Jurnal Kecerdasan Buatan dan Teknologi Informasi
Vol. 5, No. 3, September 2026, pp. 535–549
e-ISSN: 2964-2922, p-ISSN: 2963-6191

Open access

Original Research

Application of K-Means Clustering Algorithm in Edam Burger Sales Information System for Inventory Control Optimization

Muhammad Rofiq Ubaidillah [a], R Wisnu Prio Pamungkas [b*], Prio Kustanto [c]

[a,b,c] Informatics Study Program, Faculty of Computer Science, Universitas Bhayangkara Jakarta Raya, Bekasi, Indonesia

*Corresponding Author: wisnu.prio@dsn.ubharajaya.ac.id

ABSTRACT

Manual sales and inventory management in small culinary enterprises often leads to data inaccuracies, stock mismanagement, and underutilized transactional data. This study aims to design a web-based sales information system integrated with the K-Means clustering algorithm to optimize inventory control at Edam Burger & Frozen Foods. Utilizing the Waterfall methodology, the system was developed using the Laravel framework and MySQL. The analytical engine processed five months of transactional data across fifteen products, applying Min-Max Normalization to equalize the scales of sales volume, revenue, and transaction frequency. The K-Means algorithm successfully segmented the product catalog into three distinct categories based on performance: one high-selling core product (6.7%), three medium-selling secondary items (20%), and eleven low-selling complementary products (73.3%). Black Box Testing confirmed a 100% functional success rate across all system modules. The primary novelty of this research lies in seamlessly embedding the K-Means engine directly into the operational dashboard, overcoming the common barrier of offline, standalone data mining. This integration enables real-time, data-driven procurement strategies, providing actionable recommendations: prioritizing continuous stock availability for high-demand items, scheduling regular restocking for medium items, and minimizing capital tied up in low-moving inventory to reduce food waste. Ultimately, this integrated approach empowers small business owners to transition from intuition-based management to systematic, algorithm-driven inventory optimization. This study successfully bridges the gap between routine transactions and strategic analytics.

How to Cite:
M. R. Ubaidillah, R. W. P. Pamungkas, and P. Kustanto, "Application of K-Means Clustering Algorithm in Edam Burger Sales Information System for Inventory Control Optimization," *J. Kecerdasan Buatan dan Teknol. Inf.*, vol. 5, no. 3, pp. 535–549, 2026.
Doi: https://doi.org/10.69916/jkbti.v5i3.565

KEYWORDS

K-Means Clustering
Inventory Optimization
Sales Information System
Data Mining
Web Application

ARTICLE INFO

Received: June 30, 2026
Accepted: August 17, 2026
Available Online: September 1, 2026

1. INTRODUCTION

Sales and inventory management are core business activities directly related to revenue generation and operational continuity, particularly for Small and Medium Enterprises (SMEs) in the food and beverage sector. In this industry, fluctuating consumer demand and product characteristics require precise stock monitoring to prevent stockouts and overstock conditions, which can severely impact profitability and customer satisfaction [1]. Edam Burger & Frozen Foods Bekasi Utara is a small culinary business that currently manages its sales and inventory processes manually. This conventional approach has led to several critical issues, including recording errors, data inconsistencies, difficulties in real-time stock monitoring, and delays in report preparation [2]. Consequently, the business owner struggles to make data-driven decisions, and the accumulated historical sales data remains

underutilized for strategic planning, such as inventory optimization.

As information technology advances, web-based sales applications have emerged as an ideal solution for improving the efficiency of transaction management, product stock tracking, and sales reporting [2, 3]. Previous research has demonstrated that transitioning from manual to web-based systems significantly enhances transaction effectiveness and product information delivery [2, 4]. However, a purely transactional system is insufficient if the resulting sales data is not further analyzed to extract actionable business insights. To bridge this gap, data mining techniques, specifically the K-Means clustering algorithm, can be applied to group products based on underlying sales patterns. K-Means is highly effective in identifying natural groupings within datasets by categorizing products according to variables such as sales volume, total revenue, and transaction frequency [5,6]. Numerous studies have explored the efficacy of the K-Means algorithm in extracting actionable insights from sales, retail, and transactional data. Foundational reviews by Ikotun et al. [7] and Ahmed et al. [8] highlight the algorithm's broad applicability and computational efficiency in the era of big data, emphasizing its robustness in handling complex datasets while acknowledging inherent challenges such as random centroid initialization. Building on these theoretical foundations, recent empirical research has successfully deployed K-Means to optimize retail operations and inventory management. For instance, Manarung et al. [9] applied the K-Means algorithm to cluster retail product sales data, effectively categorizing items into high- and low-demand groups to optimize stock availability and inform strategic inventory decisions. Similarly, Nugroho et al. [10] utilized K-Means to segment sales transaction data in traditional markets, demonstrating the algorithm's capacity to uncover distinct purchasing patterns and identify best-selling products for targeted distribution strategies. In the specific context of food and retail inventory, Amalina et al. [11] successfully grouped frozen food products into distinct sales clusters, while Sunia et al. [12] utilized K-Means to determine optimal inventory levels in a retail mart, establishing clear categories as a foundation for stock control. Despite these successful applications in retail analytics, a significant gap remains: most existing implementations treat data mining as an isolated, offline analytical process requiring manual data extraction. The seamless integration of K-Means within a single, unified, and real-time operational web application specifically tailored for food-sector SMEs remains uncommon, presenting the core novelty of this research.

Despite the extensive literature on web-based sales systems and K-Means clustering independently, a significant research gap remains. Most existing studies treat data mining as an isolated, offline analytical process rather than an integrated component of daily operations. The application of K-Means within a single, unified system that combines operational transaction recording with automated clustering specifically tailored for food-sector SMEs remains uncommon. The novelty of this research lies in the seamless integration of a Laravel-based operational sales application with an embedded K-Means clustering engine. Unlike previous systems that require data extraction for external analysis, this proposed system processes multi-variable sales data (quantity, revenue, and frequency) directly within the application dashboard to generate real-time, actionable inventory control recommendations (High, Medium, and Low clusters) [13, 14]. This integration effectively bridges the gap between routine transaction processing and strategic inventory management for small culinary businesses. Based on the aforementioned issues and research gap, this study aims to: (1) design and develop a web-based sales application for Edam Burger capable of managing transaction data effectively; (2) apply the K-Means algorithm for multi-variable product sales segmentation; and (3) optimize product inventory control based on the generated clustering results. Through this integrated approach, the study contributes a practical, ready-to-use technological solution that empowers SME owners to transition from reactive stock management to proactive, data-driven inventory control.

2. RESEARCH METHODS

2.1. System Development Method

The research methodology is designed to provide a structured approach to solving the inventory and sales management issues at Edam Burger. The framework encompasses several phases: (1) problem identification and requirements gathering, (2) data collection and preprocessing, (3) system development using the Waterfall model [15,16], (4) implementation of the K-Means clustering algorithm for product segmentation, and (5) system testing and evaluation. This structured flow ensures that the developed system not only fulfills operational transaction needs but also provides analytical value for strategic inventory control.

2.2. K-Means Algorithm and Waterfall SDLC

The web-based sales application was developed using the Waterfall methodology [17]. This linear and sequential approach was selected because the system requirements for Edam Burger were clearly defined and stable from the outset, allowing for systematic progression through distinct phases. The development comprises six stages:

1. Requirements Analysis: Gathering system needs through observation and interviews.
2. System Design: Creating UML diagrams and database schemas.
3. Development (Coding): Implementing the design using PHP and the Laravel framework.
4. Testing: Validating system functionality using Black Box Testing.
5. Deployment: Installing the system in the local server environment for operational use.
6. Maintenance: Providing ongoing support and minor updates. As demonstrated by Kustanto et al. [18] in developing a web-based infrastructure reporting system, the sequential nature of Waterfall is highly effective for systems with well-defined functional requirements, minimizing the risk of scope creep during development.

2.3. Data Collection and Dataset

To ensure the system addresses actual business needs, data collection was conducted using three complementary techniques:

1. Direct Observation: Monitoring the daily sales and inventory processes at Edam Burger to identify operational bottlenecks.
2. Interviews and Questionnaires: Conducting structured interviews with the business owner, Mr. Ngadiman, utilizing a Likert-scale questionnaire to quantify system requirements. The analysis yielded a system necessity score of 94.28%, categorized as "Highly Needed," validating the urgency of the proposed digital solution.
3. Literature Review: Analyzing relevant scientific journals to establish the theoretical foundation for the K-Means algorithm and web-based system architecture.

The primary dataset for the clustering process consists of historical sales transaction records from Edam Burger & Frozen Foods Bekasi Utara, collected over a five-month period from February to June 2026. The raw transaction logs were aggregated at the product level to form three distinct attributes: total quantity sold, total revenue generated, and transaction frequency. The complete recapitulation of the dataset, comprising fifteen core product types, is presented in Table 1.

Table 1. Summary of Sales Data for All Products (February–June 2026)

No	Product	Qty	Revenue (Rp)	Freq.
1	Daging Patty Edam	43	860.000	10
2	Daging Patty Hemato	137	1.370.000	24
3	Daging Patty Yona	12	468.000	8
4	Kertas Logo Edam Isi 100	21	462.000	12
5	Kertas Logo Edam Isi 50	11	121.000	7
6	Paket Burger Edam	17	756.500	9
7	Paket Burger Hemato	16	552.000	10
8	Roti Burger	5.913	8.573.850	133
9	Roti Hotdog	886	1.284.700	34
10	Saos Sambal Edam	11	181.500	6
11	Saos Special Edam	81	1.336.500	24
12	Saos Tomat Edam	8	132.000	6
13	Thousand Island	11	181.500	5
14	Thousand Island & SPC Cup	10	110.000	6
15	Tortila Besar	8	136.000	6

2.4. K-Means Clustering Algorithm Framework

K-Means is a widely used unsupervised machine learning algorithm designed to partition a dataset into k distinct, non-overlapping clusters based on feature similarity [5]. The algorithm aims to minimize the within-cluster variance, ensuring that data points within the same cluster are as similar as possible. In this study, the K-Means algorithm is executed through the following iterative steps:

1. Initialization: Determine the number of clusters (k). In this research, $k = 3$ is chosen to categorize products into High, Medium, and Low sales performance. Initial centroids are assigned to representative data points.
2. Distance Calculation: Calculate the distance between each data point and all centroids.
3. Cluster Assignment: Assign each data point to the cluster whose centroid is closest.
4. Centroid Update: Recalculate the position of each cluster's centroid by computing the mean of all data points assigned to that cluster.
5. Iteration: Repeat steps 2 through 4 until convergence is achieved, which occurs when no data points change clusters or the centroids' positions no longer shift significantly.

2.5. Data Preprocessing: Min-Max Normalization

A critical challenge in clustering multi-dimensional data is the disparity in variable scales. In this dataset, Revenue values range in the millions of Rupiah, Quantity in thousands, and Frequency in hundreds. If applied directly, the distance calculation would be heavily biased toward the Revenue variable, rendering Quantity and Frequency insignificant. To resolve this, Min-Max Normalization is applied to transform all variables into a uniform scale ranging from 0 to 1 [19, 20]. The normalization formula is defined as:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

Where:

- x' = Normalized value
- x = Original value
- $x_{\min}$ = Minimum value in the variable's dataset
- $x_{\max}$ = Maximum value in the variable's dataset

2.6. Euclidean Distance Metric

To determine the similarity between a product and the cluster centroids, this study utilizes the Euclidean Distance metric. It measures the straight-line distance between two points in a multi-dimensional space [21]. The formula is expressed as:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2}$$

Where $d(x, y)$ is the distance between data point x and centroid y, and x_i and y_i are the normalized values of the variables. The product is then assigned to the cluster with the minimum Euclidean distance.

2.7. Clustering Variables and Categories

The clustering process relies on three key business metrics that reflect product performance. These variables are detailed in Table 2.

Based on the business requirements for inventory control, the resulting clusters are mapped to three actionable categories, as shown in Table 3.

2.8. Development Environment

The implementation of the proposed system and the execution of the clustering algorithm were supported by specific hardware and software environments, detailed in Tables 4 and 5.

Table 2. K-Means Algorithm Variables

Variable	Description	Business Relevance
Quantity (Qty)	Total number of products sold	Indicates physical stock turnover rate.
Revenue	Total revenue generated (Rp)	Reflects the financial contribution of the product.
Transaction Frequency	Total number of sales transactions	Shows product popularity and customer purchase consistency.

Table 3. Cluster Categories and Inventory Actions

Cluster	Category	Inventory Control Recommendation
C1	High Sales	Prioritize continuous stock availability; avoid stockouts.
C2	Medium Sales	Maintain regular restocking based on standard demand.
C3	Low Sales	Order in smaller quantities; evaluate pricing or promotional strategies.

Table 4. Hardware Specifications

No	Component	Specification
1	Processor	AMD Ryzen 5 5600 6-Core 3.50 GHz
2	RAM	16.0 GB
3	System Type	64-bit, x64-based Processor
4	OS	Windows 11

Table 5. Software Specifications

No	Tools	Category	Function
1	PHP	Programming Language	Primary server-side scripting language.
2	Laravel	Framework	PHP framework utilizing the MVC architecture.
3	VS Code	Code Editor	Integrated Development Environment (IDE).
4	MySQL	Database	Relational database management system (RDBMS).
5	XAMPP	Local Server	Localhost environment for Apache and MySQL.
6	Visual Paradigm	Design Tool	Software for UML diagram modeling.

3. RESULTS AND DISCUSSION

3.1. Data Normalization

The K-Means clustering process was applied directly to the complete dataset of fifteen products in Table 1, using quantity sold, revenue, and transaction frequency as clustering attributes. Min-Max Normalization was first applied to all fifteen products to equalize the scale of the three attributes, using Equation (1). For this dataset, the minimum and maximum values of each attribute are: $\text{Qty}_{\text{min}} = 8$, $\text{Qty}_{\text{max}} = 5.913$; $\text{Revenue}_{\text{min}} =$ Rp 110.000, $\text{Revenue}_{\text{max}} =$ Rp 8.573.850; $\text{Freq}_{\text{min}} = 5$, $\text{Freq}_{\text{max}} = 133$.
As an example, the calculation for the product Daging Patty Edam (Qty = 43, Revenue = Rp 860.000, Freq = 10) is shown below:

$$\begin{aligned} \text{Qty}' &= (43 - 8)/(5.913 - 8) = 35/5.905 = 0.006 \\ \text{Revenue}' &= (860.000 - 110.000)/(8.573.850 - 110.000) = 750.000/8.463.850 = 0.089 \\ \text{Freq}' &= (10 - 5)/(133 - 5) = 5/128 = 0.039 \end{aligned}$$

The same calculation procedure is applied to each of the remaining fourteen products. The complete normalization result for all fifteen products is presented in Table 6.

Table 6. Results of Data Normalization for All Products

No	Product	Qty (norm)	Revenue (norm)	Freq. (norm)
1	Daging Patty Edam	0.006	0.089	0.039
2	Daging Patty Hemato	0.022	0.149	0.148
3	Daging Patty Yona	0.001	0.042	0.023
4	Kertas Logo Edam Isi 100	0.002	0.042	0.055
5	Kertas Logo Edam Isi 50	0.001	0.001	0.016
6	Paket Burger Edam	0.002	0.076	0.031
7	Paket Burger Hemato	0.001	0.052	0.039
8	Roti Burger	1.000	1.000	1.000
9	Roti Hotdog	0.149	0.139	0.227
10	Saos Sambal Edam	0.001	0.008	0.008
11	Saos Special Edam	0.012	0.145	0.148
12	Saos Tomat Edam	0.000	0.003	0.008
13	Thousand Island	0.001	0.008	0.000
14	Thousand Island & SPC Cup	0.000	0.000	0.008
15	Tortila Besar	0.000	0.003	0.008

Table 6 presents the comprehensive results of the Min-Max normalization applied to all fifteen products across the three clustering variables: Quantity, Revenue, and Transaction Frequency. By transforming the raw data, which originally spanned vastly different scales, such as millions of Rupiah for revenue and thousands of units for quantity, into a uniform scale ranging from 0.000 to 1.000, the normalization process ensures that no single variable disproportionately influences the subsequent distance calculations. This step is critical for the K-Means algorithm, as it prevents variables with larger absolute magnitudes (such as Revenue) from dominating the Euclidean distance metric, thereby allowing Quantity and Frequency to contribute equally to the clustering process.
An analysis of the normalized values reveals a highly skewed distribution, which is heavily influenced by the dominant market position of "Roti Burger" (Product No. 8). As the top-selling item by a significant margin, Roti Burger achieved the maximum normalized score of 1.000 across all three attributes, establishing it as the absolute upper bound of the dataset. Consequently, the remaining products exhibit normalized values that are relatively compressed toward the lower end of the scale. For instance, secondary core products such as Roti Hotdog (No. 9) and Daging Patty Hemato (No. 2) display moderate normalized values (ranging from 0.139 to 0.227), indicating their position as secondary revenue drivers. Meanwhile, complementary items, packaging materials, and condiments like Thousand Island (No. 13), Tortila Besar (No. 15), and Saos Tomat Edam (No. 12) show values approaching 0.000, reflecting their significantly lower individual sales volumes and transaction frequencies compared to the primary products.
Despite this concentration of values near zero, the normalized dataset successfully preserves the relative variance

and hierarchical sales patterns among the products. This uniform scaling provides a mathematically balanced foundation for the Euclidean distance calculations in the next phase. By eliminating scale disparities, the K-Means algorithm can now accurately partition the products into High, Medium, and Low clusters based on their true relative performance rather than raw numerical differences, ensuring that the resulting inventory control recommendations are both statistically sound and practically relevant for the business.

3.2. Initial Centroid Determination

Three initial centroids were determined by selecting representative products with high, medium, and low sales performance: C1 (High) = Roti Burger (1.000 ; 1.000 ; 1.000), C2 (Medium) = Saos Special Edam (0.012 ; 0.145 ; 0.148), and C3 (Low) = Daging Patty Edam (0.006 ; 0.089 ; 0.039). The selection of initial centroids is a critical step in the K-Means algorithm, as it directly influences the convergence speed and the final stability of the clusters. In this study, rather than relying on random initialization—which can sometimes lead to suboptimal clustering or local minima, especially with small and highly skewed datasets—a heuristic approach based on business knowledge and data distribution was employed. By selecting actual, representative products as the starting anchors, the algorithm is guided toward a solution that is both statistically sound and practically interpretable for the business owner. The assignment of C1 (High) to Roti Burger is straightforward and necessary. As the absolute outlier and market leader, Roti Burger achieved the maximum normalized score of 1.000 across all three variables. Establishing it as the initial centroid for the "High" cluster ensures that the algorithm immediately recognizes the existence of a dominant, high-turnover segment. This prevents the "High" cluster from being diluted or merged with other categories during the early iterations, guaranteeing that the core product receives dedicated inventory priority.

The choice of C2 (Medium), represented by Saos Special Edam, is particularly strategic from an analytical perspective. While its physical quantity sold is relatively low (0.012), its normalized revenue (0.145) and transaction frequency (0.148) are significantly higher than the majority of the dataset. This selection serves a crucial purpose: it forces the K-Means algorithm to define the "Medium" cluster not merely by physical sales volume, but by a combination of economic value and customer engagement. Without this anchor, the algorithm might have incorrectly grouped profitable, medium-value items into the "Low" cluster simply because their physical quantities were small. This ensures that the resulting inventory recommendations account for both turnover speed and financial contribution.

Finally, C3 (Low) was initialized using Daging Patty Edam (0.006 ; 0.089 ; 0.039), which accurately represents the "long tail" of the product catalog. This product exhibits consistently low performance across all three metrics, making it an ideal anchor for the numerous complementary items, condiments, and packaging materials that are sold infrequently. By setting this baseline, the algorithm can efficiently group the majority of low-moving inventory together, allowing the business owner to apply a unified "order in smaller quantities" strategy without having to manage each item individually. This strategic initialization proved highly effective, as it provided well-separated starting points across the multi-dimensional space. This not only accelerated the clustering process but also ensured that the subsequent iterations converged rapidly toward a solution that directly aligns with the operational inventory control needs of Edam Burger.

3.3. Euclidean Distance Calculation – Iteration 1

Following the initialization of the three centroids, the first iteration of the K-Means algorithm involves calculating the Euclidean distance from each of the fifteen normalized products to each of the three centroids. The product is then assigned to the cluster corresponding to the centroid with the minimum distance. The Euclidean distance formula used is $d(x, y) = \sqrt{\sum(x_i - y_i)^2}$.

To illustrate the calculation process, two examples are presented: one representing a product that matches its initial centroid, and another representing a product that requires evaluation between multiple clusters.

Example 1: Daging Patty Edam (Low Cluster Anchor) For Daging Patty Edam, with normalized values

(0.006 ; 0.089 ; 0.039), the distances to the three centroids are calculated as follows:

$$\begin{aligned}
\text{Distance to C1 (High): } d &= \sqrt{(0.006-1.000)^2+(0.089-1.000)^2+(0.039-1.000)^2} \\
&= \sqrt{0.988+0.829+0.923} = \sqrt{2.740} = 1.656 \\
\text{Distance to C2 (Medium): } d &= \sqrt{(0.006-0.012)^2+(0.089-0.145)^2+(0.039-0.148)^2} \\
&= \sqrt{0.000+0.003+0.012} = \sqrt{0.015} = 0.123 \\
\text{Distance to C3 (Low): } d &= \sqrt{(0.006-0.006)^2+(0.089-0.089)^2+(0.039-0.039)^2} = \sqrt{0} = 0.000
\end{aligned}$$

Since the minimum distance is 0.000 (to C3), Daging Patty Edam is assigned to the Low cluster.
Example 2: Paket Burger Edam (Cluster Evaluation) For Paket Burger Edam, with normalized values (0.002 ; 0.076 ; 0.031), the algorithm determines the closest cluster among the three centroids:

$$\begin{aligned}
\text{Distance to C1 (High): } d &= \sqrt{(0.002-1.000)^2+(0.076-1.000)^2+(0.031-1.000)^2} \\
&= \sqrt{0.996+0.854+0.939} = \sqrt{2.789} = 1.670 \\
\text{Distance to C2 (Medium): } d &= \sqrt{(0.002-0.012)^2+(0.076-0.145)^2+(0.031-0.148)^2} \\
&= \sqrt{0.000+0.005+0.014} = \sqrt{0.019} = 0.136 \\
\text{Distance to C3 (Low): } d &= \sqrt{(0.002-0.006)^2+(0.076-0.089)^2+(0.031-0.039)^2} \\
&= \sqrt{0.000+0.000+0.000} = \sqrt{0.000} = 0.015
\end{aligned}$$

Comparing the three distances (1.670, 0.136, and 0.015), the minimum distance is to C3. Therefore, Paket Burger Edam is assigned to the Low cluster.
The same distance calculation procedure was applied to all fifteen products. The complete results of the Euclidean distance calculations and the resulting cluster assignments for Iteration 1 are presented in Table 7.

Table 7. Results of Euclidean Distance Calculations – Iteration 1 (All Products)

No	Product	d ke C1	d ke C2	d ke C3	Cluster
1	Daging Patty Edam	1.656	0.123	0.000	Low
2	Daging Patty Hemato	1.551	0.010	0.126	Medium
3	Daging Patty Yona	1.694	0.162	0.049	Low
4	Kertas Logo Edam Isi 100	1.676	0.140	0.050	Low
5	Kertas Logo Edam Isi 50	1.722	0.196	0.091	Low
6	Paket Burger Edam	1.670	0.136	0.015	Low
7	Paket Burger Hemato	1.679	0.144	0.037	Low
8	Roti Burger	0.000	1.559	1.656	High
9	Roti Hotdog	1.437	0.157	0.241	Medium
10	Saos Sambal Edam	1.722	0.196	0.086	Low
11	Saos Special Edam	1.559	0.000	0.123	Medium
12	Saos Tomat Edam	1.726	0.200	0.092	Low
13	Thousand Island	1.727	0.202	0.089	Low
14	Thousand Island & SPC Cup	1.727	0.202	0.094	Low
15	Tortila Besar	1.726	0.200	0.091	Low

3.4. Centroid Update and Euclidean Distance – Iteration 2

After the cluster assignment in Iteration 1, the K-Means algorithm proceeds to update the position of each cluster's centroid. The new centroid is calculated as the arithmetic mean of all normalized data points currently assigned to that cluster, using Equation (3):

$$C_j = \frac{1}{n_j}\sum_{i=1}^{n_j} x_i \tag{3}$$

Where C_j is the new centroid for cluster j, n_j is the number of data points in cluster j, and x_i represents the normalized value of each data point.
Based on the cluster assignments in Table 7, the High cluster (C1) contains only one product (Roti Burger), so

its centroid remains unchanged at (1.000 ; 1.000 ; 1.000). The Medium cluster (C2) contains three products, and its new centroid is calculated as follows:

$$\begin{aligned}\text{Qty: } &(0.022 + 0.149 + 0.012)/3 = 0.183/3 = 0.061\\ \text{Revenue: } &(0.149 + 0.139 + 0.145)/3 = 0.433/3 = 0.144\\ \text{Freq: } &(0.148 + 0.227 + 0.148)/3 = 0.523/3 = 0.174\end{aligned}$$

The Low cluster (C3) contains the remaining eleven products. Applying Equation (3) to their normalized values yields the new centroid:

$$C3 \text{ (Low): } (0.001; 0.030; 0.021)$$

The updated centroids for Iteration 2 are summarized in Table 8.

Table 8. Updated Centroid Values – Iteration 2

Cluster	Qty	Revenue	Freq.
C1 (High)	1.000	1.000	1.000
C2 (Medium)	0.061	0.144	0.174
C3 (Low)	0.001	0.030	0.021

Using these updated centroids, the Euclidean distance from each product was recalculated using Equation (2). As an example, the recalculation for Daging Patty Edam (0.006 ; 0.089 ; 0.039) is shown below:

$$\text{Distance to C1 (High): } d = \sqrt{(0.006 - 1.000)^2 + (0.089 - 1.000)^2 + (0.039 - 1.000)^2} = 1.656 \quad (4)$$

$$\begin{aligned}\text{Distance to C2 (Medium): } d &= \sqrt{(0.006 - 0.061)^2 + (0.089 - 0.144)^2 + (0.039 - 0.174)^2}\\ &= \sqrt{0.003 + 0.003 + 0.018} = 0.156 \quad (5)\end{aligned}$$

$$\begin{aligned}\text{Distance to C3 (Low): } d &= \sqrt{(0.006 - 0.001)^2 + (0.089 - 0.030)^2 + (0.039 - 0.021)^2}\\ &= \sqrt{0.000 + 0.003 + 0.000} = 0.062 \quad (6)\end{aligned}$$

Comparing the three distances (1.656, 0.156, and 0.062), the minimum distance is to C3. Therefore, Daging Patty Edam remains assigned to the Low cluster. The same recalculation was performed for all fifteen products, with the complete results presented in Table 9.

Table 9. Results of Euclidean Distance Calculations – Iteration 2 (All Products)

No	Product	d ke C1	d ke C2	d ke C3	Cluster
1	Daging Patty Edam	1.656	0.156	0.062	Low
2	Daging Patty Hemato	1.551	0.047	0.176	Medium
3	Daging Patty Yona	1.694	0.192	0.013	Low
4	Kertas Logo Edam Isi 100	1.676	0.168	0.036	Low
5	Kertas Logo Edam Isi 50	1.722	0.222	0.029	Low
6	Paket Burger Edam	1.670	0.169	0.048	Low
7	Paket Burger Hemato	1.679	0.174	0.029	Low
8	Roti Burger	0.000	1.515	1.702	High
9	Roti Hotdog	1.437	0.102	0.275	Medium
10	Saos Sambal Edam	1.722	0.223	0.025	Low
11	Saos Special Edam	1.559	0.055	0.172	Medium
12	Saos Tomat Edam	1.726	0.227	0.030	Low
13	Thousand Island	1.727	0.229	0.030	Low
14	Thousand Island & SPC Cup	1.727	0.229	0.032	Low
15	Tortila Besar	1.726	0.227	0.030	Low

Convergence Analysis: A comparison between the cluster assignments in Iteration 1 (Table 7) and Iteration 2 (Table 9) reveals that no product changed its cluster membership between the two iterations. All fifteen products remain in their respective clusters: C1 contains one product, C2 contains three products, and C3 contains eleven products. Since the cluster assignments are identical, the centroid values will not change in any subsequent iteration. Therefore, the algorithm is considered to have converged after Iteration 2, and the final

clustering results are established.

3.5. Final Clustering Results for All Products

Table 10 presents the final clustering result after the K-Means algorithm achieved convergence at Iteration 2, covering all fifteen products in the dataset. Each product is assigned to one of three clusters based on its multi-dimensional sales performance, along with the corresponding inventory control recommendation.

Table 10. Final Results of K-Means Clustering for All Products

No	Product	Cluster	Recommendation
1	Daging Patty Edam	Low	Order in smaller quantities / evaluate promotion
2	Daging Patty Hemato	Medium	Re-stock according to normal demand
3	Daging Patty Yona	Low	Order in smaller quantities / evaluate promotion
4	Kertas Logo Edam Isi 100	Low	Order in smaller quantities / evaluate promotion
5	Kertas Logo Edam Isi 50	Low	Order in smaller quantities / evaluate promotion
6	Paket Burger Edam	Low	Order in smaller quantities / evaluate promotion
7	Paket Burger Hemato	Low	Order in smaller quantities / evaluate promotion
8	Roti Burger	High	Maintain stock availability
9	Roti Hotdog	Medium	Re-stock according to normal demand
10	Saos Sambal Edam	Low	Order in smaller quantities / evaluate promotion
11	Saos Special Edam	Medium	Re-stock according to normal demand
12	Saos Tomat Edam	Low	Order in smaller quantities / evaluate promotion
13	Thousand Island	Low	Order in smaller quantities / evaluate promotion
14	Thousand Island & SPC Cup	Low	Order in smaller quantities / evaluate promotion
15	Tortila Besar	Low	Order in smaller quantities / evaluate promotion

Cluster Distribution Summary: The final clustering produces the following distribution: C1 (High) contains 1 product (6.67%), C2 (Medium) contains 3 products (20.00%), and C3 (Low) contains 11 products (73.33%). This distribution reflects a highly skewed sales structure, which is characteristic of food-service businesses where a small number of core items drive the majority of revenue, while a large number of complementary products serve as supporting accessories to the main offering.

Analysis of Cluster C1 (High): The High cluster contains exclusively Roti Burger, which recorded 5,913 units sold, generating Rp 8,573,850 in revenue across 133 transactions over the five-month observation period. This product's normalized values reached the maximum score of 1.000 across all three clustering dimensions, confirming its position as an extreme outlier and the undisputed revenue driver of Edam Burger. From an inventory management perspective, Roti Burger represents the critical stock item whose unavailability would directly halt the business's primary revenue stream. Any stockout event for this product would result in immediate and significant revenue loss, making continuous stock availability the highest operational priority.

Analysis of Cluster C2 (Medium): The Medium cluster comprises three products: Daging Patty Hemato (137 units, Rp 1,370,000, 24 transactions), Roti Hotdog (886 units, Rp 1,284,700, 34 transactions), and Saos Special Edam (81 units, Rp 1,336,500, 24 transactions). These products demonstrate moderate but consistent sales performance and serve as secondary revenue contributors. Notably, while Roti Hotdog shows high physical volume, Saos Special Edam achieves its Medium status through strong revenue generation and transaction frequency despite lower unit sales. This confirms that the multi-variable clustering approach successfully captures economic value beyond mere physical quantity. These products require regular monitoring and scheduled restocking to ensure they remain available without creating excessive inventory holding costs.

Analysis of Cluster C3 (Low): The Low cluster encompasses the remaining eleven products, including various condiments (Saos Sambal Edam, Saos Tomat Edam, Thousand Island), packaging materials (Kertas Logo Edam Isi 50 and Isi 100), supplementary protein items (Daging Patty Edam, Daging Patty Yona), bundled packages (Paket Burger Edam, Paket Burger Hemato), and specialty items (Tortila Besar, Thousand Island & SPC Cup). These products share common characteristics: low individual transaction volumes (ranging from 8 to 43 units), modest revenue contributions (Rp 110,000 to Rp 860,000), and infrequent purchase patterns (5 to 12 transactions). Their grouping into a single Low cluster indicates that they function primarily as complementary items purchased alongside the core products, rather than as standalone revenue generators.

Business Validation and Context: The clustering results are consistent with the general sales pattern of a burger kiosk business, where the core product (burger bread) dominates total sales while supporting products

are purchased in much smaller quantities. This pattern aligns with the Pareto principle (80/20 rule) commonly observed in retail and food-service industries, where a minority of products generates the majority of revenue. The fact that the algorithm independently arrived at this structure without manual intervention validates both the appropriateness of the K-Means approach for this dataset and the quality of the normalization process applied in Section 3.1.

Inventory Control Recommendations: Based on the clustering results, the following differentiated inventory control strategies are recommended:

1. High Cluster (C1) – Roti Burger: Implement a continuous review inventory policy with a safety stock buffer to prevent stockouts. Procurement should be prioritized, and supplier relationships should be maintained to ensure consistent delivery. The system should generate automatic low-stock alerts when inventory falls below a predefined threshold.

2. Medium Cluster (C2) – Daging Patty Hemato, Roti Hotdog, Saos Special Edam: Apply a periodic review policy with regular restocking intervals aligned with normal demand patterns. These products should be monitored weekly to ensure availability while minimizing excess inventory that could lead to waste or increased holding costs.

3. Low Cluster (C3) – Eleven complementary products: Adopt a just-in-time ordering approach with smaller procurement quantities. These products should be evaluated periodically for potential bundling, promotional pricing, or menu integration strategies to increase their sales velocity. For items with persistently low turnover, the business owner should consider whether to maintain, reposition, or discontinue them.

This differentiated approach to inventory management is supported by Shen [13], who emphasized that systematic stock control based on actual sales data can prevent both stockouts and overstock conditions in small retail businesses. Furthermore, the findings are consistent with Sunia et al. [12], who demonstrated that K-Means-based inventory grouping enables more efficient resource allocation compared to uniform stocking policies. Similarly, Dermawan et al. [4] confirmed that clustering-based stock management reduces waste and improves operational efficiency in small-scale retail environments.

Implications for the Business: The integration of these clustering results within the web-based sales application provides Edam Burger's owner with a data-driven decision support tool that transforms raw transaction records into actionable inventory strategies. Rather than relying on intuition or uniform restocking practices, the owner can now allocate procurement budgets more effectively—investing heavily in the High-cluster product, maintaining steady supply for Medium-cluster items, and minimizing capital tied up in slow-moving Low-cluster products. This targeted approach is expected to reduce inventory holding costs, minimize product waste (particularly relevant for perishable food items), and improve overall business profitability.

3.6. System Testing

To ensure the reliability and correctness of the developed application, comprehensive testing was conducted using the Black Box Testing method. Black Box Testing is a software testing technique that evaluates the functionality of a system based on its inputs and outputs, without examining the internal code structure or implementation logic [11]. This approach is particularly suitable for validating that the system meets its functional requirements from the end-user's perspective, ensuring that every feature produces the expected output for each given input. The testing process covered all six core modules of the sales application: (1) User Authentication (Login), (2) Product Management, (3) Sales Transaction Processing, (4) Stock Control and Notification, (5) K-Means Clustering Execution, and (6) Sales Report Generation. Each module was tested with multiple scenarios, including valid inputs, invalid inputs, and boundary conditions, to verify both functional correctness and error-handling capabilities. The complete test scenarios and results are summarized in Table 11.

3.7. Discussion

The findings of this study demonstrate that integrating a K-Means clustering algorithm within a web-based sales application effectively transforms raw transactional data into actionable inventory strategies for small and medium enterprises (SMEs). The clustering process successfully categorized the fifteen products into three distinct segments: one dominant high-selling product (Roti Burger), three medium-selling secondary items, and eleven low-selling complementary products. This distribution accurately reflects the typical sales architecture of

Table 11. Black Box Testing Results

No	Module	Test Scenario	Input	Expected Result	Actual Result	Status
1	Login	Valid credentials	Correct username & password	Redirect to dashboard	Redirect to dashboard	Valid
2	Login	Invalid credentials	Wrong password	Display error message	Display error message	Valid
3	Product Management	Add new product	Complete product form data	Product saved to database	Product saved successfully	Valid
4	Product Management	Edit product	Modified product details	Product data updated	Product data updated	Valid
5	Product Management	Delete product	Select product & confirm	Product removed from list	Product removed successfully	Valid
6	Sales Transaction	Create new transaction	Select product, set quantity	Transaction recorded, stock reduced	Transaction recorded, stock updated	Valid
7	Sales Transaction	Exceed stock limit	Quantity > available stock	Display warning message	Display stock alert	Valid
8	Stock Control	Low stock threshold	Stock falls below minimum	Trigger low-stock notification	Notification displayed	Valid
9	K-Means Clustering	Execute clustering	Select date range, run algorithm	Display 3 clusters with products	Clusters generated correctly	Valid
10	K-Means Clustering	View recommendations	Click cluster category	Display inventory recommendations	Recommendations displayed	Valid
11	Sales Reporting	Generate monthly report	Select month & year	Display sales summary table	Report generated accurately	Valid

a food-service business, where a core staple drives the majority of revenue while condiments, packaging, and supplementary items serve as low-volume accessories. By mathematically validating this operational reality through multi-variable analysis, considering not just physical quantity but also revenue and transaction frequency—the system provides the business owner with an objective, data-driven basis for inventory management, replacing the traditional reliance on intuition or uniform restocking practices.

A critical contribution of this research lies in its architectural approach, which bridges the persistent gap between routine transaction processing and advanced data mining. Previous studies have successfully applied K-Means to retail and frozen food inventory, but typically as an offline, standalone analytical process requiring manual data extraction and external software execution. In contrast, the novelty of this study is the seamless embedding of the K-Means engine directly into the Laravel-based operational dashboard. This integration ensures that clustering analysis is not a retrospective, isolated academic exercise, but a continuous, real-time decision support tool that dynamically updates as new daily sales data are recorded. This directly addresses a common technological barrier in SMEs, where operational data is routinely collected but rarely utilized for strategic optimization due to the complexity of external analytical tools.

From a practical business standpoint, the system's segmentation directly translates into differentiated inventory control policies that optimize capital allocation and minimize operational waste. For the high cluster, the system mandates continuous stock availability to prevent revenue-halting stockouts of the core product. For the medium cluster, it enables periodic, demand-aligned restocking to maintain steady secondary revenue streams. Crucially, for the low cluster—which constitutes the vast majority of the catalog—the system recommends reducing procurement volumes and evaluating promotional strategies. For perishable or slow-moving food items, this targeted approach prevents working capital from being unnecessarily tied up in dead stock and significantly reduces the risk of product spoilage, thereby improving the overall financial health and operational efficiency of the business.

Despite these theoretical and practical contributions, this study acknowledges several limitations that present opportunities for future research. The clustering was conducted on a relatively constrained dataset comprising only fifteen product types over a five-month observation period, which may limit the generalizability of the segmentation model to businesses with vastly larger and more complex catalogs. Furthermore, the number of clusters ($k = 3$) was predetermined based on intuitive business logic (High, Medium, Low) rather than mathematically validated. Future research should explore dynamic cluster determination using evaluation metrics such as the Elbow Method or Silhouette Index to identify the optimal number of categories based on evolving

sales data. Additionally, expanding the observation period to capture seasonal fluctuations, and integrating predictive time-series forecasting algorithms alongside the clustering module, would further enhance the system's capability to provide proactive, forward-looking inventory recommendations.

4. CONCLUSION

This study successfully designed, developed, and implemented a web-based sales information system integrated with the K-Means clustering algorithm for Edam Burger & Frozen Foods Bekasi Utara. Utilizing the Waterfall methodology, the Laravel-based application effectively digitized manual sales and inventory processes, resolving previous issues related to data inaccuracies and delayed reporting. The core analytical engine, the K-Means algorithm, successfully segmented fifteen products based on multi-dimensional sales data, comprising quantity, revenue, and transaction frequency, into three distinct clusters: one high-selling product (6.7%), three medium-selling products (20%), and eleven low-selling complementary items (73.3%). Black Box Testing confirmed a 100% functional compliance rate across all system modules, ensuring the software operates reliably according to predefined specifications. The primary contribution of this research lies in its architectural approach, which bridges the persistent gap between routine transaction processing and advanced data mining. By embedding the K-Means engine directly into the operational dashboard, this system overcomes the common technological barrier in SMEs, where data is routinely collected but rarely utilized due to the complexity of offline analytical tools. This seamless integration empowers business owners to transition from intuition-based management to data-driven inventory strategies. The system generates differentiated, actionable procurement recommendations: prioritizing continuous stock availability for high-performing items to prevent revenue-halting stockouts, maintaining regular restocking for medium items, and minimizing capital tied up in low-moving inventory to reduce food waste and improve overall financial health.

Despite these theoretical and practical achievements, this study acknowledges certain limitations. The clustering was conducted on a relatively constrained dataset comprising fifteen product types over a five-month period, and the cluster count ($k = 3$) was predetermined based on intuitive business logic rather than mathematically validated. Future research should explore dynamic cluster determination using evaluation metrics such as the Elbow Method or Silhouette Index to identify the optimal number of categories. Additionally, expanding the observation period to capture seasonal fluctuations and integrating predictive time-series forecasting algorithms alongside the clustering module would further enhance the system's capability to provide proactive, forward-looking inventory recommendations.

DECLARATIONS

Author Contributions:

Muhammad Rofiq Ubaidillah conceived the study, conducted data collection and observation at Edam Burger, implemented the K-Means clustering algorithm, developed the web-based application, and prepared the original draft of the manuscript. R Wisnu Prio Pamungkas supervised the research, contributed to the methodology design, performed data analysis and interpretation of clustering results, and critically revised the manuscript for intellectual content. Prio Kustanto contributed to the system design and architecture, validated the testing procedures, and reviewed the final manuscript. All authors read and approved the final version of the manuscript.

Funding:

The authors received no financial support for the research, authorship, and/or publication of this article.

Conflicts of Interest:

The authors declare no conflict of interest regarding the publication of this paper.

Data Availability:

The data that support the findings of this study are available from the corresponding author upon reasonable request.

AI Use Statement:

During the preparation of this work, the authors used Claude (Anthropic) to assist in structuring and improving the language and readability of the manuscript. The authors reviewed and edited all content and take full responsibility for the final published article.

Additional Information:

No additional information is available for this paper.

REFERENCES

[1] V. Y. P. Ardhana, "Application of the Extreme Programming Method in a Web-Based Sales Information System," *JISMDB*, vol. 1, no. 2, pp. 227–235, 2024.

[2] F. L. D. Cahyanti, E. Firasari, and U. Khultsum, "Web-Based Sales Information System for SME Rukun Makmur Tlingsing," *Nuansa Inform.*, vol. 18, 2024.

[3] A. E. Yanuar and M. A. Senubekti, "Perancangan Aplikasi Penjualan Online Berbasis Website (Studi Kasus: Bakso Emsa)," *J. Nuansa Inform.*, vol. 16, no. 1, 2022.

[4] H. D. Dermawan, R. Kurniawan, and Y. A. Wijaya, "Sales Data Analysis Using K-Means Clustering at Daun Indah Store on Shopee," *CESS*, vol. 9, no. 1, pp. 175–191, 2024.

[5] S. Kumar, R. Rani, S. K. Pippal, and R. Agrawal, "Customer segmentation in e-commerce: K-means vs hierarchical clustering," *Telkomnika (Telecommunication Comput. Electron. Control.)*, vol. 23, no. 1, pp. 119–128, 2025, doi: 10.12928/TELKOMNIKA.v23i1.26384.

[6] H. Mukhtar et al., "K-Means Algorithm for Customer Behavior Clustering," *SEIS*, vol. 4, no. 2, pp. 96–101, 2024.

[7] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, "K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data," *Inf. Sci. (Ny).*, vol. 622, pp. 178–210, 2023, doi: 10.1016/j.ins.2022.11.139.

[8] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means algorithm: A comprehensive survey and performance evaluation," *Electronics*, vol. 9, no. 8, pp. 1–12, 2020, doi: 10.3390/electronics9081295.

[9] R. I. Manarung, E. Widodo, and A. M. Rifai, "Sales Data Clustering Using the K-Means Algorithm to Determine Retail Product Needs," *Int. J. Softw. Eng. Comput. Sci.*, vol. 5, no. 1, pp. 226–234, 2025, doi: 10.35870/ijsecs.v5i1.4090.

[10] B. I. Nugroho, A. Rafhina, P. S. Ananda, and G. Gunawan, "Customer segmentation in sales transaction data using k-means clustering algorithm," *J. Intell. Decis. Support Syst.*, vol. 7, no. 2, pp. 130–136, 2024, doi: 10.35335/idss.v7i2.236.

[11] T. Amalina, D. B. A. Pramana, and B. N. Sari, "Metode K-Means Clustering dalam Pengelompokan Penjualan Produk Frozen Food," *J. Ilm. Wahana Pendidik.*, vol. 8, pp. 574–583, 2022.

[12] D. Sunia, J. Jasmir, and B. Purnama, "Implementation of Data Mining to Determine Inventory at Mahkota Mart Using K-Means," *JAKAKOM*, vol. 6, no. 1, 2026.

[13] B. Shen, "E-commerce Customer Segmentation via Unsupervised Machine Learning," in *The 2nd International Conference on Computing and Data Science (CONF-CDS 2021)*, New York, NY, USA: ACM, 2021. doi: 10.1145/3448734.3450775.

[14] B. Sembiring, S. Natalia, H. Winata, and S. Kusnasari, "Pengelompokan Prestasi Siswa Menggunakan Algoritma K-Means," *J. Sist. Inf. Triguna Dharma (JURSI TGD)*, vol. 1, no. 1, p. 31, 2022, doi: 10.53513/jursi.v1i1.4784.

[15] H. Humaidi and I. F. Hanif, "Perancangan Sistem Informasi Keuangan dan Aset Berbasis Web (Studi Kasus: PT Informatika Media Pratama)," vol. 1, no. 1, pp. 25–33, 2022.

[16] R. Triyanto, "Rancang Bangun Aplikasi Penjualan Berbasis Website (Studi Kasus: Toko Waroeng Bola)," *J. Sist. Inf. dan Sains Teknol.*, vol. 2, no. 1, pp. 1–9, 2020, doi: 10.31326/sistek.v2i1.670.

[17] S. F. Fakhirah et al., "Design and Development of an E-Commerce Website Using the Waterfall Method with the Laravel Framework," *JTOS*, vol. 8, no. 2, pp. 441–452, 2025.

[18] P. Kustanto, W. A. Kurniawan, A. Noe'man, Nurfiyah, and Ridwan, "Implementation of Priority Scheduling Algorithm in Designing Infrastructure Damage Reporting System," *Greenation Int. J. Eng. Sci.*, vol. 2, no. 3, pp. 98–112, 2024.

[19] E. Bu'ulolo, "K-NN Algorithm with Min-Max Normalization for Selecting Eligible Students for the KIP Kuliah Merdeka Scholarship," *SIMKOM*, vol. 9, no. 2, pp. 184–194, 2024.

[20] X. Pei et al., "Robustness of machine learning to color, size change, normalization, and image enhancement on micrograph datasets with large sample differences," *Mater. Des.*, vol. 232, p. 112086, 2023, doi: 10.1016/j.matdes.2023.112086.

[21] F. P. Dewi, P. S. Aryni, and Y. Umaidah, "Implementasi Algoritma K-Means Clustering Seleksi Siswa Berprestasi Berdasarkan Keaktifan dalam Proses Pembelajaran," *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 7, no. 2, pp. 111–121, 2022, doi: 10.14421/jiska.2022.7.2.111-121.