Optimizing Filling Shed Reconfiguration through K-Means Clustering with Silhouette, Association Rule Mining, and Artificial Neural Network

UNDERGRADUATE THESIS

Submitted to the Undergraduate Program in Industrial Engineering in Partial Fulfilment of Requirement for the Degree of Sarjana Teknik at the Faculty of Industrial Technology
Universitas Islam Indonesia

Name : Aprillia Rosalind Ann Sophie
Student Number : 20522333

UNDERGRADUATE PROGRAM IN INDUSTRIAL ENGINEERING
DEPARTMENT OF INDUSTRIAL ENGINEERING
FACULTY OF INDUSTRIAL TECHNOLOGY
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2024

AUTHENTICITY STATEMENT

For the sake of Allah SWT, I admit this work is the result of my own work, except for the excerpts and summaries from which I have explained the source. If, in the future, it turns out that my confession is proven to be untrue and violates the legal regulations in the paper and intellectual property rights. In that case, I am willing to get a diploma that I have received to be withdrawn by Universitas Islam Indonesia.

Yogyakarta, 28-09-2024

(Aprillia Rosalind Ann Sophie)

20522333

SUPERVISOR APPROVAL SHEET

Optimizing Filling Shed Reconfiguration through K-Means Clustering with Silhouette, Association Rule Mining, and Artificial Neural Network

UNDERGRADUATE THESIS

By

Name : Aprillia Rosalind Ann Sophie
Student Number : 20522333

Yogyakarta, October 22nd 2024

Supervisor,

(Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM)

EXAMINER'S APPROVAL PAGE

Optimizing Filling Shed Reconfiguration through K-Means Clustering with Silhouette, Association Rule Mining, and Artificial Neural Network

UNDERGRADUATE THESIS

Written by:

Name : Aprillia Rosalind Ann Sophie
Student Number : 20522333

Has been defended before the board of examiners in partial fulfillment of the requirement for a Bachelor Degree in Teknik Industri at the Faculty of Industrial Technology Universitas Islam Indonesia

Yogyakarta, October 17th 2024

Board of Examiners

Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM. Chairman	
Dr. Drs. Imam Djati Widodo, M.Eng.Sc. Member I	
Dr. Harwati, S.T., MT Member II	

Acknowledged by,
Head of Undergraduate Program in Industrial Engineering
Faculty of Industrial Technology
Universitas Islam Indonesia

Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM.
015220101

DEDICATION PAGE

Alhamdulillahirabbil ‘alamin
This undergraduate thesis that the author has spent a lot of time, energy, and emotion to finish is dedicated to myself and my family. To the author’s closest friends and persons who have accompanied and supported the author during the making of this thesis. Also, to the author’s supervisor, Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM, who guided and taught the author during this thesis journey.

MOTTO

"*So, surely with hardship comes ease. Surely, with that hardship comes more ease*"
(Q.S Al - Insyirah, 5-6)

"And provide for them from sources they could never imagine. And whoever puts their trust in Allah, then He ˹alone˺ is sufficient for them. Certainly, Allah achieves His Will. Allah has already set a destiny for everything.
(Q.S At - Talaq, 3)

PREFACE

Assalamu'alaikum Warahamatullahi Wabarakatuh

Praise and gratitude go to the presence of Allah SWT, who has given His mercy, grace, and guidance. With Allah SWT's permission and blessing, the author was able to complete the undergraduate thesis with the title of **"Optimizing Filling Shed Configuration through K-Means Clustering with Silhouette, Association Rule, and Artificial Neural Network".** The author would like to express gratitude to all parties involved in the making of the author's undergraduate thesis, namely:

1. Prof. Dr. Ir. Hari Purnomo, M.T. as Dean of Industrial Technology Faculty, Universitas Islam Indonesia.
2. Dr. Drs. Imam Djati Widodo, M.Eng.Sc. as the Head of the Department of Industrial Engineering Universitas Islam Indonesia.
3. Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM as the Head of the Undergraduate Program in Industrial Engineering Study Program, Universitas Islam Indonesia, also as the author undergraduate thesis supervisor that has guided, taught, and given knowledge to the author during the making to finish the undergraduate thesis.
4. Devy Nurrahmah, S. Kom. as the staff of the International Undergraduate Program in Industrial Engineering who always helps and supports the author in providing information and administration assistance during the undergraduate thesis and studies of the author.
5. My family who always pray and cheer for the author during the making of the undergraduate thesis.
6. The author's closest friends always supported, cheered, and comforted the author whenever the author felt down during the making and finishing of the undergraduate thesis.
7. Classmates from IP IE 2020, also the seniors who help the author during the studies. Also, fellow friends from AKSI TI, SC 2020, Perempuan Tangguh, PSDM HMTI 2023/2024, and Data Mining Laboratory have taught the author many lessons, bring laughter during the making of this undergraduate thesis, and give unforgettable memories for the author.
8. Other parties involved that the author can't mention one by one during the author's studies, making of the undergraduate thesis that has accompanied the author and gave huge support to the author.

The author realizes that this undergraduate thesis still has many shortcomings. Therefore, criticism and suggestions are expected to make the undergraduate thesis even better. Hopefully, this undergraduate thesis can be useful for readers or future researchers. Aamiin. ***Wassalamu'alaikum Warahmatullahi Wabarakatuh***.

Yogyakarta, September 23rd, 2024

Aprillia Rosalind Ann Sophie
NIM 20522333

ABSTRACT

Fuel oil is an essential resource in Indonesia, widely used in various sectors, including industry, transportation, and personal consumption. The company operates numerous branches, including integrated terminals throughout Indonesia, and is responsible for receiving products from refineries, storing them, and delivering them to consumers. The current system's inefficiencies obstruct the company's capacity to meet increasing demands, which could result in customer dissatisfaction and revenue loss. This study aims to propose a reconfiguration of the filling shed by segmenting, finding patterns and relationships between each transaction, and forecasting. The clustering result, by using 19 days of historical data, formed ten clusters, the association rules formed 248 rules, and the ANN showthat the model has an effective predicting ability. These results will be used to propose a reconfiguration that helps to overcome the bottleneck dealt with by the company.

Keywords: Artificial Neural Network, Association Rules, Clustering, Customer Segmentation.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Research Background

Fuel oil is an essential resource in Indonesia, widely used in various sectors, including industry, transportation, and personal consumption. The oversight of fuel oil distribution and management in the country falls under the jurisdiction of BPH Migas, an organization established by Government Regulation No. 67 of 2002, which governs oil and gas distribution as well as downstream transportation activities (A. S. Putri, 2020). BPH Migas collaborates with PT Pertamina (Persero) to ensure the nationwide distribution of fuel oil. Pertamina operates numerous branches, including integrated terminals throughout Indonesia, and is responsible for receiving products from refineries, storing them, and delivering them to consumers. Distribution is carried out via land, air, and sea routes. For land transport, processes similar to those used at gas stations are employed, where tanker trucks load fuel at designated bays and then transport it to gas stations or commercial clients (Ramadhan, 2018).

At the Integrated Terminal, nine filling bays house 17 product pumps. The distribution process, which includes health checks, filling tanker trucks, and performing final inspections, can lead to significant delays, particularly during busy periods when there are long queues for certain products. This problem is exacerbated by the increasing demand for fuel oil, which surpasses the existing production capacity. When a specific product has a long queue, it can cause longer wait times for other tanker trucks as well. Additionally, if a tanker truck is assigned to deliver the same product to multiple gas stations, its limited capacity necessitates several trips back to the terminal. Each of these trips requires repeating the entire process before the fuel can be delivered to the customer or company.

The demand for fuel oil in Indonesia is rising annually and is projected to continue increasing, consistently surpassing production levels, according to the Director General of Oil and Gas at the Ministry of Energy and Mineral Resources (C. A. Putri, 2020).

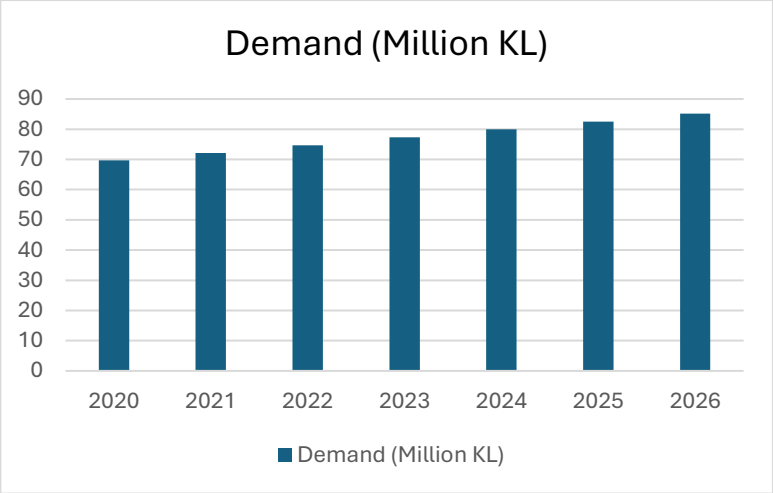


Figure 1. 1 Fuel Oil Demand in Indonesia

The current system's inefficiencies obstruct the company's capacity to meet increasing demands, which could result in customer dissatisfaction and revenue loss. Specifically, the inefficiency in fuel distribution at the Integrated Terminal Balikpapan adversely affects PT Pertamina's potential profits. Distribution delays lead to longer wait times for tanker trucks, which in turn decreases the overall fuel delivery throughput. This bottleneck can result in unmet customer demands, translating into potential sales and market share losses. Moreover, frequent delays and inefficiencies can increase operational costs, further eroding profitability.

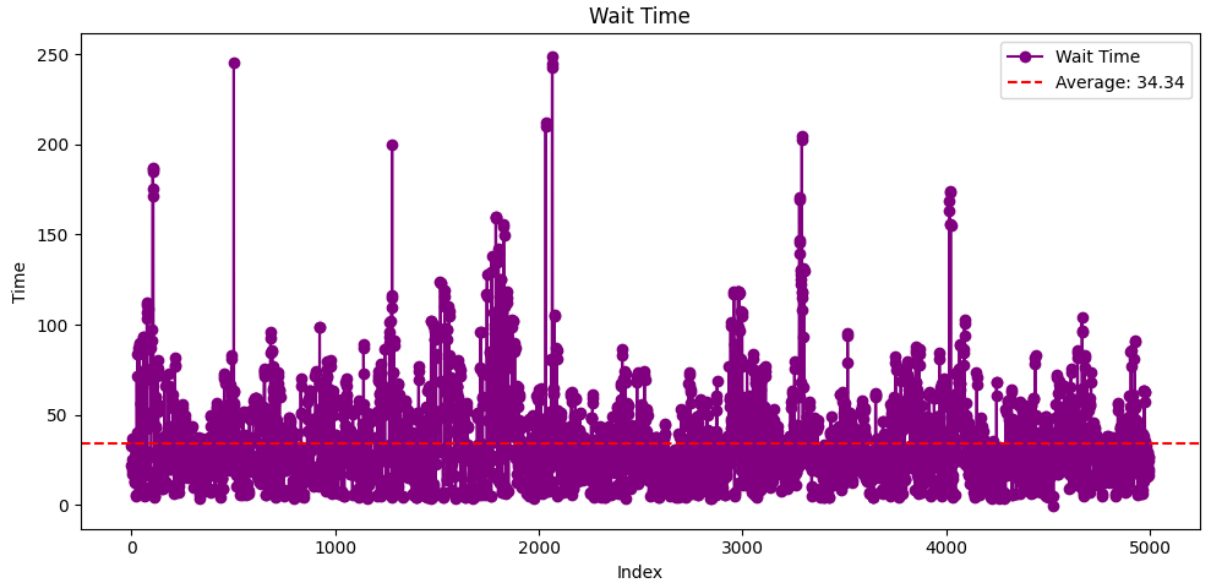


Figure 1. 2 Average Wait Time

Figure 1.2 depicts the average wait time for tanker trucks, recorded at 34.34 minutes. The most commonly observed wait time is 25.26 minutes, suggesting that most wait times fall below the overall average. The pattern of wait times shows fluctuations, with some instances exceeding 100 minutes, underscoring the variability in wait times. These variations imply that

specific conditions or factors contribute to considerable delays, resulting in inconsistent wait times.

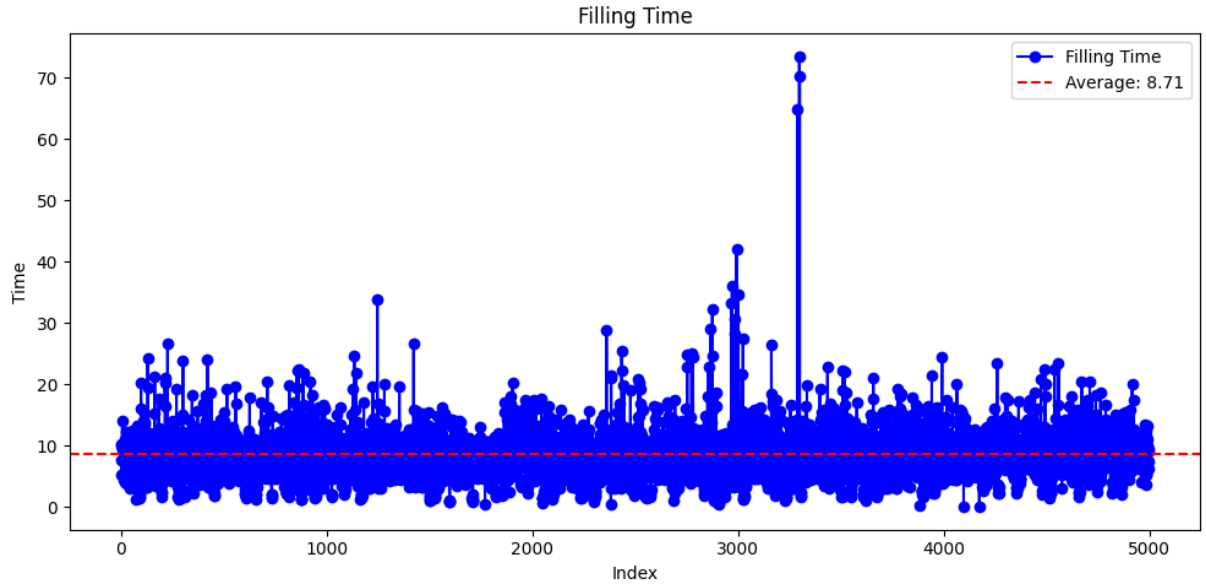


Figure 1. 3 Average Filling Time

Figure 1.3 depicts the average filling time as 8.71 minutes, while the most frequently observed filling time is 7.51 minutes, aligning with earlier findings that show most filling times are below the average. Nonetheless, there are instances where filling times surpass 10 minutes, with occasional spikes reaching as high as 70 minutes, highlighting inconsistencies in the filling process. These fluctuations indicate that various factors may be causing delays, resulting in irregular filling times.

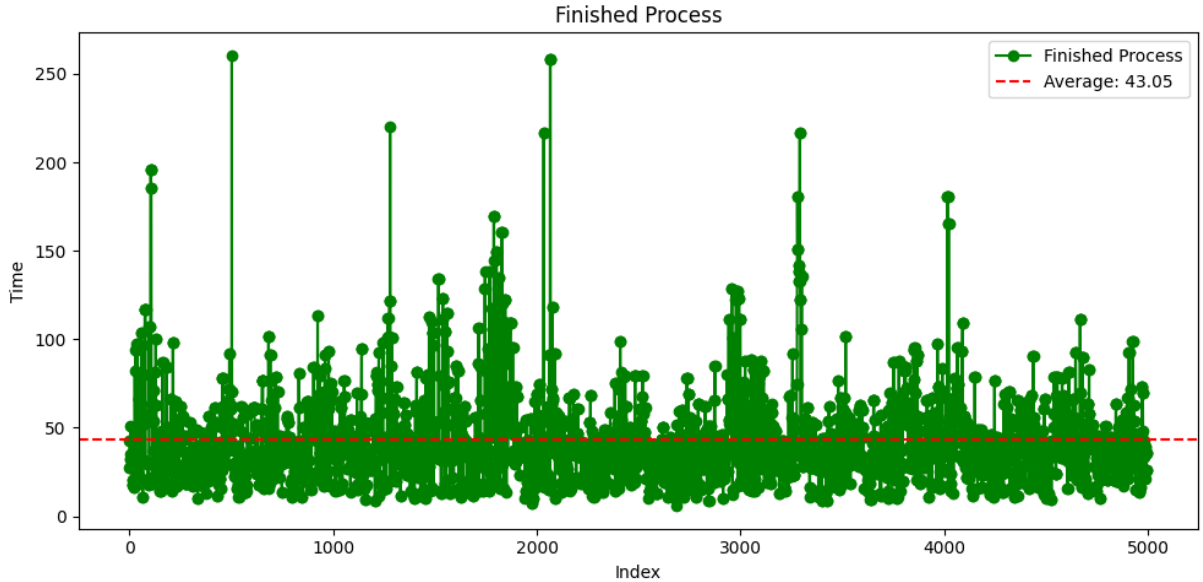


Figure 1. 4 Average Finished Process

Figure 1.4 illustrates that the average finished process time is 43.05 minutes, whereas the most common finished process time is 23.48 minutes, which is below the average. Nonetheless, there are instances where finished processes take longer than 50 minutes, highlighting inconsistencies within the process.

The company has various Key Performance Indicators (KPIs), including loading and unloading times, which gauge the duration required to load and unload product to enhance throughput. Furthermore, they monitor the average loading time for tanker trucks, which indicates the time taken to fill a tanker until it is ready to leave, along with truck turnaround time, which measures the total period a truck remains at the terminal from arrival to departure. Another crucial metric is the operational excellence index, which evaluates the company's operational efficiency and effectiveness.

These key performance indicators (KPIs) are vital for assessing the company's performance. While the wait time, filling time, and completed process times often fall below average, there are still cases where they surpass the targets. The company has established a goal of achieving 70% of its KPIs; however, based on the average results, the wait time, filling time, and completed processes do not reach this target. The company is focused on gradually enhancing its performance to boost efficiency, particularly in anticipation of projected demand and future objectives. As the sole licensed producer and supplier of fuel oil in Indonesia, achieving these targets is essential for maintaining operational excellence and fulfilling market requirements.

By addressing these issues, the company has the opportunity to improve its operational efficiency, better satisfy customer needs, and maximize its profit potential, thereby preventing substantial losses. While increasing the number of filling bays could be a potential solution, without a clear understanding of the optimal product mix for each bay, the same issues may continue to arise. Given that the company must serve both industrial and public demands, it is essential to have a comprehensive understanding of its market. To enhance the system and reduce complications, the company can implement a new configuration based on historical demand patterns to tackle these challenges effectively.

To tackle these challenges, it is essential to optimize the configuration of the filling shed to improve efficiency and minimize bottlenecks. This study suggests utilizing RStudio, RapidMiner, and Google Colab to apply three analytical techniques: K-Means clustering with the silhouette method, association rule mining using the FP-Growth algorithm, and Artificial

Neural Networks (ANN) using backpropagation. K-Means clustering will used to categorize similar product orders, facilitating better product distribution across various filling bays. The silhouette method will help ensure that the clusters are well-defined and significant. Additionally, association rule mining with the FP-Growth algorithm will reveal patterns and relationships among different products, allowing for more informed decisions regarding the setup of filling bays. Finally, ANN using backpropagation will be implemented for pattern recognition and predictive analysis.

By addressing the identified problems with the proposed methods, PT Pertamina can significantly improve its filling shed configuration. This optimization will enhance operational efficiency, reduce bottlenecks, and ensure timely fuel distribution, ultimately increasing customer satisfaction and maximizing profits. The research aims to provide actionable insights that align with the company's goal of effectively meeting the growing demand for fuel oil in Indonesia while avoiding substantial financial losses.

1.2 Problem Formulation

Based on the background that is presented, the problem formulations are revealed as follows:

1. What is the optimal number of clusters that can be formed based on the historical data, and what is the result of the clustering?
2. What are the association results and patterns that can be identified based on the historical data?
3. How effectively can an Artificial Neural Network predict the efficiency of the filling shed based on the historical data from the previous method?
4. Based on the calculation from the previous method, what is the proposed reconfiguration?

1.3 Research Objective

Based on the problem formulation above, below are the research objectives of this study:

1. To determine and analyze the optimal number of clusters formed based on the historical data.
2. To identify and analyze the association rules results and patterns derived from the historical data.
3. To evaluate the effectiveness of the Artificial Neural Network in predicting the efficiency of the filling shed based on the historical data.

4. To provide a new recommendation of the configuration in the filling shed based on the previous calculation.

1.4 Scope of Research

There are several limitations to this undergraduate research. The limitations are as follows:

1. The methods used for this undergraduate research are theK-Means clustering silhouette method using Rstudio, Association Rules FP-Growth using Rapidminer, and Artificial Neural Network using Google Colab.
2. The research was conducted at Integrated Terminal Balikpapan in the Distribution Department from January 2024 to May 2024.
3. The object of this study is the filling shed configuration at Integrated Terminal Balikpapan.
4. The data used in this undergraduate research are received during the fieldwork of the author and can't be explicitly presented or shared with other parties.

1.5 Research Benefit

Several beneifts can result for several parties, the benefits are as follows:

1. For companies, the result can provide further information and can be used as consideration for the reconfiguration of the filling shed. The company also helps to educate students using real-world problems.
2. Students understand the operation and industrial management further using statistics and data science based on real-world problems. It is also can be used to measure the student's capability in applying the knowledge received in university in to the real world.

CHAPTER II
LITERATURE REVIEW

2.1 Deductive Review

2.1.1 Clustering

Clustering is a technique used to organize data objects into clusters or groups, where the resulting clusters demonstrate a higher degree of similarity or homogeneity among their members than with objects in other clusters. Cluster analysis is identified as an unsupervised learning approach and is used to explore the relationships between patterns by organizing them (Nepal et al., 2019).

Clustering methods are generally divided into two categories: hierarchical and non-hierarchical. Hierarchical clustering consists of nested clusters, allowing clusters to be components of larger clusters, with the grouping strategy influencing the cluster formation. Examples of hierarchical clustering include agglomerative and divisive methods. On the other hand, non-hierarchical clustering relies on centroids and requires that the number of clusters (k) be predetermined. However, there is no single method for determining the optimal k value. Non-hierarchical clustering techniques include K-Means and Fuzzy c-means clustering (Et-taleby et al., 2020) This research specifically examines the K-Means clustering method.

2.1.2 K-Means

K-Means is a clustering algorithm that employs a centroid model. It offers a simple method for organizing data based on centroids and the distances from each data point. The centroids for each cluster are typically initialized randomly, often by designating the first centroid to the first data point, the second centroid to the second data point, and so forth. Subsequently, the distances between each data point and the centroids are computed, and the data is clustered according to the shortest distance to the nearest centroid (Nainggolan et al., 2019). K-Means aim to enhance the similarity of data points within a cluster while reducing the similarity between different clusters. This is accomplished by utilizing a distance function to assess similarity, which is based on the shortest distance to the centroid (Nainggolan et al., 2019). In this undergraduate thesis, the author has chosen four variables: Product, Quantity, Filling Time, and Finished Process. These variables were selected because the research seeks to develop a new configuration in which each filling bay can accommodate a maximum of two product types. Product was identified as a key variable, along with Quantity, Filling Time, and

Finished Process, as these factors are important in creating the new configuration. By incorporating these variables, the author aims to identify trends, such as which products tend to have high quantities, shorter filling times, and faster finishing processes to be taken into consideration.

2.1.3 Silhouette Method

The silhouette method employs a silhouette coefficient to assess the effectiveness of object grouping within clusters. This coefficient takes into account both the similarity of an object to its cluster (cohesion) and its dissimilarity to other clusters (separation). The silhouette coefficient is computed by dividing the separation measure by the cohesion measure. If the separation measure exceeds the cohesion measure, one is deducted from the result. On the other hand, if the cohesion measure surpasses the separation measure, the quotient of the cohesion measure divided by the separation measure is subtracted from 1 (Saputra et al., 2020).

The silhouette coefficient assesses the connection between an object and its corresponding cluster, with values that range from -1 to 1. A higher coefficient signifies a stronger association, whereas a value close to 0 implies that the object might belong to another cluster. A value nearing -1 suggests that the object is probably in an incorrect cluster. The silhouette method produces a graph to compare coefficient values among various clusters, identifying the cluster with the highest coefficient as the most appropriate one (Saputra et al., 2020).

2.1.4 Association Rule Mining

Association rule mining is a method employed to identify relationships or combinations of items, commonly known as frequent item sets. Usually, the outcomes of association rule mining are expressed in an IF-THEN format, which aids in interpreting and comprehending the relationships among items. (Santoso, 2021). The association rule has three statistical indicators that are as follows:

a. Support

Support is an indicator that shows how often a rule occurs in the dataset. The formula for calculating support is as follows:

$$Support\ (A \rightarrow B) = P(A \cup B)$$

b. Confidence

Confidence is an indicator that shows how reliable the rule calculation is. The formula for calculating confidence is as follows:

$$Confidence\ (A \rightarrow B) = \frac{P(A \cup B)}{P(A)}$$

c. Lift Ratio

The lift ratio is an indicator that shows the strength of the dependence between the antecedents and the consequences of the association rule. The formula for calculating the lift ratio is as follows:

$$Lift\ Ratio\ (A \rightarrow B) = \frac{P(A \cup B)}{P(A)\ P(B)}$$

Association rule mining features two key algorithms: FP-Growth and the Apriori algorithm. In this research, the author opted for FP-Growth due to its efficiency compared to the Apriori algorithm when handling large datasets. The Apriori algorithm is recognized for its ability to identify common item sets within data and has found widespread application across numerous fields. It comprises two primary steps: the first step focuses on identifying all frequently occurring item sets in the dataset, while the second step involves generating association rules based on these frequently occurring item sets (Sivasankaran et al., 2020).

2.1.5 Artificial Neural Network (ANN)

An Artificial Neural Network (ANN) is a reasoning technique inspired by the human brain, composed of simple processing units known as interconnected neurons. These neurons are linked by weights that transmit signals from one neuron to another. ANNs can mimic the behavior of biological neural networks. During the learning process, an ANN adjusts its outputs to align with various inputs. Each neuron in an ANN is designed and trained to operate like human neurons, enabling it to receive multiple inputs and generate a single output. The inputs can be either raw data or the output from a preceding neuron, while the output may represent either the final result or serve as the input for the subsequent neuron. (Galih Pradana et al., 2022).

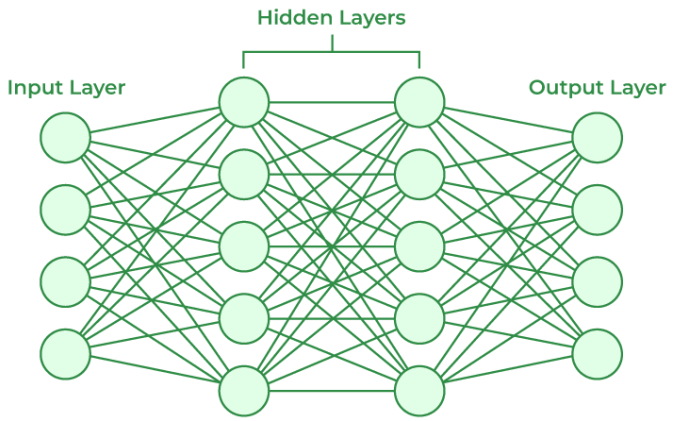


Figure 2. 1 Artificial Neural Network Architecture

ANN has three layers as follows:

a. Input Layer

   The input layer is the initial layer of the network, tasked with receiving data from external sources. The neurons in this layer do not process the data,-; they merely transmit it to the subsequent layer.

b. Hidden Layer

   Hidden layer is a layer that can comprise one or more layers, or it may not exist at all. In cases where there are several hidden layers, the lowest one receives input directly from the input layer. These hidden layers are where the majority of the network's computations take place, converting the input data into a format that the output layer can utilize.

c. Output Layer

   The output layer has a similar principle as the hidden layer, but it has a crucial distinction as its output represents the final result of the entire process. The data that has been processed through the hidden layers is sent to the output layer, which produces the final output of the network.

Artificial Neural Networks (ANNs) utilize several algorithms, such as backpropagation, particle swarm optimization, and simulated annealing. This undergraduate research focuses specifically on backpropagation, a supervised training algorithm that calculates the gradient of the error function. The training process involves measuring the difference between the target value and the model's output, followed by adjusting the weights of each neuron based on the calculated gradient to reduce the loss. This iterative process continues until the error drops below a predefined threshold, or the model achieves improved accuracy. Backpropagation is valued for its ability to handle complex data and generate accurate predictions while considering various factors (Irianti et al., 2022; Widhi Aryanti & Nur Azizah Komara Rifai, 2023).

In data processing using ANN, some parameters need to be determined before, such as epoch and learning rate. Epoch is the times the dataset passes through the network where epochs help the model to learn the pattern within the data. Epoch is a crucial part of the processing because if epoch score is higher, it can cause over-fitting, which may cause the model to be overly complex and lose the power to tests the data later (Hosseinzadeh et al.,

2021). The learning rate controls how much the model weights are adjusted to the gradient loss during each training step. A higher training rate can help the model train faster, but it can cause training loss to fluctuate and overshoot the optimal point. Al ower training rate makes the model train longer to achieve an optimal solution, but it can risk over-fitting and take longer to reduce a loss (Basodi et al., 2020). Finding the optimal number of epochs and learning rate is crucial since it affects the result of the prediction.

The output for the ANN can be assessed using several validations. The parameters are as follows:

a. Accuracy

   Accuracy quantifies how close the predicted values are to the actual values. It is defined as the proportion of correctly predicted observations compared to the total number of observations.

b. Precision

   Precision refers to the ratio of relevant items selected to the total number of items selected. It can also be understood as the correspondence between an information request and the response given by the system.

c. Recall

   Recall is the proportion of relevant items that are accurately identified compared to the total number of relevant items present. It assesses the model's capability to identify all relevant instances within the dataset.

d. F1-Score

   F1-Score is the harmonic mean of precision and recall. It offers a single metric that balances both precision and recall, making it particularly valuable when the distribution of classes is imbalanced.

2.2 Inductive Study

An inductive study involves reviewing previous literature that correlates with the current research. This approach helps the author gain a deeper understanding of the methods used or similar topics that have been explored in the past.

For instance, previous research by Fansyuri (2023), titled "Analisis Algoritma Neural Network untuk Identifikasi Jenis Apel Berbasis Ekstraksi Fitur Bentuk dan Warna," employed a neural network algorithm to identify different types of apples based on feature extraction of shape

and color. In this study, the researcher used 160 photos of two different apples, dividing the data into testing and training datasets. The training data was segmented into two clusters: foreground and background. The next step involved extracting color and shape features to remove noise from the images, with the results forming the basis for training image extraction data using K-Means Clustering. The testing process followed the same steps as the training. The neural network method was then applied, resulting in a 99.23% accuracy rate, along with precise predictions, recall, and precision regarding the classification of the apples.

Kaoungku et al. (2024) conducted a study titled “The Silhouette Width Criterion for Clustering and Association Mining to Select Image Features.” In their research, the authors employed discretization through the chi-square method to handle numerical data. They subsequently identified patterns for association rules using the Apriori algorithm, which was presented in an IF-THEN format, yielding association rule scores and average confidence levels. Clustering was executed utilizing the silhouette coefficient, and K-Means clustering was implemented with the optimal k value determined by the silhouette method. This approach resulted in the identification of the best features for image classification.

Fadilah (2023) conducted research titled “Implementasi Algoritma K-Means Clustering Untuk Targeting Ads.” The study aimed to explore the implementation of customer segmentation and evaluate the performance of the silhouette coefficient. The researcher performed data preprocessing by selecting relevant attributes and normalizing the data, ensuring that it was numerical. Following this, linear regression was applied to identify which independent variables had strong correlations with the dependent variable. Subsequently, silhouette analysis was conducted to determine the optimal k-value. This optimal k-value was then used to perform K-Means clustering, with iterations carried out to update the centroid positions until cluster stability was achieved. The K-Means model was utilized to predict new cluster labels, and dimensionality reduction was applied using t-SNE to enhance the visualization of the results.

Husein et al. (2022) carried out a study titled “Combining Grouping Techniques and Association Rules for Marketing Analysis Based on Customer Segmentation.” The researchers initiated the process with data cleaning, which included eliminating missing values, smoothing noisy data, identifying and removing outliers, and addressing inconsistencies. Once they achieved a better understanding of the data through exploratory analysis, they utilized K-Means clustering within the RFM (Recency, Frequency, Monetary) model. The clusters generated were

subsequently analyzed using the Apriori algorithm to identify patterns that could aid in marketing management.

In a separate study, Ula et al. (2023) investigated the application of "Machine Learning Clustering K-Means and Linear Regression in Determining the Risk Levels of Pulmonary Tuberculosis." The researchers utilized K-Means clustering to categorize the data into three clusters, with the centroids initially selected at random based on the nearest variables using Euclidean Distance. By the fourth iteration, the centroids had stabilized, signaling that no further alterations were necessary. The team integrated K-Means clustering with linear regression, and their findings indicated that the predictive model explained 57% of the patients' risk levels, while the remaining 43% were influenced by factors outside the independent variables examined in the study.

Safara et al. (2020) carried out a study entitled “Improved Intrusion Detection Method for Communication Networks Using Association Rule Mining and Artificial Neural Networks.” The researchers applied association rule mining to extract features from the dataset, which were subsequently used to build and assess a model. The dataset was divided into training and testing sets. The researchers utilized neural networks along with AdaBoost to create and evaluate the model. The findings revealed a detection accuracy of 99.55% and included a comparison of precision among AdaBoost, the neural network, and the proposed method.

Rahman et al. (2024) conducted a study titled “Unsupervised Machine Learning Approach for Tailoring Educational Content to Individual Student Weaknesses.” The research commenced with data preprocessing to eliminate any missing information. Utilizing Google Collaboratory, the researchers tested the Apriori algorithm by transforming the data into transaction data and organizing it within a binary matrix. They then employed the Apriori algorithm to identify frequent item sets and repeated the process using the FP-Growth algorithm. The data was filtered to retain only passing grades from one subject and failing grades from another, which were subsequently clustered using the elbow method based on the within-cluster sum of squares. Due to the lack of a clear outcome from the elbow method, the researchers used a silhouette graph to ascertain the optimal number of clusters. The algorithm's output revealed associations among various variables. The study incorporated both qualitative analysis through interviews and quantitative analysis via A/B testing. The researchers observed that FP-Growth effectively tackled the issue of generating a large number of candidate frequent item sets, providing more efficient solutions, particularly as the size of the data increased.

In their 2021 study titled "Unsupervised Learning as a Data Sharing Model in the FP-Growth Algorithm for Determining Optimal Transaction Data Patterns," Mustakim et al. examined the integration of clustering techniques with the FP-Growth algorithm to uncover transaction data patterns. The researchers employed K-Means, K-Medoids, and FCM clustering methods before applying the FP-Growth algorithm to extract association rules. The data was divided for the clustering process, after which the FP-Growth algorithm was utilized. The study assessed the outcomes of the FP-Growth algorithm both with and without prior clustering. The results revealed that, in the absence of clustering, the FP-Growth algorithm failed to identify any rules based on the specified minimum support and confidence thresholds. Conversely, when clustering was performed beforehand, the method successfully generated 12 association rules.

In a separate study conducted by Al et al. (2023), titled “Product Recommendations using Market Basket Analysis with FP-Growth and Clustering Techniques,” the researchers aimed to develop product recommendations for a café. The study commenced with data cleaning and reduction, followed by the selection of data based on RFM (Recency, Frequency, Monetary) attributes to streamline the clustering process. The data was subsequently transformed to support processing through the Association Rule Market Basket Analysis using the FP-Growth method and was then normalized. Clustering was performed utilizing the Euclidean distance measure, and a validation test was executed using a performance vector. The FP-Growth algorithm was implemented with a minimum support threshold of 50% and a minimum confidence level of 95%. The study ultimately produced a set of product promotion recommendations for the café.

In a study titled “Data Mining Technique for Grouping Products Using Clustering Based on Association,” Mandala & Putri (2023) conducted data preprocessing followed by the implementation of the FP-Growth algorithm, setting a minimum support value of 33.33%. They discovered strong association rules with a minimum confidence level of 50%. The researchers then employed the K-Means algorithm to categorize the products into two clusters, calculating distances to determine new centroids based on each product's sales frequency. The study's objective was to enhance data processing to directly obtain clustered products. The authors utilized a hybrid data mining approach that is appropriate for large datasets, although it necessitates a longer processing time.

In a separate study, Boyko and Zhyhailo (2021) investigated the “Comparison of Algorithms for Associative Rules Search Apriori and FP-Growth about Time Dependence on Database Parameters.” The objective of the research was to compare and analyze the features of

the Apriori and FP-Growth algorithms. The researchers carried out multiple experiments using a database to assess the execution speed of both algorithms with medium-sized datasets and minimal support. The findings revealed notable differences, with FP-Growth demonstrating greater efficiency for large datasets, while the Apriori algorithm excelled with small to medium-sized databases. The study concluded that the FP-Growth algorithm offers a significant advantage in execution time, especially when dealing with large datasets.

In their previous research titled “The Comparison of Apriori Algorithm with Preprocessing and FP-Growth Algorithm for Finding Frequent Data Patterns in Association Rule,” Wicaksono et al. (2020) compared the performance of the Apriori and FP-Growth algorithms across various scenarios, including different minimum frequencies, minimum supports, minimum confidences, and varying dataset sizes. The findings indicated that FP-Growth performs better with small datasets due to its efficient handling of complex tree structures; however, it becomes time-consuming and memory-intensive when applied to larger datasets. In contrast, the Apriori algorithm is more effective with large datasets because it utilizes a repeated scanning process to identify rules. When the Apriori algorithm was enhanced with preprocessing, it could uncover rules in large datasets more effectively than either the standard Apriori or the FP-Growth algorithms. The study concluded that FP-Growth is most suitable for small datasets, whereas the Apriori algorithm is more efficient for larger datasets.

In a separate study conducted by Nurfalah et al. (2021), titled "Identifikasi Citra Beras Menggunakan Algoritma Multi-SVM dan Neural Network pada Segmentasi K-Means," the researchers concentrated on identifying images of rice grains by utilizing Multi-SVM and Neural Network algorithms. The research process began with preprocessing, which involved dividing the data into training and testing sets, followed by image preprocessing. K-Means clustering was employed to segment the images, distinguishing the rice grains from the background and noise. Subsequently, the images were analyzed to extract features such as area, perimeter, major axis length, minor axis length, metric, eccentricity, contrast, homogeneity, and correlation. The identification process utilized both Multi-SVM and Neural Network algorithms, and the results included a comparison of accuracy between the two techniques.

The research conducted by Tambunan et al. (2020), titled “Electrical Peak Load Clustering Analysis Using K-Means Algorithm and Silhouette Coefficient,” focused on analyzing and categorizing electrical peak loads during the COVID-19 pandemic. The study examined six years of data, which were divided into three clusters based on load levels. The

simulation outcomes revealed that the highest silhouette scores were achieved with three clusters, which corresponded to the classification of data into low, intermediate, and high load levels. The low peak load was generally linked to national holidays, whereas the greatest variation between maximum and minimum peak loads was observed in January, June, and July.

In a separate study conducted by Hadi Nasyuha et al. (2021), entitled "Frequent Pattern Growth Algorithm for Maximizing Display Items," the authors sought to enhance the organization of display items. They utilized the FP-Growth algorithm, incorporating FP-Tree formation. The data underwent preprocessing to create a transaction database, and the support for each item was manually computed. Subsequently, the researchers identified frequent item sets based on these manual support calculations. The outcomes included support and confidence values, leading to the conclusion that the new arrangement of displays should be founded on combinations that meet or surpass the minimum confidence threshold.

Table 2. 1 Inductive Study

No	**Title**	**Method**		
		K-Means	**Artificial Neural Network**	**AR**
1.	Analisis Algoritma Neural Network untuk Identifikasi Jenis Apel Berbasis Ekstraksi Fitur Bentuk dan Warna (Fansyuri, 2023)	✓	✓	
2.	The Silhouette Width Criterion for Clustering and Association Mining to Select Image Features (Kaoungku et al., 2018)	✓		✓
3.	Implementasi Algoritma K-Means Clustering untuk Targeting Ads (Fadilah, 2023)	✓		
4.	Combination Grouping Techniques and Association Rules for Marketing Analysis based Customer Segmentation (Husein et al., 2022)	✓		✓

No	Title	Method		
		K-Means	**Artificial Neural Network**	**AR**
5.	Penerapan Machine Learning Clustering K-Means dan Linear Regression dalam Penentuan Tingkat Resiko Tuberkulosis Paru (Ula et al., 2023)	✓		
6.	Improved Intrusion Detection Method for Communication Networks Using Association Rule Mining and Artificial Neural Networks (Safara et al., 2020)		✓	✓
7.	Unsupervised Machine Learning Approach for Tailoring Educational Content to Individual Student Weaknesses (Rahman et al., 2024)	✓		✓
8.	Unsupervised Learning as a Data Sharing Model in the FP – Growth Algorithm in Determining the Best Transaction Data Pattern (Mustakim et al., 2021).	✓		✓
9.	Product Recommendations Using Market Basket Analysis With FP-Growth and Clustering Techniques (AL et al., 2023)	✓		✓
10.	Data Mining Technique for Grouping Products using Clustering based on Association (Mandala & Putri, 2023)	✓		✓

No	Title	Method		
		K-Means	**Artificial Neural Network**	**AR**
11.	Comparison of Algorithms of Associative Rules Search Apriori and FP-Growth for Investigation of Time Dependence of Their Execution on Database Parameters (Boyko & Zhyhaylo, 2021)			✓
12.	The Comparison of Apriori Algorithm with Preprocessing and FP-Growth Algorithm for Finding Frequent Data Pattern in Association Rule (Wicaksono et al., 2020)			✓
13.	Identifikasi Citra Beras Menggunakan Algoritma Multi-SVM dan Neural Network pada Segmentasi K-Means (Nurfalah et al., 2021)	✓	✓	
14.	Electrical Peak Load Clustering Analysis Using K-Means Algorithm and Silhouette Coefficient (Tambunan et al., 2020)	✓		
15.	Frequent Pattern Growth Algorithm for Maximizing Display Items (Hadi Nasyuha et al., 2021)			✓
16.	Optimizing Filling Shed Reconfiguration through K-Means Clustering with Silhouette,	✓	✓	✓

No	Title	Method		
		K-Means	Artificial Neural Network	AR
	Association Rule Mining, and Artificial Neural Network.			

The explanation of previous research alsocan be seen in Table 2.1, several differences differentiate the author's research and the previous research. Previous studies, such as Fansyuri (2023), primarily focused on applying neural networks for specific tasks like apple classification through feature extraction, while Kaoungku et al. (2024) utilized the silhouette method and clustering for selecting image features. Similarly, Fadilah (2023) and Husein et al. (2022) explored K-Meansclustering for customer segmentation and marketing analysis, often integrating other techniques such as Apriori algorithms. Additionally, Ula et al. (2023) integrated K-Means with linear regression, and Safara et al. (2020) employed neural networks for intrusion detection.

The author's research seeks to optimize the reconfiguration of filling sheds by concentrating on product distribution to improve allocation within the filling bays. In contrast, previous studies mainly utilized clustering techniques for product segmentation or risk assessment. This research distinguishes itself through the application of Association Rule Mining to reveal patterns and relationships, which aids in deciding the placement of products in the proposed filling bays. Furthermore, it adopts a more sophisticated approach by integrating Artificial Neural Networks (ANN) for predictive analysis, moving beyond simple analysis and categorization of product distribution. It also uncovers patterns to recommend a new operational layout aimed at enhancing efficiency.

CHAPTER III
RESEARCH METHOD

3.1 Research Subject and Object

The research subject of this research is Integrated Terminal Balikpapan, located in Jl. Yos Sudarso No. 148 Balikpapan, South Borneo, Indonesia. The object of this research is product distribution in the filling shed, where there is a product delivered every day for the product to be distributed to the gas station and private sectors. This research focused on product segmentation in the filling shed to evaluate whether the current filling shed can accommodate the existing product distribution in the terminal.

3.2 Data Collection Method

The author used several data collection methods in this research. The method used is mentioned as follows:

3.2.1 Primary Data

Primary data is data given directly to the author based on observation, test, interview, etc (Hikmawati, 2020). The primary data of this research is collected from the Distribution (P1) Department based on the product order and distribution to the car tanks the company received from May to August 2023.

3.2.2 Secondary Data

Secondary data is data given indirectly to the author that is usually used to fulfill the author's knowledge and information based on journals, articles, thesis, etc (Hikmawati, 2020). In this research, the author used journals, articles, and books from existing sources.

3.3 Research Flow

Below is the research flow on this research in Figure 3.1 below:

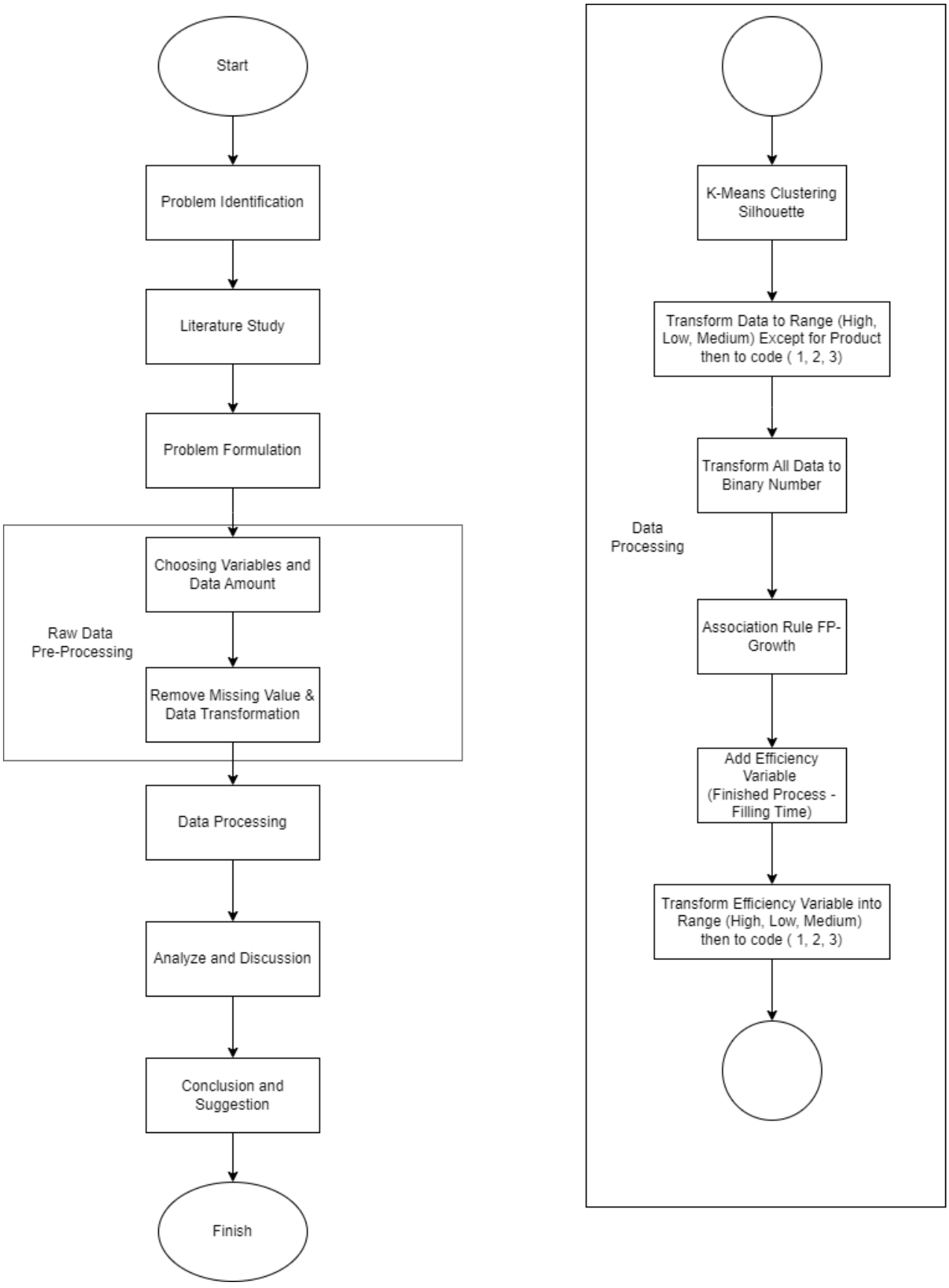


Figure 3. 1 Research Flow

Below is the explanation of the research flow above:

1. Start

   The undergraduate research begins.

2. Problem Identification

   Problem identification is done by observing the issues faced by the company during the business process. In this research, the author identifies the problem to be analyzed and formulates the problem statement, objectives, and limitations.

3. Literature Study

   The author conducts a literature review to understand the problem and the methods used, while also gathering knowledge and references to support the research.

4. Problem Formulation

   Based on the previous problem identification and literature review, the author formulates the final problem to be researched and determines the appropriate method to use.

5. Receive Product Order & Distribution Data

   In this phase, the author obtains the data from the company.

6. Raw Data Pre-Processing

   In this phase, there are two steps, namely:

   a. The author selects the variables to be used in the undergraduate research and determines the amount of data to be analyzed.
   b. The author pre-processes the data by removing missing values and transforming the data. Only the Product variable is transformed, converting it into corresponding code numbers based on the frequency of product orders, from the most ordered to the least.

7. Data Processing

   In data processing, there are several steps to be done:

   a. After completing data pre-processing, the author conducts K-Means Clustering using the Silhouette method in R Studio. At this stage, the author determines the number of clusters (k) and obtains the cluster members.
   b. Before applying Association Rules, the author conducts further preprocessing. First, the data is transformed into ranges (High, Medium, Low) based on the median for all variables except the Product variable. These ranges are then converted into numerical codes (1, 2, and 3). The transactions are then transformed into binary numbers, which are

used to apply the FP-Growth Association Rules to identify patterns and relationships that support the clustering results.

c. Prior to applying the Artificial Neural Network (ANN), the author defines an Efficiency variable, which is determined by the difference between the Finished Process and Filling Time. This Efficiency variable is classified into three categories: High, Medium, and Low, and is subsequently encoded into numerical values: 1, 2, and 3. The dataset is then divided into training and testing sets using a 75/25 ratio, with 75% designated for training and 25% for testing. This distribution ensures that the model has ample data to learn patterns during the training phase while reserving a smaller portion for validation through testing. The data for both the training and testing sets is randomly distributed to prevent any bias.

8. Analyze and Discussion

   The author analyzes the results based on data processing and discusses the findings.

9. Conclusion and Suggestion

   The author provides a conclusion based on the analysis and discussion of the research problem and objectives. Suggestions for the company, based on the conclusion, are also provided.

10. Finish

    The undergraduate research is completed.

CHAPTER IV
DATA COLLECTING AND PROCESSING

4.1 Data Collection and Pre-Processing

Data collection is conducted at Integrated Terminal Balikpapan. The data used is historical data that the company owned regarding distribution data for fuel distribution using tanker trucks. The author used several variables such as Product, Quantity, Filling Time, and Finished Process. For the products variable, there are several products distributed by the company, the products are as follows:

Table 4. 1 Product Transformation Code

Product Name	Transformation
Pertalite	1
Biosolar B35	2
Avtur	3
Dexlite	4
Pertamax	5
Pertamina Dex 50 PPM, Bulk	6
Solar	7

Before conducting clustering using R Studio, the author conducts pre-processing, such as removing non-numeric columns, removing missing values, and transforming the product variable into the transformation code in Table 4.1.

4.2 K-Means Clustering Using Silhouette

After conducting the pre-processing, the author can conduct the silhouette method to determine the amount of k for the clustering later. The result is as follows:

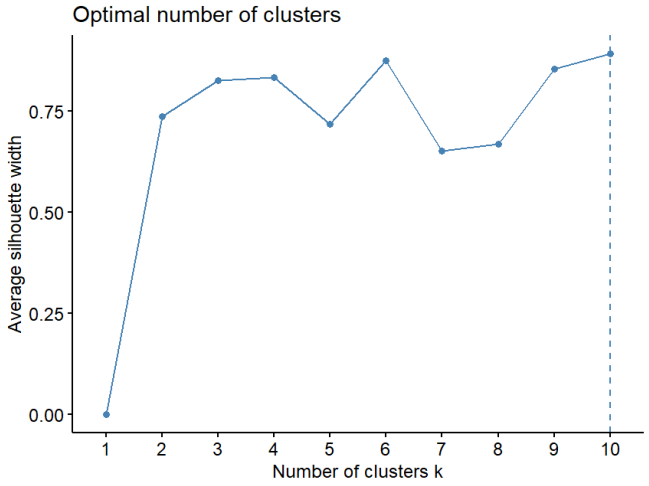


Figure 4. 1 Silhouette Result

Figure 4.1 shows the silhouette result to determine the amount of k to be used in the clustering. The amount of k that will be used is the result that has the highest average silhouette result or closest to 1. Based on Figure 4.1, the value of k = 10 is the optimal number since it has the average silhouette result, aligned with the company plan to add the filling bay into ten, the complexity of the data also can be overcome by having more accurate groupings, also having more number than the product the company has meaning that one product that often has product order can have two product pumps or more. After the amount of k is found, the author can conduct the K-Means clustering. The result is as follows:

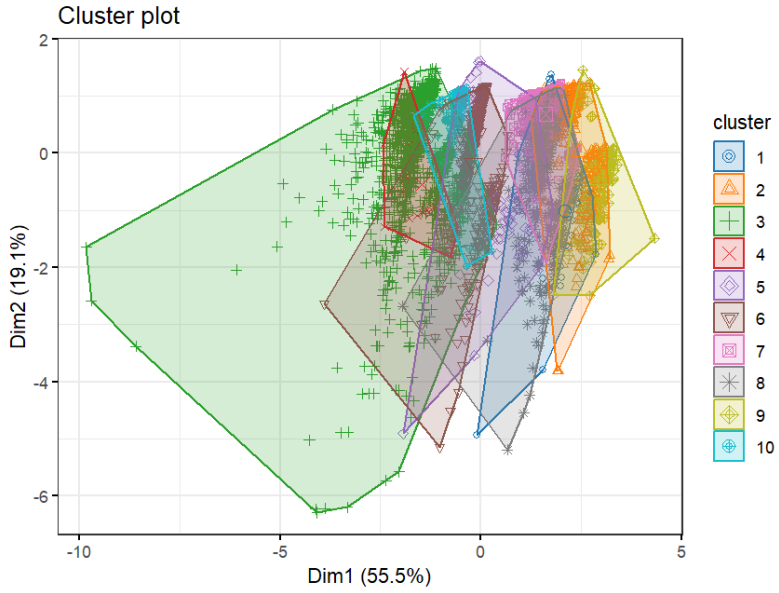


Figure 4. 2 Cluster Visualization Result

Figure 4.2 above visualizes the result of the clustering. This cluster plot visualizes data that has been segmented into ten distinct clusters, each represented by a different color and symbol. The x-axis, labeled 'Dim1', explains 55.5% of the variance in the dataset, while the y-axis, labeled 'Dim2', explains 19.1% of the variance. It suggests that 'Dim1' is the most significant dimension for differentiating the clusters.

Most of the clusters are in one area and overlap with one another. While several data points for cluster three are far from each other, most of the cluster members are still in a similar area. This could suggest the clusters are not distinctly separated.

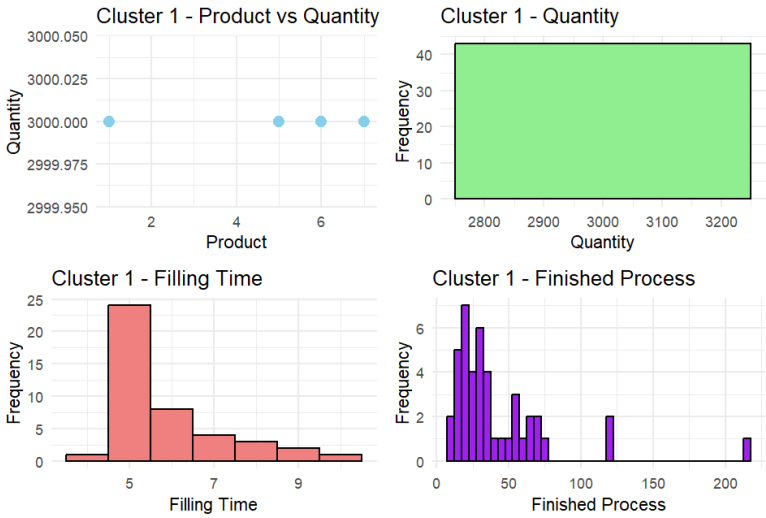


Figure 4. 3 Cluster 1 Characteristics

Figure 4.3 shows the characteristics of cluster one based on each variable in the research. Based on the product, the products that exist in this cluster are product 1 (Pertalite), product 5 (Pertamax), product 6 (Pertamina Dex 50 PPM, Bulk), and product 7 (Solar). The quantity at 3000 where all the cluster member has the same value of quantity. The filling time is the highest frequency at 5 minutes and gets lower to the next; also, the process ranges from four to 10 minutes. The finished process is mostly highest under 100 minutes, where the highest frequency is under 50 minutes. Based on that, cluster 1 is considered as a cluster that has low quantity, low filling time, and high finished process.

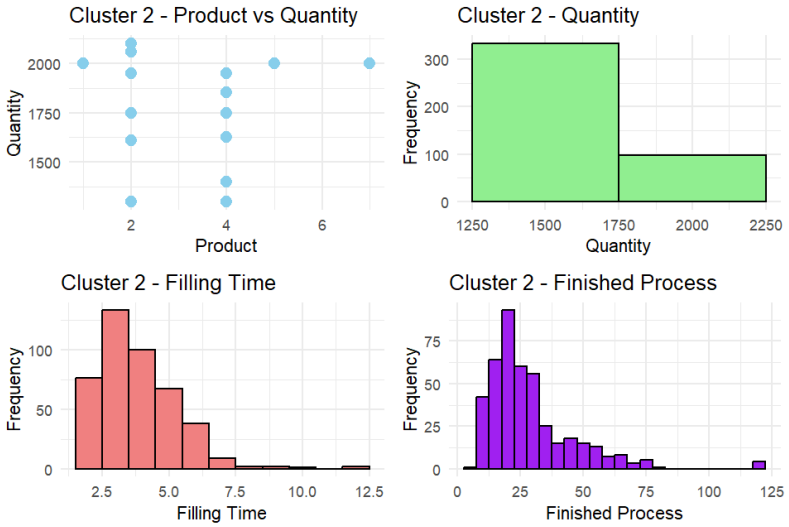


Figure 4. 4 Cluster 2 Characteristics

Figure 4.4 shows the characteristics of cluster two, where the product that is most commonly shown is product 2 (Biosolar B35), 4 (Dexlite), product 5 (Pertamax), and 7 (Solar). The quantity is high at 1500 and the second is 2000. Filling time, the highest frequency is 2.5 minutes and ranges between 2 to 10 minutes also, there is a process in 12.5 minutes. The finished process has the highest frequency in under 25 minutes and ranges between 25 – 75 minutes also, there is a process that falls in 125 minutes. Then, cluster 2 is considered as a cluster that has low quantity, low filling time, and medium finished process.

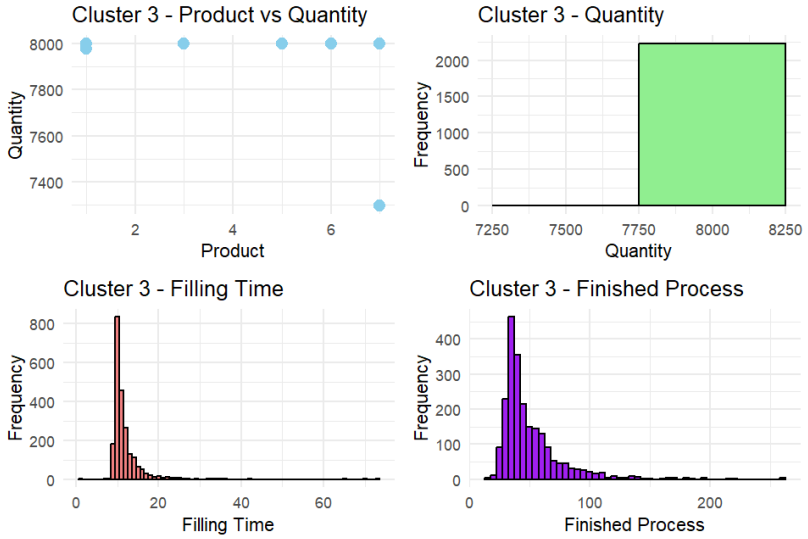


Figure 4. 5 Cluster 3 Characteristics

Figure 4.5 shows the characteristics of cluster 3 where the product that exists in this cluster is Product 1 (Pertalite), product 3 (Avtur), product 5 (Pertamax), product 6 (Pertamina Dex 50 PPM, Bulk), and product 7 (Solar). The quantity that existed in this cluster is 8000. The filling time is at the highest frequency, around 10 minutes, and ranges around 10 to 30 minutes. The finished process is at the highest frequency, around 30 minutes, and ranges from 10 to 100 minutes the most. Then, cluster 3 is considered as the cluster that has high quantity, high filling time, and low finished process.

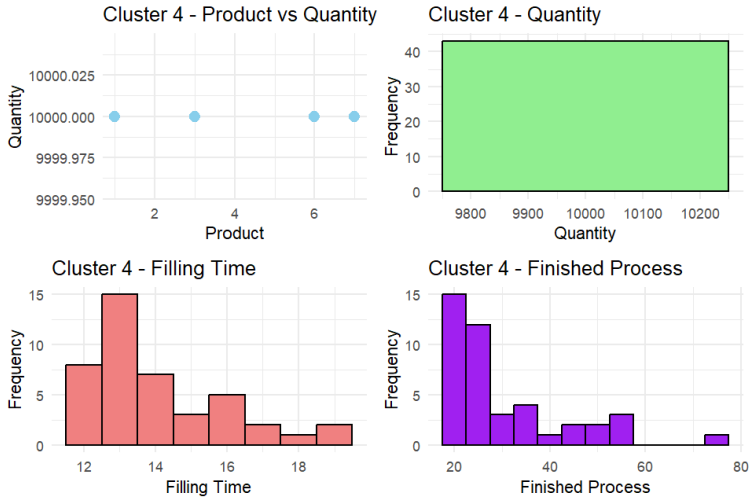


Figure 4. 6 Cluster 4 Characteristics

Figure 4.6 shows the characteristics of cluster 4, where the product that existed in this cluster is product 1 (Pertalite), product 3 (Avtur), product 6 (Pertamina Dex 50 PPM, Bulk), and product 7 (Solar). The quantity is all fall in 10000. The filling time has the highest frequency of 13 minutes and ranges from 12 to 19 minutes. The finished process has the highest frequency of 20 minutes and ranges from 20 to 50 minutes. Also, there is a process of 70 minutes. Consequently, cluster 4 is considered as the cluster that has high quantity, high filling time, and low finished process.

Figure 4. 7 Cluster 5 Characteristics

Figure 4.7 shows the characteristics of cluster 5, where the product that exists in this cluster is product 1 (Pertalite), product 2 (Biosolar B35), product 4 (Dexlite), product 5 (Pertamax), product 6 (Pertamina Dex 50 PPM, Bulk), and product 7 (Solar). The quantity distribution is at the highest frequency of 5000, and there is also some in 4500. The filling time mostly ranges from 5 to 10 minutes and at the highest frequency at 6 minutes. There are also several filling times in 15 minutes 0 that could be caused by filling time under 2 minutes. The finished process is at the highest rate, probably in 20 minutes and ranging from 0 to 30 minutes, then 40 to 60 minutes, and 80 to 100 minutes. There is also a finished process above 200 minutes. Then, cluster 5 is considered as the cluster that has medium quantity, low filling time, and low finished process.

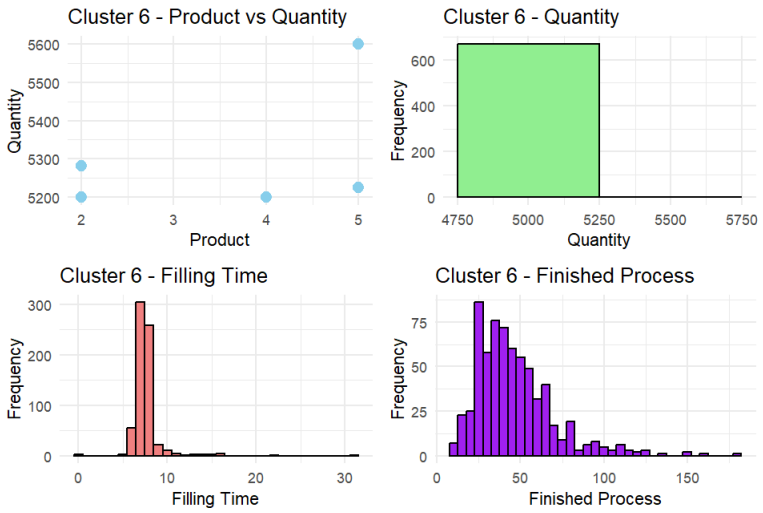
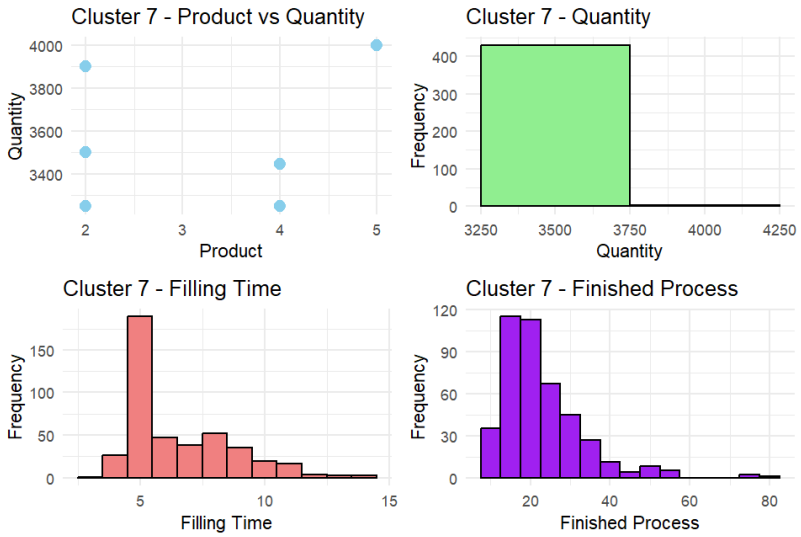


Figure 4. 8 Cluster 6 Characteristics

Figure 4.8 shows the characteristics of cluster 6, where most of the products that exist in this cluster are product 2 (Biosolar B35), product 4 (Dexlite), and product 5 (Pertamax). The quantity that existed in this cluster is 5000. The filling time ranges from 5 to 10 minutes and, at the highest frequency, approximately 7 minutes. The finished process is ranging from 0 to 125 minutes and at the highest frequency at 25 minutes. Cluster 6 is considered as the cluster that has a medium quantity, medium filling time, and medium finished process.



,

Figure 4. 9 Cluster 7 Characteristics

Figure 4.9 shows the characteristics of cluster 7, where the products that existed in this cluster are product 2 (Biosolar B35), product 4 (Dexlite), and product 5 (Pertamax). The quantity that existed in this cluster is 3500, which reaches the frequency of 400. The filling time ranges from 4 to 14 minutes, where the highest frequency is at 5 minutes. The finished process ranges

from 10 to 50 minutes, where the highest peak is at 15 minutes. Cluster 7 is considered as the cluster that has low quantity, low filling time, and low finished process.

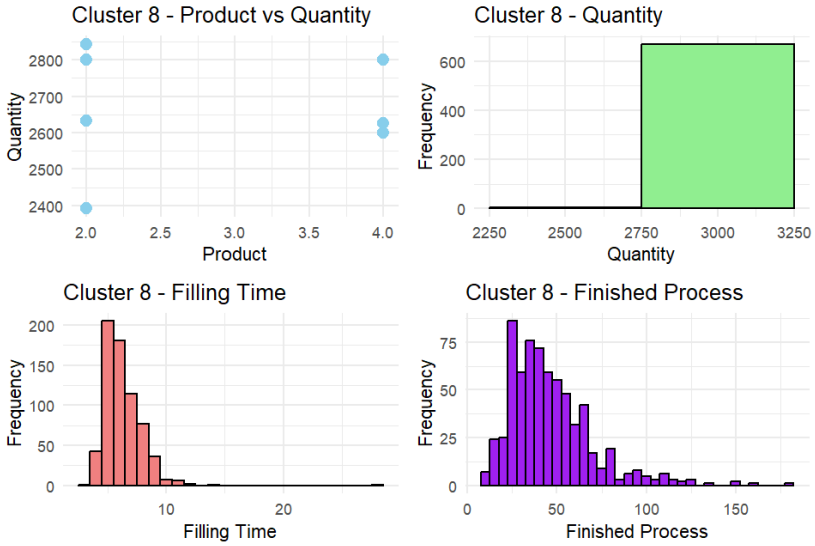


Figure 4. 10 Cluster 8 Characteristics

Figure 4.10 shows the characteristics of cluster 8, where the product that exists in this cluster is product 2 (Biosolar B35) and product 4 (Dexlite). The quantity that existed in this cluster is 3000. The filling time is ranging from 4 to 11 minutes, and the highest frequency is at 5 minutes. The finished process had a wide range from 15 to 100 minutes and more. The highest frequency is at 30 minutes. Cluster 8 is considered as the cluster that has low quantity, low filling time, and medium finished process.

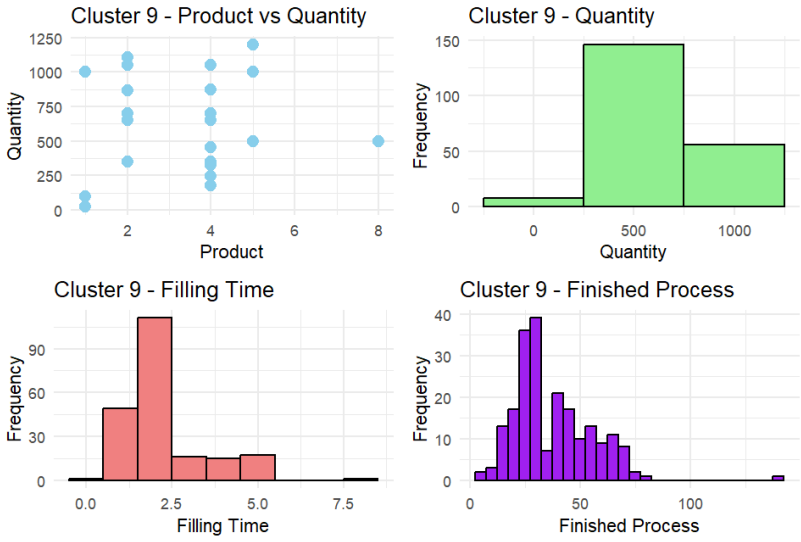


Figure 4. 11 Cluster 9 Characteristics

Figure 4.11 shows the characteristics of cluster 9, where the product that exists in this cluster is product 1 (Pertalite), product 2 (Biosolar B35), product 4 (Dexlite), and product 5

(Pertamax). The quantity is at the highest frequency at 500 and ranges from 500 – 1000 units. The filling time ranges from 2 to 5 minutes and at the highest peak at around 2 minutes. The finished process ranges from 10 to 60 minutes, and the highest frequency is approximately 25 minutes. Cluster 9 is considered as the cluster that has low quantity, low filling time, and low finished process.

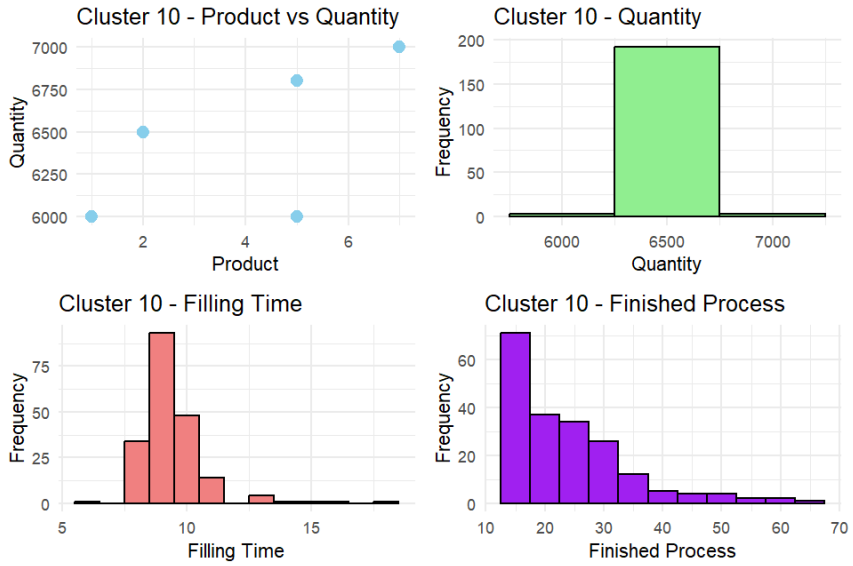


Figure 4. 12 Cluster 10 Characteristics

Figure 4.12 shows the characteristics of cluster 10, where the product that existed in this cluster is product 1 (Pertalite), product 2 (Biosolar B35), product 5 (Pertamax), product 7 (Solar). The quantity is in the highest frequency at 6500. The filling time is ranging from 8 minutes to 11 minutes and has the highest frequency in 9 minutes. The finished process is ranging from 10 to 60 minutes, and the highest frequency is in 10 minutes. Cluster 10 is considered as the cluster that has medium quantity, medium filling time, and low finished process.

4.3 Association Rule Using FP-Growth Algorithm

The author conducts the association rules using FP-Growth using Rapidminer. Before conducting the data processing, the author transformed the data into binary values for each variable. After the transformation is done, the author conducts the association rules using several amounts of confidence and support as follows:

Table 4. 2 Association Rules Parameter Trial

Trial No	Minimum Support	Minimum Confidence	Rules Formed
1	0,1	0,5	52

Trial No	Minimum Support	Minimum Confidence	Rules Formed
2	0,01	0,5	148
3	0,001	0,5	248

Based on the parameter above, the author has changed the support to gain more coverage of products. Since the first and second trials don't contain any rules that can mention all the product kinds, the author chose the third trial and used 0,001 as the minimum support and 0,5 as the minimum confidence since it's able to provide the information regarding the relationship clearer than others. Below is the result of the association rules.

Table 4. 3 Association Rules Result

No	Premises	Conclusion	Support	Confidence	Lift
1	Product_7	Filling Time_Low	0,0028	0,5000	1.310
2	Quantity_Low, Product_1	Finished Process_High	0,0022	0,5000	0.999
3	Filling Time_High, Product_6	Finished Process_Low	0,0046	0,5000	1.196
4	Product_7	Finished Process_Low, Filling Time_Low	0,0028	0,5000	2.372
5	Finished Process_Low, Product_7	Filling Time_Low	0,0028	0,5000	1.310
6	Filling Time_Low, Product_6	Quantity_Low	0,0010	0,5000	1.394
7	Filling Time_Low, Product_6	Quantity_Medium	0,0010	0,5000	2.680
8	Filling Time_High, Quantity_Medium, Product_6	Finished Process_Low	0,0010	0,5000	1.196
9	Quantity_Low, Product_5	Finished Process_Low, Filling Time_Low	0,0088	0,5000	2.372
10	Filling Time_Low, Product_6	Finished Process_Low, Quantity_Low	0,0010	0,5000	2.298
11	Filling Time_Low, Product_2, Quantity_Medium	Finished Process_High	0,0248	0,5020	1.003
12	Finished Process_High, Filling Time_Medium	Product_2, Quantity_Medium	0,0248	0,5020	3.583
13	Filling Time_High, Product_2	Finished Process_Low, Quantity_Medium	0,0222	0,5023	5.638
14	Finished Process_Low	Filling Time_Low	0,2108	0,5046	1.322
15	Finished Process_Low, Product_5	Filling Time_Low	0,0088	0,5057	1.325
…	….....	….....	…..	…..	…..

No	Premises	Conclusion	Support	Confidence	Lift
237	Finished Process_Low, Filling Time_Low, Product_5	Quantity_Low	0,0088	1,0000	2.790
238	Quantity_Low, Product_6	Finished Process_Low, Filling Time_Low	0,0010	1,0000	4.745
239	Finished Process_Low, Quantity_Low, Product_6	Filling Time_Low	0,0010	1,0000	2.621
240	Filling Time_Low, Quantity_Low, Product_6	Finished Process_Low	0,0010	1,0000	2.393
241	Finished Process_Low, Quantity_Medium, Product_4	Filling Time_Low	0,0016	1,0000	2.621
242	Quantity_Medium, Product_7	Finished Process_Low, Filling Time_Low	0,0022	1,0000	4.745
243	Finished Process_Low, Quantity_Medium, Product_7	Filling Time_Low	0,0022	1,0000	2.621
244	Filling Time_Low, Quantity_Medium, Product_7	Finished Process_Low	0,0022	1,0000	2.393
245	Filling Time_Medium, Product_6	Finished Process_Low, Quantity_Medium	0,0026	1,0000	1.122
246	Finished Process_Low, Filling Time_Medium, Product_6	Quantity_Medium	0,0026	1,0000	5.361
247	Quantity_Medium, Filling Time_Medium, Product_6	Finished Process_Low	0,0026	1,0000	2.393
248	Quantity_Low, Product_4, Finished Process_Medium	Filling Time_Low	0,0030	1,0000	2.621

Figure 4.3 above is the association rules result using Rapidminer, where all of the rules resulted in all rules passing the lift ratio > 1. Since most of the rules are in 1.000 to 2000, meaning all the rules above are valid except for rule number two, which only has the lift ratio of 0,999, it's considered as not valid. The rule that has the highest lift ratio with a value of 8.821 is rule 161, which is low filling time, product 7, and low finished process, medium quantity. The rule that has the lowest lift ratio with a value of 1.003 is rule 11 which, is low filling time, product 2, medium quantity, and high finished process.

Below is the visualization:

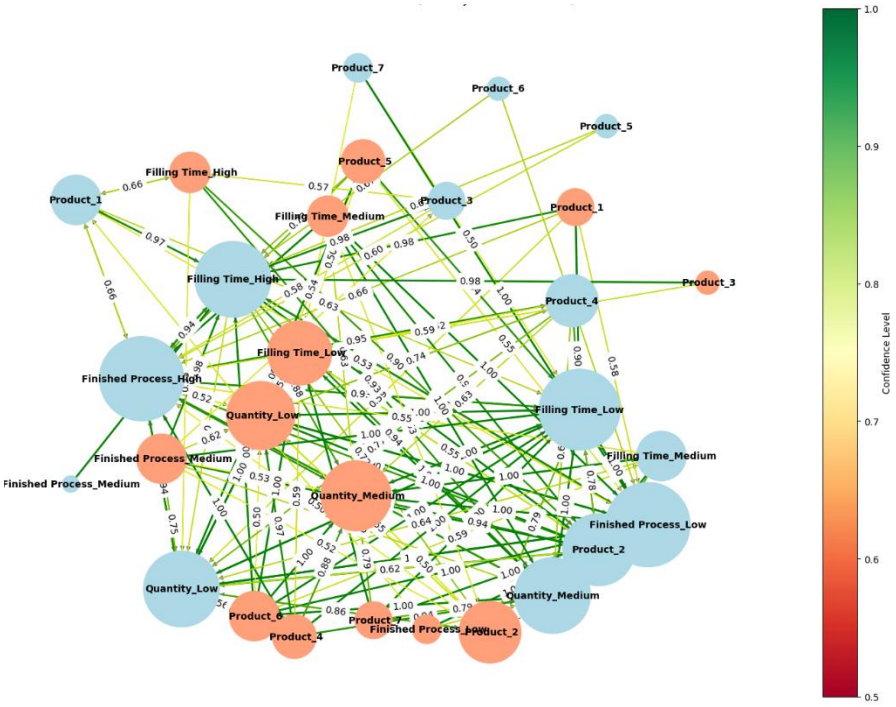


Figure 4. 13 Association Rule Visualization

Figure 4.13 is the visualization for the association rules result where the node's color has different sizes that represent how connected they are to other rules. The light blue color represents premises (antecedents), and the light orange represents consequences. The lines that connect the nodes are represented in a heatmap where green indicates high confidence, and yellow and red indicate lower confidence. For example, finished process_high and filling time_high have a high confidence level of 0.94, Product_1 and filling time_high has lower confidence of 0.66, Product_1 and finished process_high has lower confidence of 0.66, and filling time_high to Product_3 has a high confidence level of 0.98.

Based on the rules above, we can also find the association between each product to the quantity, filling time, and finished process as follows:

Table 4. 1 Association Rules Conclusions for Each Product

No	Product Name	Quantity	Finished Process	Filling Time	Confidence	Support	Lift Ratio
1	Pertalite	Medium	Low	Low	0,5263 – 0,7000	0,0020 – 0,0356	1.259 – 1.834
2	Biosolar B35	Low	Low	Low	0,5492 – 0,6944	0,0022 – 0,1841	1.452 – 2.605
3	Avtur	Medium	High	High	0,9740 – 0,9805	0,0659 – 0,1123	1.919 – 1.960

No	Product Name	Quantity	Finished Process	Filling Time	Confidence	Support	Lift Ratio
4	Dexlite	Low	Low	Low	0,6316 – 0,8969	0,0739 – 0,1271	2.441 – 2.529
5	Pertamax	Medium	High	High	0,6008 – 0,6324	0,0304 – 0,0320	1.200 – 1.264
6	Pertamina Dex, 50 PPM Bulk	Low	Low	High	0,6522 – 0,6667	0,0090 – 0,0092	1.332 – 1.561
7	Solar	Medium	Low	Low	0,5000 – 0,7857	0,0022 – 0,0056	2.393 – 8.821

4.4 Artificial Neural Network (ANN)

Before processing the data, the author performs data pre-processing. This step is conducted using R Studio, where the data is transformed into categorical ranges for all variables, except for the product variable, which is converted into corresponding numerical values. After pre-processing, the transformed data is processed using Google Colab. The author also conducts a sensitivity analysis on the number of epochs and learning parameters. The specific values for the epoch count and learning parameters tested are as follows:

Table 4. 4 Epoch and Learning Rate Trial

No	Epoch	Learning Rate
1.	30	0.01
2.	50	0.001
3.	100	0,0005

Table 4.4 displays the outcomes of experiments conducted with various epochs and learning rates. In the initial trial, 30 epochs were implemented with a learning rate of 0.01, leading to a consistent training accuracy of 0.9 and an increase in validation accuracy from 0.83 to 0.90. The training loss showed a steady decline from 0.40 to 0.30 without any fluctuations, while the validation loss began at over 0.6 and fell to 0.35. This elevated learning rate enabled the model to learn rapidly during the first trial.

In the second trial, a total of 50 epochs were tested using a learning rate of 0.001. The training accuracy began at approximately 0.85 and steadily rose to 0.90, whereas the validation accuracy remained consistently at 0.90. The model's training loss decreased from 0.6 to 0.35,

while the validation loss fell from 0.39 to 0.32, demonstrating minor fluctuations within the range of 0.3 to 0.4.

In the third trial, we utilized 100 epochs and maintained a learning rate of 0.001. This led to an increase in training accuracy from 0.6 to 0.9, while the validation accuracy remained around 0.90. The model loss consistently decreased to 0.32, starting from a validation loss of 0.61 and also dropping to 0.32. This trial showed a more precise convergence, likely attributed to the reduced learning rate and the increased number of epochs, which contributed to the model's steady improvement.

All trials exhibited a similar trend of increasing accuracy, albeit with varying values, and remained stable without significant fluctuations. However, the loss patterns differed: the first trial displayed some variability, while the second and third trials followed a comparable trend, albeit within different ranges. Based on these findings, the second trial was selected as the optimal configuration for the artificial neural network (ANN). The learning rate was moderate neither too low (as seen in the third trial) nor too high (as observed in the first trial) which helped avoid overshooting. Furthermore, the number of epochs allowed the model to evade over-fitting (as in the third trial) or under-fitting (as in the first trial). The second trial yielded more consistent results, with a lower loss value and a high accuracy of approximately 90.9%. The training and validation losses suggested that the model achieved better convergence, exhibiting no signs of over-fitting or under-fitting.

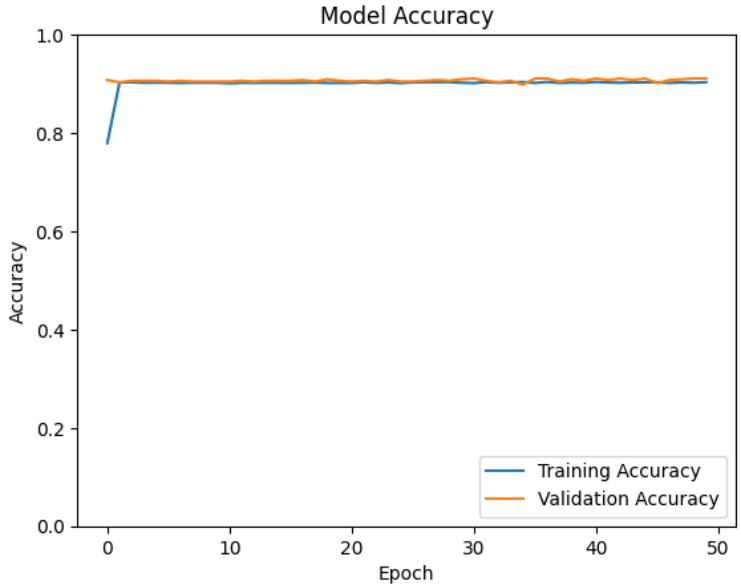


Figure 4. 14 Training Model Accuracy

Figure 4.13 provides the result for the accuracy of the training model. The accuracy is at a high level where it is around 0.8 or 80% and the accuracy validation goes even higher at 0.9 or 90%. By the training model accuracy, it can be assumed that the model has achieved optimal learning and there is no fluctuation where the validation stays consistent since there is no significant increase or decrease in validation.

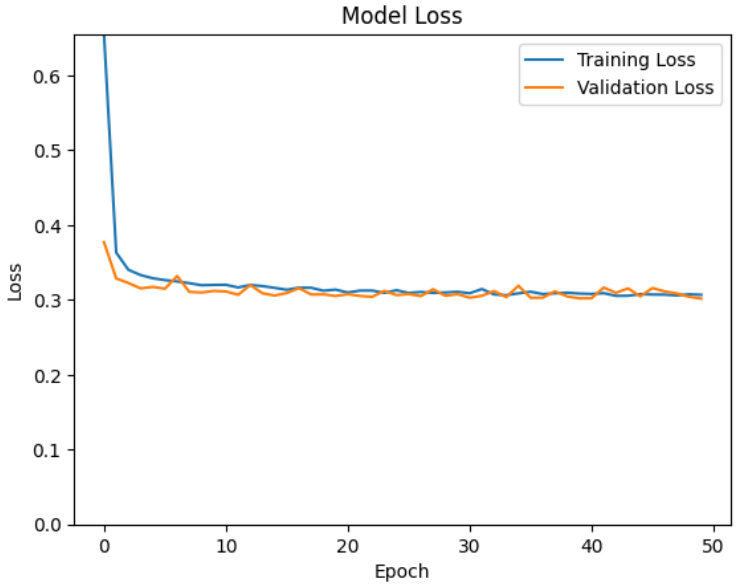


Figure 4. 15 Training Model Loss

Figure 4.14 shows the loss of the training model where the loss is decreased rapidly which could indicate the model to minimize the error of the prediction and outcome. The loss validation also declined but also changed but stayed within 0.3 to 0.4 which could indicate it reached minimal loss. The validation loss and the loss then stay close throughout the drop which could indicate the model does not overfit and stays stable.

Table 4. 5 Training Epoch

Epoch	Accuracy	Val_Accuracy	Loss	Val_Loss
1	0,779429	0,907812	0,654621	0,377216
2	0,90262	0,903125	0,363268	0,328748
3	0,903402	0,90625	0,340249	0,322597
4	0,902229	0,90625	0,333243	0,315403
5	0,90262	0,90625	0,328961	0,317414
6	0,902229	0,904688	0,326452	0,315016
7	0,901838	0,90625	0,324691	0,331905
8	0,902229	0,904688	0,322381	0,310752
9	0,902229	0,904688	0,319635	0,309949
10	0,902229	0,904688	0,320074	0,311999
11	0,901056	0,904688	0,320192	0,311354

Epoch	Accuracy	Val_Accuracy	Loss	Val_Loss
12	0,902229	0,90625	0,316643	0,306837
13	0,901838	0,904688	0,319986	0,31991
14	0,902229	0,90625	0,318469	0,308845
15	0,902229	0,90625	0,316122	0,305855
16	0,901838	0,90625	0,313587	0,30919
17	0,902229	0,907812	0,316322	0,31571
18	0,90262	0,904688	0,316242	0,307268
19	0,901838	0,909375	0,31228	0,307437
20	0,901838	0,90625	0,313719	0,305248
21	0,901838	0,904688	0,309935	0,307669
22	0,903402	0,90625	0,312407	0,30535
23	0,901838	0,904688	0,312482	0,304149
24	0,903011	0,907812	0,309505	0,312132
25	0,901447	0,904688	0,312943	0,306408
26	0,903011	0,904688	0,309301	0,307916
27	0,903402	0,90625	0,310615	0,305018
28	0,903402	0,907812	0,309414	0,314212
29	0,903402	0,90625	0,309909	0,305572
30	0,902229	0,909375	0,310902	0,307696
31	0,901447	0,910937	0,308923	0,302952
32	0,903402	0,90625	0,314531	0,305522
33	0,902229	0,903125	0,307354	0,311908
34	0,903011	0,90625	0,306282	0,303792
35	0,903794	0,898438	0,308771	0,318919
36	0,901838	0,910937	0,31093	0,302867
37	0,903794	0,910937	0,307809	0,302959
38	0,901838	0,904688	0,308812	0,311537
39	0,90262	0,909375	0,309674	0,304699
40	0,902229	0,90625	0,308504	0,302095
41	0,903794	0,910937	0,307969	0,302196
42	0,90262	0,907812	0,309055	0,316515
43	0,902229	0,910937	0,305415	0,309446
44	0,903011	0,907812	0,30556	0,315441
45	0,903011	0,910937	0,307735	0,304746
46	0,903402	0,901563	0,307191	0,315916
47	0,901838	0,907812	0,307058	0,311361
48	0,903011	0,909375	0,306021	0,308673
49	0,902229	0,910937	0,307527	0,304193
50	0,903402	0,910937	0,306827	0,301709
Accuracy	0.9046	**New Test Accuracy**		0.9239

Epoch	Accuracy	Val_Accuracy	Loss	Val_Loss
Loss	0.3233	**New Test Loss**		0.2657

Throughout Table 4.3, the accuracy from epoch one to epoch 50 is kept increased as well as the accuracy that stays within the same range. The loss however decreased rapidly from epoch one to epoch two and kept decreasing until epoch 50. The validation loss also decreases and increases but is not as significant and stays within the same range. Based on the accuracy, the model achieved 90.46% accuracy and had a loss of 32.33%. On the new test, the accuracy has increased to 92.39% and the loss has lowered to 26.57%. Since the result of the loss decreased, the new testing model is better than the previous model.

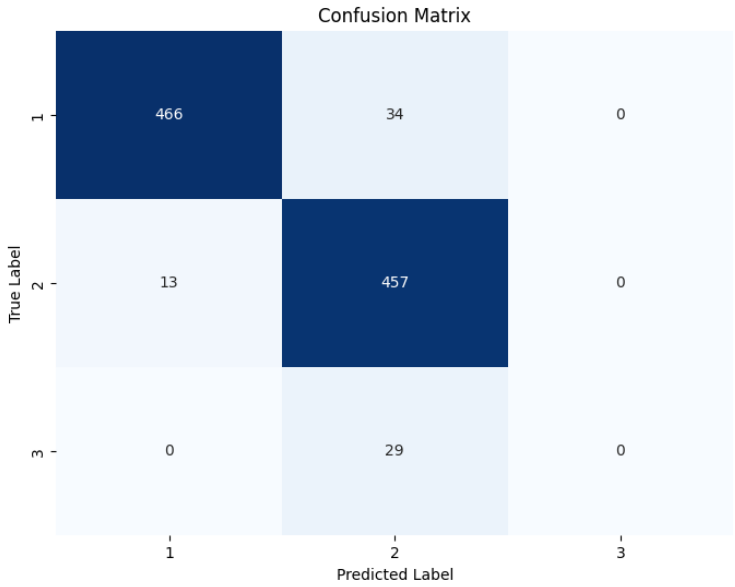


Figure 4. 16 Confusion Matrix Testing Data

Notes:

1: Low

2: Medium

3: High

Figure 4.15 is the confusion matrix of the testing data result where class 1 (Low) has 466 correctly classified with 34 misclassified as class 2 (Medium). Class 2 (Medium) has 457 correctly classified with 13 misclassified as class 1 (Low). Lastly for class 3 (High) 29 is misclassified as class 2.

Table 4. 6 Classification Summary

Class	Precision	Recall	F1-Score	Support
1	0.97	0.93	0.95	500

2	0.88	0.97	0.92	470
3	0.00	0.00	0.00	29
Accuracy			0.92	999
Macro Avg	0.62	0.63	0.63	999
Weighted Avg	0.90	0.92	0.91	999

Table 4.4 is the classification summary result for the testing data. The accuracy for the overall model is 92%. For class 1, the model demonstrates strong performance, achieving a precision of 97%, meaning it correctly predicts class 1 nearly all the time. Its recall is 93%, indicating that the model successfully identifies most instances of class 1. The F1-score of 0.95 reflects a good balance between precision and recall. Similarly, for class 2, the model performs well with a precision of 88% and a recall of 97%, leading to an F1-score of 0.92. This suggests that while the model sometimes misclassifies instances of class 2, it still captures the majority of them correctly. However, the model struggles significantly with class 3, which is a minority class with only 29 instances in the dataset. It has a precision, recall, and F1-score of 0, indicating that the model fails to correctly identify any instances of class 3.

The macro average metrics show lower scores around 0.63 for precision, recall, and F1-score. In contrast, the weighted average metrics that account for the class distribution remain high at around 0.90 for precision and 0.91 for the F1-score, reflecting the model's strong performance on the more frequent classes (1 and 2).

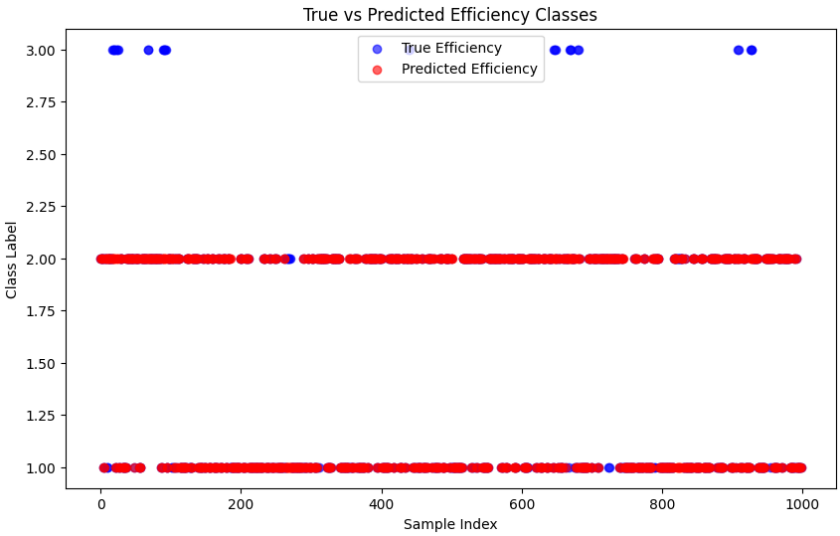


Figure 4. 17 True vs Predicted Efficiency Class

Figure 4.16 visualizes the result of the true vs predicted efficiency class. Where 1 represents low, 2 represents medium, and 3 represents high. Based on the graph, most of the predicted

labels are only shown on medium and low where it aligned with the true labels. The high is misclassified since there is no predicted efficiency there.

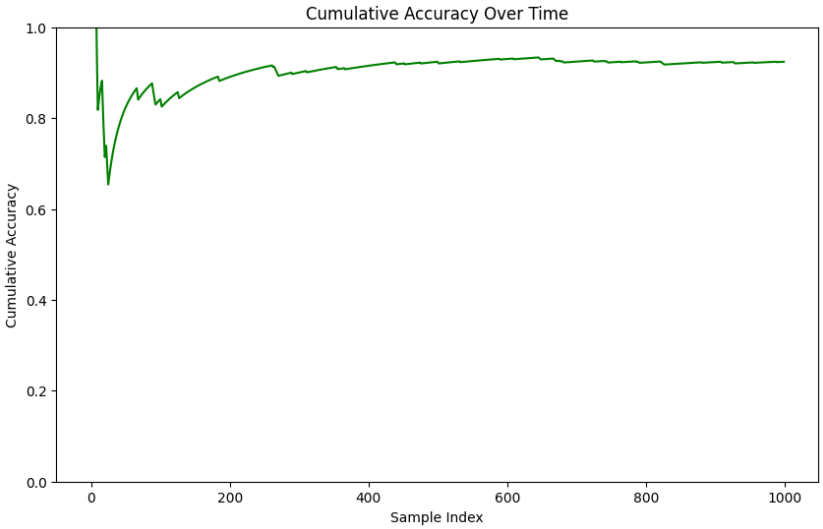


Figure 4. 18 Cumulative Accuracy

Figure 4.8 shows the cumulative accuracy of the whole model where the model has an unstable accuracy since it fluctuates significantly. However, the more the sample the more stable the model became where it maintained accuracy at 90%. This result mostly stabilizes after more data is processed. This could indicate that class 1 (Low) and class 2 (Medium) are more common, and class 3 (High) is less common. Which also shown in the previous output where class 3 has no predicted output that could be caused by the class imbalance.

Below is the result of the turnoff ratio in Figure 4.20:

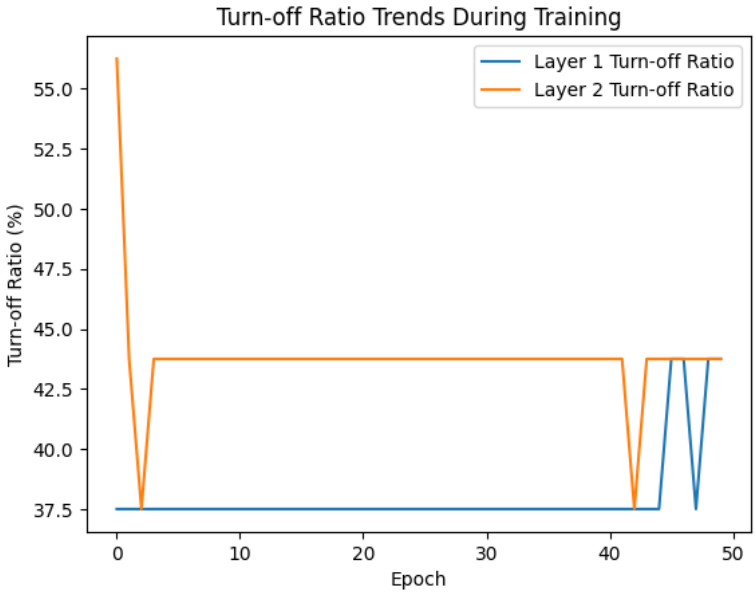


Table 4. 7 Turn off Ratio

The turn-off ratio is the proportion of inactive neurons that result in zero output. The second layer has a high turn-off ratio during the first epoch, starting from 55%. In the accuracy of the model, the first epoch has an accuracy of around 77.9% but improves to 90%, also indicated by the second layer turn-off ratio decreasing rapidly and increaseing to around 43%. The first layer has a lower turn-off, around 37.5%, until the last few epochs, where it increases to 43% and fluctuates after. This fluctuation also occurs in the second layer which causes in slight changes in accuracy.

Based on the result, the turn-off ratio is consistent throughout the whole model with minimum fluctuation through the end of the model. The accuracy that improves in the early epoch with a high turn-off ratio indicates the model is able to find useful patterns within the data altough some of the neurons is inactive.

4.5 Relationship Between Association Rules and K-Means Clustering Result

The result for each product association can be compared with the previous clustering result. Product 1 (Pertalite) is a member in several clusters such as cluster 1, cluster 3, cluster 4, and cluster 5. The result for the association rules to the K-Means clustering has some discrepancies where clusters one, three, and four have different quantities, and filling times. Whereas for cluster 5, the clustering characteristic is the same as the association rules result for product 1. Product 2 (Biosolar B35) is a member of cluster 2, cluster 6, cluster 7, cluster 8, cluster 9, cluster 10. The result for the cluster characteristics to the association rules has several discrepancies in Cluster 2, cluster 6, cluster 8, and Cluster 10, whereas cluster 7 and cluster 9 have the same result.

Product 3 (Avtur) is a member of clusters 3 and 4 where the association rules result, and the cluster characteristics have discrepancies in the quantity and finished process. Product 4 (Dexlite) is a member in cluster 2, cluster 6, cluster 7, cluster 8, and cluster 9. The comparison between the association rules result with the clustering has differences in Cluster 2, cluster 6, cluster 8, and cluster 9 where cluster 7 has the same result.

Product 5 (Pertamax) is a member in cluster 1, cluster 2, cluster 3, cluster 5, cluster 6, cluster 7, and cluster 10. The association result has similarities in cluster 5, 6, and 7 quantities also cluster 3 filling time, and cluster 1 finished process. Whereas the similarity with the cluster characteristics in quantity, filling time, and finished process. Product 6 (Pertamina Dex, 50 PPM Bulk) is a member of clusters 1, 3, 4, 5, and 9. The K-Means clustering result doesn’t have any

exact similarity with the association rules result, however, it matches with cluster 1 and 9 quantity, cluster 3 and 4 filling time, and cluster 3, 4, 5, and 9 finished processes. Product 7 (Solar) is a member of clusters 1, 3, 4, 5, 7, and 10. The K-Means clustering result with the association result has exact matches in cluster 5, whereas, for the other cluster, some of it matches with the quantity in cluster 10, filling time in clusters 1, 5, and 7, and finished process in clusters 3, 4,5,7, and 10.

4.6 Recommended Filling Shed Reconfiguration

Based on the calculation that has been conducted, the author can recommend the filling shed configuration. The recommendation is as follows:

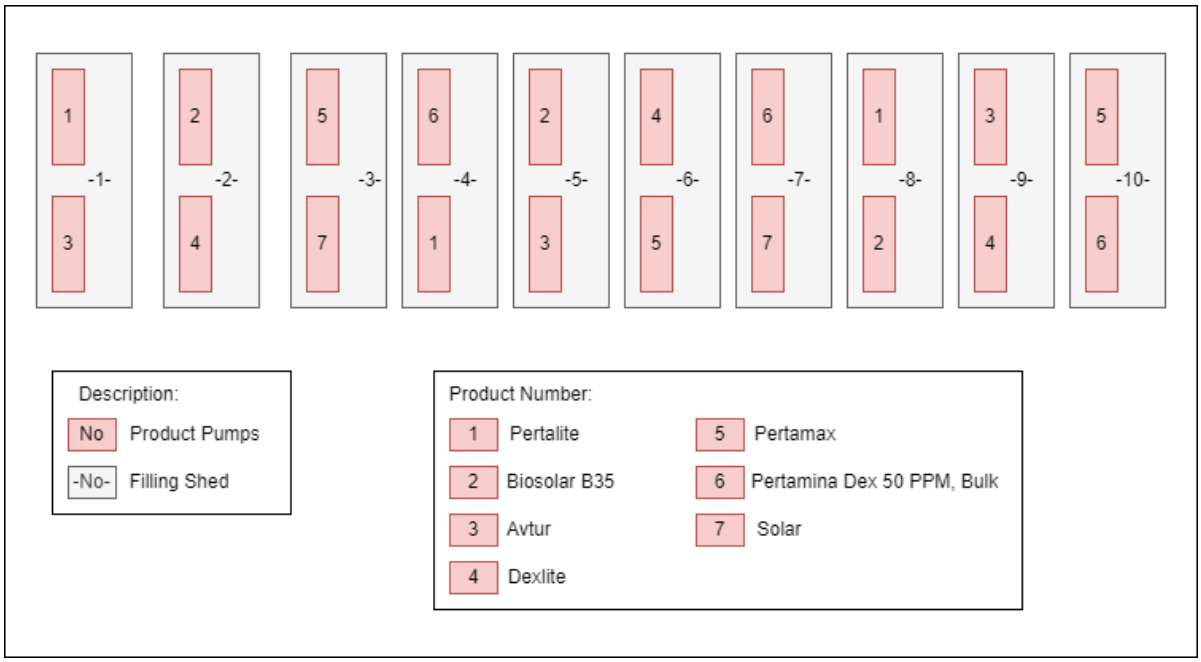


Figure 4. 19 Proposed Filling Shed Configuration

Figure 4.18 shows the proposed filling shed configuration where this new configuration resulted in several changes that has differences with the clustering result for the products. The first filling shed consists of Product 1 (Pertalite) and Product 3 (Avtur) where both of the products have similar characteristics but differ in the finished process where Pertalite has a low finished process and Avtur has a high finished process. This pair could balance the efficiency of the filling time. The second filling shed consists of Product 2 (Biosolar B35) and Product 4 (Dexlite) where both of the products have an exact similarity in the quantity, filling time, and finished process that could benefit in increasing the volume for this product order.

The third filling shed consists of Product 5 (Pertamax) and Product 7 (Solar) that has similarities in quantity but differences in the filling time where Pertamax is high and Solar is low, which also goes with the finished process. This configuration can help to balance the workload of the filling shed since Pertamax has higher efficiency than Solar. The fourth filling shed consists of Product 6 (Pertamina Dex, 50 PPM Bulk) and Product 1 (Pertalite). Both of these products have low-finished processes but have differences in quantity and filling time. This configuration is chosen since the Pertamina Dex, 50 PPM Bulk has a higher filling time than Pertalite to balance the workload of the finished process.

The fifth filling shed consists of Product 2 (Biosolar B35) and Product 3 (Avtur) where both of it has differences in quantity, filling time, and finished process. Avtur has higher efficiency that could help balance the Biosolar B35 since it has low efficiency overall. The sixth filling shed consists of Product 4 (Dexlite) and Product 5 (Pertamax) which also have differences in all the characteristics to help balance the filling time between low-efficiency and high-efficiency products.

The seventh filling shed consists of Product 6 (Pertamina Dex, 50 PPM Bulk) and Product 7 (Solar) where both of the has low finished processes but Pertamina Dex, 50 PPM Bulk has a high filling time to manage the filling shed flow more effectively. The eighth filling shed consists of Product 1 (Pertalite) and Product 2 (Biosolar B35) where both products have low filling time and finished process, but the Pertalite has a medium quantity.

The ninth filling shed consists of Product 3 (Avtur) and Product 4 (Dexlite) where which is one of the configurations that have high and low filling times and processes to help balance the filling shed. The tenth filling shed consists of Product 5 (Pertamax) and Product 6 (Pertamina Dex, 50 PPM Bulk) where both products have high filling times but differences in the finished process where Pertamax is higher since it has medium quantity and the Pertamina Dex, 50 PPM Bulk has a low quantity.

Below is the calculation for the filling rate result between the previous and proposed configuration:

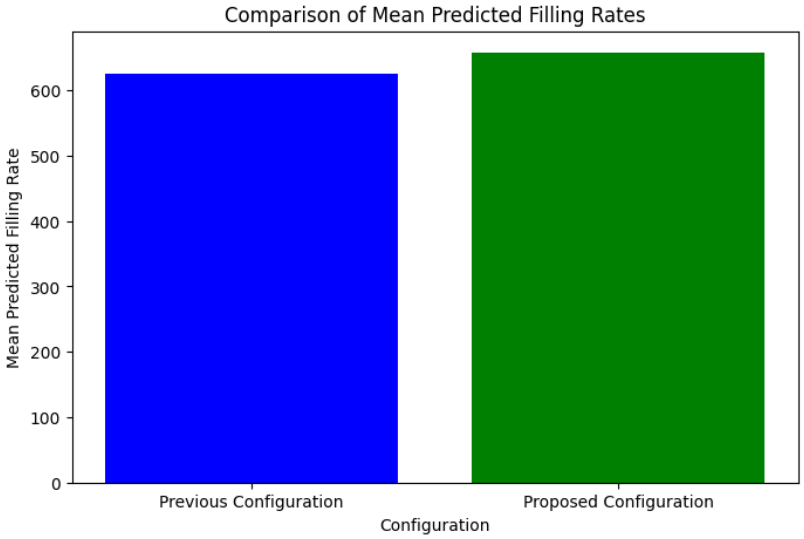


Figure 4. 20 Comparison of Mean Predicted Rate

Figure 4.20 illustrates a comparison of the performance between the previous and proposed filling shed configurations. The method employed involved calculating the filling rate by dividing the quantity by the time required for filling. Furthermore, an artificial neural network (ANN) was utilized for regression tasks to forecast continuous values. The findings reveal that the mean filling rate for the previous configuration was 626.02, whereas the proposed configuration reached a mean rate of 657.48. This suggests that the proposed configuration can enhance the efficiency of the filling shed.

CHAPTER V
DISCUSSION

5.1 K-Means Clustering using Silhouette Result Discussion

This research aims to determine the number of clusters to improve the current configuration that exists at the company since the segmentation can help to understand the buying pattern based on 500 historical data. The author conducts pre-processing to help standardize and remove noise from the data. Then, using the silhouette method to find the optimum amount of cluster (k) based on the highest silhouette value or closest to one. The highest amount of cluster in this thesis is 10 in which the silhouette value is approximately above 0.80. The amount of k is then used to conduct the K-Means clustering where the visualization shows the cluster grouping. The graph shows that cluster 3 has several data that are further than other data points indicated by cluster 3 to be broader than another cluster. Another cluster however groups in a similar area and overlaps with one another. This condition can be caused by the cluster is not distinct enough to be spread throughout the graph.

Cluster 1 has the characteristics of low quantity, low filling time, and high finished process since the product quantity is pretty much on the same amount of 3000 with four product kinds that are Pertalite, Pertamax, and Pertamina Dex 50 PPM Bulk, and Solar. The filling time also has a close range with most of the filling time taking 5 minutes and the finished process mostly under 25 minutes.

Cluster 2 has the characteristics of low quantity, low filling time, and medium finished processed since the product quantity is only 1500 and 2000 kL with four products that are Biosolar B35, Dexlite, Pertamax, and Solar. The filling time is also mostly under 5 minutes supported by the finished process which is also mostly under 50 minutes.

Cluster 3 has the characteristics of high quantity, high filling time, and low finished process since the product quantity is the same amount of 8000 kL with five products that are Pertalite, Avtur, Pertamax, Pertamina Dex 50 PPM Bulk, and Solar. The filling time is mostly under 20 minutes supported by the finished process which is mostly under 100 minutes.

Cluster 4 has the characteristics of high quantity, high filling time, and low finished process indicated by the product quantity is on 10000 kL and has four products that are Pertalite, Avtur, Product 6, and Solar. The filling time is on a close range with most of the process being under 18 minutes supported by the finished process that is mostly under 60 minutes.

Cluster 5 has the characteristics of medium quantity, low filling time, and low finished process indicated by the quantity of 5000 kL and has six products that are Pertalite, Biosolar B35, Dexlite, Pertamax, Pertamina Dex 50 PPM Bulk, and Solar. The filling time is mostly under 10 minutes also the finished process is mostly under 50 minutes.

Cluster 6 has the characteristics of medium quantity, low filling time, and low finished process shown by the quantity of 5000 kL with three products that are Biosolar B35, Dexlite, and Pertamax. The filling time is mostly under 10 minutes with the finished process mostly under 50 minutes.

Cluster 7 has the characteristics of low quantity, low filling time, and low finished process indicated by the quantity of 3500 kL with three products that are Biosolar, Dexlite, and Pertamax. The filling time is mostly under 10 minutes also the finished process is mostly under 40 minutes.

Cluster 8 has the characteristics of low quantity, low filling time, and medium finished process shown by the quantity of 3000 kL with two products that are Biosolar B35 and Dexlite. The filling time is mostly under 10 minutes with the finished process that mostly falls under 100 minutes.

Cluster 9 has the characteristics of low quantity, low filling time, and low finished process indicated by the quantity of 500 kL and 1000 kL with four products that are Pertalite, Biosolar B35, Dexlite, and Pertamax. The filling time mostly falls under 5 minutes with the finished process that is mostly under 75 minutes.

Cluster 10 has the characteristics of medium quantity, medium filling time, and low finished process shown by the quantity of 6500 kL with four amounts of product are Pertalite, Biosolar B35, Pertamax, and Solar. The filling time is mostly under 10 minutes also the finished process is mostly under 40 minutes.

This method of clustering enables the company to categorize transactions according to their similarities, thereby aiding in the identification of patterns among products and the attributes of each cluster. This understanding empowers the company to make well-informed decisions regarding the arrangement of products and the selection of complementary items to pair with them. Consequently, the company can enhance efficiency and reduce bottlenecks in the filling process.

5.2 Association Rules using FP-Growth Algorithm Result Discussion

The association rules using FP-Growth are conducted in several trials by changing the minimum support to get the optimum result that could represent the provided historical data. The first trial used minimum support of 0,1 and minimum confidence of 0,5 where it formed 52 rules. However, the rules in this first trial do not cover all the product kind. Then, the second trial was done using minimum support of 0,01 and minimum confidence of 0,5 which formed 148 rules. This second trial already covers all the products but product 7 only exists in 2 rules. The last trial was done by changing the minimum support to 0,001 and minimum confidence of 0,5 where it formed 248 rules. These rules already cover all the data as well where all the products exist several times in the rules. The author used the last trial with a minimum support of 0,001 and a minimum confidence of 0,5 as the minimum threshold for this research. This threshold resulted in all 248 rules being considered valid since all of it has a lift ratio > 1. The rule of "Filling time low, Product 7 => Finished process low, Quantity Medium" with confidence of 78,57%, support of 0,22% from the total transaction, and lift ratio of 8.821. This rule indicates that "Filling time low and Product 7" have the likelihood for the "Finished process low and Quantity Medium" to occur. The second highest rule is "Product 4, Filling time medium => Finished process high, Quantity Medium" with a confidence of 64,16%, support of 0,72% from total transaction, and lift ratio of 7.338. This rule indicates that "Product 4, Filling time medium" has the likelihood for the "Finished process high and Quantity medium" to occur.

The third highest rule is "Filling time high, Product 2 => Finished process low, Quantity Medium" with confidence of 50,23%, support of 2,22% from the total transaction, and lift ratio of 5.638. This rule indicates that "Filling time high and Product 2" have the likelihood for the "Finished process low and Quantity medium" to occur.

Association rules can help both the company, and the author identify connections among products, quantities, filling times, and completed processes. By utilizing this method, the author can ascertain which products generally have larger quantities, extended filling times, or a greater number of completed processes. This information is crucial for making informed decisions, such as pairing products with shorter filling times with those that have longer ones, thereby optimizing the overall process by balancing these elements.

5.3 Relationship Between Association Rules and K-Means Clustering Discussion

Based on the result comparison between the association rules and K-Means clustering mentioned in the previous chapter, it can be said that there are some results the same on several clusters, but it has differences in other clusters. This result can be caused by different product characteristics that can be seen in the association rules result where several products that have different characteristics go into the same cluster. This can cause discrepancies in the result. The other results also can be affected by real-life conditions such as Avtur that mostly ordered in an exact amount of 8000 kL or 10000 kL. Also, the finished process can be affected by some car tanks re-entering the filling shed if they have to deliver more than one kind of product. Therefore, in deciding which product should go in the proposed recommendation, it is crucial to also take into account the real-life condition and balance it with the calculation result. Since the company, considered adding the filling shed from 9 to 10 filling shed, where the current filling shed contains 17 product pumps it can be said that one filling shed contains two product pumps.

In this study, K-Means clustering and association rules serve as complementary techniques that can be utilized independently but also function effectively together. K-Means clustering emphasizes the grouping of data based on similarities, whereas association rules examine and quantify the relationships between products. To suggest a reconfiguration, both methods can inform decisions regarding product placement and combinations.

The number of clusters (k) identified by the silhouette algorithm forms the foundation for the number of filling sheds, as it relies on the highest silhouette value. This reflects a real-world situation in which the addition of an extra filling shed enables a greater variety of product combinations. The clustering results, which categorize products according to their characteristics, can guide decisions regarding the appropriate products to include in each cluster.

Association rules evaluate the connections between products based on quantity, filling time, and filling shed. The rules generated from these analyses can subsequently be utilized to identify optimal product pairings. Therefore, both methods—K-Means clustering and association rules—are interrelated and inseparable in the context of this research. Together, they offer a holistic approach to reconfiguration and product management.

5.4 Artificial Neural Network (ANN) Result Discussion

For the ANN, the author divides the data into training data and testing data by the ratio of 70:30. The training has more data since it helped the ANN to learn the patterns and relationships from

the dataset resulting in a strong predicted outcome. The author then conducts training model accuracy which resulted in the training accuracy of around 80% with an accuracy of 90%, indicating that the training model has achieved the optimal learning and stays consistent in both accuracy and accuracy validation. The model loss also stays within 30% to 40% where it reaches minimal loss.

Then, the testing data resulted in the epoch accuracy also increasing from epoch one to 50, where the loss is decreasing. The accuracy previously on 90,46% and the new accuracy is 92,39%. The previous loss is 32,33% and the new loss is 26,57%. The confusion matrix also shows that there is a misclassified label in the Low there are 24 misclassified as Medium, in Medium there are 13 misclassified as Low, and in High, there are 29 misclassified as Medium.

The model has a strong prediction for the Low class, moderate for the medium class, and weak for the High class. Based on the result of Precision, Recall, and F1-Score the high class resulted in zero which indicates the model failed to identify any instances. This result is also supported by Figure 4.16 where the predicted efficiency is only able to predict for the Low and Medium Class mostly. The model can be improved by having more balanced datasets.

5.5 Proposed Filling Shed Configuration Discussion

The proposed configuration is developed to reflect real-world conditions, integrating a variety of products with both quick and slow filling times to balance the completion of processes while taking into account the differing volume requirements of each product. By merging high- and low-efficiency items, the configuration seeks to enhance filling times and evenly distribute the workload across the filling sheds, while also allowing for flexibility in managing seven distinct product types. Furthermore, the configuration streamlines product management by categorizing products according to their characteristics, particularly aiming to improve the handling of high-demand items. This setup is also less complicated, as it minimizes variability within each filling shed.

The results of this configuration are in line with the company's future strategy to increase the number of filling sheds. This method offers flexibility, as modifying the configuration only involves redirecting product pumps to various filling sheds. An analysis of projected filling rates, based on averages, demonstrates that the suggested configuration enhances filling rates compared to the earlier setup. This enhancement is due to the greater availability of

product pumps and the optimized product combinations in each filling shed, which contribute to minimizing bottlenecks during the tanker truck refilling process.

CHAPTER VI
CONCLUSIONS AND SUGGESTIONS

6.1 Conclusions

Based on the discussion in the previous chapter, below are the conclusions:

1. Based on the data processing for the K-Means clustering silhouette, the optimal number of k is 10 since it has the highest silhouette value that is closer to one than another k. Then, the clustering result using k of 10 resulted in 10 clusters. Most of the clusters are grouped similarly and overlapped to one another which can indicate the data points are not far from each other and not distinctly separated. Each cluster has its characteristics that are grouped based on the similarity of the data. The amount of k will be used as a base amount for the filling shed reconfiguration.
2. The author conducts several trials to determine the parameter. The minimum support that will be used in this research is 0,001 and 0,5 as the minimum confidence resulting in 248 rules. All the rules are valid since the lift ratio is greater than one, indicating that the patterns were statistically significant. The association rules also provided more information regarding the association between each product the quantity, filling time, and finished process that also used as one of the considerations for the filling shed recommendation since the result of the association rules provided information regarding the characteristics of each product.
3. The ANN result shows that it has high accuracy for low and medium efficiency. However, the data that has high efficiency resulted in false predictions. The model resulted in a training accuracy of 90,46% and the validation improved to 92,39%, also the loss previously was 32,33% and the new result is 26,57%. This can indicate that the predicting ability of the ANN is effective.
4. The proposed reconfiguration of the filling sheds was based on balancing the different characteristics of the products (such as quantity, filling time, and finished process) identified through clustering and association rule analysis. The proposed configuration aimed to optimize the workflow by pairing products with complementary characteristics, such as balancing products with high filling times with those that have low filling times. This reconfiguration was designed to enhance overall efficiency by leveraging the strengths and mitigating the weaknesses identified in the clustering and association rules.

6.2 Suggestions

Future studies should prioritize gathering more data on products that have longer filling times and completed processes. This information could help in forecasting a more efficient configuration by taking these extended times into account. Furthermore, it is important to collect more detailed information regarding inspection durations before products enter the filling shed, as bottlenecks during inspections can affect overall efficiency. Additionally, gathering comprehensive data on the tanker trucks dispatched to different gas stations would be beneficial for accurately estimating the time needed for product refills.

The author also suggests carrying out real-world tests to assess whether the proposed configuration successfully addresses bottlenecks. Additionally, the company should consider product order quantities, as a high demand for certain products could worsen bottlenecks. To improve the accuracy of calculations, future research could incorporate discrete event simulation for configuration analysis, utilize optimization models such as linear programming for capacity and demand distribution, and apply queuing theory, especially multi-channel queuing models, to forecast waiting times and optimize the number of filling sheds based on arrival rates and service times.

REFERENCES

AL, R. M., Sembiring, M. T., & Maulana, R. G. R. (2023). *Product Recommendations Using Market Basket Analysis With FP-Growth and Clustering Techniques*. 250–260. https://doi.org/10.46254/au01.20220085

Basodi, S., Ji, C., Zhang, H., & Pan, Y. (2020). Gradient amplification: An efficient way to train deep neural networks. *Big Data Mining and Analytics*, *3*(3), 196–207. https://doi.org/10.26599/BDMA.2020.9020004

Boyko, N., & Zhyhaylo, Y. (2021). Comparison of Algorithms of Associative Rules Search Apriori and Fp-Growth for Investigation of Time Dependence of Their Execution on Database Parameters. *2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT)*, 44–51. https://doi.org/10.1109/CSIT52700.2021.9648732

Et-taleby, A., Boussetta, M., & Benslimane, M. (2020). Faults Detection for Photovoltaic Field Based on K-Means, Elbow, and Average Silhouette Techniques through the Segmentation of a Thermal Image. *International Journal of Photoenergy*, *2020*, 1–7. https://doi.org/10.1155/2020/6617597

Fadilah, A. T. (2023). *IMPLEMENTASI ALGORITMA K-MEANS CLUSTERING UNTUK TARGETING ADS*.

Fansyuri, M. (2023). Analisis Algoritma Neural Network Untuk Identifikasi Jenis Apel Berbasis Ekstraksi Fitur Bentuk Dan Warna. In *Jurnal Ilmu Komputer dan Pendidikan* (Vol. 1, Issue 6). https://journal.mediapublikasi.id/index.php/logic

Galih Pradana, D., Alghifari, M. L., Farhan Juna, M., & Dwisiwi Palaguna, S. (2022). Klasifikasi Penyakit Jantung Menggunakan Metode Artificial Neural Network. *Indonesian Journal of Data and Science (IJODAS)*, *3*(2), 55–60.

Hadi Nasyuha, A., Jama, J., Abdullah, R., Syahra, Y., Azhar, Z., Hutagalung, J., & Hasugian, S. (2021). Frequent pattern growth algorithm for maximizing display items. *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, *19*(2), 390–396. https://doi.org/10.12928/TELKOMNIKA.v19i2.16192

Hikmawati, Dr. F. (2020). *Metodologi Penelitian* (1st ed., Vol. 4). PT. Raja Grafindo Persada.

Hosseinzadeh, M., Ahmed, O. H., Ghafour, M. Y., Safara, F., hama, H. kamaran, Ali, S., Vo, B., & Chiang, H. Sen. (2021). A multiple multilayer perceptron neural network with an

adaptive learning algorithm for thyroid disease diagnosis in the internet of medical things. *Journal of Supercomputing*, *77*(4), 3616–3637. https://doi.org/10.1007/s11227-020-03404-w

Husein, A. M., Setiawan, D., Sumangunsong, A. R. K., Simatupang, A., & Yasmin, S. A. (2022). Combination Grouping Techniques and Association Rules For Marketing Analysis based Customer Segmentation. *SinkrOn*, *7*(3), 1998–2007. https://doi.org/10.33395/sinkron.v7i3.11571

Irianti, A., Rantelinggi, P. H., Barat, U. S., & Zulkarnaim, N. (2022). *IMPLEMENTATION OF BACKPROPAGATION ARTIFICIAL NEURAL NETWORK FOR FOOD PRICE PREDICTION IN MAJENE CENTRAL MARKET Alief Taufik*. https://doi.org/10.20884/1.jutif.2022.3.3.226

Kaoungku, N., Suksut, K., Chanklan, R., Kerdprasop, K., & Kerdprasop, N. (2018). The silhouette width criterion for clustering and association mining to select image features. *International Journal of Machine Learning and Computing*, *8*, 69–73. https://doi.org/10.18178/ijmlc.2018.8.1.665

Mandala, E. P. W., & Putri, D. E. (2023). Data mining technique for grouping products using clustering based on association. *Indonesian Journal of Electrical Engineering and Computer Science*, *31*(2), 835. https://doi.org/10.11591/ijeecs.v31.i2.pp835-844

Mustakim, Khairunnisa, U., Wenda, A., Ilham, A., Laumal, F. E., Daengs GS, A., Putra, D. S., Iswara, I. B. A. I., Fitriatien, S. R., & Rahim, R. (2021). UNSUPERVISED LEARNING AS A DATA SHARING MODEL IN THE FP-GROWTH ALGORITHM IN DETERMINING THE BEST TRANSACTION DATA PATTERN. *Journal of Theoretical and Applied Information Technology*, *15*(11). www.jatit.org

Nainggolan, R., Perangin-angin, R., Simarmata, E., & Tarigan, A. F. (2019). Improved the Performance of the K-Means Cluster Using the Sum of Squared Error (SSE) optimized by using the Elbow Method. *Journal of Physics: Conference Series*, *1361*(1), 012015. https://doi.org/10.1088/1742-6596/1361/1/012015

Nepal, Yamaha, Sahashi, & Yokoe. (2019). Analysis of Building Electricity Use Pattern Using K-Means Clustering Algorithm by Determination of Better Initial Centroids and Number of Clusters. *Energies*, *12*(12), 2451. https://doi.org/10.3390/en12122451

Nurfalah, R., Dwiza Riana, & Anton. (2021). Identifikasi Citra Beras Menggunakan Algoritma Multi-SVM Dan Neural Network Pada Segmentasi K-Means. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *5*(1), 55–62. https://doi.org/10.29207/resti.v5i1.2721

Putri, A. S. (2020, January 7). Pengelolaan Bahan Bakar Minyak (BBM) di Indonesia. *Https://Www.Kompas.Com/Skola/Read/2020/01/07/200000769/Pengelolaan-Bahan-Bakar-Minyak-Bbm-Di-Indonesia?Page=all*.

Putri, C. A. (2020, November 16). *Sampai 2025, RI Bakal Masih Impor BBM 13 Juta KL*. Https://Www.Cnbcindonesia.Com/News/20201116161459-4-202192/Sampai-2025-Ri-Bakal-Masih-Impor-Bbm-13-Juta-Kl.

Rahman, S. I., Ahmed, S., Fariha, T. A., Mohammad, A., Haque, M. N. M., Chellappan, S., & Noor, J. (2024). Unsupervised machine learning approach for tailoring educational content to individual student weaknesses. *High-Confidence Computing*, 100228. https://doi.org/https://doi.org/10.1016/j.hcc.2024.100228

Ramadhan, M. R. (2018). *Prototipe Automatic Volume Gauge Pada Tangki Berbasis Mikrokontroler*. https://elibrary.unikom.ac.id/id/eprint/378/

Safara, F., Souri, A., & Serrizadeh, M. (2020). Improved intrusion detection method for communication networks using association rule mining and artificial neural networks. *IET Communications*, *14*(7), 1192–1197. https://doi.org/10.1049/iet-com.2019.0502

Santoso, M. H. (2021). Application of Association Rule Method Using Apriori Algorithm to Find Sales Patterns Case Study of Indomaret Tanjung Anom. *Brilliance: Research of Artificial Intelligence*, *1*(2), 54–66. https://doi.org/10.47709/brilliance.v1i2.1228

Saputra, D. M., Saputra, D., & Oswari, L. D. (2020). *Effect of Distance Metrics in Determining K-Value in KMeans Clustering Using Elbow and Silhouette Method* (Vol. 172).

Sivasankaran, S. K., Natarajan, P., & Balasubramanian, V. (2020). Identifying Patterns of Pedestrian Crashes in Urban Metropolitan Roads in India using Association Rule Mining. *World Conference on Transport Research – WCTR 2019, Mumbai, 26-30 May 2019*.

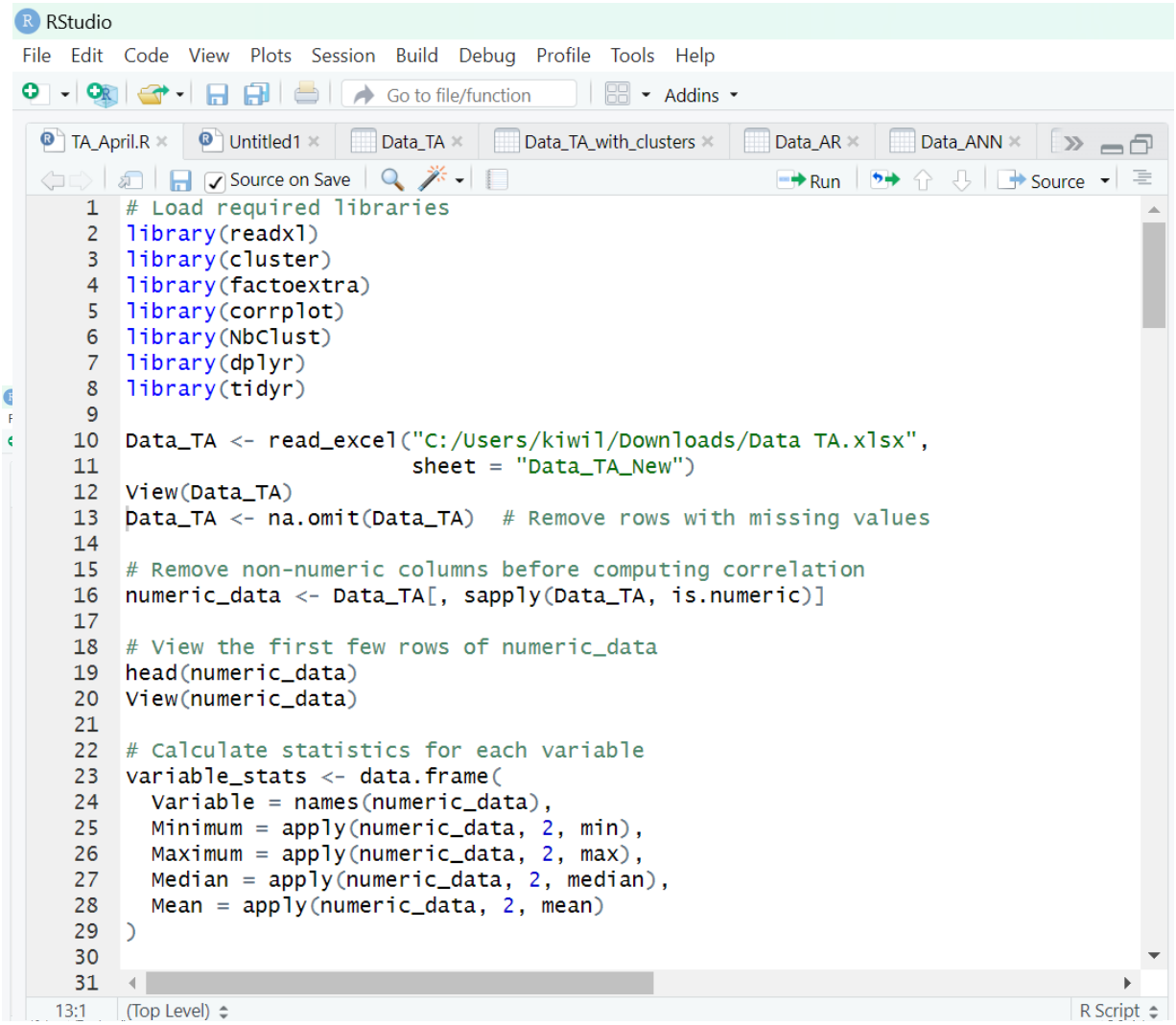
Tambunan, H. B., Barus, D. H., Hartono, J., Alam, A. S., Nugraha, D. A., & Usman, H. H. H. (2020). Electrical peak load clustering analysis using K-Means algorithm and silhouette coefficient. *Proceeding - 2nd International Conference on Technology and Policy in Electric Power and Energy, ICT-PEP 2020*, 258–262. https://doi.org/10.1109/ICT-PEP50916.2020.9249773

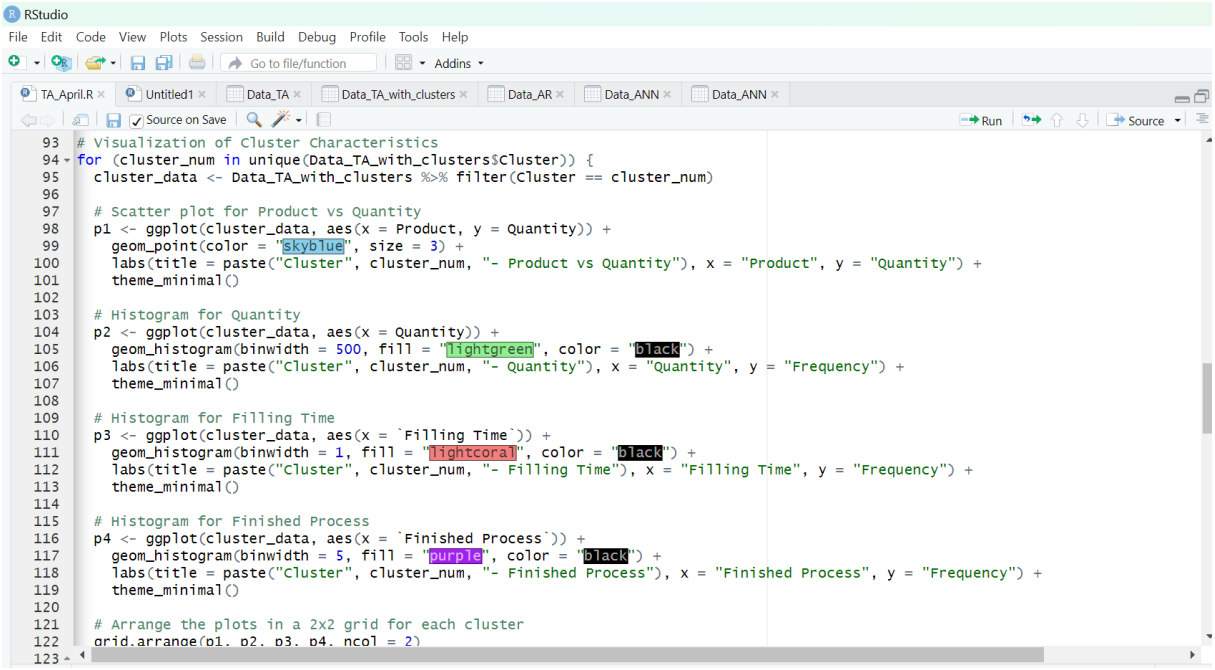
Ula, M., Zulfikri, A., Ulva, A. F., & Rizal, R. A. (2023). Penerapan Machine Learning Clustering K-Means dan Linear Regression Dalam Penentuan Tingkat Resiko Tuberkulosis Paru. *Indonesian Journal of Computer Science*, *12*(1). https://doi.org/10.33022/ijcs.v12i1.3162

Wicaksono, D., Jambak, M. I., & Saputra, D. M. (2020). Advances in Intelligent Systems Research. *Sriwijaya International Conference on Information Technology and Its Applications (SICONIAN 2019)*, *172*.

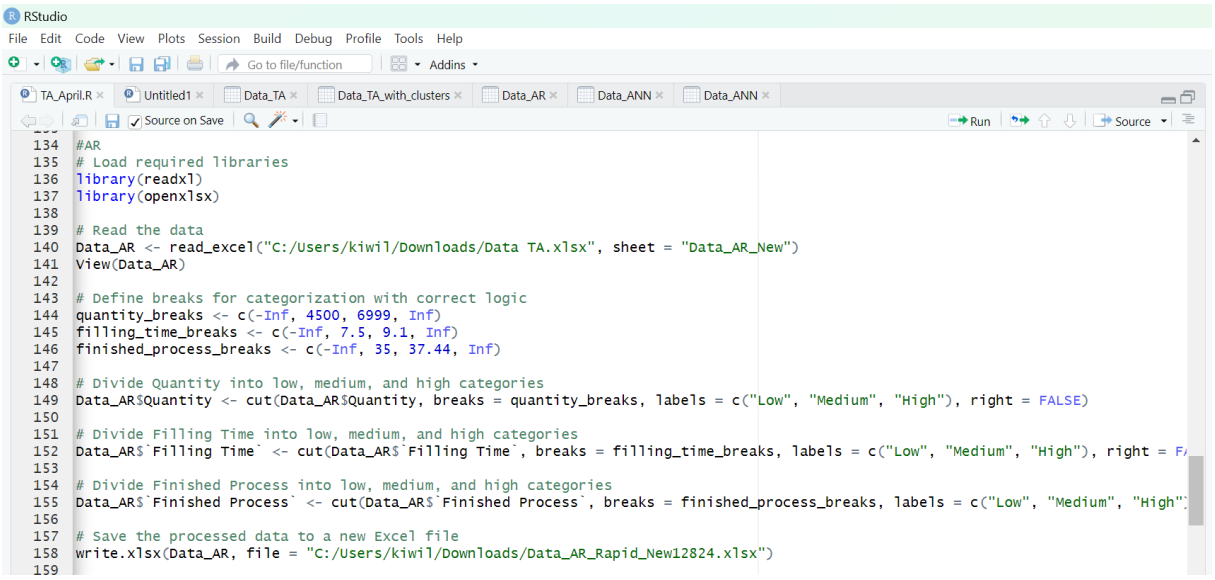
Widhi Aryanti, & Nur Azizah Komara Rifai. (2023). RISK ANALYSIS OF THE IMPACT OF THE COVID 19 PANDEMIC BY HOUSE OF RISK METHOD AGAINST DELAYS IN DELIVERY OF EXPORTED GOODS TO PT. INDONESIAN OCEAN. *Jurnal Riset Statistika*, 107–118. https://doi.org/10.29313/jrs.v3i2.2953

APPENDIX

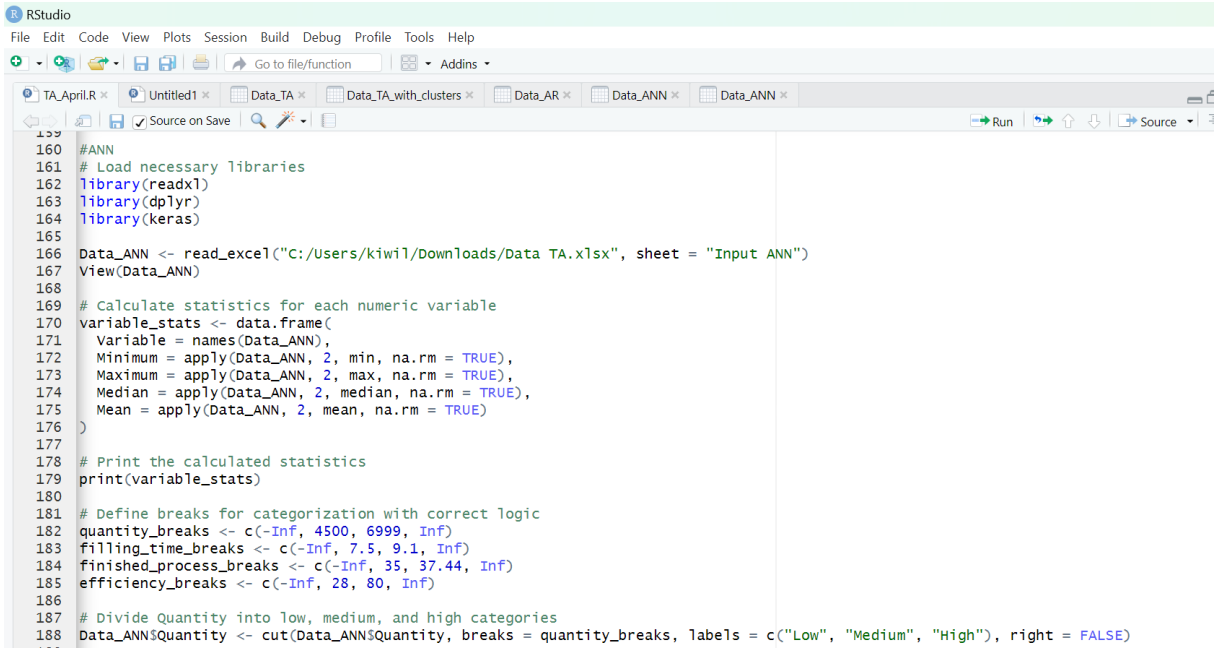

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
TA_April.R
Untitled1
Data_TA
Data_TA_with_clusters
Data_AR
Data_ANN
Source on Save
Run
Source
# Load required libraries
library(readxl)
library(cluster)
library(factoextra)
library(corrplot)
library(NbClust)
library(dplyr)
library(tidyr)
Data_TA <- read_excel("C:/Users/kiwil/Downloads/Data TA.xlsx",
sheet = "Data_TA_New")
View(Data_TA)
Data_TA <- na.omit(Data_TA) # Remove rows with missing values
# Remove non-numeric columns before computing correlation
numeric_data <- Data_TA[, sapply(Data_TA, is.numeric)]
# View the first few rows of numeric_data
head(numeric_data)
View(numeric_data)
# Calculate statistics for each variable
variable_stats <- data.frame(
Variable = names(numeric_data),
Minimum = apply(numeric_data, 2, min),
Maximum = apply(numeric_data, 2, max),
Median = apply(numeric_data, 2, median),
Mean = apply(numeric_data, 2, mean)
)
13:1
(Top Level)
R Script

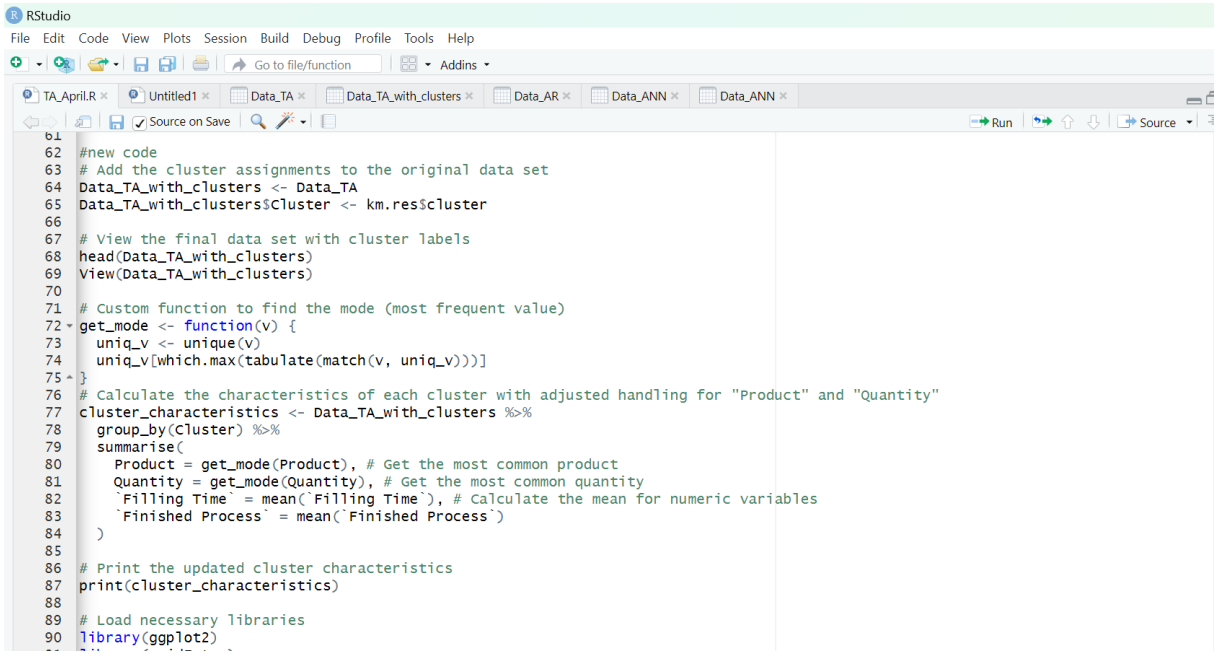


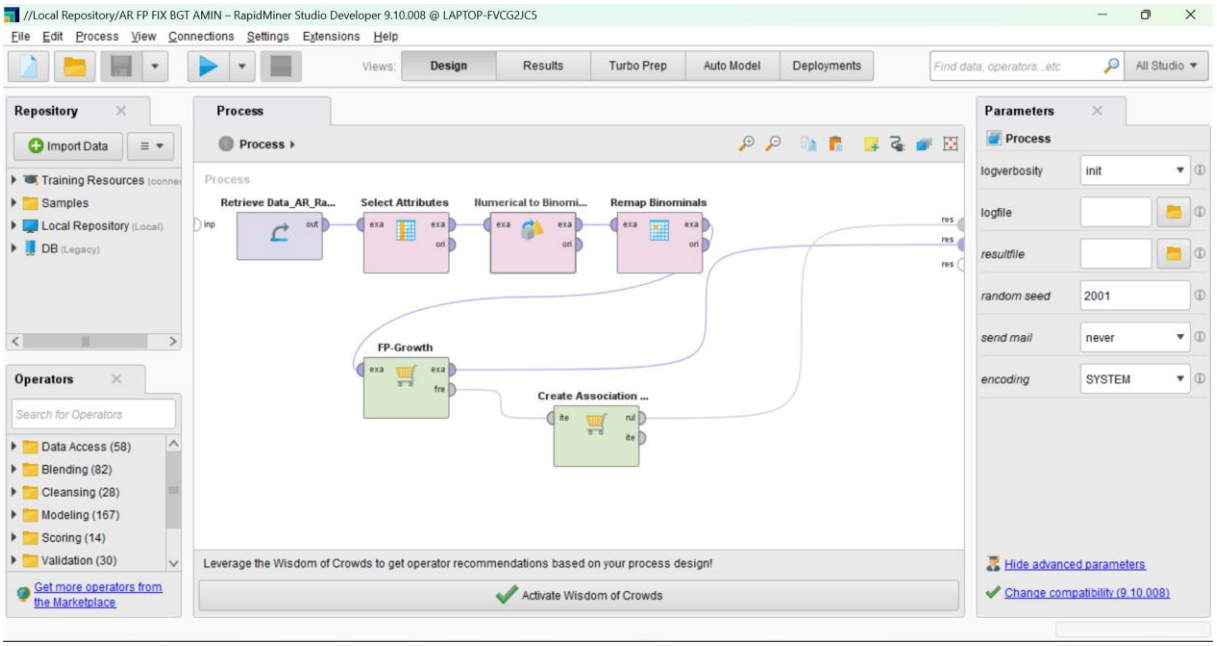
```
# Visualization of Cluster Characteristics
for (cluster_num in unique(Data_TA_with_clusters$Cluster)) {
  cluster_data <- Data_TA_with_clusters %>% filter(Cluster == cluster_num)

  # Scatter plot for Product vs Quantity
  p1 <- ggplot(cluster_data, aes(x = Product, y = Quantity)) +
    geom_point(color = "skyblue", size = 3) +
    labs(title = paste("Cluster", cluster_num, "- Product vs Quantity"), x = "Product", y = "Quantity") +
    theme_minimal()

  # Histogram for Quantity
  p2 <- ggplot(cluster_data, aes(x = Quantity)) +
    geom_histogram(binwidth = 500, fill = "lightgreen", color = "black") +
    labs(title = paste("Cluster", cluster_num, "- Quantity"), x = "Quantity", y = "Frequency") +
    theme_minimal()

  # Histogram for Filling Time
  p3 <- ggplot(cluster_data, aes(x = `Filling Time`)) +
    geom_histogram(binwidth = 1, fill = "lightcoral", color = "black") +
    labs(title = paste("Cluster", cluster_num, "- Filling Time"), x = "Filling Time", y = "Frequency") +
    theme_minimal()

  # Histogram for Finished Process
  p4 <- ggplot(cluster_data, aes(x = `Finished Process`)) +
    geom_histogram(binwidth = 5, fill = "purple", color = "black") +
    labs(title = paste("Cluster", cluster_num, "- Finished Process"), x = "Finished Process", y = "Frequency") +
    theme_minimal()

  # Arrange the plots in a 2x2 grid for each cluster
  grid.arrange(p1, p2, p3, p4, ncol = 2)
```

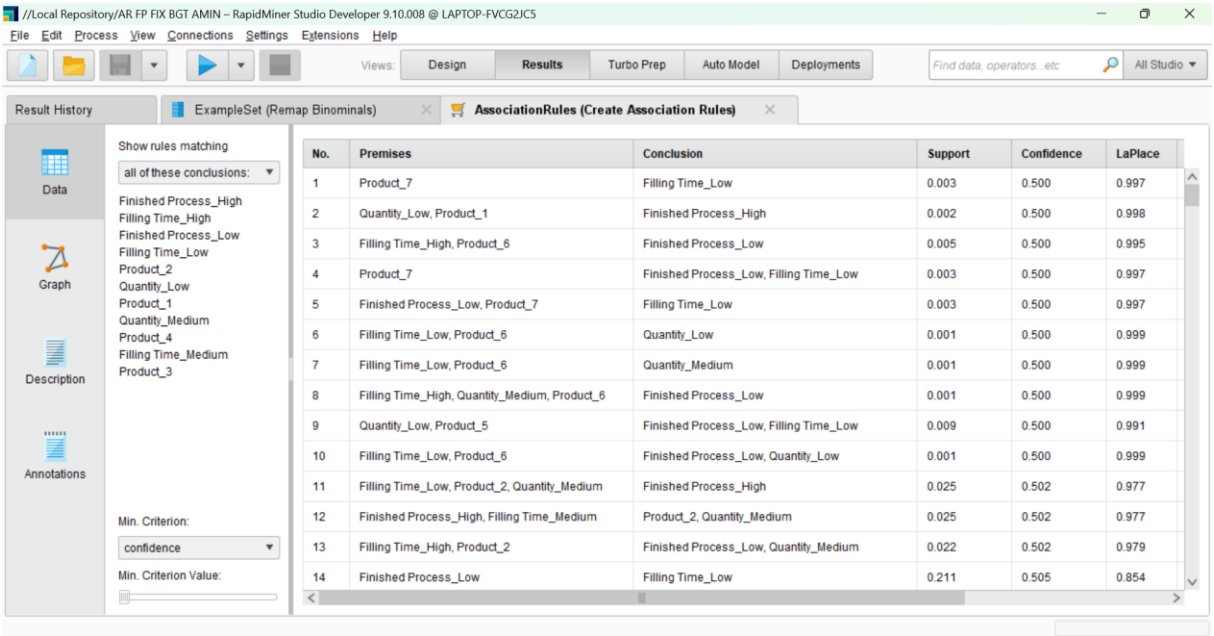


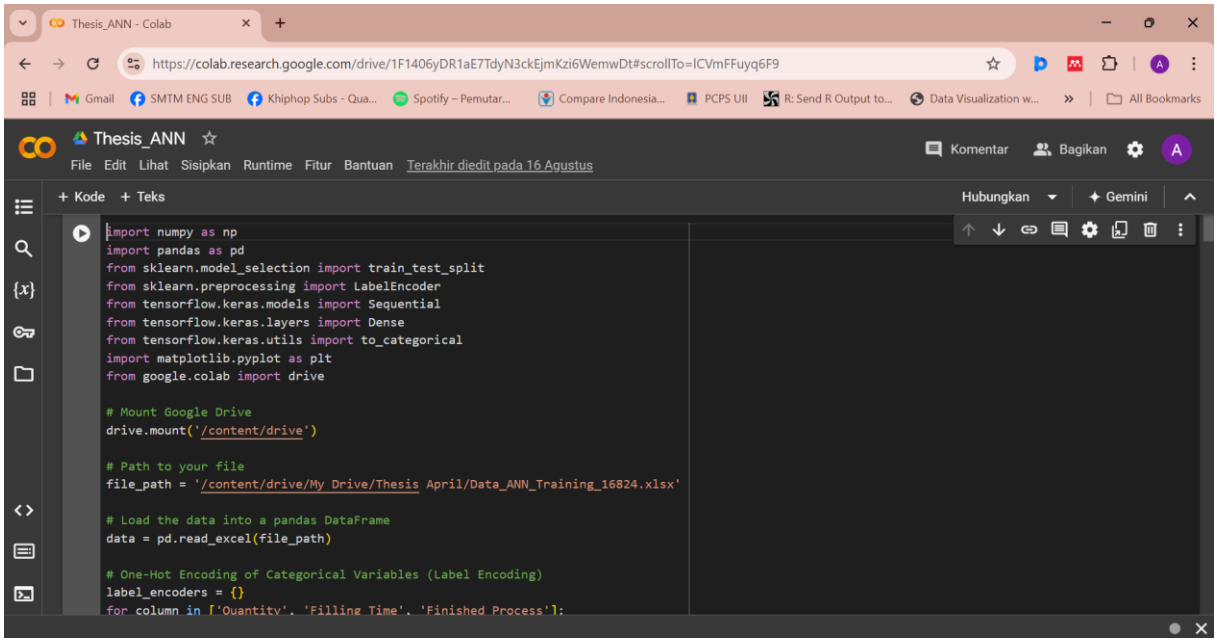
```
#AR
# Load required libraries
library(readxl)
library(openxlsx)

# Read the data
Data_AR <- read_excel("C:/Users/kiwil/Downloads/Data TA.xlsx", sheet = "Data_AR_New")
View(Data_AR)

# Define breaks for categorization with correct logic
quantity_breaks <- c(-Inf, 4500, 6999, Inf)
filling_time_breaks <- c(-Inf, 7.5, 9.1, Inf)
finished_process_breaks <- c(-Inf, 35, 37.44, Inf)

# Divide Quantity into low, medium, and high categories
Data_AR$Quantity <- cut(Data_AR$Quantity, breaks = quantity_breaks, labels = c("Low", "Medium", "High"), right = FALSE)

# Divide Filling Time into low, medium, and high categories
Data_AR$`Filling Time` <- cut(Data_AR$`Filling Time`, breaks = filling_time_breaks, labels = c("Low", "Medium", "High"), right = F

# Divide Finished Process into low, medium, and high categories
Data_AR$`Finished Process` <- cut(Data_AR$`Finished Process`, breaks = finished_process_breaks, labels = c("Low", "Medium", "High"

# Save the processed data to a new Excel file
write.xlsx(Data_AR, file = "C:/Users/kiwil/Downloads/Data_AR_Rapid_New12824.xlsx")
```

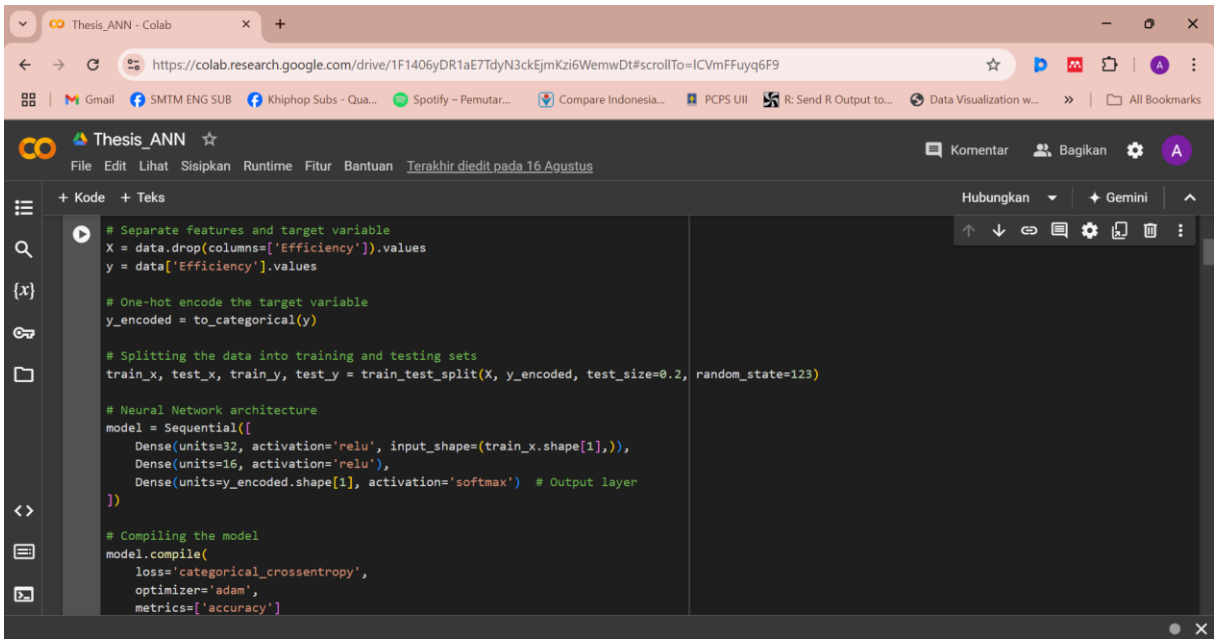

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
TA_April.R
Untitled1
Data_TA
Data_TA_with_clusters
Data_AR
Data_ANN
Data_ANN
Source on Save
Run
Source
#ANN
# Load necessary libraries
library(readxl)
library(dplyr)
library(keras)
Data_ANN <- read_excel("C:/Users/kiwil/Downloads/Data TA.xlsx", sheet = "Input ANN")
View(Data_ANN)
# Calculate statistics for each numeric variable
variable_stats <- data.frame(
Variable = names(Data_ANN),
Minimum = apply(Data_ANN, 2, min, na.rm = TRUE),
Maximum = apply(Data_ANN, 2, max, na.rm = TRUE),
Median = apply(Data_ANN, 2, median, na.rm = TRUE),
Mean = apply(Data_ANN, 2, mean, na.rm = TRUE)
)
# Print the calculated statistics
print(variable_stats)
# Define breaks for categorization with correct logic
quantity_breaks <- c(-Inf, 4500, 6999, Inf)
filling_time_breaks <- c(-Inf, 7.5, 9.1, Inf)
finished_process_breaks <- c(-Inf, 35, 37.44, Inf)
efficiency_breaks <- c(-Inf, 28, 80, Inf)
# Divide Quantity into low, medium, and high categories
Data_ANN$Quantity <- cut(Data_ANN$Quantity, breaks = quantity_breaks, labels = c("Low", "Medium", "High"), right = FALSE)

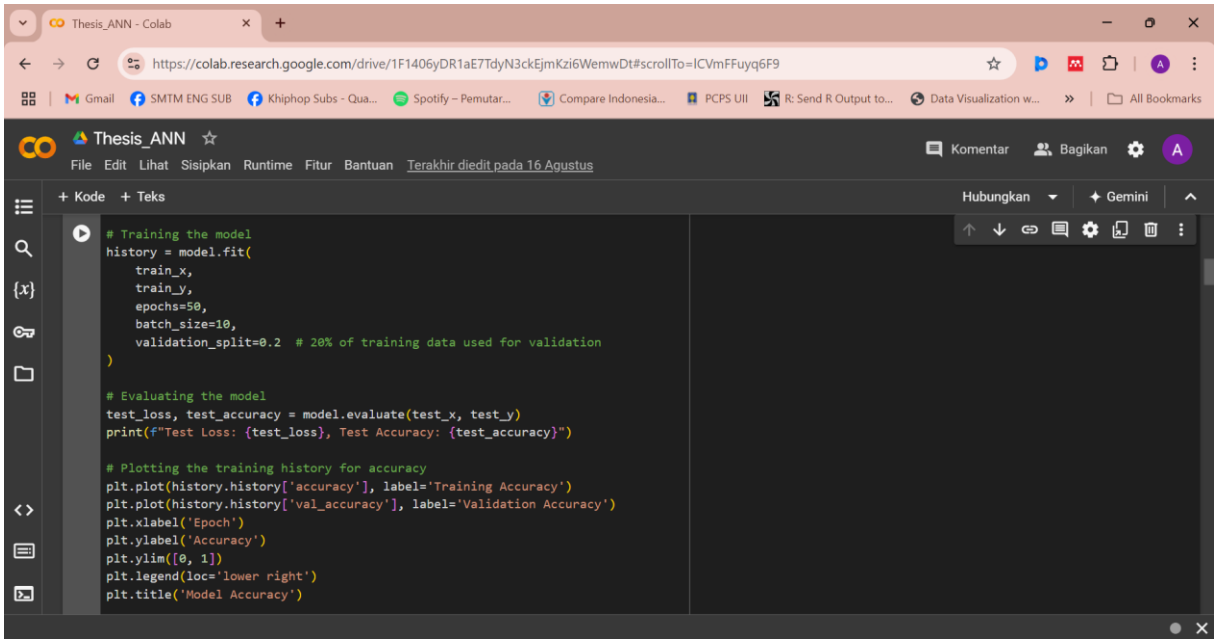

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
TA_April.R
Untitled1
Data_TA
Data_TA_with_clusters
Data_AR
Data_ANN
Data_ANN
Source on Save
Run
Source
#new code
# Add the cluster assignments to the original data set
Data_TA_with_clusters <- Data_TA
Data_TA_with_clusters$Cluster <- km.res$cluster
# View the final data set with cluster labels
head(Data_TA_with_clusters)
View(Data_TA_with_clusters)
# Custom function to find the mode (most frequent value)
get_mode <- function(v) {
uniq_v <- unique(v)
uniq_v[which.max(tabulate(match(v, uniq_v)))]
}
# Calculate the characteristics of each cluster with adjusted handling for "Product" and "Quantity"
cluster_characteristics <- Data_TA_with_clusters %>%
group_by(Cluster) %>%
summarise(
Product = get_mode(Product), # Get the most common product
Quantity = get_mode(Quantity), # Get the most common quantity
`Filling Time` = mean(`Filling Time`), # Calculate the mean for numeric variables
`Finished Process` = mean(`Finished Process`)
)
# Print the updated cluster characteristics
print(cluster_characteristics)
# Load necessary libraries
library(ggplot2)

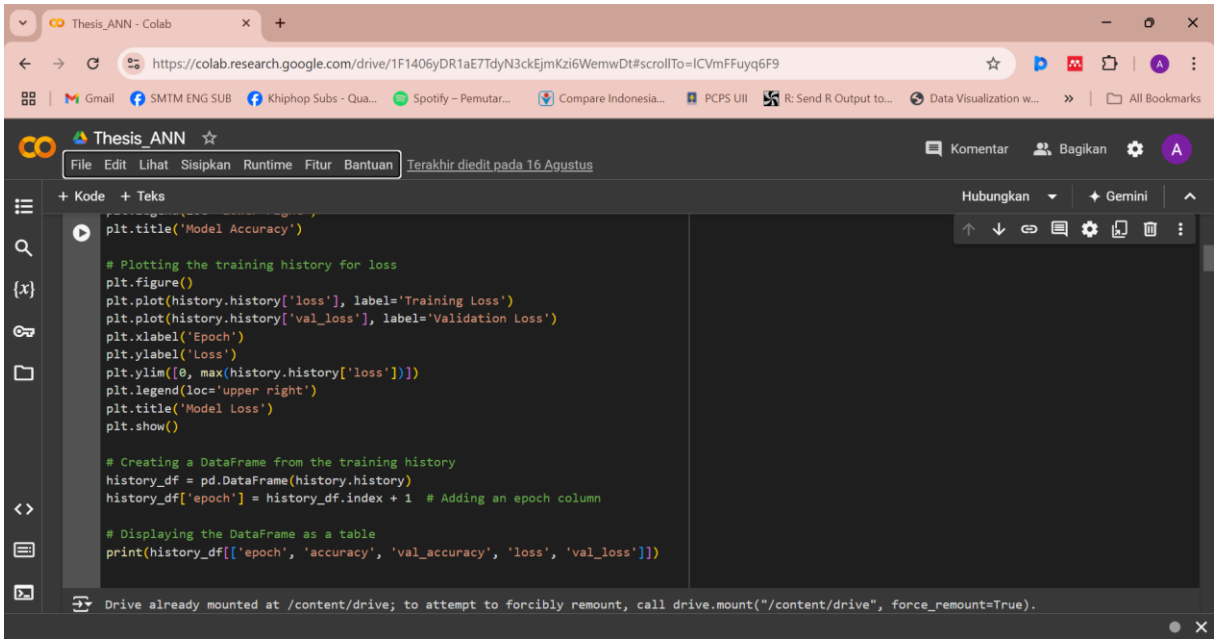

//Local Repository/AR FP FIX BGT AMIN – RapidMiner Studio Developer 9.10.008 @ LAPTOP-FVCG2JC5
File Edit Process View Connections Settings Extensions Help
Views: Design Results Turbo Prep Auto Model Deployments
Find data, operators...etc
All Studio
Repository
Import Data
Training Resources
Samples
Local Repository (Local)
DB (Legacy)
Operators
Search for Operators
Data Access (58)
Blending (82)
Cleansing (28)
Modeling (167)
Scoring (14)
Validation (30)
Get more operators from the Marketplace
Process
Process
Retrieve Data_AR_Ra...
Select Attributes
Numerical to Binomi...
Remap Binominals
FP-Growth
Create Association ...
Leverage the Wisdom of Crowds to get operator recommendations based on your process design!
Activate Wisdom of Crowds
Parameters
Process
logverbosity
init
logfile
resultfile
random seed
2001
send mail
never
encoding
SYSTEM
Hide advanced parameters
Change compatibility (9.10.008)

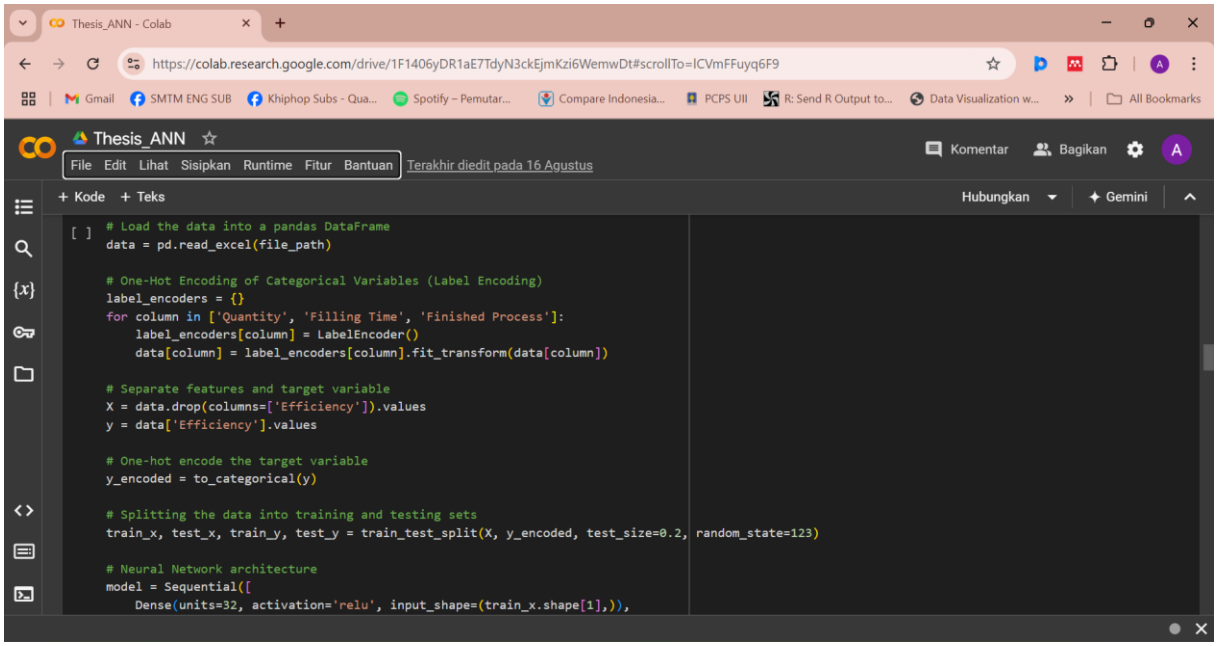

//Local Repository/AR FP FIX BGT AMIN – RapidMiner Studio Developer 9.10.008 @ LAPTOP-FVCG2JC5
File Edit Process View Connections Settings Extensions Help
Views: Design Results Turbo Prep Auto Model Deployments
Find data, operators...etc
All Studio
Result History
ExampleSet (Remap Binominals)
AssociationRules (Create Association Rules)
Data
Graph
Description
Annotations
Show rules matching
all of these conclusions:
Finished Process_High
Filling Time_High
Finished Process_Low
Filling Time_Low
Product_2
Quantity_Low
Product_1
Quantity_Medium
Product_4
Filling Time_Medium
Product_3
Min. Criterion:
confidence
Min. Criterion Value:
No. Premises Conclusion Support Confidence LaPlace
1 Product_7 Filling Time_Low 0.003 0.500 0.997
2 Quantity_Low, Product_1 Finished Process_High 0.002 0.500 0.998
3 Filling Time_High, Product_6 Finished Process_Low 0.005 0.500 0.995
4 Product_7 Finished Process_Low, Filling Time_Low 0.003 0.500 0.997
5 Finished Process_Low, Product_7 Filling Time_Low 0.003 0.500 0.997
6 Filling Time_Low, Product_6 Quantity_Low 0.001 0.500 0.999
7 Filling Time_Low, Product_6 Quantity_Medium 0.001 0.500 0.999
8 Filling Time_High, Quantity_Medium, Product_6 Finished Process_Low 0.001 0.500 0.999
9 Quantity_Low, Product_5 Finished Process_Low, Filling Time_Low 0.009 0.500 0.991
10 Filling Time_Low, Product_6 Finished Process_Low, Quantity_Low 0.001 0.500 0.999
11 Filling Time_Low, Product_2, Quantity_Medium Finished Process_High 0.025 0.502 0.977
12 Finished Process_High, Filling Time_Medium Product_2, Quantity_Medium 0.025 0.502 0.977
13 Filling Time_High, Product_2 Finished Process_Low, Quantity_Medium 0.022 0.502 0.979
14 Finished Process_Low Filling Time_Low 0.211 0.505 0.854

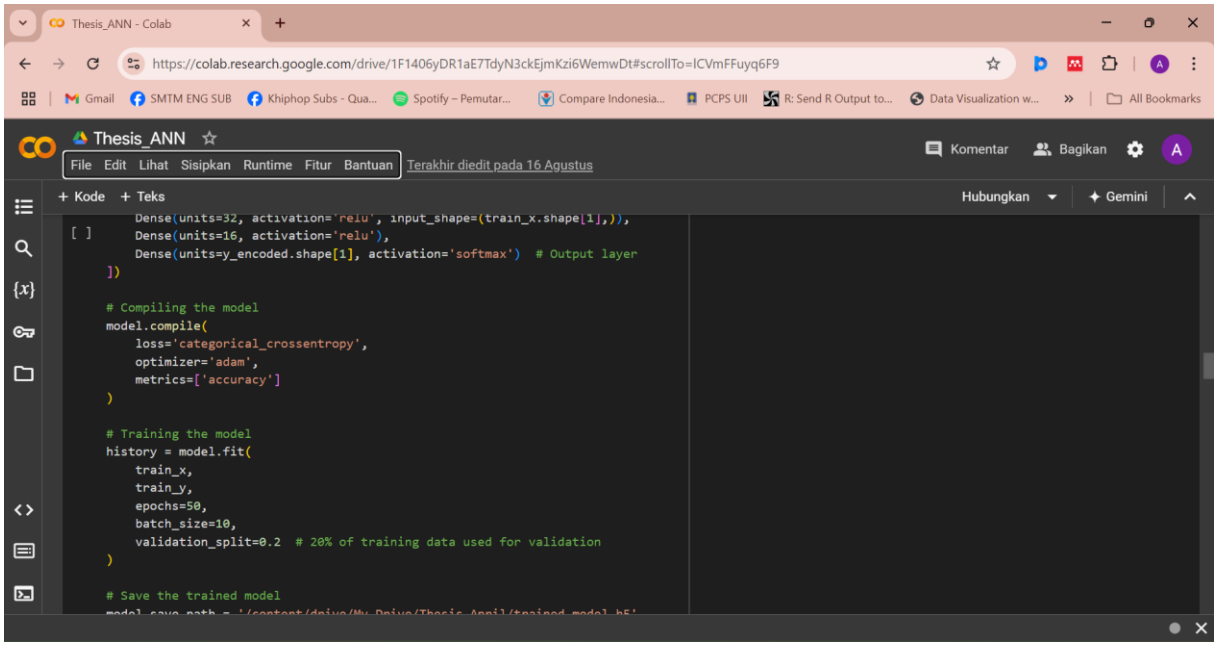

Thesis_ANN - Colab
https://colab.research.google.com/drive/1F1406yDR1aE7TdyN3ckEjmKzi6WemwDt#scrollTo=ICVmFFuyq6F9
Thesis_ANN
File Edit Lihat Sisipkan Runtime Fitur Bantuan Terakhir diedit pada 16 Agustus
Komentar Bagikan
+ Kode + Teks
Hubungkan Gemini
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.utils import to_categorical
import matplotlib.pyplot as plt
from google.colab import drive
# Mount Google Drive
drive.mount('/content/drive')
# Path to your file
file_path = '/content/drive/My Drive/Thesis April/Data_ANN_Training_16824.xlsx'
# Load the data into a pandas DataFrame
data = pd.read_excel(file_path)
# One-Hot Encoding of Categorical Variables (Label Encoding)
label_encoders = {}
for column in ['Quantity', 'Filling Time', 'Finished Process']:

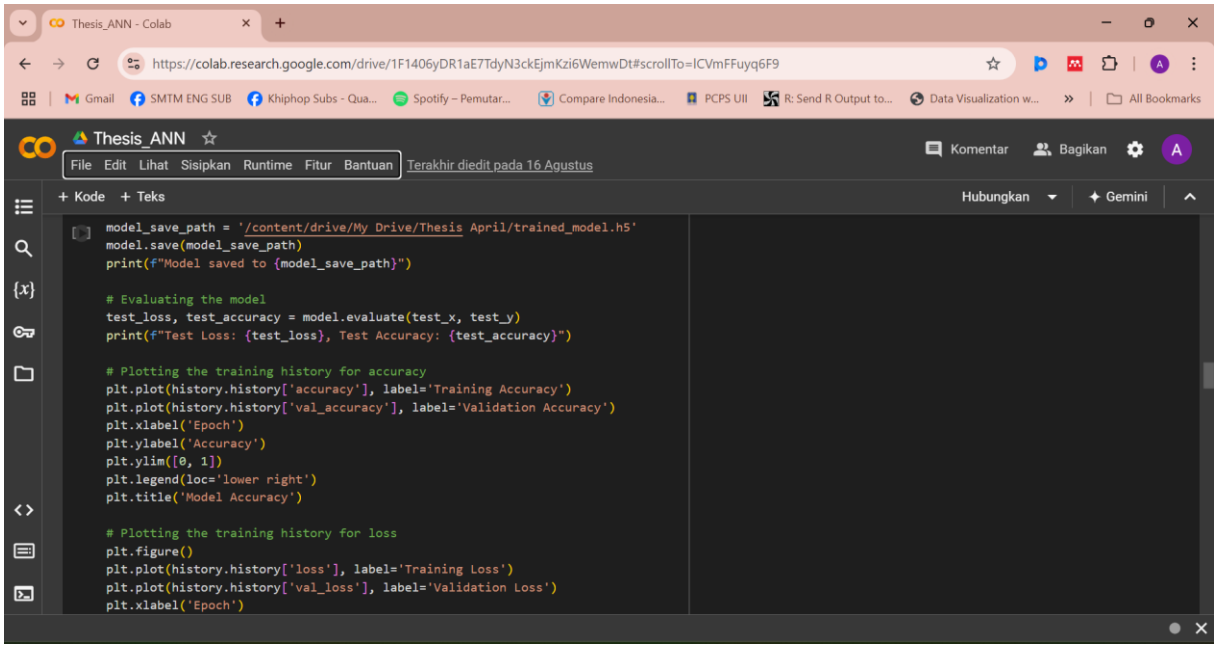

Thesis_ANN - Colab
https://colab.research.google.com/drive/1F1406yDR1aE7TdyN3ckEjmKzi6WemwDt#scrollTo=ICVmFFuyq6F9
Thesis_ANN
File Edit Lihat Sisipkan Runtime Fitur Bantuan Terakhir diedit pada 16 Agustus
Komentar Bagikan
+ Kode + Teks
Hubungkan Gemini
# Separate features and target variable
X = data.drop(columns=['Efficiency']).values
y = data['Efficiency'].values
# One-hot encode the target variable
y_encoded = to_categorical(y)
# Splitting the data into training and testing sets
train_x, test_x, train_y, test_y = train_test_split(X, y_encoded, test_size=0.2, random_state=123)
# Neural Network architecture
model = Sequential([
Dense(units=32, activation='relu', input_shape=(train_x.shape[1],)),
Dense(units=16, activation='relu'),
Dense(units=y_encoded.shape[1], activation='softmax') # Output layer
])
Compiling the model
model.compile(
loss='categorical_crossentropy',
optimizer='adam',
metrics=['accuracy']

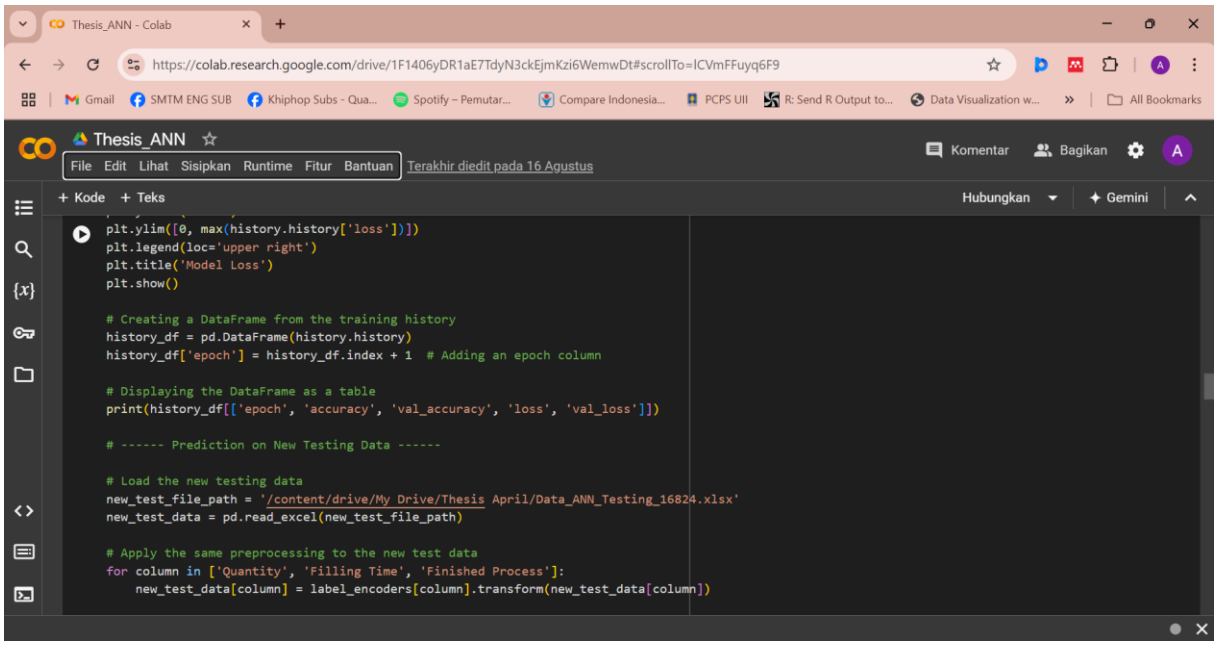

Thesis_ANN - Colab
https://colab.research.google.com/drive/1F1406yDR1aE7TdyN3ckEjmKzi6WemwDt#scrollTo=ICVmFFuyq6F9
Thesis_ANN
File Edit Lihat Sisipkan Runtime Fitur Bantuan Terakhir diedit pada 16 Agustus
Komentar Bagikan
+ Kode + Teks
Hubungkan Gemini
# Training the model
history = model.fit(
train_x,
train_y,
epochs=50,
batch_size=10,
validation_split=0.2 # 20% of training data used for validation
)
Evaluating the model
test_loss, test_accuracy = model.evaluate(test_x, test_y)
print(f"Test Loss: {test_loss}, Test Accuracy: {test_accuracy}")
# Plotting the training history for accuracy
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.ylim([0, 1])
plt.legend(loc='lower right')
plt.title('Model Accuracy')

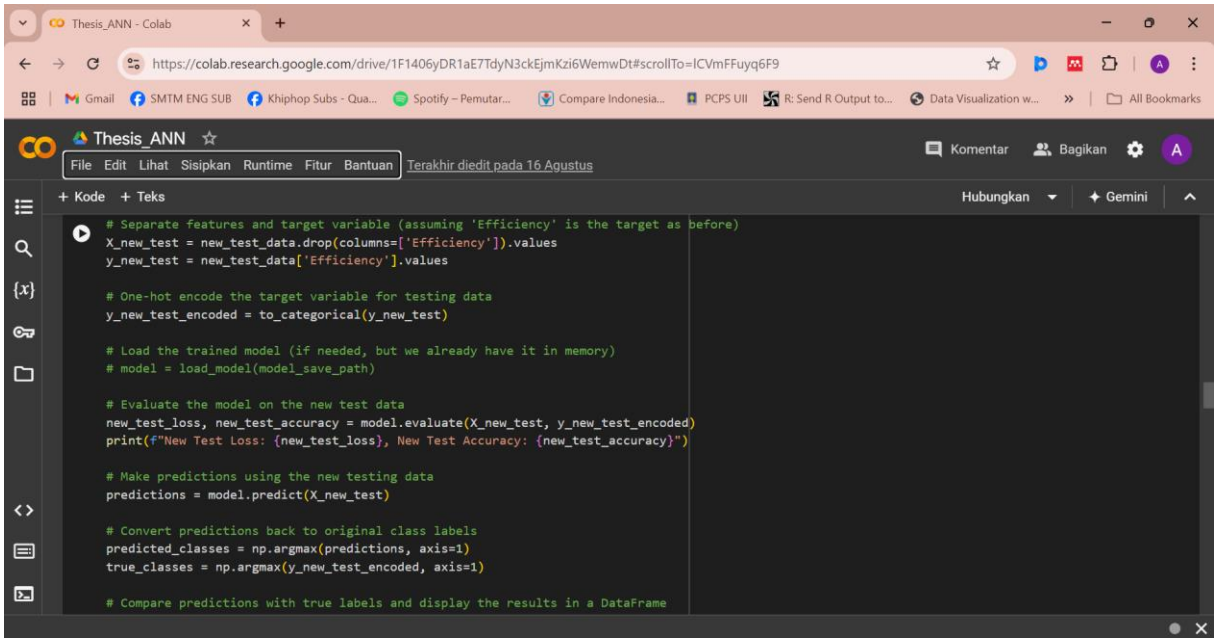

Thesis_ANN - Colab
https://colab.research.google.com/drive/1F1406yDR1aE7TdyN3ckEjmKzi6WemwDt#scrollTo=ICVmFFuyq6F9
Thesis_ANN
File Edit Lihat Sisipkan Runtime Fitur Bantuan Terakhir diedit pada 16 Agustus
Komentar Bagikan
+ Kode + Teks
Hubungkan Gemini
plt.title('Model Accuracy')
# Plotting the training history for loss
plt.figure()
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.ylim([0, max(history.history['loss'])])
plt.legend(loc='upper right')
plt.title('Model Loss')
plt.show()
# Creating a DataFrame from the training history
history_df = pd.DataFrame(history.history)
history_df['epoch'] = history_df.index + 1 # Adding an epoch column
# Displaying the DataFrame as a table
print(history_df[['epoch', 'accuracy', 'val_accuracy', 'loss', 'val_loss']])
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

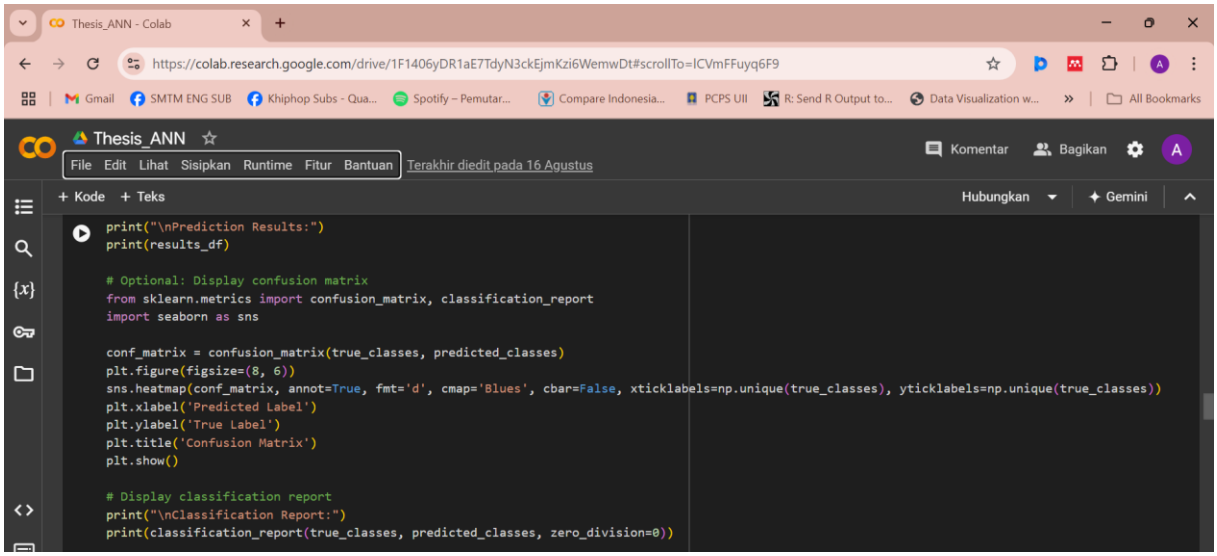

Thesis_ANN - Colab
https://colab.research.google.com/drive/1F1406yDR1aE7TdyN3ckEjmKzi6WemwDt#scrollTo=ICVmFFuyq6F9
Thesis_ANN
File Edit Lihat Sisipkan Runtime Fitur Bantuan
Terakhir diedit pada 16 Agustus
Komentar
Bagikan
+ Kode + Teks
Hubungkan
Gemini
# Load the data into a pandas DataFrame
data = pd.read_excel(file_path)
# One-Hot Encoding of Categorical Variables (Label Encoding)
label_encoders = {}
for column in ['Quantity', 'Filling Time', 'Finished Process']:
label_encoders[column] = LabelEncoder()
data[column] = label_encoders[column].fit_transform(data[column])
# Separate features and target variable
X = data.drop(columns=['Efficiency']).values
y = data['Efficiency'].values
# One-hot encode the target variable
y_encoded = to_categorical(y)
# Splitting the data into training and testing sets
train_x, test_x, train_y, test_y = train_test_split(X, y_encoded, test_size=0.2, random_state=123)
# Neural Network architecture
model = Sequential([
Dense(units=32, activation='relu', input_shape=(train_x.shape[1],)),


Thesis_ANN - Colab
https://colab.research.google.com/drive/1F1406yDR1aE7TdyN3ckEjmKzi6WemwDt#scrollTo=ICVmFFuyq6F9
Thesis_ANN
File Edit Lihat Sisipkan Runtime Fitur Bantuan
Terakhir diedit pada 16 Agustus
Komentar
Bagikan
+ Kode + Teks
Hubungkan
Gemini
Dense(units=32, activation='relu', input_shape=(train_x.shape[1],)),
Dense(units=16, activation='relu'),
Dense(units=y_encoded.shape[1], activation='softmax') # Output layer
])
Compiling the model
model.compile(
loss='categorical_crossentropy',
optimizer='adam',
metrics=['accuracy']
)
# Training the model
history = model.fit(
train_x,
train_y,
epochs=50,
batch_size=10,
validation_split=0.2 # 20% of training data used for validation
)
Save the trained model


Thesis_ANN - Colab
https://colab.research.google.com/drive/1F1406yDR1aE7TdyN3ckEjmKzi6WemwDt#scrollTo=ICVmFFuyq6F9
Thesis_ANN
File Edit Lihat Sisipkan Runtime Fitur Bantuan
Terakhir diedit pada 16 Agustus
Komentar
Bagikan
+ Kode + Teks
Hubungkan
Gemini
model_save_path = '/content/drive/My Drive/Thesis April/trained_model.h5'
model.save(model_save_path)
print(f"Model saved to {model_save_path}")
# Evaluating the model
test_loss, test_accuracy = model.evaluate(test_x, test_y)
print(f"Test Loss: {test_loss}, Test Accuracy: {test_accuracy}")
# Plotting the training history for accuracy
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.ylim([0, 1])
plt.legend(loc='lower right')
plt.title('Model Accuracy')
# Plotting the training history for loss
plt.figure()
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.xlabel('Epoch')


Thesis_ANN - Colab
https://colab.research.google.com/drive/1F1406yDR1aE7TdyN3ckEjmKzi6WemwDt#scrollTo=ICVmFFuyq6F9
Thesis_ANN
File Edit Lihat Sisipkan Runtime Fitur Bantuan
Terakhir diedit pada 16 Agustus
Komentar
Bagikan
+ Kode + Teks
Hubungkan
Gemini
plt.ylim([0, max(history.history['loss'])])
plt.legend(loc='upper right')
plt.title('Model Loss')
plt.show()
# Creating a DataFrame from the training history
history_df = pd.DataFrame(history.history)
history_df['epoch'] = history_df.index + 1 # Adding an epoch column
# Displaying the DataFrame as a table
print(history_df[['epoch', 'accuracy', 'val_accuracy', 'loss', 'val_loss']])
# ------ Prediction on New Testing Data ------
# Load the new testing data
new_test_file_path = '/content/drive/My Drive/Thesis April/Data_ANN_Testing_16824.xlsx'
new_test_data = pd.read_excel(new_test_file_path)
# Apply the same preprocessing to the new test data
for column in ['Quantity', 'Filling Time', 'Finished Process']:
    new_test_data[column] = label_encoders[column].transform(new_test_data[column])


Thesis_ANN - Colab
https://colab.research.google.com/drive/1F1406yDR1aE7TdyN3ckEjmKzi6WemwDt#scrollTo=ICVmFFuyq6F9
Thesis_ANN
File Edit Lihat Sisipkan Runtime Fitur Bantuan
Terakhir diedit pada 16 Agustus
Komentar
Bagikan
+ Kode + Teks
Hubungkan
Gemini
# Separate features and target variable (assuming 'Efficiency' is the target as before)
X_new_test = new_test_data.drop(columns=['Efficiency']).values
y_new_test = new_test_data['Efficiency'].values
# One-hot encode the target variable for testing data
y_new_test_encoded = to_categorical(y_new_test)
# Load the trained model (if needed, but we already have it in memory)
# model = load_model(model_save_path)
# Evaluate the model on the new test data
new_test_loss, new_test_accuracy = model.evaluate(X_new_test, y_new_test_encoded)
print(f"New Test Loss: {new_test_loss}, New Test Accuracy: {new_test_accuracy}")
# Make predictions using the new testing data
predictions = model.predict(X_new_test)
# Convert predictions back to original class labels
predicted_classes = np.argmax(predictions, axis=1)
true_classes = np.argmax(y_new_test_encoded, axis=1)
# Compare predictions with true labels and display the results in a DataFrame


Thesis_ANN - Colab
https://colab.research.google.com/drive/1F1406yDR1aE7TdyN3ckEjmKzi6WemwDt#scrollTo=ICVmFFuyq6F9
Thesis_ANN
File Edit Lihat Sisipkan Runtime Fitur Bantuan
Terakhir diedit pada 16 Agustus
Komentar
Bagikan
+ Kode + Teks
Hubungkan
Gemini
print("\nPrediction Results:")
print(results_df)
# Optional: Display confusion matrix
from sklearn.metrics import confusion_matrix, classification_report
import seaborn as sns
conf_matrix = confusion_matrix(true_classes, predicted_classes)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', cbar=False, xticklabels=np.unique(true_classes), yticklabels=np.unique(true_classes))
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix')
plt.show()
# Display classification report
print("\nClassification Report:")
print(classification_report(true_classes, predicted_classes, zero_division=0))