AITI: Jurnal Teknologi Informasi, Volume 22 No. 1 Maret 2025, 29-45
ISSN 1693-8348 E-ISSN 2615-7128

JURNAL TEKNOLOGI INFORMASI

Product bundling and substitution recommendation system: Facilitating marketing improvement strategy of retail business

Ellysa Tjandra [1], Liliana [2]
[1]Program Studi Teknik Informatika, Fakultas Teknik, Universitas Surabaya
[2]Program Studi Sistem Informasi, Fakultas Teknik, Universitas Surabaya
Jl. Raya Kalirungkut, Surabaya, Indonesia
Email: [2]lili@staff.ubaya.ac.id

Article History

Received:	Revised:	Accepted:
03-07-2024	02-08-2024	06-08-2024

Abstract

A retail store is a place where a variety of everyday items are sold. Supermarkets need to use different tactics to sell their products and attract consumers. According to an interview with an employee of Supermarket X, one strategy supermarkets use is to bundle products. This involves combining two or more products into one package and selling them at a discounted price. Supermarkets sometimes need help choosing the products to bundle and finding similar products due to limited stock. Based on the problem description, it can be inferred that the supermarket system will utilize the Apriori algorithm for creating product bundling and the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) method to showcase similar products and predict if any products are running low on stock. The system's performance has been evaluated through trials and validation at the supermarket. The findings showed that they were able to determine which products may be included in the package with the use of this product packaging automation system. The results were promising, indicating that choosing replacement products can aid in lowering the inventory of goods that are not doing well.

Keywords: *apriori, information system, market basket, product bundling, TOPSIS*

Abstrak

Toko retail adalah tempat di mana berbagai macam barang sehari-hari dijual. Supermarket perlu menggunakan taktik berbeda untuk menjual produknya dan menarik konsumen. Berdasarkan wawancara dengan karyawan Supermarket X, salah satu strategi yang digunakan supermarket adalah dengan menggabungkan produk. Dua atau lebih produk digabungkan ke dalam satu paket dan menjualnya dengan harga diskon. Supermarket terkadang membutuhkan bantuan dalam memilih produk yang akan digabung dan menemukan produk serupa karena keterbatasan stok. Berdasarkan uraian masalah, dapat disimpulkan bahwa sistem pada supermarket menggunakan algoritma Apriori untuk membuat *bundling* produk dan metode *Technique for Order of Preference by Kemiripan ke Solusi Ideal* (TOPSIS) untuk menampilkan produk serupa dan memprediksi apakah ada produk yang stoknya sedikit. Kinerja sistem telah dievaluasi melalui uji coba dan validasi di supermarket. Temuan menunjukkan bahwa algoritma mampu menentukan produk mana yang dapat dimasukkan ke dalam kemasan

menggunakan sistem otomasi pengemasan produk. Hasilnya cukup menjanjikan, menunjukkan bahwa memilih produk pengganti dapat membantu menurunkan persediaan barang yang tidak berfungsi dengan baik.

Kata kunci: apriori, sistem informasi, *market basket*, produk *bundling*, TOPSIS

Introduction

The intense competition in the information goods market forces retailers to continuously enhance their features and pricing methods [1], [2]. Product bundling combines two or more distinct products into a single bundle and sells them at a discount [3] or provides the customer with additional value [4], [5]. It is often used because its profitability increases both parties' payoffs by making rivals less aggressive in price due to greater product differentiation [6], [7]. Product bundling can increase profitability and promote a cooperative atmosphere, resulting in a win-win situation and cultivating a collaborative atmosphere that leads to a win-win solution [1]. Sellers must carefully consider their pricing strategy since it might affect customer behavior and, in turn, their profitability [8]. Many retailers use bundling as a marketing strategy to encourage customers to purchase two or more products [9]–[11].

This research analyzes product bundling strategy in a supermarket as a retailer system case study. Several problems are found in product bundling in the supermarket, i.e., errors in combining products to make packages because there is not enough data analysis to combine packaged products. In this supermarket, bundling products from well-known brands causes products from lesser-known brands to sell less, resulting in deadstock. These products are ultimately sold at discounted prices but are still above capital prices, which can decrease supermarkets' profits [12]. Many bundling products are also created without a thorough analysis of consumer buying patterns, which are not aligned with customers' needs and preferences [13], [14]. It is also hard to promote slow-selling products, leading to missed opportunities and finances. In addition, if a well-known brand product has limited stock in the supermarket, it will be sold separately or not packaged. Packaging will be done by looking for substitute products with a lot of stock or products that are difficult to sell.

This study performs a requirement analysis of bundling products and then implements a web-based product recommendation system. The system will be designed to analyze consumer purchase patterns and suggest product bundles based on these patterns. Product pairs are selected from products often purchased together by consumers in one transaction [15], [16], and substitute products (replacement products) are also considered to be used as packages [17], [18]. The chosen product is a product nearing its expiration date. The data is processed using market basket analysis utilizing the Apriori algorithm [19]–[21] to find product pairs often purchased together and the TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) method

[22]–[24] to create substitute products. Our study aims to help the supermarket sell bundling products tailored to consumer needs [25], [26], thereby increasing sales for more product variants so that stock turnover can be well managed. The limitation of this study is that the trial was conducted with limited data because the trial respondents were unwilling to provide original sales data.

This paper is divided into four parts. The second part discusses the research methodology. The third section discusses the findings from the research conducted, and the final section discusses the research conclusions.

Research Method

The research used the highly regarded waterfall method [27], with a supermarket as a case study, as depicted in Figure 1. The research begins with a comprehensive literature study, ensuring that the product integration needs are well understood and that the procedures and algorithms chosen are appropriate for the case. In the next stage, further analysis was carried out, considering performance factors and criteria using the market basket, Apriori, and TOPSIS methods. During the analysis stage, data was collected by distributing questionnaires to respondents who had shopped at supermarkets. On the supermarket side, our research team directly observed operations in the field and conducted interviews with officers.

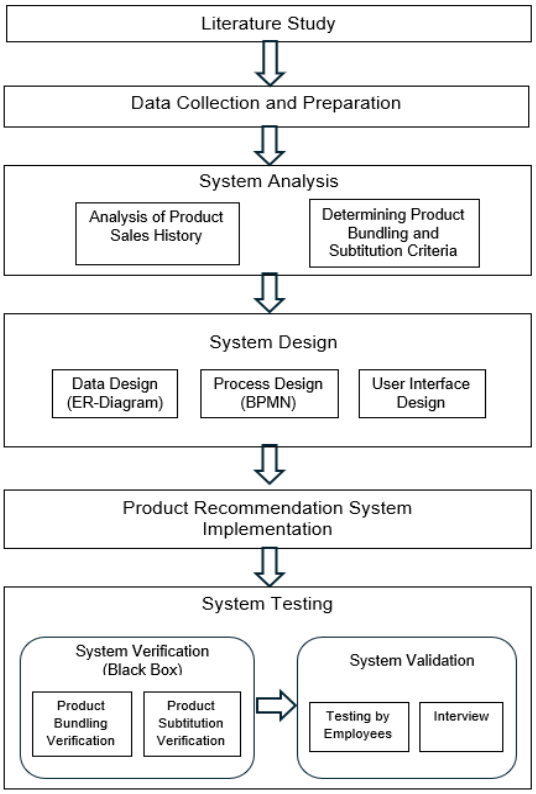

Figure 1 Research methodology

57.5% of respondents stated they were interested in packaged products because the price they had to pay was cheaper than buying them separately [7]. Respondents also said that with packaged products, they could try new brands at more affordable prices. Two supermarkets were analyzed. Both sources indicated that they implemented a package system in their supermarkets. The problems in making packages are errors in selecting and combining products to create packages that do not suit consumer needs because they combine these products manually.

Another problem is that there is a possibility that the product that will be made into a package product has limited stock, so the number of package products does not match. The next step is to conduct a benchmark using a similar system, namely the Fajar Stainless Steel application with the Apriori algorithm and smartphone selection using the TOPSIS method. The output of the analysis stage is the identification of these system requirements: (1) Determine product pairs that consumers often buy together in one transaction; (2) Identify the expiration date of each product so that tracking can be done when the product is approaching its expiration date; (3) Determine recommendations for substitute products by looking at the weight, price, and hashtag criteria for products in the same subcategory; (4) Implementing the Apriori algorithm to determine product pairs often purchased in one transaction and the TOPSIS method to find substitute products in the package.

Furthermore, system design was performed, including data design [28], process design [29], and user interface design [30], to produce a system that users can use easily and meets the requirements—the design data using a crow's foot diagram. The process design is divided into several parts: determining products often purchased together using the Apriori method, determining substitute products using the TOPSIS method, and determining package contents based on stock and expiration date.

After the process design is completed, the system is implemented by translating the design results into software ready for the supermarket to use. After that, testing is carried out through system verification with a BlackBox mechanism. Testing is done by entering a collection of scenarios used to run the system, including product bundling and product substitution verification, to ensure the resulting system is error-free. System validation is carried out by conducting direct testing of employees and interviews to ensure that the resulting system meets the requirements.

Results and Discussions

This section will discuss the results of each stage of the research methodology. Figure 2 shows the results of the design data using the crow's foot notation diagram. There are 13 entities needed to store system data. The results of the overall process design stage can be seen in Figure 3. Three main processes are carried out, including

the bundling process using the Apriori algorithm, determining substitute products with the TOPSIS algorithm, and combining substitute products into the resulting bundling products. More specifically, the process design using the Apriori algorithm is explained in Figure 4.

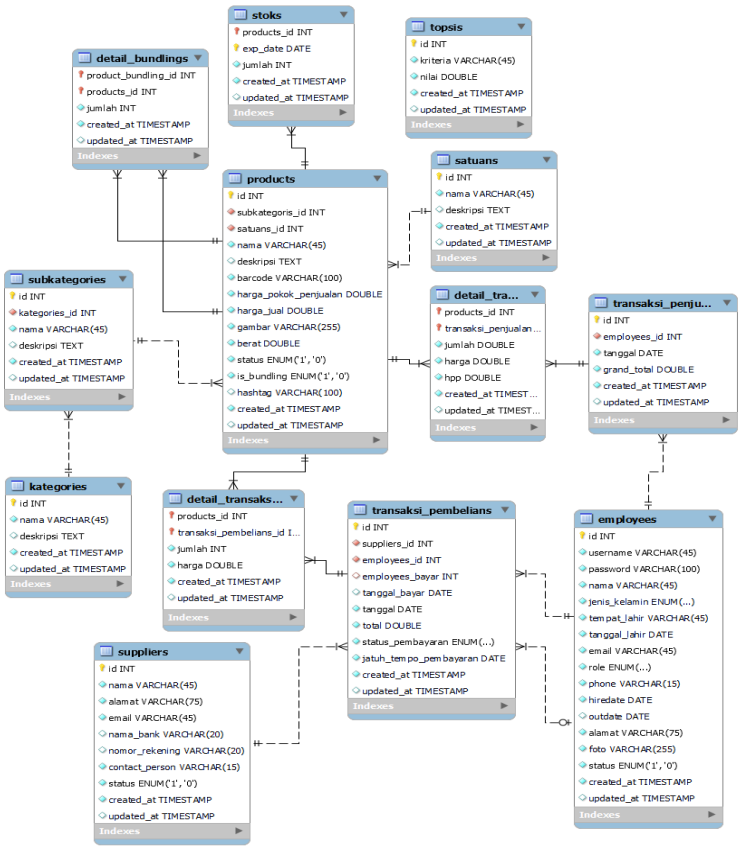

Figure 2 Data design

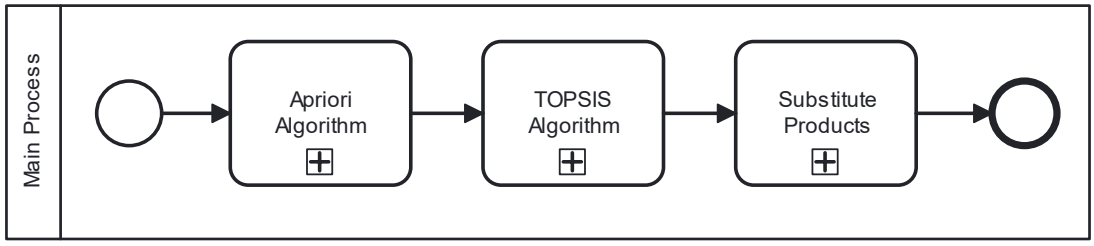

Figure 3 Main process

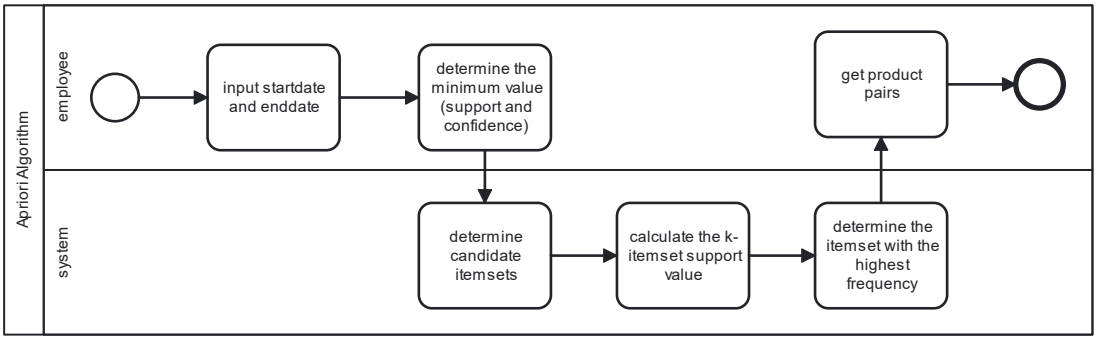

Figure 4 Apriori algorithm process design for product bundling systems

The TOPSIS approach, with its multiple calculation phases, is designed to identify replacement items [24]. Figure 5 shows the details of the TOPSIS algorithm used for product substitution. The algorithm plays a crucial role in this process, first assigning a score to each product based on each criterion and creating a decision matrix. The system then normalizes each product's price, weight, and hashtag in the choice matrix. The next step is to produce a normalized decision matrix, where the system must first square each score from the choice matrix and total each criterion's weight, price, and hashtag columns. The results from the sum of each criterion are then rooted. The score in the original matrix is then divided by the root result for every criterion. The system generates a weighted, normalized decision matrix in the third stage. The weight assigned to each criterion is multiplied by each score in the algorithm's normalized decision matrix. The values of the positive and negative ideal solutions are ascertained using the outcomes of the weighted normalized choice matrix. The system then confidently identifies the optimal solution that is positive. The system's ability to find the lowest score for the cost criteria and the highest score for the profit criteria for each criterion when determining the positive ideal solution instills confidence in its capabilities. The greater the profit criteria and the lower the cost criteria, the better the result is in identifying the positive optimum solution. The lower the product's price and weight in the positive ideal solution, the better the price and weight requirements are; therefore, they are classified as cost criteria. In the meantime, the more similar hashtags in the positive ideal solution, the better the hashtag criteria are classified as profit criteria.

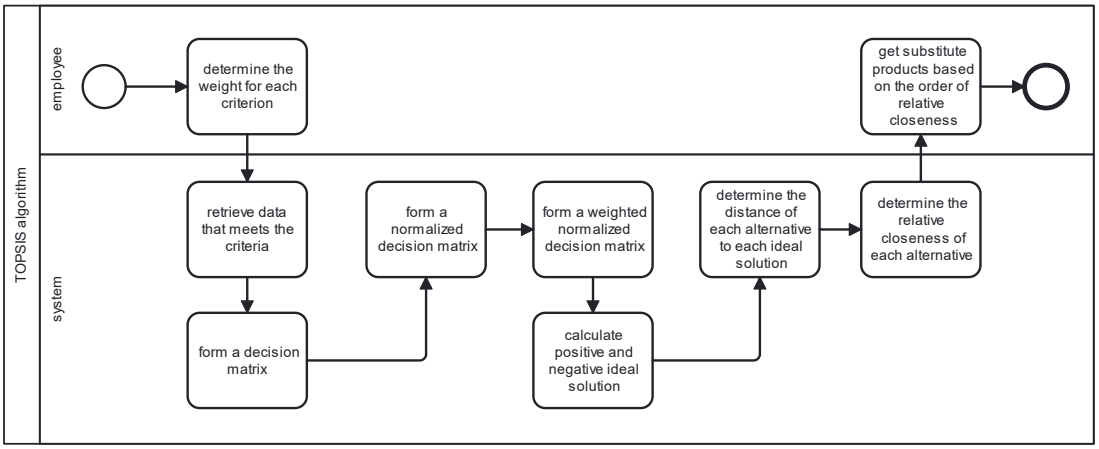

Figure 5 Topsis method process design for selection of substitute products

Next, users can search for substitute products in the bundling products package. Users are asked to input the number of the first product, the number of a second product, the number of product packages, the name of the product packages, the selling price, the description, and the product image. From the product selected on the package process page, the system will look for substitute products from the same subcategory, and the system will display substitute products in the combo box for each product. There are two options for displaying substitute products: weight, price, hashtag, expiration date, and number of sales. Figure 6 illustrates all the steps to combine substitute products into bundling products.

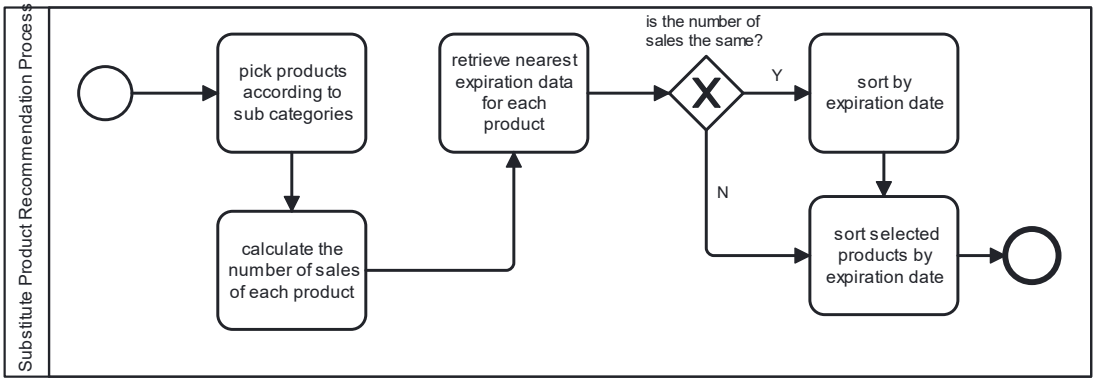

Figure 6 Process design for substitute products in bundling

After completing the process design results, the system was developed using the Laravel framework and MySQL database. In the packaging process, the user must fill in some data, namely the transaction date period, minimum support, and minimum confidence. The system then processes the packaging according to the user's input, displaying one or more frequently purchased product pairs. The packaging process page and detailed calculations are shown in Figure 7. The product pair results contain actions the user can carry out, such as using the main product for the product pair or

finding substitute products. A snippet of the program for searching for products purchased simultaneously can be seen in Algorithm 1, based on the Apriori algorithm [31]. In this study, 31 sales transaction data were used, involving 15 types of goods. The data processing results using the Apriori algorithm obtained four pairs of goods often purchased together, as seen in Table 1.

Table 1 Results of processing paired items with the apriori algorithm

1st product	2nd product	Confidence	Lift
Beras BMW 10Kg	*Minyak Goreng* Sunco 2L	70.00	1.808333
Minyak Goreng Sunco 2L	*Beras* BMW 10Kg	58.33	1.808333
Frestea *Minuman Teh Melati* 350Ml	Indomie *Instan Kaldu Ayam* 75Gr	100.00	2.384615
Indomie *Instan Kaldu Ayam* 75Gr	Frestea *Minuman Teh Melati* 350Ml	76.92	2.384615

According to the a priori algorithm flow, the process is as follows: the system determines products that meet the specified minimum support, and the list of products that meet the minimum support will have their confidence value calculated for other products. In the final process, a lift test is carried out on products that pass the confidence calculation to determine the increase in the sales ratio of packaged products compared to sales of separate products. Next, the system will identify the negative ideal solution in the following step of TOPSIS. The system will search for each criterion's maximum score for the cost and lowest scores for the profit criteria to identify the negative optimal solution. The more favorable the cost criterion, the more favorable the profit criterion when establishing the negative optimum solution. Since the higher the price and weight of the goods, the better the price and weight criteria are, therefore, classified as cost criteria in the negative ideal solution. In the meantime, the hashtag criteria fall within the category of profit criteria as, in the negative optimum solution, the fewer hashtags are used, the better.

Algorithm 1 Determining products purchased simultaneously

```
Method CalculatingSupport ($startdate, $enddate, $transCount, $minSupport)
{
    Calculate the number of transactions for each product, based on the
     start and end dates.
    For every product in the database
       Calculate support = transCount / number of all transactions
       if support >= minSupport, set status = okay, insert product into
        itemset
       else set status = failed
}
Method CalculatingConfidence($startDate, $endDate, $transCount, $itemset,
$minConfiddence)
{
    For every product combination in the itemset
      Calculate confidence = number of trans containing both products/number
       of trans containing first products
      If confidence >= minConfiddence, set status = okay, insert combination
       into itemsetUsed
      Else set status = failed
```

```
}
Method LiftTest ($itemsetUsed)
{
    For every product combination in itemsetUsed:
        Lift = count support(product1 dan product2) / (support(product1) *
         support(product2))
}
```

The system will next decide whether the positive ideal solution is adequate. The system will square the results after computing the difference in scores from the weighted normalized choice matrix with the positive perfect solution results for each criterion to determine a positive ideal solution. Subsequently, the algorithm will collect the points assigned to each option and calculate the overall outcomes. The system will ascertain the similarity to the negative ideal solution aside from that. The system will square the results after computing the difference in scores from the weighted normalized choice matrix with the negative ideal solution results for each criterion to arrive at a distinct negative ideal solution. Subsequently, the algorithm will collect the points assigned to each option and calculate the overall outcomes. The system will calculate the relative closeness in the last phase. The values of the positive and negative ideal solution separations will be added together by the system, and the total will then be divided by the negative ideal solution separation value. The product most similar to the original has the most significant relative proximity value, which is determined by the system for each option. The user interface implementation example of the search page for substitute products based on weight, price, and hashtag is shown in Figure 8.

System testing is done on the software to ensure the resulting system is running well and following specifications. System testing was developed using 31 sales transaction data involving various product combinations, where the data was processed using manual methods (paper-based calculations) and data processing in a system that had been prepared. Black box testing was used to perform system verification by entering multiple scenarios. The system will determine if the products included in the sales transaction are supported once it has received data from the sales transaction list within a predetermined time frame. By dividing the number of transactions involving a particular product by the total number of transactions, the system will determine support for each product. Once the system has determined the support for every product, it will decide whether or not the product is eligible for the itemset if the support value is less than the minimum value of the user input. The system will execute all possible product combinations that meet the requirements for the itemset. After sorting the products, the system calculates each pair's confidence and lift value. At this point in the test, the system and manual computations yielded the same product combination, yielding appropriate results.

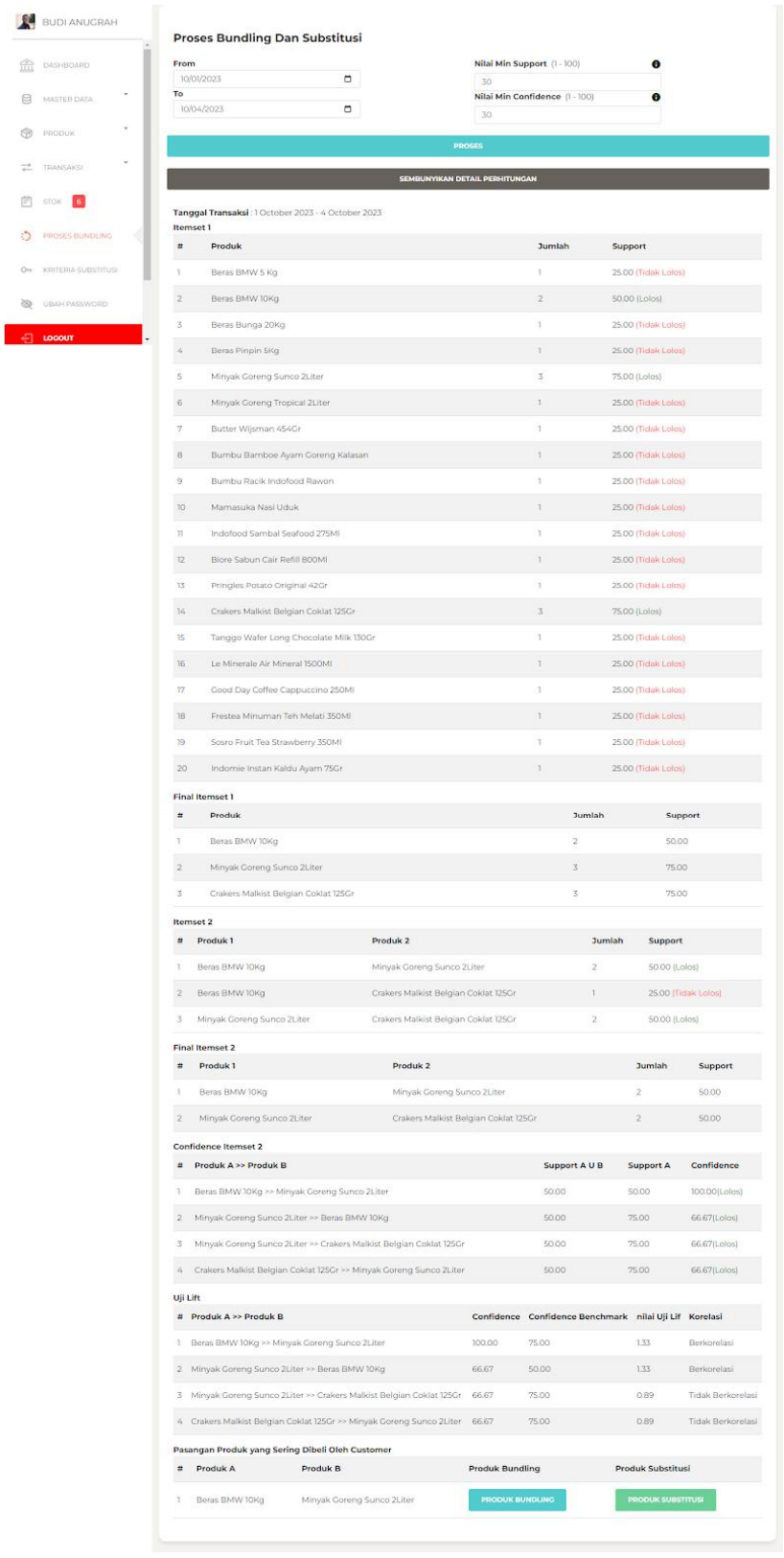

Figure 7 Bundling process

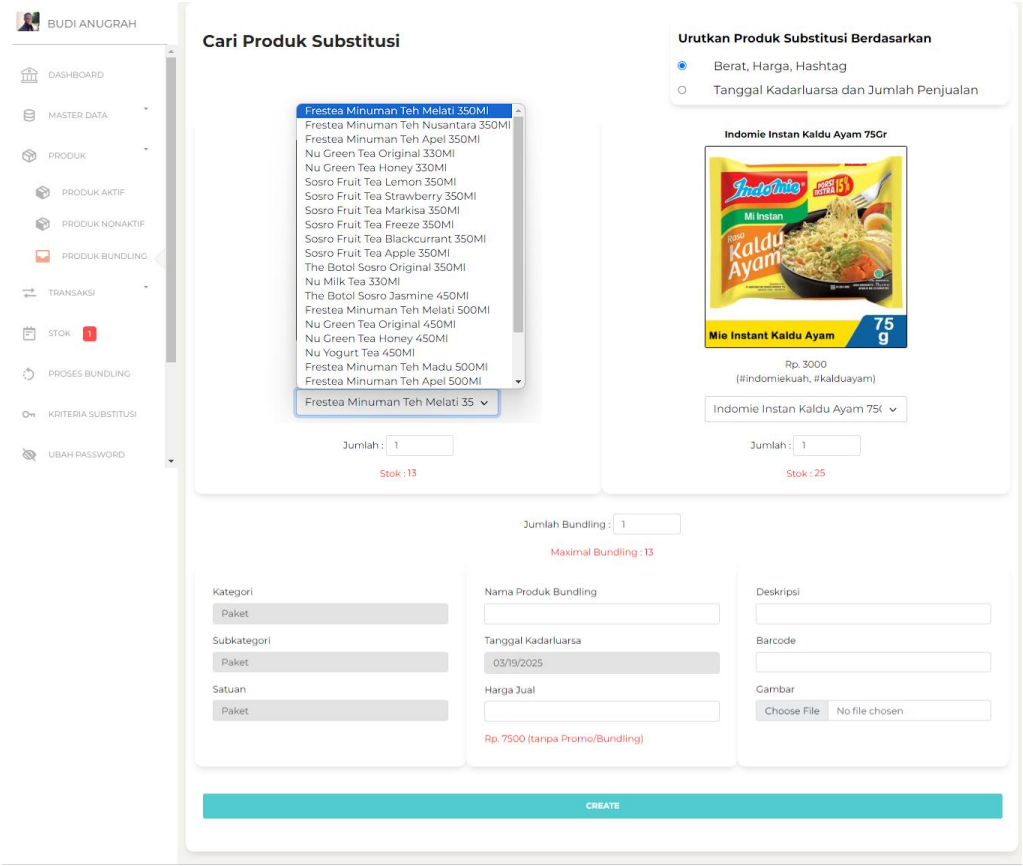

Figure 8 Search for product substitution

Data processing is done in the following testing step to find alternatives to the current items. During the testing phase, the system that has been developed will be manually compared to the calculation results to test the product substitution process feature. Using weight, price, and hashtags as criteria, replacement items for 10 kg of Rojolele rice were sought out for the research. Several products fall under the same rice subcategory—10 kilograms of Rojolele rice (Table 2). The price, weight, and hashtag were normalized for each product using the available data to generate a decision matrix when the product data was collected to find a replacement for 10 kg of Rojolele rice. A relative closeness table is created using the TOPSIS procedure and sorted in descending order (Table 3). Figure 9 illustrates how calculations made by the system follow the same pattern as calculations made by hand. As shown in Figure 10, choosing replacement products is not limited to comparing price, weight, and kind; it can also consider factors like stock availability and expiration date.

The experiment proceeded by verifying and validating the users who had participated at the beginning of the research once it was established that the application complied with manual computations [32]. Supermarket workers were requested to test the system with scenarios designed specifically for each store's flow. Employees at both supermarkets were interviewed, and the findings showed they could determine which

products may be included in the package using this packaging automation system. The results were promising, indicating that choosing replacement products can aid in lowering the inventory of goods that are not doing well. Ultimately, this procedure can reduce the likelihood of product waste or returns to suppliers while boosting store earnings, offering a bright future for the retail industry.

Table 2 List of substitute products

Name	Price	Weight	Hashtag
Rice BMW 5 Kg	75,000	5	#*beras*bmw, #*beras*5kg
Beras BMW 10Kg	150,000	10	#*beras*bmw, #*beras*10kg
Beras BMW 20Kg	300,000	20	#*beras*bmw, #*beras*20kg
Beras BMW 25Kg	378,000	25	#*beras*bmw, #*beras*25kg
Beras Rojolele 5Kg	75,000	5	#*beras*rojolele, #*beras*5kg
Beras Rojolele 20Kg	280,000	20	#*beras*rojolele, #*beras*20kg
Beras Rojolele 25Kg	370,000	25	#*beras*rojolele, #*beras*25kg
Beras Bunga 5Kg	60,000	5	#*beras*bunga, #*beras*5kg
Beras Bunga 10Kg	130,000	10	#*beras*bunga, #*beras*10kg
Beras Bunga 20Kg	288,000	20	#*beras*bunga, #*beras*20kg
Beras Bunga 25Kg	350,000	25	#*beras*bunga, #*beras*25kg
Beras Sumo 5 Kg	79,000	5	#*beras*sumo, #*beras*5kg
Beras Sumo 10Kg	150,000	10	#*beras*sumo, #*beras*10kg
Beras Sumo 20Kg	297,000	20	#*beras*sumo, #*beras*20kg
Beras Sumo 25KgA	387,000	25	#*beras*sumo, #*beras*25kg
Beras Maknyuss 5Kg	80,000	5	#*beras*maknyuss, #*beras*5kg
Beras Maknyuss 10Kg	148,000	10	#*beras*maknyuss, #*beras*10kg
Beras Pinpin 5Kg	77,000	5	#*beras*pinpin, #*beras*5kg
Beras Pinpin 10Kg	150,000	10	#*beras*pinpin, #*beras*10kg
Beras Pinpin 25Kg	367,000	25	#*beras*pinpin, #*beras*25kg
Beras Sania 5Kg	75,000	5	#*beras*sania, #*beras*5kg
Beras Sania 10Kg	150,000	10	#*beras*sania, #*beras*10kg
Beras Sania 20Kg	350,000	20	#*beras*sania, #*beras*20kg
Beras Top Koki 5Kg	70,000	5	#*beras*topkoki, #*beras*5kg
Beras Top Koki 10Kg	140,000	10	#*beras*topkoki, #*beras*10kg
Beras Top Koki 20Kg	298,000	20	#*beras*topkoki, #*beras*20kg
Beras Top Koki 25Kg	425,000	25	#*beras*topkoki, #*beras*25kg

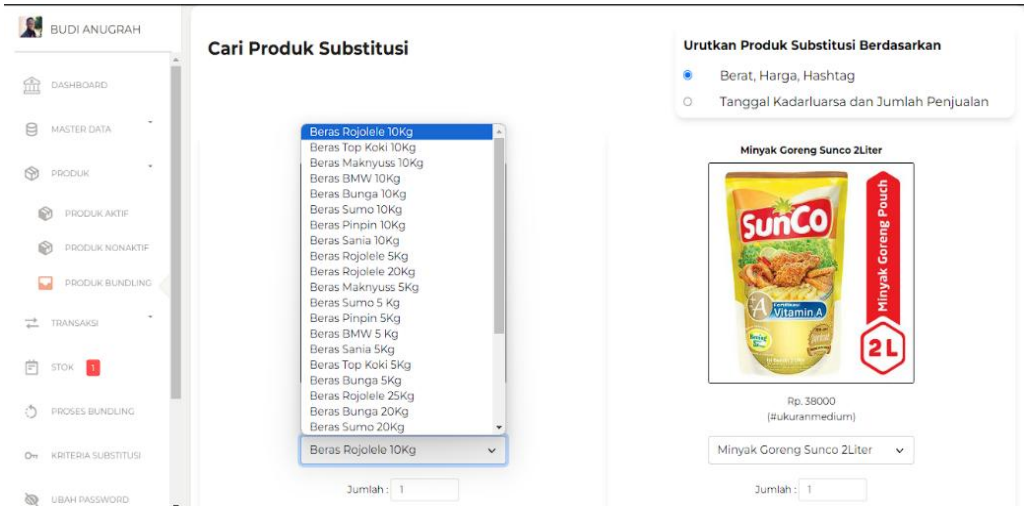

Figure 9 Substitution product list

Table 3 Relative closeness results

Name	Relative Closeness
Beras Top Koki 10Kg	1.000
Beras Maknyuss 10Kg	0.988
Beras BMW 10Kg	0.985
Beras Bunga 10Kg	0.985
Beras Sumo 10Kg	0.985
Beras Pinpin 10Kg	0.985
Beras Sania 10Kg	0.985
Beras Rojolele 5Kg	0.762
Beras Rojolele 20Kg	0.563
Beras Maknyuss 5Kg	0.478
Beras Sumo 5 Kg	0.477
Beras Pinpin 5Kg	0.476
Beras BMW 5 Kg	0.475
Beras Sania 5Kg	0.475
Beras Top Koki 5Kg	0.473
Beras Bunga 5Kg	0.468
Beras Rojolele 25Kg	0.427
Beras Bunga 20Kg	0.286
Beras Sumo 20Kg	0.279
Beras Top Koki 20Kg	0.278
Beras BMW 20Kg	0.277
Beras Sania 20Kg	0.243
Beras Bunga 25Kg	0.104
Beras Sania 25Kg	0.104
Beras Pinpin 25Kg	0.082
Beras BMW 25Kg	0.067
Beras Sumo 25Kg	0.055
Beras Top Koki 25Kg	0.000

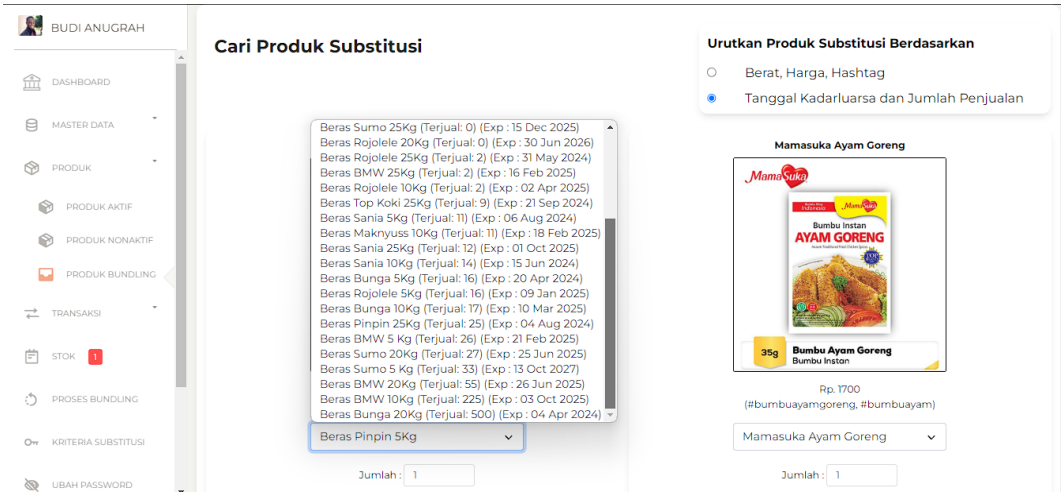

Figure 10 List of substitute products based on expiration date

Conclusion

Based on the implementation and testing results, our study concludes that applying an Apriori algorithm to identify frequently purchased product combinations, in conjunction with the TOPSIS method to identify substitute products, can assist retailers, like supermarkets, in choosing product packaging. This can lower the risk of stock storage losses, particularly for less well-known products, and ultimately boost profits for the company. However, the findings still need to be reviewed further, using more sales data to obtain higher accuracy. Based on the product packaging examined in this research, future research can identify another product promotion strategy, such as product event promo or discounts.

References

[1] C. H. Wu and P. D. L. Pambudi, "On-Premise Software vs. Cloud-Based Software under the Presence of Product Bundling," *Procedia Comput. Sci.*, vol. 234, pp. 180–187, 2024, doi: 10.1016/j.procs.2024.02.164.

[2] J. Yap and T. Wiradinata, "A comparative study on classification models for stock rating prediction," *Aiti*, vol. 21, no. 1, pp. 140–151, 2024, doi: 10.24246/aiti.v21i1.140-151.

[3] M. Mutasar and C. Niesa, "Analisis Transaksi Konsumen Bidang Data Mining Menggunakan Algoritma Apriori Untuk Rekomendasi Bundling Produk Pada 212 Mart Kota Lhokseumawe," *J. Tika*, vol. 6, no. 02, pp. 92–98, 2021, doi: 10.51179/tika.v6i02.463.

[4] A. Vamosiu, "Optimal bundling under imperfect competition," *Int. J. Prod. Econ.*, vol. 195, pp. 45–53, 2018, doi: 10.1016/j.ijpe.2017.09.016.

[5] A. D. Puspitasari, H. Riofita, and H. Com, "Strategi Pemasaran Product Bundling pada Produk Kosmetik Wardah dalam Meningkatkan Penjualan

Produk Retail," *Ekonodinamika J. Ekon. Din.*, vol. 6, no. 2, pp. 139–146, 2024, [Online]. Available: https://journalpedia.com/1/index.php/jed.

[6] P. T. Pungkasanti, N. Wakidah, and R. R. F. Kurniawan, "Penerapan metode Weighted Aggregated Sum Product Assessment (WASPAS) dalam menentukan reseller terbaik," *Aiti*, vol. 20, no. 2, pp. 206–219, 2023, doi: 10.24246/aiti.v20i2.206-219.

[7] Z. Sun, J. Yang, K. Feng, H. Fang, X. Qu, and Y. S. Ong, *Revisiting Bundle Recommendation: Datasets, Tasks, Challenges and Opportunities for Intent-aware Product Bundling*, vol. 1, no. 1. Association for Computing Machinery, 2022.

[8] D. Lukyani, R. Arifin, and A. B. Primanto, "Pengaruh Pemasaran Cross selling, Pemasaran Viral, dan Pemasaran Produk Bundling Terhadap Keputusan Pembelian Pada Produk Skintific (Studi Pada Konsumen di Kecamatan Lowokwaru Kota Malang)," *e – J. Ris. Manaj.*, vol. 12, no. 01, pp. 510–518, 2023, doi: http://dx.doi.org/10.24912/jseb.v1i1.22734.

[9] T. Chen, F. Shan, F. Yang, and F. Xu, "Online retailer bundling strategy in a dual-channel supply chain," *Int. J. Prod. Econ.*, vol. 259, 2023, doi: https://doi.org/10.1016/j.ijpe.2023.108821.

[10] T. Nie, B. Song, and J. Zhang, "Sales pricing models based on returns: Bundling vs. add-on," *Omega*, vol. 125, 2024, doi: https://doi.org/10.1016/j.omega.2024.103038.

[11] J. Li, L. Liu, X. Luo, and S. X. Zhu, "Interactive bundle pricing strategy for online pharmacies," *Transp. Res. Part E Logist. Transp. Rev.*, vol. 177, no. July, p. 103223, 2023, doi: 10.1016/j.tre.2023.103223.

[12] O. K. Sari and A. Cherid, "Aplikasi Berbasis Web Menggunakan Apriori Untuk Rekomendasi Bundling Produk Sembako," *Bit (Fakultas Teknol. Inf. Univ. Budi Luhur)*, vol. 20, no. 2, p. 95, 2023, doi: 10.36080/bit.v20i2.2444.

[13] F. Anggraini and A. Budiarti, "Pengaruh Harga, Promosi, dan Kualitas Pelayanan Terhadap Loyalitas Pelanggan Dimediasi Kepuasan Pelanggan Pada Konsumen Gojek," *J. Pendidik. Ekon.*, vol. 8, no. 3, pp. 86–94, 2020, doi: 10.26740/jupe.v8n3.p86-94.

[14] S. Bera and B. C. Giri, "Impact of consumer preferences on pricing and strategic decisions in a triopoly with heterogeneous smart sustainable supply chains," *Expert Syst. Appl.*, vol. 247, no. January, p. 123348, 2024, doi: 10.1016/j.eswa.2024.123348.

[15] S. Aryanti, D. Mahdiana, and A. Setiadi, "Penerapan Metode K-Means Dan Apriori Untuk Pemilihan Produk Bundling," *J. CERITA*, vol. 8, no. 1, pp. 1–12,

2022, doi: 10.33050/cerita.v8i1.2126.

[16] V. Venkatesh, H. Hoehle, J. A. Aloysius, and H. R. Nikkhah, "Being at the cutting edge of online shopping: Role of recommendations and discounts on privacy perceptions," *Comput. Human Behav.*, vol. 121, no. August 2020, p. 106785, 2021, doi: 10.1016/j.chb.2021.106785.

[17] M. Zhang and J. Bockstedt, "Complements and substitutes in online product recommendations: The differential effects on consumers' willingness to pay," *Inf. Manag.*, vol. 57, no. 6, p. 103341, 2020, doi: 10.1016/j.im.2020.103341.

[18] V. Gupta, D. Ivanov, and T. Choi, "Competitive pricing of substitute products under supply disruption," *Omega*, vol. 101, no. January, 2021.

[19] R. R. Ramadina, T. H. Pudjiantoro, and I. Santikarama, "Pembangunan Sistem Customer Relationship Management (CRM) Menggunakan Metode Asosiasi Algoritma Apriori," *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 2, no. 2, p. 78, 2020, doi: 10.36499/jinrpl.v2i2.3539.

[20] A. R. Riszky and M. Sadikin, "Data Mining Menggunakan Algoritma Apriori untuk Rekomendasi Produk bagi Pelanggan," *J. Teknol. dan Sist. Komput.*, vol. 7, no. 3, pp. 103–108, 2019, doi: 10.14710/jtsiskom.7.3.2019.103-108.

[21] S. B. Aher and L. M. R. J. Lobo, "Combination of machine learning algorithms for recommendation of courses in E-Learning System based on historical data," *Knowledge-Based Syst.*, vol. 51, pp. 1–14, 2013, doi: https://doi.org/10.1016/j.knosys.2013.04.015.

[22] S. Setiawansyah, "Sistem Pendukung Keputusan Rekomendasi Tempat Wisata Menggunakan Metode TOPSIS," *J. Ilm. Inform. dan Ilmu Komput.*, vol. 1, no. 2, pp. 54–62, 2022, doi: 10.58602/jima-ilkom.v1i2.8.

[23] Y. Çelikbilek and F. Tüysüz, "An in-depth review of theory of the TOPSIS method: An experimental analysis," *J. Manag. Anal.*, vol. 7, no. 2, pp. 281–300, 2020, doi: 10.1080/23270012.2020.1748528.

[24] S. Chakraborty, "TOPSIS and Modified TOPSIS: A comparative analysis," *Decis. Anal. J.*, vol. 2, no. September 2021, p. 100021, 2022, doi: 10.1016/j.dajour.2021.100021.

[25] E. F. L. Awalina and W. I. Rahayu, "Optimalisasi Strategi Pemasaran dengan Segmentasi Pelanggan Menggunakan Penerapan K-Means Clustering pada Transaksi Online Retail," *J. Teknol. dan Inf.*, vol. 13, no. 2, pp. 122–137, 2023, doi: 10.34010/jati.v13i2.10090.

[26] C.-H. Wang, "Considering economic indicators and dynamic channel interactions to conduct sales forecasting for retail sectors," *Comput. Ind. Eng.*,

vol. 165, 2022, doi: https://doi.org/10.1016/j.cie.2022.107965.

[27] A. Alshamrani and A. Bahattab, "A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model," *IJCSI Int. J. Comput. Sci. Issues*, vol. 12, no. 1, pp. 106–111, 2015, [Online]. Available: https://www.academia.edu/10793943/A_Comparison_Between_Three_SDLC_Models_Waterfall_Model_Spiral_Model_and_Incremental_Iterative_Model.

[28] P. Dybka, "Crow's Foot Notation," *Vertabelo SA*, 2020. https://www.vertabelo.com/blog/crow-s-foot-notation/.

[29] BPMN, "BPMN," *Object Management Group, Inc.*, 2020. http://www.bpmn.org/.

[30] D. S. S. Sahid, P. I. Santosa, R. Ferdiana, and E. N. Lukito, "Evaluation and measurement of Learning Management System based on user experience," *Proc. - 2016 6th Int. Annu. Eng. Semin. Ina. 2016*, pp. 72–77, 2017, doi: 10.1109/INAES.2016.7821910.

[31] I. A. Ashari, A. Wirasto, and D. N. Triwibowo, "Implementasi Market Basket Analysis dengan Algoritma Apriori untuk Analisis Pendapatan Usaha Retail," *Matrik J. Manajemen, Tek. Inform. dan Rekayasa Komput. Vol.*, vol. 21, no. 3, 2022, doi: 10.30812/matrik.v21i3.1439.

[32] S. M. Katrachanca and A. J. Koleske, "Causal Interpretations of Black-Box Models," *J. Bus. Econ. Stat.*, 2019, doi: https://doi.org/10.1080/07350015.2019.1624293.