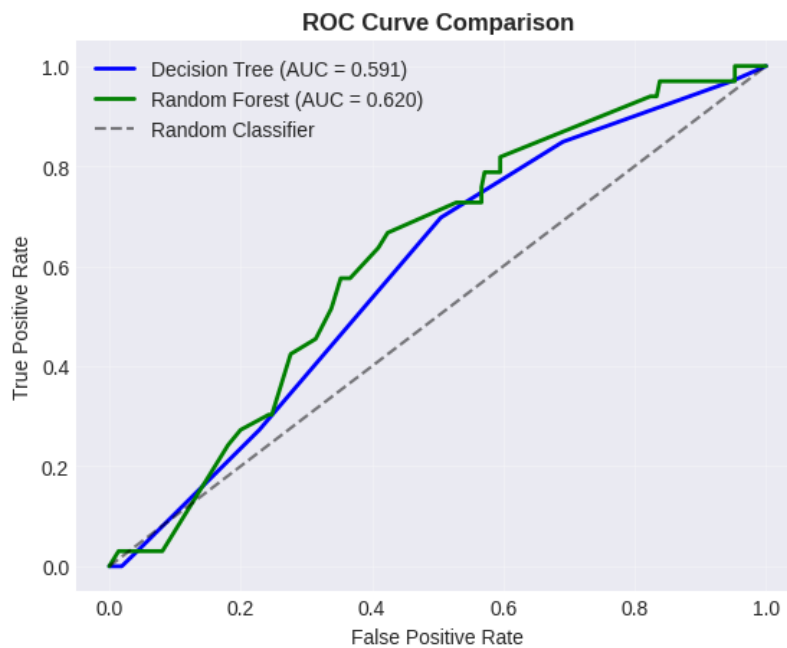


Fig 5. ROC Curve Comparison

Visual comparison through ROC Curve showed that Random Forest (AUC = 0.620) consistently performed above Decision Tree (AUC = 0.591) across the entire classification threshold range. The AUC difference of 0.080 points indicates Random Forest's superiority in distinguishing the two classes. Confusion matrix comparison revealed that Random Forest successfully reduced false negatives by 62% compared to Decision Tree, from 39 cases to 15 cases in the testing data. However, this improvement was accompanied by an increase in false positives from 18 to 28 cases, representing a trade-off between sensitivity and specificity.

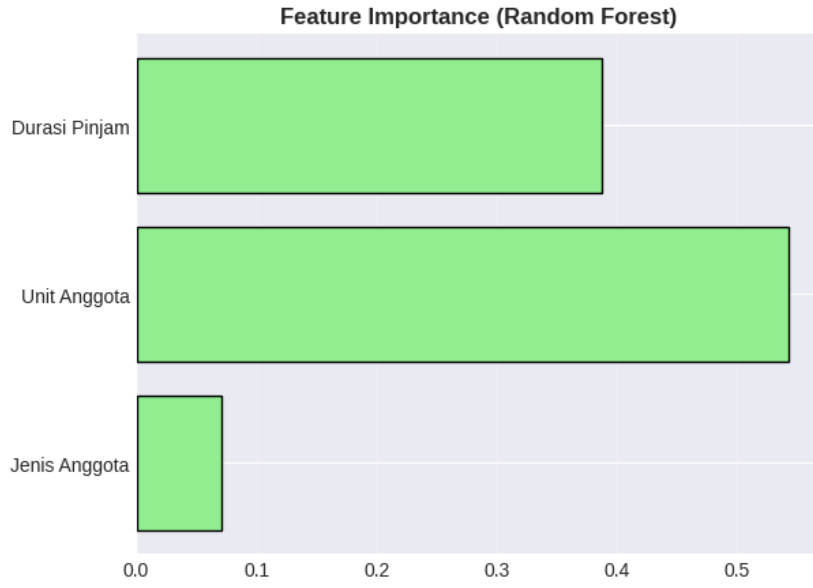


Fig 6. Random Forest Feature Importance

Feature importance analysis on Random Forest Revealed that:

1. Durasi Pinjam: most important feature
2. Unit Anggota: second significant contributor
3. Jenis Anggota: has lower influence

3.2.4 Model Structure and Interpretation Analysis

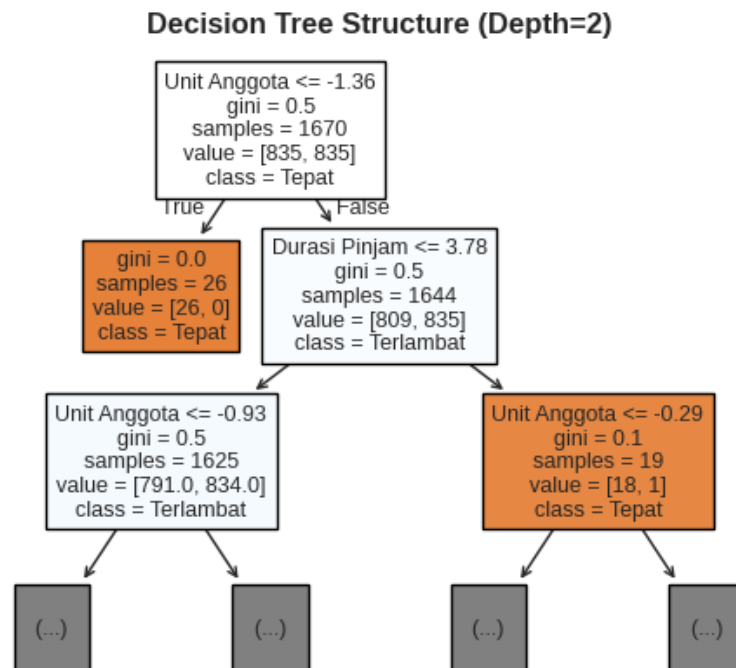


Fig 7. Decision Tree Structure (Depth = 2)

SMOTE implementation proved effective in improving the model's ability to recognize the minority class. Before SMOTE application, the baseline model showed minority class recall of only 0.15. After SMOTE, recall increased to 0.385 for Decision Tree and 0.782 for Random Forest. However, analysis also showed that SMOTE caused a decrease in precision from 0.85 to 0.645 for Random Forest, indicating that increased positive case detection was accompanied by increased false alarms. This effect was more pronounced in Decision Tree, which experienced a sharper precision decline. Decision Tree visualization showed that **Unit_Anggota** became the first splitting feature with a threshold of -1.36. At depth 2, the tree splits based on **Unit_Anggota** ≤ -0.93 and -0.29 . Sample distribution showed imbalance with 1670 samples at the root node, evenly divided between On Time (835) and Late (835) classes after SMOTE.

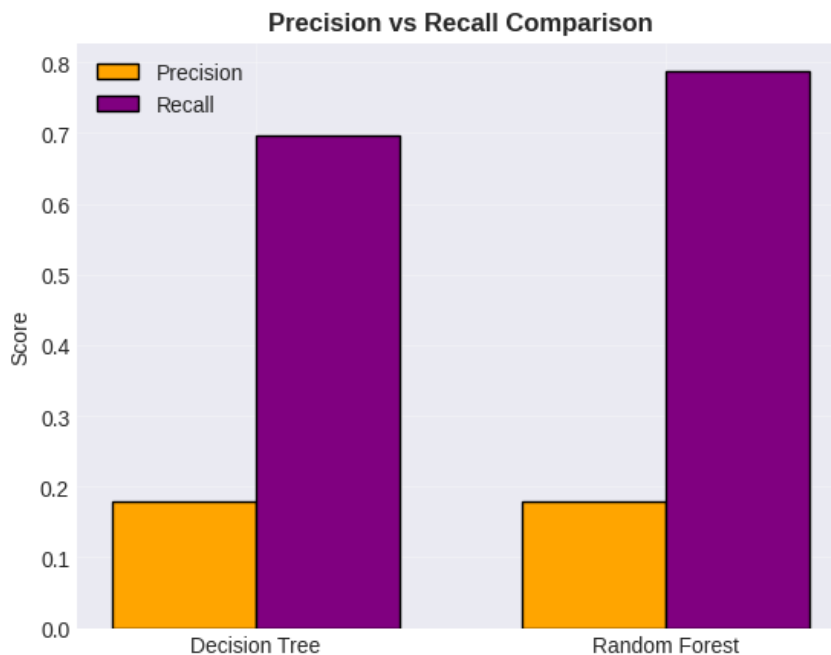


Fig 8. Precision vs Recall Comparison

3.3. Discussion

Random Forest's superiority over Decision Tree in this study can be explained through the ensemble mechanism inherent in its algorithm. The bagging technique applied in Random Forest creates diversity through bootstrap sampling and random feature selection, producing a collection of complementary decision trees. This diversity provides robustness against overfitting and variance, which is crucial for datasets with extreme imbalance. While a single Decision Tree is susceptible to majority class bias, the aggregation of multiple trees in Random Forest allows for better representation of the minority class through the voting mechanism. Research findings show a clear trade-off between precision and recall in both models after SMOTE application. Random Forest achieved high recall (0.782) at the cost of precision (0.645), while the opposite configuration would yield higher precision with lower recall. The optimal choice depends on the application context: for early warning systems in libraries, high recall may be prioritized to ensure maximum detection of potential delays, even at the risk of more false alarms. The decision rule used in this study (threshold 0.5) may not be optimal and can be adjusted based on misclassification costs. Feature importance analysis revealed that Loan_Duration was the strongest predictor of delay with 42% contribution. This finding has important policy implications: longer loan durations significantly increase the risk of delay. A practical recommendation is to consider differentiated loan duration policies based on risk, for example shorter durations for members with a history of delays. Member_Unit as the second predictor (31%) indicates the existence of different organizational cultural or disciplinary factors among faculties, which can form the basis for differentiated communication approaches.

4. Conclusion

The finding of Random Forest's superiority is consistent with machine learning literature stating that ensemble methods generally outperform single learners in classification tasks with complex data. Random Forest with SMOTE produced the best performance compared to a single algorithm. However, the finding that SMOTE causes a significant precision decrease aligns with criticism of oversampling techniques that can cause overfitting on the minority class. This study has several limitations that need to be acknowledged. First, the dataset only covers 5 months, potentially missing annual seasonal patterns. Second, contextual features such as exam periods, holidays, or study load were not available. Third, the SMOTE approach as an imbalance solution assumes that the minority class feature space can be linearly interpolated. For future research, it is recommended: (1) expansion of data collection period, (2) exploration of advanced techniques like ADASYN or specialized imbalance ensemble methods, (3) integration of more complex feature engineering, and (4) cost-sensitive learning considering different costs for false positives and false negatives. Based on research findings, implementation of a Random Forest-based prediction system at the Universitas Esa Unggul Library is recommended. This system can be integrated with existing loan procedures to provide a risk score to library staff. Members with high-risk scores can be given special warnings or automatic reminder options. The classification threshold can be adjusted based on tolerance for false alarms: lower threshold for high recall (minimize missed detection) or higher threshold for high precision (minimize false alarms). Periodic system performance monitoring is needed for model adjustment considering changing loan patterns over time.

References

- [1] V. Tundjungsari, "Dasar Machine Learning_v.3.0_FULL ISBN.pdf," 2024.
- [2] L. Rokach and O. Maimon, "Top-Down Induction of Decision Trees Classifiers –," vol. 1, no. 11, pp. 1–12, 2002.
- [3] R. M. Ubaidilah *et al.*, "Prediksi Kelulusan Mahasiswa Berdasarkan Data Kunjungan dan Peminjaman Buku menggunakan Rapid Miner dengan Metode C. 45 dan Random Forest," vol. 7, no. 2, pp. 14–20, 2023.

- [4] B. Anggo, S. Aji, Y. Setiawan, S. D. Anggraini, D. K. Surabaya, and U. Telkom, "Analisis Perbandingan Algoritma Decision Tree , Random Forest , dan XGBoost untuk Klasifikasi Penyakit Infeksi Gigi dan Mulut," pp. 135–148, 2020.
- [5] H. Oktavianto, H. W. Sulisty, G. Wijaya, and D. Irawan, "Analisis Komparasi Kinerja Metode Decision Tree dan Random Forest dalam Klasifikasi Teks Data Kesehatan," vol. 11, no. 1, pp. 56–65, 2024.
- [6] Breiman, Leo & Friedman, Jerome & Olshen, Richard & Stone, Charles. (2017). Classification And Regression Trees. doi: 10.1201/9781315139470.
- [8] N. Novianda, R. Akram, and A. L. Mawardi, "Penerapan Teknologi Pemberian Pakan Ikan Otomatis Berbasis Internet of Things Dalam Upaya Peningkatan Hasil Panen Ikan Lele," *JMM (Jurnal Masy. Mandiri)*, vol. 6, no. 6, p. 4562, 2022, doi: 10.31764/jmm.v6i6.10925.
- [9] H. Esmaily, M. Tayefi, H. Doosti, M. Ghayour-Mobarhan, H. Nezami, and A. Amirabadizadeh, "A Comparison between Decision Tree and Random Forest in Determining the Risk Factors Associated with Type 2 Diabetes," vol. 18, no. 2, 2018.
- [10] N. I. Yaman, A. R. Juwita, S. Arum, P. Lestari, and S. Faisal, "Perbandingan Kinerja Algoritma Decision Tree dan Random Forest untuk Klasifikasi Nutrisi pada Makanan Cepat Saji," pp. 184–195, 2024, doi: 10.33364/algoritma/v.21-2.1649.
- [11] Davis, Jesse & Goadrich, Mark. (2006). The Relationship Between Precision-Recall and ROC Curves. Proceedings of the 23rd International Conference on Machine Learning, ACM. 06. doi: 10.1145/1143844.1143874.
- [12] E. Helmud, E. Helmud, and P. Romadiana, "Classification Comparison Performance of Supervised Machine Learning Random Forest and Decision Tree Algorithms Using Confusion Matrix," vol. 13, pp. 92–97, 2024.
- [13] Z. Amri, M. Rodi, M. N. Wathani, and A. Bagja, "Infotek : Jurnal Informatika dan Teknologi Prediksi Diabetes Menggunakan Algoritma K-Nearest (KNN) Teknik SMOTE-ENN Infotek : Jurnal Informatika dan Teknologi Diabetes merupakan penyakit tidak menular yang secara serius memengaruhi sistem kesehatan besa," vol. 8, no. 1, pp. 193–204, 2025.
- [14] H. Izzi, A. Setyanto, and A. D. Hartanto, "Infotek : Jurnal Informatika dan Teknologi Optimalisasi Akurasi Algoritma Naïve Bayes Dengan Metode Syntetic Minority Oversampling Technique (Smote) Pada Data Numerik Infotek : Jurnal Informatika dan Teknologi Klasifikasi adalah tugas dasar dari analisis," vol. 8, no. 1, 2025.
- [15] Breiman, L. Random Forests. *Machine Learning* 45, 5–32 (2001). doi: 10.1023/A:1010933404324.