Optimizing Procurement Efficiency by Implementing K-Means and Random Forest in Kopegtel Samarinda’s Warehouse System

Fernando Nikolas R[1]*, Islamiyah[2], Vina Zahrotun Kamila [3]
Department of Information Systems, Department of Engineering Faculty, Mulawarman University [1], [2], [3]
Samarinda, Indonesia
fnikolabs@gmail.com[1], islamiyah.unmul@gmail.com [2],vinakamila@ft.unmul.ac.id[3]

Procurement is a company’s activity to purchase goods or equipment needed in operations. In the management process, a procurement management system is often used to facilitate this management, such as at CV Indocitra Multi Artha, which uses the “Sistem Warehouse Kopegtel Samarinda.” The system provides significant assistance to the company, but large requests can be overwhelming to be handled by the manager and can cause an overload information problem. Research was conducted to deal with these problems by implementing a data mining algorithm as a procurement recommendation system. K-means and Random Forest algorithms were chosen as methods for the research. The algorithm is processed within two critical steps, first by K-Means to get cluster data and then by predicting it with Random Forest to get a recommendation for whether the object should be bought or not. Hyperparameter tuning was performed to optimize the model’s performance, yielding an F1-Score of 86.95%, representing the balance between precision and recall, and an ROC AUC value of 82.34%. These substantial metric outcomes indicate that the model can provide practical recommendations.

Keywords— CRISP-DM, K-Means, Procurement, Random Forest, Warehouse

I. Introduction

The process by which a business buys the products, equipment, supplies, and services it needs is known as procurement [1]. For CV Indocitra, a company that works in the construction and telecommunications industries, the procurement procedure is essential. In order to avoid project operational delays, procurement activities must be accelerated as the company grows [2]. Making decisions is a major obstacle that comes up during this process, especially when management is presented with large requests that might cause information overload [3].

One approach to handling this issue is using machine learning algorithms [2]. A number of studies indicate that integrating machine learning into the procurement process can lead to enhancements in business value [4]. Among the algorithms utilized for procurement recommendations are K-Means and Random Forest [4].

This study employs both K-Means and Random Forest algorithm as part of its methodological framework. The selection of these algorithms is informed by existing literature that indicates a notable enhancement in predictive accuracy, with reported improvements from 88% to 99.86%, as well as an increase in the F1-Score from 88% to 100% when utilizing a hybrid approach that combines K-Means and Random Forest, in contrast to the application of Random Forest in isolation [5]. Furthermore, this research is also supported by other comparative analyses conducted by Elzeheiry et al. and Brahmana et al. that showed K-Means and Random Forest algorithms produced better metric values than other models tested [6],[7]. In the modeling process, the SciKit-Learn library is used. This library was chosen because it offers a variety of data mining algorithms with a simple and easy-to-use interface [8],[9],[10].

Based on the description above, a study has been undertaken with the objective of developing a procurement recommendation system using K-Means and Random Forest. This research is titled " Optimizing Procurement Efficiency with K-Means and Random Forest in Kopegtel Samarinda Warehouse System."

II. Research Methodology

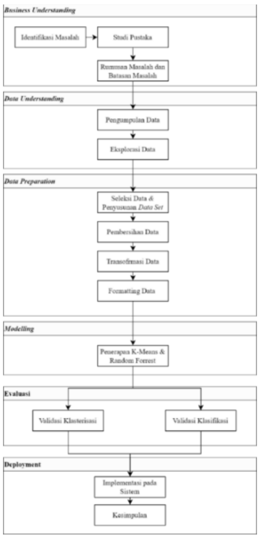


Fig. 1. The Research Process

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v13i3.2288, Copyright ©2024
Submitted : October 3, 2024, Revised : November 5, 2024, Accepted :November 9, 2024 , Published : November 21, 2024

The CRISP-DM method, which has multiple steps from business knowledge to deployment, is used in this study. The figure below illustrates the research process

A. *Data Collection*

Data Collection is one stage of data understanding in CRISP-DM. This process involves collecting data from Warehouse Kopegtel Samarinda's database. By employing join queries, multiple tables are integrated to yield procurement historical data recorded between July 2022 and April 2024.

B. *Data Exploration*

Data exploration is another stage in data understanding in CRISP-DM. This process involves visualizing the data to obtain a comprehensive and descriptive overview of the dataset under consideration. Various libraries, including Pyplot and Seaborn, are employed to carry out this process.

C. *Feature Selection*

After data is collected through the data understanding process, the next step is data preparation. Data preparation plays a vital role in the CRISP-DM process and can affect the result of the model [11]. Feature selection is one of the stages in data preparation. This process is carried out by selecting features in the data that has been collected to determine what features should be included and not [12]. The exclusion of certain feature from the dataset was informed by the outcomes of interviews conducted with representatives from the company.

D. *Cleaning Data*

Cleaning data is the next step conducted after feature selection process. The aim of this stage is to fix or remove incorrect, corrupted, duplicate, or incomplete data within a dataset [11]. In this research, several processes are conducted, as explained below.

1) Handling Missing Value

This process is carried out by filling empty data using several strategies, such as median, mode, and average, or more sophisticated methods, like the imputer library available at Scikit-Learn. [13], [14].

2) Handing Duplication

This process is conducted to minimize the chance for the model to be overfitting [15]. In this stage, a duplicate row is removed, leaving only one unique combination for each row. This process is carried out by removing a duplicated row in the dataset [11]

E. *Data Transformation*

Creating new variables or merging variables based on ones that already exist in the dataset are two methods used to modify data. As explained below, a number of procedures are being carried out.

1) Adjust Feature Datatype

This state is carried out by converting the value for each feature to categorical or numerical. In this research, “created_at” is one of the features that will be converted. As a feature with the DateTime data type, this feature will be converted to numerical by converting datetime to epoch timestamp, leaving only a numerical value for it.

2) Labelling

Labeling is a critical stage in this research. There are two features that need to be validated. One is a priority, which consists of numbers 1 to 4 that determine this procurement item’s priority. Moreover, the other is a label that is a historical value for procurement items that signifies that this item is labeled as “penting” or “tunda.” This process is conducted by interviewing and observing CV Indocitra Multi Artha.

F. *Formatting Data*

Formatting data is carried out by adjusting the data based on the model to be used. There are several processes in the data formatting stage, such as encoding and normalization [16].

1) Encoding

2) The encoding process uses one-hot encoding, changing categorical variables into several binary variables [16]. This process is done using the “get_dummies” method from pandas.

3) Normalization

The normalization process is carried out using standard normalization or z-score normalization [8]. This process is done using the StandardScaller function already in the Scikit-Learn library.

G. *Modelling*

Modelling is the process of implementing the algorithm to the processed dataset. At this stage, several processes are performed, which can be seen in Fig 2.

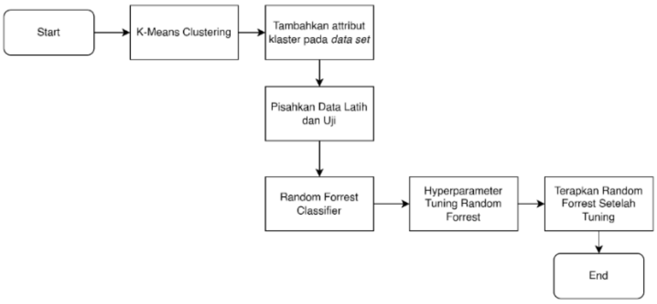

Start
K-Means Clustering
Tambahkan attribut klaster pada *data set*
Pisahkan Data Latih dan Uji
Random Forrest Classifier
Hyperparameter Tuning Random Forrest
Terapkan Random Forrest Setelah Tuning
End

Fig. 2. Modelling Process

1) K-Means Clustering

In this stage, data is being processed using the K-Means algorithm. This process is done using the KMeans method available in the Scikit-Learn Library. The result from this clustering will be stored in a feature called cluster in the dataset [16].

2) Train Test Split

In this stage, data is split into two types: train data and test data. Train data will be used for training data for Random Forest later, and testing data will be used to determine model effectiveness by using some well-known evaluation method [16].

3) Random Forest Classification

A Random Forest classification algorithm is applied to the train data at this stage. This process is done using the RandomForestClassifier method from Scikit-Learn [16].

4) Hyperparameter Tuning

This stage is being done to find the most optimal hyperparameter configuration that can be used to get the highest possible metric value. AUC and Accuracy score is used as a metric for this hyperparameter tuning stage.

H. Model Evaluation

Model evaluation is carried out to measure the model or algorithm and determine whether its quality meets the research's initial objectives [12], [17]. Both classification and clustering have different metrics, so they are relative to the model type.

1) K-Means Evaluation

K-means evaluation is done by comparing the intrinsic and extrinsic quality of the clusters. This research will use several metrics. Davies-Bouldin Index, the Silhouette Coefficient for intrinsic measurements, Jaccard Index, and the Folkes-Mallows Score for extrinsic measurements [16].

a) Davies-Bouldin Index

A cluster's density in proportion to the mean separation between other groups is depicted by the Davies-Bouldin index [16]. The metric value is the sum of the highest values for all existing ratios [18]..

b) Silhouette Coefficient

A measurement of the cohesion and distance between clusters is called the silhouette coefficient. Its foundation is the discrepancy between the average distance of points inside the same cluster and the largest average distance at points close to the cluster [16]. The average silhouette coefficient value for every point is used to get this metric value [16].

c) Rand Statistics

Rand statistics measure the fraction of true positive and true negative pairs at all defined points. The higher the value of the Rand statistic, the better; the maximum value for this metric is one [16].

d) Folkes-Mallows Measure

The Folkes-Mallows Measure is the geometric mean of pairwise precision and recall. Its highest value is one, which indicates the absence of false positives and negatives [16].

2) Random Forest Evaluation

Several evaluation metrics are often used for random forest evaluation, like accuracy, error rate, recall, sensitivity or true positive rate, specificity or true negative rate, precision, and F1-Score [16], [19] .

a) Accuracy

Accuracy is defined as the value for true prediction divided by the total item in the training dataset [16]. It shows how well the model correctly predicts the outcome [19].

b) Precision

Precision is defined as the value for true positive divided by true positive and false positive[16]. It shows how well the model correctly predicts positive classes in total [19].

c) Recall

Recall is defined as the value for true positive divided by true positive and false negative [16]. It shows the ability of the model to correctly identify true positives from all positive predicted samples [19].

d) F1-Score

F1-Score is the trade-off ratio between the precision and recall of a classifier [16]. It is measured by doing a weighted average between precision and recall [19].

e) AUC ROC Score

The area under the ROC score curve is known as the AUC ROC score. Receiver Operating Characteristics, or ROC, is a metric that is calculated by mapping every conceivable threshold value that currently exists. A classifier must produce a score value for the positive class for every point in the testing set in order to perform ROC analysis. The classifier algorithm's prediction accuracy increases with its AUC value.[16].

I. System Implementation

The system implementation or deployment stage includes the process of applying an algorithm that has been evaluated into a finished product that the company can use. The algorithm is implemented on the backend using Django. Each component used in program implementation is planned to be connected, as in Fig 3. This stage is divided into three processes: exporting the model, creating an API for the backend, and integrating the model into the existing system.

1) Exporting Model

Exporting a model is conducted using the joblib library from Python. This library enables the model to be saved and reused in another environment.

2) Creating Backend API

The creation of a backend API is being conducted using Django, a Python web framework that provides a robust component for web development yet is small in size. Django is being used in this research to adjust the machine learning-

Submitted : October 3, 2024, Revised : November 5, 2024, Accepted :November 9, 2024 , Published : November 21, 2024

based language that is being used, Python, so problems like dependency issues can be minimalized.

3) Integrating Model to Existing System

This stage is conducted by creating a new menu in an existing system, which in this case is Kopegtel Samarinda's Warehouse System. The process in this stage includes creating a front-end for predicting and calling the API that has been created before.

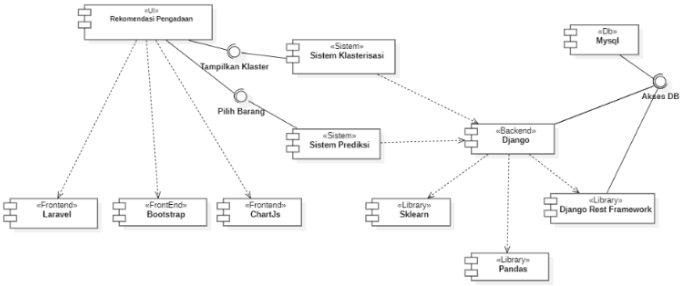


Fig. 3. Model Deployment System Architecture

III. RESULT AND DISCUSSION

In this research, the implementation process begins with collecting request history data on the Kopegtel Samarinda Warehouse System database, which produces 5660 rows of data from July 2022 to mid-April 2024. The data exploration results show that there were 2495 requests made by CV Indocitra Multi Artha, of which 2097 requests were made. Excluding companies, there were 678 requests made by Kopegtel and 390 by PT Putra Bistel Solusindo as seen in Fig 4. This figure shows that that Indocitra is requesting more procurement.

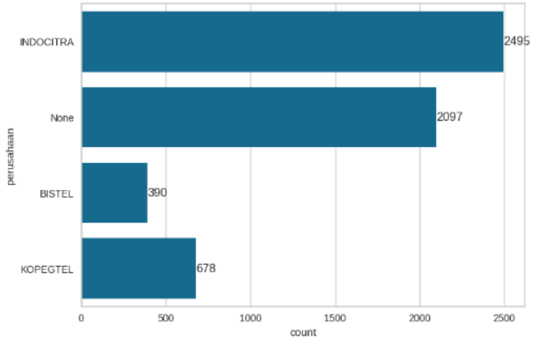


Fig. 4. Company Composition from Dataset

Based on the requests each year, were 2330 requests made in 2022, 2326 requests made in 2023, and 1004 requests made in 2024 (until April 2024). The details and visualization can be seen in Fig 5.

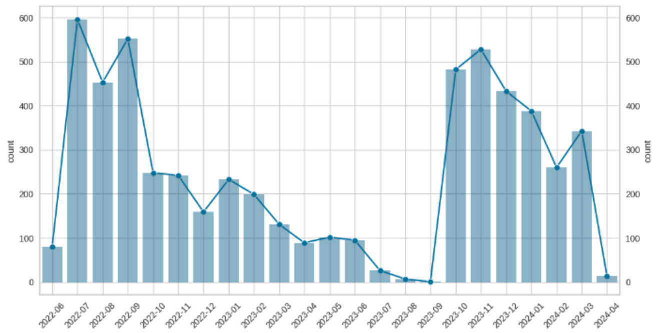


Fig. 5. Composition of Request Each Month

In Fig.5, there is a downtrend in requests at the end of 2023. Based on observations, this happened because the system was migrated from the old Warehouse System to the new Warehouse System. This causes some data to be stored in offline files like Excel and not in the database.

After the data collection process, the data cleaning and transformation process is carried out. The data cleaning process is carried out in several stages, from handling missing values and data redundancies to simplifying variable values, handling data errors, feature extraction, feature selection, and filling in "priority" variables and labeling. Through this process, a dataset is produced, as in Table 2.

TABLE I. SAMPLE DATASET

price	***qty_diminta***	**nama_barang**	**…**	**...**	...	**perusahaan**	**prioritas**	**label**
539 000	10	tiang besi 7m	..	..	..	INDO CITRA	4	1
235 000	8	helm biru	..	..	..	STAKEHOLDER	1	0
....	....	....	..	..	..	....	....	....
....	....	....	..	..	..	....	....	....
90 850	1	35 mikro +3 micro pin 3	..	..	..	INDO CITRA	4	1

After the data cleaning and transformation process, the data is processed again at the data formatting stage. The process includes normalization and encoding. Normalization is carried out with Z-Score Normalization on numerical variables such as "price," "timestamp," and "qty_dminta" to produce normalization results as in Table 3.

TABLE II. NORMALIZATION RESULT

sum	**harga**	**qty_diminta**	***timestamp***
-0.0403	0.3499	-0.0737	-1.3768
-0.0420	0.0132	-0.0737	-1.3600
-0.0437	-0.2837	-0.0753	-1.3599
...	...	...	...
...	...	...	...
0.9084	0.2639	2.4177	1.6110

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v13i3.2288, Copyright ©2024
Submitted : October 3, 2024, Revised : November 5, 2024, Accepted :November 9, 2024 , Published : November 21, 2024

-0.0429	-0.1933	-0.0705	1.6255

The modeling process begins with K-means clustering. The cluster is set to 4 to adjust the number of possible values in the “priority” feature. This process results in a cluster label that will later be stored in the dataset as a new feature. Overall, this process can be seen from the code below.

```
from sklearn.cluster import KMeans
x = pd.get_dummies(x)
kmeans = KMeans(n_clusters=4, random_state=42)
kmeans.fit(x)
x['cluster'] = kmeans.labels_
```

After the clustering process, the data is divided into train and test data. The proportion between train and test data is 80:20: 4508 rows for training data and 1128 rows for test data. After separating train and test data, a random forest is implemented using the RandomForestClassifier from the Scikit-Learn library. For starters, no specific hyperparameter will be defined, and this result will be compared to hyperparameter tuning later on.

```
# Train Test Split
x_train,x_test,y_train,y_test                                            =
train_test_split(x,y1,test_size=0.2,random_state=42)

# Random Forest
rf = RandomForestClassifier()
rf.fit(x_train,y_train)
y_pred = rf.predict(x_test)
```

After fitting the dataset to the Random Forest algorithm, a hyperparameter tuning is carried out to compare and observe if there is any improvement after implementing that. This process was carried out using GridSearchCV, which contains the Scikit-Learn library with a list of search items, as in Table 4.

TABLE III. HYPER PARAMETER TUNING SEARCH GRID LIST

Hyperparameter	Search Value
n_estimator	[200,300,400,500]
max_features	[‘auto’,’sqrt’,log2’]
max_depth	[4,5,6,7,8]
min_samples_split	[2,3,4,5]
min_samples_leaf	[1,2,3,4,5]

After implementing hyperparameter tuning, we evaluated model performance for both K-Means and Random Forest. In the K-Means clustering evaluation, the silhouette coefficient value was 0.268. This value is obtained from the average silhouette index for each point.

$$SC = \frac{1}{n}\sum_{i=1}^{n} s_i = \frac{s_1 \ + s_2 \ + \cdots + \ s_{5634} \ + \ s_{5635}}{5635}$$
$$= \frac{0.350 \ + 0.307 \ + \cdots + \ 0.174 \ + \ 0.433}{5635}$$
$$= \ 0.268$$

For Davies-Bouldin metric, it shows a value of 1.681, this obtained from the ratio of the distance between clusters to the intercluster distance of each combination of points.

$$DB = \frac{1}{4}\sum_{i=1}^{k} \max_{j\neq i}\{F_{Ch}\} \ = \ \frac{1}{4}\ (1.892 + \ 1.475 + \ 1.892 + \ 1.469)$$
$$= \ 1.681$$

Extrinsic cluster measurements were also carried out on the model. Two metrics are used in the measurements Rand Index, which has a value of 0.568, and the Folkes Mallows Measure, which has a value of 0.432.

$$Rand\ Index = \frac{TP \ + \ TN}{N} = \frac{5006034 + \ 13039246}{31758860} = \ 0.568$$

$$Folkes\ Mallows\ Score$$
$$= \frac{5006034}{\sqrt{(5006034 + 9443972)(5006034 + 4269608)}} \ = \ 0.432$$

Both silhouette and David Poulin’s scores show low segregation and separation for each item in the cluster. That can be seen in Silhouette’s score below 0.5 and David Bouldin’s score higher than 1. The result in the extrinsic cluster also shows a low score with a value lower than 80, indicating that the cluster pattern still does not follow the ‘ground truth.’.

For the RandomForest models, several metrics are being used, such as accuracy, precision, recall, specificity, F1-score, and ROC AUC. During the hyperparameter tuning phase, the values derived from the confusion matrix were as follows: true positives amounted to 763, true negatives to 136, false positives to 137, and false negatives to 92, as illustrated in Fig 6..

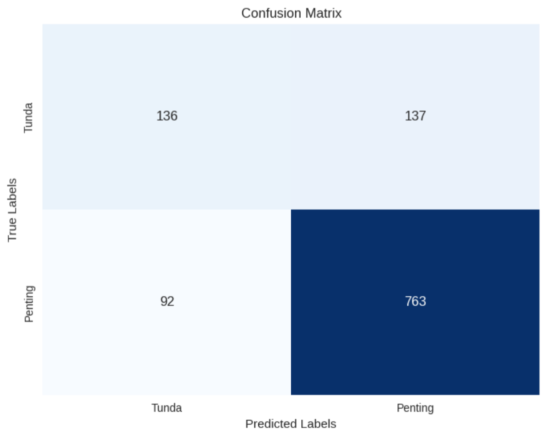


Fig. 6. Confusion Matrix Random Forest

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v13i3.2288, Copyright ©2024
Submitted : October 3, 2024, Revised : November 5, 2024, Accepted :November 9, 2024 , Published : November 21, 2024

According to the confusion matrix, an accuracy value of 79.69%, precision of 20.30%, recall of 89.23%, and F1-Score value of 86.95% were obtained. This value is obtained based on the following calculations:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}} = \frac{763 + 136}{1128} = 0.7969$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{763}{763 + 137} = 0.8477$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{763}{763 + 88} = 0.8924$$

$$\text{F1} - Score = \frac{2.\,\text{Precision}\,.\,\text{Recall}}{\text{precision} + \text{recall}}$$

$$= \frac{2(0.8477)\ (0.8924)}{(0.8477) + (0.8924)} = 0.8695$$

Accuracy results in random forest tell a valid accuracy between label and prediction. This result was also followed by considerably good results in precision and recall, which tells that the model can tell true positive and false positive, which implies a good F1-Score metric. Then, a ROC curve evaluation was carried out using the "predict_proba" method, and an AUC measurement was performed using "roc_auc_score" in the Sci-Kit Learn library. The result of this measurement is an AUC value of 0.8234 or 82.34%, and the results and visualization of these metrics can be seen in Fig 7.

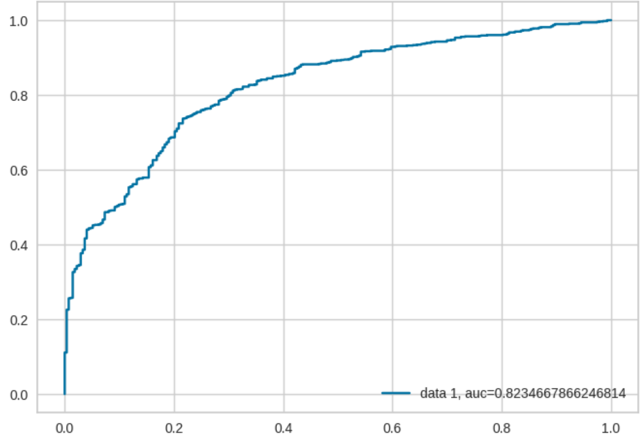


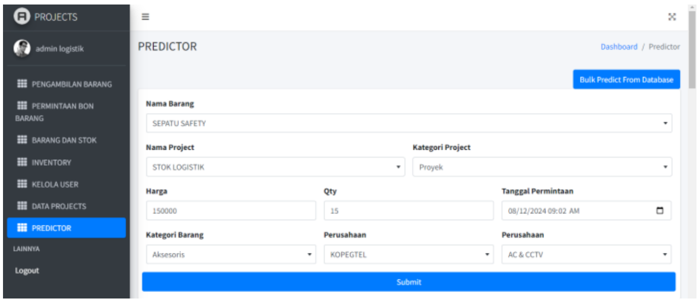
Fig. 7. Random Forest ROC Diagram

In this study, the metric values before and after hyperparameter tuning have also been compared, and the results can be seen in Table 5. This table shows some improvement in accuracy, recall, F1-Score, and ROC AUC, with approximately a 1% improvement on average.

TABLE IV. HYPER PARAMETER TUNING SEARCH GRID LIST

Metrics	*Before Hyperparameter Tuning*	*After Hyperparameter Tuning*	Difference
Accuracy	0.7854	0.7969	0.0115
Precision	0.8479	0.8477	-0.0002
Recall	0.8736	0.8924	0.0188
F1-*Score*	0.8625	0.8695	0.007
ROC AUC	0.8131	0.8234	0.0103

The next stage is implementation into the Kopegtel Samarinda Warehouse System. Implementation is carried out using Django as a backend, which will then be connected to the existing system using an API. Then, several components are formed in the recommendation display. In the recommendation input component, several fields need to be entered, such as item name, project name, project category, price, qty, request date, item category, company, and company, as shown in Fig 8.



PREDICTOR

Nama Barang: SEPATU SAFETY

Nama Project: STOK LOGISTIK

Kategori Project: Proyek

Harga: 150000

Qty: 15

Tanggal Permintaan: 08/12/2024 09:02 AM

Kategori Barang: Aksesoris

Perusahaan: KOPEGTEL

Perusahaan: AC & CCTV

Fig. 8. Random Forest ROC Diagram

The cluster component contains a list of existing clusters and the position of each prediction history for the cluster that has been determined. The cluster can be seen in Fig 9.

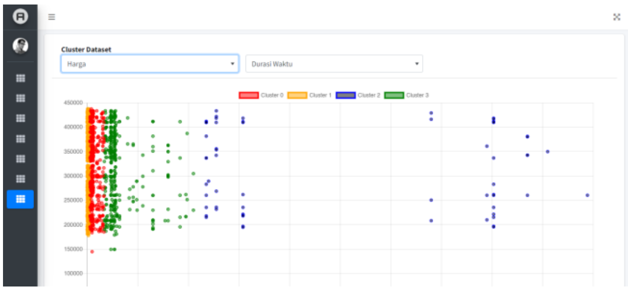
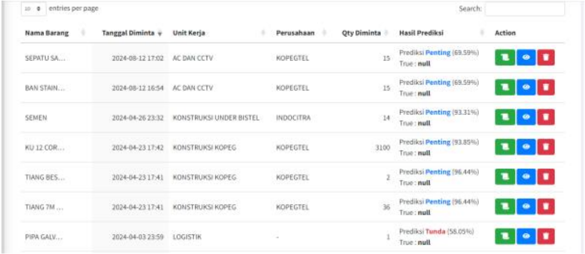


Fig. 9. Random Forest ROC Diagram

Apart from that, a recommendation history display also describes all the predictions made. In the recommendation history display, there are several pop-ups, such as item details to display the type of item, prediction evaluation to evaluate the prediction results, and delete to delete the history, as shown in Fig 10.



Nama Barang	Tanggal Diminta	Unit Kerja	Perusahaan	Qty Diminta	Hasil Prediksi	Action
SEPATU SA...	2024-08-12 17:02	AC DAN CCTV	KOPEGTEL	15	Prediksi **Penting** (69.59%) True : **null**	
BAN STAIN...	2024-08-12 16:54	AC DAN CCTV	KOPEGTEL	15	Prediksi **Penting** (69.59%) True : **null**	
SEMEN	2024-04-26 23:32	KONSTRUKSI UNDER BISTEL	INDOCITRA	14	Prediksi **Penting** (93.31%) True : **null**	
KU 12 COR...	2024-04-23 17:42	KONSTRUKSI KOPEG	KOPEGTEL	3100	Prediksi **Penting** (93.85%) True : **null**	
TIANG BES...	2024-04-23 17:41	KONSTRUKSI KOPEG	KOPEGTEL	2	Prediksi **Penting** (96.44%) True : **null**	
TIANG 7M ...	2024-04-23 17:41	KONSTRUKSI KOPEG	KOPEGTEL	36	Prediksi **Penting** (96.44%) True : **null**	
PIPA GALV...	2024-04-03 23:59	LOGISTIK	-	1	Prediksi **Tunda** (58.05%) True : **null**	

Fig. 10. Random Forest ROC Diagram

IV. CONCLUSION

Implementing K-Means and Random Forest Algorithms for the Procurement Recommendation System in the Kopegtel Samarinda Warehouse System at CV Indocitra Multi Artha has been successfully implemented. Intrinsic evaluation of the K-Means algorithm shows a Silhouette Coefficient value of 0.2682 and a Davies-Bouldin score of 1.681. In contrast, extrinsic evaluation produces a Rand Index of 56.8% and a Folkes Mallows value of 0.432. This shows a low segregation and separation for each cluster, indicating that the K-Means process still has much room for improvement. Furthermore, the Random Forest algorithm shows a precision value of 84.77%, a recall value of 89.24%, and an accuracy rate of 79.69%. This result is limited to the model's ability to predict the label, so the classification metrics result is considered essential. Overall, this model performs considerably well in providing procurement recommendations with an F1-Score value of 86.95% and ROC AUC of 82.34%. Further research is needed to improve this research, especially with a more comprehensive dataset and different clustering techniques or cluster numbers approaches..

REFERENCES

[1] G. Sugiyanto *et al.*, *Manajemen Sistem Informasi*, 1st ed. Padang: Global Eksekutif Teknologi, 2022.

[2] D. García-Barrios, K. Palomino, E. García-Solano, and A. Cuello-Quiroz, “A Machine Learning Based Method for Managing Multiple Impulse Purchase Products: an Inventory Management Approach,” *J. Eng. Sci. Technol. Rev.*, vol. 14, no. 1, pp. 25–37, 2021, doi: 10.25103/jestr.141.02.

[3] Z. Fayyaz, M. Ebrahimian, D. Nawara, A. Ibrahim, and R. Kashef, “Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities,” *Appl. Sci.*, vol. 10, no. 21, pp. 1–20, 2020, doi: 10.3390/app10217748.

[4] J. M. Spreitzenbarth, C. Bode, and H. Stuckenschmidt, “Artificial Intelligence and Machine Learning in Purchasing and Supply Management: a Mixed-methods Review of the State-of-the-art in Literature and Practice,” *J. Purch. Supply Manag.*, vol. 30, no. 1, p. 100896, 2024, doi: 10.1016/j.pursup.2024.100896.

[5] S. S. Yassin and Pooja, “Road Accident Prediction and Model Interpretation Using a Hybrid K-means and Random Forest Algorithm Approach,” *SN Appl. Sci.*, vol. 2, no. 9, pp. 1–13, 2020, doi: 10.1007/s42452-020-3125-1.

[6] H. A. Elzeheiry, S. Barakat, and A. Rezk, “Different Scales of Medical Data Classification Based on Machine Learning Techniques: a Comparative Study,” *Appl. Sci.*, vol. 12, no. 2, p. 919, Jan. 2022, doi: 10.3390/app12020919.

[7] R. W. S. B. Brahmana, F. A. Mohammed, and K. Chairuang, “Customer Segmentation Based on RFM Model Using K-Means, K-Medoids, and DBSCAN Methods,” *Lontar Komput. J. Ilm. Teknol. Inf.*, vol. 11, no. 1, p. 32, 2020, doi: 10.24843/lkjiti.2020.v11.i01.p04.

[8] A. Pajankar and A. Joshi, *Introduction to Machine Learning with Scikit-learn*. Berkeley, CA: Apress, 2022. doi: 10.1007/978-1-4842-7921-2_5.

[9] V. Z. Kamila and E. Subastian, “KNN vs Naive Bayes Untuk Deteksi Dini Putus Kuliah Pada Profil Akademik Mahasiswa,” *J. Rekayasa Teknol. Inf.*, vol. 3, no. 2, pp. 116–121, 2019, doi: 10.30872/jurti.v3i2.3097.

[10] V. Z. Kamila, E. Subastian, and Rosmasari, “KNN and Naive Bayes for Optional Advanced Courses Recommendation,” in *2019 International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, Oct. 2019, pp. 306–309. doi: 10.1109/ICEEIE47180.2019.8981450.

[11] IBM, *IBM SPSS Modeller CRISP-DM Guide*, 18.4. New York, NY: IBM, 2023. [Online]. Available: https://www.ibm.com/docs/it/SS3RA7_18.4.0/pdf/ModelerCRISPDM.pdf

[12] C. Schröer, F. Kruse, and J. M. Gómez, “A Systematic Literature Review on Applying CRISP-DM Process Model,” *Procedia Comput. Sci.*, vol. 181, no. 2019, pp. 526–534, 2021, doi: 10.1016/j.procs.2021.01.199.

[13] J. L. Nielson *et al.*, “Statistical Guidelines for Handling Missing Data in Traumatic Brain Injury Clinical Research,” *J. Neurotrauma*, vol. 38, no. 18, pp. 2530–2537, 2021, doi: 10.1089/neu.2019.6702.

[14] R. Rodríguez *et al.*, “Water-quality Data Imputation With a High Percentage of Missing Values: a Machine Learning Approach,” *Sustain.*, vol. 13, no. 11, pp. 1–17, 2021, doi: 10.3390/su13116318.

[15] F. Sigrist, “A Comparison of Machine Learning Methods for Data with High-Cardinality Categorical Variables,” *Digit. Gov. Res. Pract.*, pp. 1–8, 2023, doi: 10.48550/arXiv.2307.02071.

[16] M. J. Zaki and M. J. Meira, *Data Mining and Analysis: Fundamental Concepts and Algorithms*, 2nd ed. Cambridge: Cambridge University Press, 2020.

[17] Islamiyah, P. L. Ginting, N. Dengen, and M. Taruk, “Comparison of Priori and FP-Growth Algorithms in Determining Association Rules,” *ICEEIE 2019 - Int. Conf. Electr. Electron. Inf. Eng. Emerg. Innov. Technol. Sustain. Futur.*, pp. 320–323, 2019, doi: 10.1109/ICEEIE47180.2019.8981438.

[18] K. Golalipour, E. Akbari, S. S. Hamidi, M. Lee, and R. Enayatifar, “From Clustering to Clustering Ensemble Selection: a Review,” *Eng. Appl. Artif. Intell.*, vol. 104, no. November 2020, p. 104388, 2021, doi: 10.1016/j.engappai.2021.104388.

[19] M. Arhami and M. Nasir, *Data Mining - Algoritma dan Implementasi*. Yogyakarta: Andi Offset, 2020.