International Journal of Entrepreneurship, Business and Creative Economy, Vol. 5 No. 1 (2025) https://doi.org/10.31098/ijebce.v5i2.3152

Check for updates

Research Paper

Mobile Based Application for Loan Approval and Loan Distribution Using Machine Learning in Savings and Loan Cooperatives

Benni Agung Nugroho[1], Rinanza Zulmy Alhamri[1], Toga Aldila Cinderatama[1]
[1]Politeknik Negeri Malang, Indonesia

Received: April 30, 2025	Revised: June 5, 2025	Accepted: July 9, 2025	Online: July 31, 2025

Abstract

Several savings and loan cooperatives (KSP) in Kediri City, Indonesia, have utilised a website-based information system to enhance efficiency. However, the financial health of KSP in Kediri City remains low due to numerous delays in credit payments, and even instances of bad credit have occured. Manual profiling for approving a loan application can lead to poor decisions. The management needs a function to obtain recommendations for approving loan applications and to automatically distribute loan services to potential members. The purpose of this research is to develop a mobile-based application for loan approval recommendation and loan distribution utilising Machine Learning (ML) in KSP and to study the performance of the model using the Support Vector Machine (SVM) method. It adopted the Waterfall Method, including analysis, design, implementation, and testing, for two purposes: SVM model development and Android-based application development. The dataset experienced preprocessing, including data cleaning, label encoding, and normalisation. It obtained 150 data points for loan approval recommendation and 150 data points for loan distribution. The implementation stage includes developing an Android-based application and a Python-based ML. The testing stage utilises functional testing for the Android application and K-Fold Cross Validation for ML performance evaluation. An Android application has two users: the first is an administrator who can manage members, retrieve loan approval recommendations, and manage loan applications, and the second is a member who can retrieve loan distributions and apply for loans. The performance of the ML model using SVM includes an accuracy of 90% in loan approval recommendations, while loan distribution reaches 85%.

Keywords: *Android; Machine Learning; Support Vector Machine; Savings and Loan Cooperatives.*

INTRODUCTION

Since 2021, the Kediri City Government has encouraged cooperatives under Dewan Koperasi Indonesia (Dekopinda) of Kediri City to utilise information technology for business operations, aiming to enhance work efficiency (Pemkot Kediri, 2021). In 2023, 400 cooperatives were registered under the Depkopinda of Kediri City (Putra, 2023), representing more than 600 cooperatives in Kediri City (BPS Kota Kediri, 2018). From those 400 cooperatives, currently only a small number have utilised information technology, especially savings and loan cooperatives (KSP). KSP have used a website-based information system to increase efficiency by providing robust, precise data and simplifying processes. The use of information technology in KSP has been proven to have a positive influence on work efficiency according to research (Saputra & Rizaldi, 2021; Dharmayanti et al., 2023).

The opposite finding was revealed through the analysis of 5 KSPs in Kediri City, conducted by Dwiana and Sari (2022), where inly one KSP has a healthy predicate based on the financial efficiency aspect; no KSP has a healthy predicate based on the liquidity aspect, and only one KSP has a healthy predicate based on the aspects of independence and growth. Additionally, a direct interview with an employee cooperative at a health agency in Kediri city, revealed that 60% of members who applied for credit failed to return it on time on time, resulting in financial losses, particularly among honorary, especially honorary employees in 2023. They manually profiled

Corresponding author's email: rinanza.z.alhamri@polinema.ac.id

members who applied for credit by verifying the data provided by the web-based information system. Manual member profiling can lead to errors in loan approval decisions, resulting in poor credit, which can cause a decline in financial health. The management needs a function to obtain recommendations for approving loan applications and also to automatically distribute loan services to potential members.

The purpose of this research is to enable cooperative management to determine loan feasibility and to automatically distribute loan services to potential members. With the availability of members' data from a web-based information system, the data can be used as a dataset to classify members as to whether they are eligible to apply for a loan. This study implements Machine Learning (ML) for loan approval and loan distribution inference as a server. It develops an Android-based application for loan approval and loan distribution as an ML client. The proposed classification employs the Support Vector Machine (SVM) method, which is capable of achieving high accuracy in determining loan eligibility (Pernama & Purnomo, 2023; Tamami & Kharisudin, 2023). Accordingly, this study hypothesizes that employing an SVM-based machine learning approach will result in loan approval and distribution accuracy exceeding 75%. This study presents the performance of an ML model using the SVM method to answer the hypothesis. Furthermore, the achieved ML model is utilized to develop a mobile-based application for loan approval and loan distribution. This study aims to support the creation of an Android-based system that assists in assessing loan eligibility and delivering loan services to qualified members, with the goal of minimizing the risk of bad credit and enhancing both operational efficiency and financial stability within cooperatives.

LITERATURE REVIEW

Relevant Research

Research on applying the SVM method to determine creditworthiness has been widely conducted and is currently being developed as a case study solution. According to prior research by Putri et al. (2021), a credit risk analysis was conducted on bank customer data using SVM. Six hundred ten data points were used, 80% for training and 20% for testing. The result is that by using linear, polynomial, RBF, and sigmoid SVM types in sequence, the accuracy reached 92.62%, 95.08%, 89.34%, and 83.61%. Then, research by Riyadi et al. (2022) compared the use of the SVM and Naive Bayes (NB) algorithms to classify bad credit in KSP. Sixty-one data with 8 attributes were used, with 80% for training and 20% for testing. The results show that SVM achieved an accuracy of more than 90%, but the highest accuracy was achieved using NB, reaching 95.51%.

Another study by Pernama and Purnomo (2023) analyzed loan risk by actively comparing SVM, Artificial Neural Network (ANN), and NB using an open dataset from an Indian bank comprising 250 data points with six attributes, allocating 75% for training and 25% for testing and found that SVM achieved the highest accuracy at 92%. The last is research which classifies the eligibility of loan applications at KSP (Asana & Yanti, 2023). Using 200 data of 6 attributes with 80% for training and 20% for testing. The result is a recommendation application for determining loan application eligibility with an accuracy of 85%.

Based on relevant research, the supervised ML method can support creditworthiness grouping in credit case studies. This becomes relevant if the supervised ML method is used in decision-making for KSP loan approval and loan distribution. The difference between this study and previous studies is that in previous studies, the ML method was used only to classify creditworthiness based on specified features. In contrast, this study not only classifies applicants based on certain features but also provides decisions on actual conditions outside of historical data. Not only that, but this study also provides decisions on loan eligibility for KSP members, classifies members who have the potential to make loans, and makes decisions on loan distribution.

The use of the SVM algorithm in the supervised ML method is widely employed in previous studies because it offers high accuracy compared to other algorithms. Except in the study by Riyadi et al. (2022), the use of the SVM algorithm has lower accuracy than NB because the study used a limited dataset, specifically fewer than 100 data points. In this study, over 100 data points were utilised. Therefore, referring to the research conducted by Putri et al. (2021), Pernama et al. (2023) and Asana et al. (2023) this study proposes the use of the SVM algorithm as a relevant approach, which forms the basis for Hypothesis 1, stating that the model's accuracy will exceed 75%. This threshold aligns with Cabitza et al. (2020), who argue that an accuracy above 75% in supervised ML is generally considered acceptable for application in contextual settings or prototype development.

In addition, other relevant research that helps determine the best features for KSP loan approval and distribution is based on Widyarini et al. (2020) research, where in assessing the feasibility of KSP loans, it is necessary to check historical data based on the amount of the loan, employee status, salary, amount of savings, loan history, and additional literature studies. Because this research is applied to the needs of a KSP case study, a combination of features is carried out based on interviews and direct observations at the KSP.

While the focus on applying classification to a system ready for use by users remains low. The proposed research focuses on the development of a supporting system, including an ML server and an Android-based application as a client, that enables users to provide recommendations for loan eligibility at cooperatives using the SVM method. Furthermore, the application is developed for mobile devices, enabling more flexible data access, which contributes to the high novelty of this research.

Decision Support System

A Decision Support System (DSS) is a computer system used to help users solve problems by analysing data mathematically based on historical data (Prahartiwi & Fatimah, 2023). The stages of decision making in general include problem identification, data collection, model selection, model implemantation, evaluation, and the final solution, as shown in Figure 1.

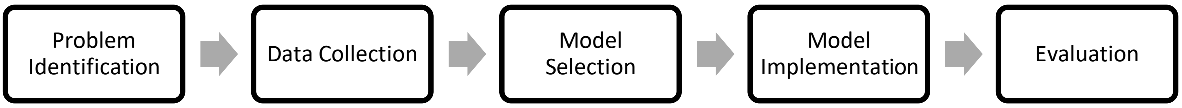


Figure 1. DSS Development

The support Vector Machine (SVM) is a supervised machine learning method that classifies data by creating a dividing line in the form of a hyperplane, allowing it to group data into two parts (binary) (Asana & Yanti, 2023). SVM classification is appropriate for the case of credit application eligibility because it requires a decision whether a member is classified as creditworthy or not. The best classification is achieved by maximising the margin value between the outermost support vector data and the hyperplane, as shown in Figure 2.

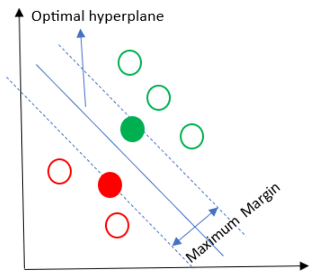


Figure 2. SVM Hyperplane

Mobile-Based Application

Research on Android-based mobile applications for KSP has been conducted previously using the Waterfall method. The Waterfall method includes Requirements Analysis and Definition, System and Software Design, Implementation and Unit Testing, Integration and System Testing, and Operation and Maintenance (Akbar et al., 2024). The use of the Waterfall method has been proven to be effective in developing KSP mobile applications according to planned functional needs, provided that all stages of the Waterfall method are followed without missing or prioritising any steps. The Waterfall method is employed in this study to develop an Android-based mobile application that utilises an ML model.

KSP Study Case Observation

This study involves five types of application users including the Chairperson, Secretary, Treasurer, Members, and New Members. Specifically, the loan application business process involves interactions between Treasurer and Member users. For the application process, Members can apply for a loan, and then the Treasurer can approve it. Meanwhile, to distribute loans, the Treasurer can manage news and information and Members can read the news or information.

Based on the case study from one of the employee cooperatives in Kediri City, there are specific requirements for members to apply for a loan, including being an active member, receiving a recommendation from the company treasurer, and having existing savings. Important considerations are whether there is still a salary to pay, a history of loans outside the cooperative, existing savings, and employee status. In detail, the loan application process in KSP is as follows.

- Members request a loan application form from the officer at the Cooperative Office
- Members fill in the data and information on the form according to the truth
- Members request a recommendation from the company treasurer,
- Members submit the application form to the cooperative management
- The cooperative treasurer submits the member's loan application to the cooperative chairperson by attaching data and information related to the applicant's condition (finances, track record, etc.) and the cooperative's finances, including:
    - Members have paid off all obligations arising from the ongoing loan
    - The applicant has a satisfactory payment record, on time, no arrears and demonstrates a cooperative attitude (good cooperation)
    - Members' Statement that they are willing to pay the loan through a monthly salary deduction system until the loan period ends
    - Statement not to borrow funds or money elsewhere that can interfere with financing at the cooperative

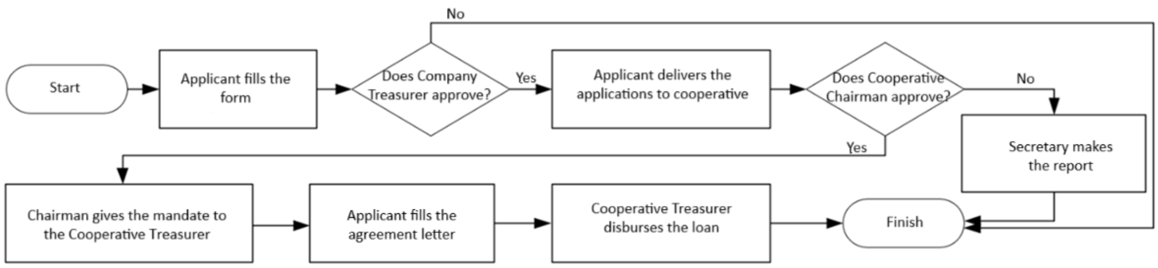


Figure 3. Loan Application Business Process

The credit application procedure must be understood to comprehend what data is a KSP factor in approving a credit application. The general credit application procedure at KSP, which involves profiling, is carried out by employees, and the chairperson makes the supervisory decision.

This aligns with the strong business process analysis results from previous research, which include the procedure of application submission by the applicant, analysis by cooperative management, and decision-making by the cooperative chairperson (Widyarini et al., 2020). Meanwhile, as a creditworthiness assessment factor, the existing condition of the KSP interview results considers four factors: the total salary, a history of loans outside the cooperative, total existing savings, and employee status. The study results of Widyarini et al. (2020) identified nine factors that need to be considered to ensure KSP finances are strong, including age, marital status, employee tenure, member tenure, salary, credit history, collateral, external information, and field condition. For this reason, a combination of assessment factors from KSP and literature studies will be used, with adjustments made to accommodate the availability of data in the field.

RESEARCH METHOD

This research was commonly conducted in three main stages: data collection, SVM Model Development, and Android-based application development. SVM Model Development and Android-based application development adopted the Waterfall Method (Akbar et al., 2024). This study employs a descriptive quantitative methodological approach to explain the study findings by analysing the collected data using an ML-based classification that yields accurate results. Furthermore, a detailed explanation of the ML model implementation for the mobile application will be provided. The following details the stages of the research methodology.

Data Collection

Data collection was conducted by directly collecting data from one of the employee cooperatives at a health agency in Kediri City. The data is in the form of CSV files from the database, consisting of member data and credit application data from 2023 to 2024. There are 300 credit application data where the data is in raw condition. The interviews were conducted with cooperative management to determine the loan distribution criteria and to assess the loan applications. Loan service distribution is designed to offer and provide opportunities for eligible members to apply, while loan approval recommendation is used to provide a recommendation for loan applications when members submit their credit requests. The interview obtained that there are main features in assessing loan application in general including the amount of the loan, employee status, salary, amount of savings, loan history, and additional literature studies (Widyarini et al., 2020) in the form of gender, age, education, marital status, dependents, collateral, external information, and field conditions. From those 300 data points, 150 data points were selected to be used as a loan distribution dataset, and the other 150 data points to be used as a loan approval recommendation dataset. Each data point is assigned a creditworthiness status label for both loan distribution and loan approval recommendations in cooperation with management.

Based on the data collection, the demographic variables include gender, marital status, and educational attainment. The data comprises two sets: one for the loan approval model and another for the loan distribution model. In the loan approval dataset, which consists of 150 entries, 82% of the participants are male and 18% are female; 68% are married, while 32% are unmarried; and 82% are university graduates, whereas 18% have completed high school or lower levels of education. Similarly, in the loan distribution dataset, which also consists of 150 entries, 86% of the participants are male and 14% are female; 74% are married, while 26% are unmarried; and 88% hold a university degree, while 12% have completed only high school or lower.

Although this research follows the features used in literature studies, some data are not available. Data anticipation is carried out to increase the use of features on ML by replacing available data. Data anticipation involves replacing collateral data with property ownership information, and then external information data is replaced with the remaining repayment time.

While field condition data or condite and member age data cannot be entered into the feature because they are not yet available. In this study, the features used include 10 variables: loan amount, employee status, salary, loan history, marital status, education, number of dependents, property ownership, remaining repayment time, and gender.

SVM Model Development

Using the Waterfall Method for adoption, the stage of SVM Model Development includes Data Preprocessing, Training Implementation, Model Testing, and Model Deployment, as shown in Figure 4. The entire process of model development is done using the Python programming language.

Figure 4. SVM Model Development Method

Data Preprocess

The raw CSV data contains 300 credit application records for the period 2023-2024, which are divided into two datasets: the loan distribution dataset and the loan approval recommendation dataset. The loan distribution dataset is used to offer loans to members who meet the criteria, as determined by the ceiling used for loan distribution to potential members. Meanwhile, the loan approval recommendation dataset is used to assess the eligibility of loan applications from members, which is used to predict the possibility of bad credit. The raw data includes 10 features: loan amount, employee status, salary, loan history, marital status, education, dependents, property, remaining repayment period, and gender.

From the 300 data points in the raw dataset, the data is divided into two subsets, where 150 data points are used for the loan distribution dataset and 150 data points for the loan approval recommendation dataset. In the loan distribution dataset, seven features are used as independent variables, including gender, marital status, dependents, education, employee status, salary, and property. The loan status label is then used as the dependent variable, which is determined manually by cooperative management, with values of "appropriate" and "inappropriate". Meanwhile, the loan approval recommendation dataset utilises all 10 features, including loan amount, employee status, salary, loan history, marital status, education, dependents, property, remaining repayment period, and gender, as independent variables. Then, the loan status label is added as a dependent variable, which is determined according to the actual conditions, with values of "good" and "bad". Two datasets that are ready to use undergo several preprocessing methods, including cleaning, checking data types, and normalisation. The following is the detailed explanation.

a. Cleaning: Data cleaning is carried out where empty data or NaN is using the dropna() function.
b. Data type checking: Object and category data types are labeled as integer using the LabelEncoder() function.
c. Normalization: Normalization is carried out by checking for inappropriate data and incorrect scales manually.

Training Implementation

The SVM method is implemented using the Scikit-Learn SVM library, which utilises the SVC() function in detail. The kernel used to clarify the hyperplane uses the Polynomial kernel (Putri et al., 2021). SVM method training uses the fit() function where the dataset has previously been divided into two parts with 80% of the data for training and 20% of the data for testing (Putri et al. 2021; Asana & Yanti, 2023). The data division in the dataset uses the train_test_split() function with a test_size value of 0.2. This applies to both datasets, both the credit distribution dataset and the credit risk dataset. For testing, the predict() function is used to compare the prediction results with the actual labels. The SVM model is divided into two, including the SVM model for loan distribution and the SVM model for loan approval assessment.

Model Testing

Testing is carried out using the classification_report() function, which displays the accuracy value based on the K-Fold Cross Validation evaluation in the form of accuracy, precision, recall, and F1 score on the confusion matrix, as shown in Table 1. For the loan distribution model, the "appropriate" label indicates that the member meets the necessary conditions to apply for the loan offering, while the "inappropriate" label indicates the opposite. For loan approval recommendations, the "bad" label indicates that the member has an inappropriate condition for loan approval and poses an actual risk. In contrast, the "good" label means the opposite, indicating a low risk and thus approval is possible

Table 1. Confusion Matrix

Loan Distribution		
Real Label	**Prediction Result**	
	Appropriate	**Inappropriate**
Appropriate	True Positive	False Positive
Inappropriate	False Negative	True Negative
Loan Approval		
Real Label	**Prediction Result**	
	Bad	**Good**
Bad	True Positive	False Positive
Good	False Negative	True Negative

This section will also explain the test results on the two datasets used, namely the loan distribution dataset and the loan approval assessment dataset. The following is a detailed explanation of the test evaluating accuracy, precision, recall, and F1 score on the SVM model created (Alhamri et al., 2024).

a. Accuracy: The percentage of successfully predicted results according to the actual label. The higher, the more accurate the model classification prediction.
b. Precision: The ratio between accurate positive prediction results and all positive prediction results. The higher the precision, the more accurate it is in predicting positive results and minimising false positives.
c. Recall: The ratio between accurate positive prediction results and all positive labels. The higher the recall, the more accurate it is in predicting positive instances and minimising false negatives.
d. F1 Score: The average precision and recall, which produces a harmonic mean. The higher the F1 score, the more accurate it is in predicting positive results, even though the data is not balanced.

Model testing aims to obtain the performance of each machine learning method for classifying the data. Still using the scikit-learning library, the model testing can be done. The testing includes:

a. Accuracy means the percentage of labels that the model successfully predicted. If the model can classify 80 data accurately from 100 data then the accuracy is 0.80, as shown in Formula (1)

$$A = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Where TP is True Positive, TN is True Negative, FP is False Positive, and FN is False Negative. A high accuracy score means the model is generally more accurate.

b. Precision measures the ratio between true positive predictions and all positive result predictions. If the model can classify 80 data into positive prediction but only 75 data that true positive then the precision is 0.9375, as shown in Formula (2)

$$P = \frac{TP}{TP+FP} \quad (2)$$

Where TP is True Positive and FP is False Positive. A high precision score means a more accurate model to predict positive results.

c. Recall or sensitivity measures the ratio between true positive prediction and all positive data. If the model consists of 90 positive data but only 75 data that are true positive, then the recall is 0.83 as shown in Formula (3)

$$R = \frac{TP}{TP+FN} \quad (3)$$

Where TP is True Positive and FN is False Negative. A high recall score means the model correctly predicts positive instances.

d. F1Score is a comparison of the weighted average precision and recall that concludes a harmonic mean. The F1Score formula is shown in Formula (4).

$$R = \frac{2\times(R\times P)}{R+P} \quad (4)$$

Where R is Recall and P is Precision. A high F1score means a more accurate model to predict the positive results of a minority in an imbalanced data.

Model Development

This research develops two SVM models, including an SVM model for loan distribution and an SVM model for loan approval recommendation analysis. By using the joblib.The dump() function allows the Python model to be saved to a file. pkl format. Using the Python programming language, an ML server application was developed with the obtained SVM model. pkl is embedded in that server application. The user of the ML server application is the admin, serving as the cooperative manager. The looping function, using the while() function with an actual condition, will execute the SVM model command, including both the loan distribution SVM model and the loan approval recommendation SVM model. The two functions of the ML server application include:

a. In the loan distribution model function, both old and new data will be updated with the credit suitability classification by the learning machine periodically, so that the classification results depend on changes in the variables of cooperative members. The looping function is performed on all data.

b. In the loan approval recommendation model function, every time there is new data, an analysis will be carried out, while the old data is stored without any updates. The looping function is performed depending on new data.

There is one user who activates the server application of ML as a deployment model. This user has the function to activate the loan distribution model and activate the loan approval recommendation model. Figure 5 illustrates the use case diagram of the Support Vector Machine model server application, developed using Python programming.

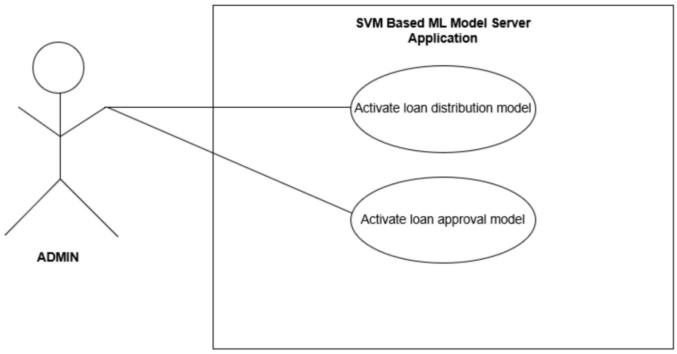


Figure 5. Python-Based SVM Model Server Application Use Case Diagram

There are three interrelated tables used to determine loan distribution decisions and loan approval recommendations, including the member table, credit distribution table, and credit risk table. Figure 6 is the relationship between tables used by the server application of Machine Learning.

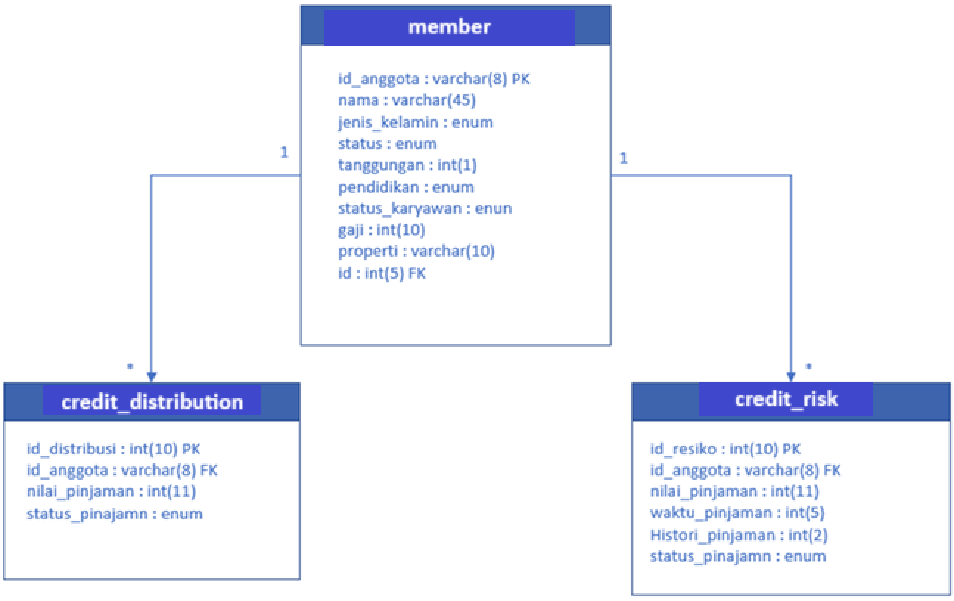


Figure 6. SVM Model Server Application Tables Relationship

Android-Based Application Development

The stage of Android-based Application Development, including Analysis, Design, Implementation, and Testing, is shown in Figure 7. The entire process of application development using the Dart language programming in the Flutter Framework.

Figure 7. Android-Based Application Development Method

Analysis

Application users consist of two users, namely Admin and Member. The administrator is the cooperative manager tasked with assessing and approving the feasibility of loan applications. At the same time, members are all cooperative members who have registered for the application. The functional requirements of the two users are as follows.

1. Admin
   a. Performing authentication: Used to log in and log out using the appropriate username and password
   b. Managing members: Can add member data and update member data related to credit data, including credit that has been given
   c. Managing loan applications: Can view and approve credit applications from members and view credit risk assessments as decision support
2. Member
   a. Performing authentication: Used to log in and log out using the appropriate username and password for those who have been registered on the application by Admin
   b. Obtaining credit offers: View the loan service distribution at certain ceilings and submit if the loan application is appropriate
   c. Managing loan applications: Make loan applications with custom values

Design

Figure 8 illustrates the Android-based application design using a use case diagram, according to the functional needs of each user.

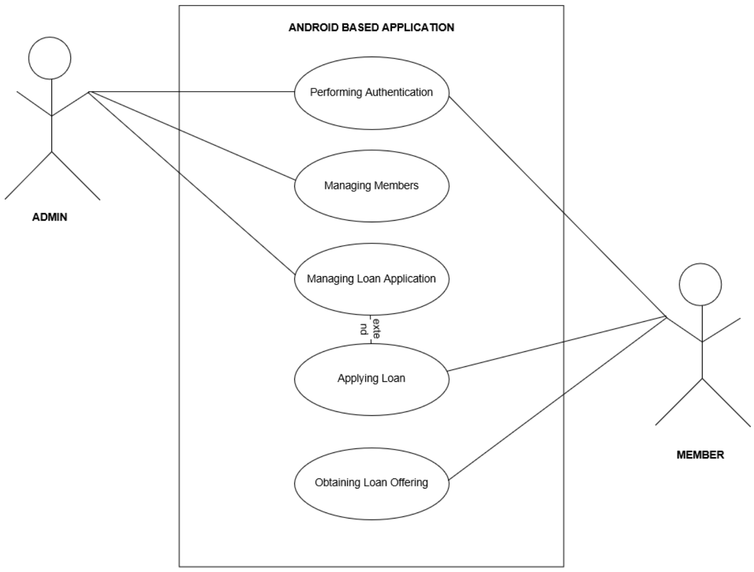


Figure 8. Android-Based Application Use Case Diagram

The following shows the implementation of an Android-based mobile application that supports loan application decisions for users, including Admin and Member. The implementation is carried out with a mobile application connected to a desktop application for SVM-based ML model server through data transactions on a MySQL Database locally, with a system architecture as shown in Figure 9.

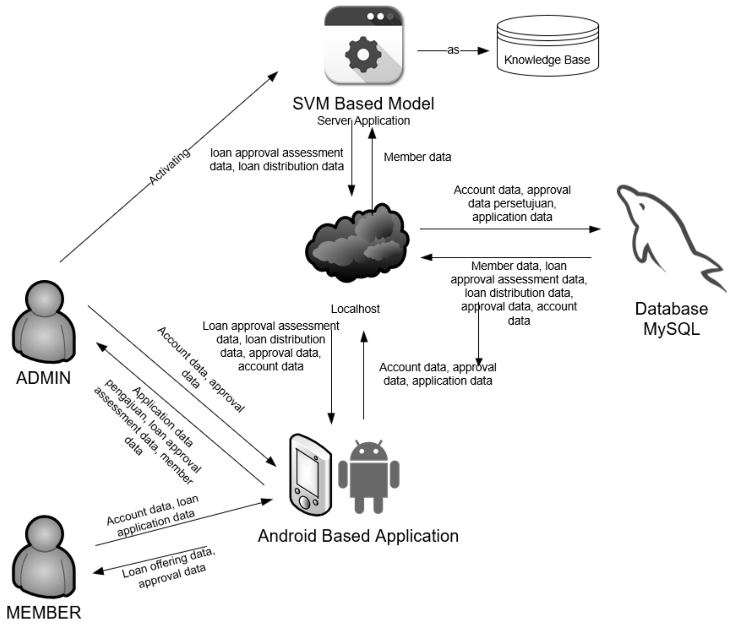


Figure 9. System Architecture

Implementation

The implementation stage involved coding for Android mobile devices using the Flutter framework and configuring local networks so that the Android-based application, acting as a client, could connect locally with the SVM-based model, which was hosted as a server, and save data in a MySQL Database.

Testing

Application functional testing refers to the application functional requirement at analysis stage. The testing scenario is explained in Table 2.

Table 2. Android Based Application Functional Testing

No	Function	Scenario	Expected Result
1	Performing authentication	Logging in account	Sucessfully logging in and showing the landing page
2	Managing members	Creating, reading, updating, and deleting member	Successfully create, read, update, and delete member data
3	Managing loan application	Reading loan application, reading loan approval recommendation, and approving loan application	Successfully read new loan application, reading approval recoomendation, and approving the loan application
4	Applying loan	Creating loan application through loan offering or loan custom	Successfully customizing the loan value and applying it
5	Obtaining loan offering	Reading loan service offering and applying the loan	Successfully read loan service offering and appliyng it

FINDINGS AND DISCUSSION

The findings will be presented in two separate chapters: the findings of the SVM model and the Android-based application. The discussion will focus on the system's deficiencies, particularly the deployment of Android applications.

Findings of the SVM Model

Model Testing

Figure 10 presents the results of testing the SVM model on the loan distribution dataset, where the accuracy reaches 85%. However, the precision percentage is only 45%, the recall is 47%, and the F1 score is 46%. Moreover, Figure 11 presents the results of testing the SVM model on the loan approval recommendation dataset, achieving an accuracy of 90%. However, the precision percentage is only 45%, the recall is 50%, and the F1 score is 47%.

	precision	recall	f1-score	support
Appropriate	0.89	0.94	0.92	18
Inapproriate	0.00	0.00	0.00	2
accuracy			0.85	20
macro avg	0.45	0.47	0.46	20
weighted avg	0.81	0.85	0.83	20

Figure 10. Result of The SVM Model Testing Loan Distribution

	precision	recall	f1-score	support
Good	0.90	1.00	0.95	18
Bad	0.00	0.00	0.00	2
accuracy			0.90	20
macro avg	0.45	0.50	0.47	20
weighted avg	0.81	0.90	0.85	20

Figure 11. Result of The SVM Model Testing Loan Approval Recommendation

The appearance of the confusion matrix for testing the SVM model on loan distribution is shown in Figure 12, and the appearance of the confusion matrix for testing the SVM model on loan approval recommendations is shown in Figure 12.

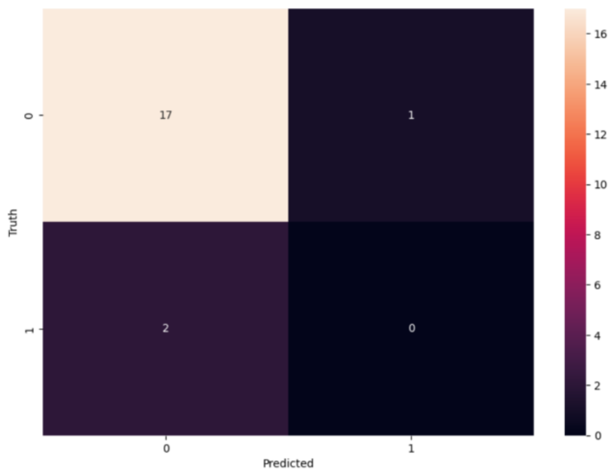


Figure 12. Confusion Matrix of the SVM Model Loan Distribution

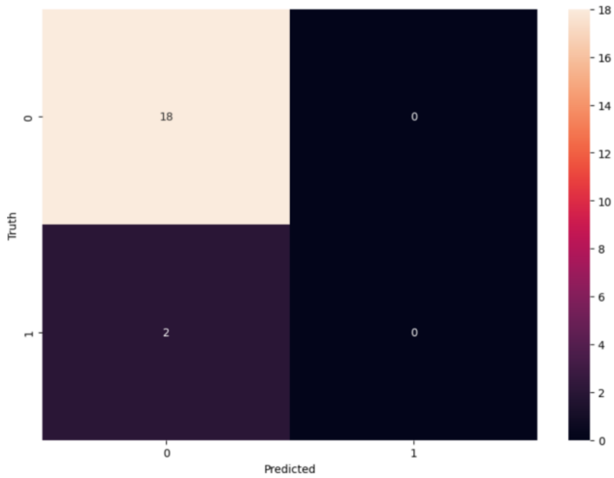


Figure 13. Confusion Matrix of the SVM Model Loan Approval Recommendation

SVM Model Server

The database implementation using MySQL is illustrated in Figures 14, 15, and 16. To determine the loan distribution, a join is performed between the member table and the credit_distribution table, where the loan_status becomes the classification result value. Meanwhile, to analyse the loan approval recommendations for member applications, a join is used between the member table and the credit_risk table, with the loan_status column serving as the predicted result value.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_anggota	varchar(8)	utf8mb4_general_ci		No	None			Change Drop More
2	nama	varchar(45)	utf8mb4_general_ci		No	None			Change Drop More
3	jenis_kelamin	enum('Laki-laki', 'Perempuan')	utf8mb4_general_ci		No	None			Change Drop More
4	status	enum('Belum kawin', 'Kawin')	utf8mb4_general_ci		No	None			Change Drop More
5	tanggungan	int(1)			No	None			Change Drop More
6	pendidikan	enum('Sekolah menengah', 'Perguruan tinggi')	utf8mb4_general_ci		No	None			Change Drop More
7	status_karyawan	enum('ASN', 'Honorer')	utf8mb4_general_ci		No	None			Change Drop More
8	gaji	int(10)			No	None			Change Drop More
9	properti	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
10	id	int(5)			No	None			Change Drop More

Figure 14. Member

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_distribusi	int(10)			No	None		AUTO_INCREMENT	Change Drop More
2	id_anggota	varchar(8)	utf8mb4_general_ci		No	None			Change Drop More
3	nilai_pinjaman	int(11)			No	None			Change Drop More
4	status_pinjaman	enum('Sesuai', 'Tidak sesuai')	utf8mb4_general_ci		No	None			Change Drop More

Figure 15. Credit Distribution

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_resiko	int(10)			No	None		AUTO_INCREMENT	Change Drop More
2	id_anggota	varchar(8)	utf8mb4_general_ci		No	None			Change Drop More
3	nilai_pinjaman	int(10)			No	None			Change Drop More
4	waktu_pinjaman	int(5)			No	None			Change Drop More
5	histori_pinjaman	int(2)			No	None			Change Drop More
6	status_pinjaman	enum('Bagus', 'Buruk')	utf8mb4_general_ci		No	None			Change Drop More

Figure 16. Credit Approval

The Python-based Desktop Agent Application is an ML server application that continuously classifies loan distribution to members and assesses credit risk for members applying for loans through loan approval recommendations. It can help Android-based applications run the functional requirements. The server application is developed using Python with a desktop platform, as shown in Figure 16. The server application, as an ML model, runs two models: the loan distribution SVM model and the loan approval recommendation SVM model. The SVM model is obtained from the training results in the previous stage using the joblib library's dump () function, where the output of the SVM model is in .pkl format.

Figure 17. View of ML Server Application

Findings of Android-Based Application

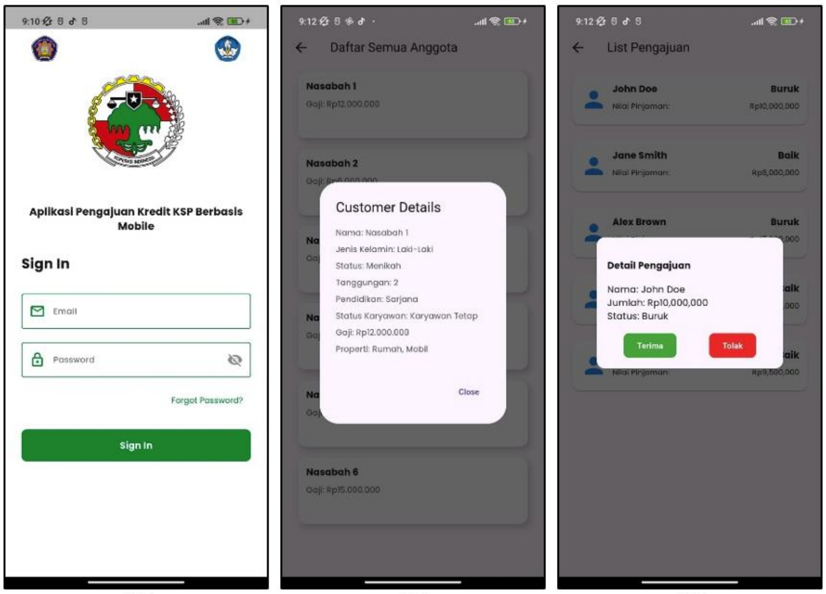
Application Result

The Android-Based application view of the admin user is shown in Figure 18, including:

- Performing authentication in Figure 18(a)
- Managing members in Figure 18(b)
- Managing loan application by approving and retrieving the loan approval recommendation in Figure 18(c).

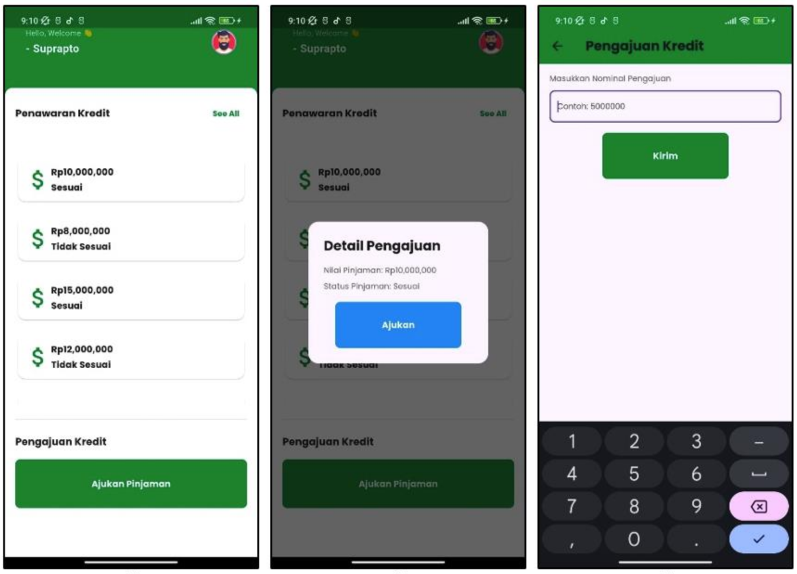
While the view of the member user is shown in Figure 19, including:

- Performing authentication in Figure 19(a).
- Obtaining loan service offers in Figure 19(b).
- Managing the loan application by applying the loan in Figure 19(c).



(a) (b) (c)

Figure 17. The View of Android-Based Application for Admin



(a) (b) (c)

Figure 18. The View of Android-Based Application for Member

Application Functional Testing

Table 3 is application functional testing result that refers to the application functional requirements at analysis stage.

No	Function	Scenario	Testing Result
1	Performing authentication	Logging in account	Success
2	Managing members	Creating, reading, updating, and deleting members	Success
3	Managing a loan	Reading the loan application, reading the loan	Success

No	Function	Scenario	Testing Result
	application	approval recommendation, and approving a loan application	
4	Applying loan	Creating loan application through a loan offering or a loan custom	Success
5	Obtaining a loan offering	Reading loan service offering and applying for the loan	Success

Discussion

Based on the results of this study on the SVM algorithm test for the ML model, the accuracy obtained for the loan approval model reached 90%, while for the loan distribution model, it reached 85%. This result supports the hypothesis, which states that the use of the SVM algorithm for the ML model in the approval and distribution of KSP loans achieves a rate of more than 75%. Compared to the research by Putri et al. (2021), which achieved an SVM accuracy of 95.08% and Pernama et al. (2023) with an accuracy of 92%, this study has a lower accuracy. While this study achieves the same results as Asana and Yanti's (2023) study, the SVM accuracy reached 85%, surpassing the accuracy of the loan distribution model and outperforming the loan approval model.

The loan distribution model testing result showed high accuracy, but low precision, recall, and F1 score. This means that the SVM model has high accuracy because the number of datasets is appropriate; however the number of positive and negative labels remains unbalanced. The data imbalance occurs because the resource cooperative more often approves loans positively for members than negatively. So that the "appropriate" label data is more than the "inappropriate" label data. High accuracy with less balancing is still used as an SVM model in loan distribution on the ML model created. Like loan distribution model testing, loan approval recommendation testing results also achieved high accuracy, but low precision, recall, and F1 score. This means that the SVM model has high accuracy because the number of datasets is appropriate, but the number of positive and negative labels is not balanced. The data imbalance is due to the creditworthiness assessment by the source cooperative, which more often results in negative risk decisions (loan rejections) for member applications than in positive risk decisions (loan approvals). So that the "good" label data exceeds the data with the "bad" label. High accuracy with less balancing is still used as an SVM model in the loan approval recommendation created by ML.

CONCLUSIONS

An Android application has two users: the first is Admin, who can manage members, retrieve loan approval recommendations, and manage loan applications; the second is member, who can retrieve loan distributions and apply for loans. The performance of the ML using SVM includes the accuracy of loan approval recommendation reaching 90%, while loan distribution reached 85%. The hypothesis is true that the accuracy of an ML model using an SVM algorithm for determining loan approval and loan distribution is more than 75%. By successfully developing a mobile application for loan approval and distribution using ML, where the accuracy of assessing the eligibility of members' loans based on historical data is relatively high, the management of savings and loan cooperatives (KSP) can quickly determine loan approval from members. The impact is that the business process for loan services at KSP can be faster and reduce the failure to pay for applicants. Additionally, the distribution of loan information to potential cooperative members based on historical data will also increase KSP's income from loan services.

LIMITATION & FURTHER RESEARCH

The limitations of this study include limited data availability, the use of a polynomial kernel SVM algorithm, and the functional requirements of the mobile application regarding the specific business processes of the KSP. For further research, it is necessary to collect more data about loan and credit applications in cooperatives, especially KSP. More than one source cooperative will produce balanced data with more than 300 data points. Further research can also involve a study on system implementation, allowing the system to be accessed on the internet by integrating cloud computing.

ACKNOWLEDGEMENT

Thank you to the Ministry of Higher Education, Research, and Technology, Republic of Indonesia, for providing financial support that enabled this research to be carried out. Thank you also to P3M Politeknik Negeri Malang for providing guidance that facilitated the smooth execution of the research.

REFERENCES

Akbar, R., Putra, A., & Agussalim. (2024). Rancang bangun aplikasi koperasi simpan pinjam berbasis Android menggunakan React Native. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *8*(3), 4135–4142. https://doi.org/10.36040/jati.v8i3.9855

Alhamri, R. Z., Cinderatama, T. A., Eliyen, K., & Izzah, A. (2024). Supervised learning methods comparison for Android malware detection based on system calls referring to ARM (32-bit/EABI) table. *Journal of Information Technology and Cyber Security*, *2*(1), 15–24.

Asana, I. M., & Yanti, N. P. (2023). Sistem klasifikasi pengajuan kredit dengan metode support vector machine (SVM). *Jurnal Sistem Cerdas*, *6*(2), 123–133.

BPS Kota Kediri. (2018, March 23). *Jumlah koperasi menurut jenis koperasi dan kecamatan di Kota Kediri*. https://kedirikota.bps.go.id/statictable/2018/03/23/87/jumlah-koperasi-menurut-jenis-koperasi-dan-kecamatan-di-kota-kediri-2016.html

Cabitza, F., Campagner, A., Zotti, F. D., Ravizza, A., & Sternini, F. (2020). All you need is higher accuracy? On the quest for minimum acceptable accuracy for medical artificial intelligence. In *International Conference on e-Health* (pp. 159–166). Zagreb: IEEE.

Dharmayanti, N., Putra, C., & Ayu, P. (2023). Pengaruh pemanfaatan teknologi informasi, partisipasi pemakai sistem informasi akuntansi, dan kemampuan teknik pemakai terhadap efektivitas sistem informasi akuntansi di koperasi simpan pinjam se - Kecamatan Sukawati. *Hita Akuntansi dan Keuangan*, *4*(3), 324–335. https://doi.org/10.32795/hak.v4i3.3876

Dwiana, O., & Sari, R. (2022). Analisis implementasi penilaian kesehatan keuangan pada koperasi simpan pinjam di Kota Kediri. *Jambura: Economic Education Journal*, *4*(2), 142–153. https://doi.org/10.37479/jeej.v4i2.13805

Pemkot Kediri. (2021, March 15). *Dekopinda Kota Kediri diminta hadirkan koperasi berbasis teknologi*. https://www.kedirikota.go.id/p/dalamberita/8539/dekopinda-kota-kediri-diminta-hadirkan-koperasi-berbasis-teknologi

Pernama, B., & Purnomo, H. (2023). Analisis risiko pinjaman dengan metode support vector machine, artificial neural network dan naïve Bayes. *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, *7*(1), 92–99. https://doi.org/10.35870/jtik.v7i1.693

Prahartiwi, L. I., & Fatimah, N. (2023). Sistem pendukung keputusan pemberian kredit pada Koperasi Karyawan Aneka Pangan Nusantara menggunakan metode SAW. *Jurnal Teknik Komputer AMIK BSI*, *9*(1), 33–40.

Putra, Y. L. (2023, July 12). *Harkopnas ke-76, Dekopinda Kota Kediri berharap koperasi jadi pengungkit ekonomi*. https://timesindonesia.co.id/indonesia-positif/461065/harkopnas-ke76-dekopinda-kota-kediri-berharap-koperasi-jadi-pengungkit-ekonomi

Putri, N., Fatekurohman, M., & Tirta, I. (2021). Credit risk analysis using support vector machines algorithm. *Journal of Physics: Conference Series*, *1836*(1), 012039. https://doi.org/10.1088/1742-6596/1836/1/012039

Riyadi, S., Siregar, M. M., Margolang, K. F., & Andriani, K. (2022). Analysis of SVM and Naive Bayes algorithm in classification of bad loans in save and loan cooperatives. *JURTEKSI*, *8*(3), 261–270.

Saputra, A. S. (2021). Pengaruh teknologi informasi pada koperasi di era industri 4.0. *Transekonomika: Akuntansi, Bisnis dan Keuangan*, *1*(5), 505–510. https://doi.org/10.55047/transekonomika.v1i5.77

Tamami, M., & Kharisudin, I. (2023). Komparasi metode support vector machine dan naive Bayes classifier untuk pemodelan kualitas pengajuan kredit. *Indonesian Journal of Mathematics and Natural Sciences*, *46*(1), 38–44. https://doi.org/10.15294/ijmns.v46i1.46174

Widyarini, L. A., Agung, D. A., & Agrippina, Y. R. (2020). Penguatan sistem dan prosedur pinjaman di Koperasi Saka Tata Makmur di Kediri dengan memanfaatkan sistem informasi manajemen. *PeKA: Jurnal Pengabdian Kepada Masyarakat*, *3*(2), 109–125. https://doi.org/10.33508/peka.v3i2.3000