Journal of Information Systems and Informatics
Vol. 6, No. 1, March 2024 e-ISSN: 2656-4882 p-ISSN: 2656-5935
DOI: 10.51519/journalisi.v6i1.671 Published By DRPM-UBD

Utilizing ORB Algorithm in Web-Based Sales Application

Edward Brainard Pranata[1], Tony[2]

[1,2]Information System, Tarumanagara University, Jakarta, Indonesia
Email: [1]edward.825200025@stu.untar.ac.id, [2]tony@fti.ac.id

Abstract

E-commerce has become common and important for businesses, but Jaya Sentosa Store has not implemented it. E-commerce commonly has only a search by keyword feature, but that cannot replicate Jaya Sentosa Store order process. An image-based search is needed to replicate the order process. Our research purpose is to develop a web-based sales application and an image search feature for Jaya Sentosa Store. We apply Scrum when developing this application. We use Javascript (JS) programming language. Back-end and front-end development employ Express JS and React JS framework, respectively. To get the right feature-matching algorithm, we conduct a test between the SIFT, KAZE, and ORB algorithms. We write Python scripts to implement ORB algorithm in image-based search feature. Our test shows that the ORB algorithm has the fastest average running time, i.e., 3.415 s, compared to SIFT and KAZE. Black box testing of the sales application shows that all cases are valid. It means that our application can replicate Jaya Sentosa Store order process and gain a competitive advantage.

Keywords: e-commerce, image-based search, ORB algorithm

1. INTRODUCTION

Due to shifting trends from shopping in traditional markets to online marketplaces, the e-commerce sector is expanding quickly. During the COVID-19 epidemic, the usage of e-commerce also developed more quickly. Work had to be done from home, and physical separation was implemented throughout the pandemic. This led to the near doubling of e-commerce in Indonesia [1]. The Internet has proven to be one of the most effective and efficient information-spreading media that can be accessed by anyone, anywhere, and anytime [2].

The Jaya Sentosa Store is the case study of this research. Jaya Sentosa Store is a store that focuses on selling office stationery and is located in Curup, Bengkulu. The products sold by Jaya Sentosa Store vary from various types of paper, ink, pens, and books to various electronic devices needed by an office. Although the rapid development of e-commerce provides great potential, Jaya Sentosa Store has not utilized it. Until now, Jaya Sentosa Store only sold goods in the physical store and received orders on the spot.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

E-commerce initially functions as a list of products, which then develops to accommodate image data for review in the form of words, videos, and images [3]. E-commerce then develops into a place that can fully carry out the function of buying and selling goods. The word e-commerce refers to online commerce or online buying and selling, so e-commerce can be interpreted as a place to buy and sell goods, products, or services online [4].

The transaction process at Jaya Sentosa Store is carried out directly in the physical store and via telephone and Whatsapp messages. Buyers will come directly to the store and mention the items they want to buy or show examples of goods, either in the form of pictures or physical goods. After that, the staff will take the appropriate items and write the order on the note. Customers then pay for the order and get all the ordered items. All payments occur directly at the Jaya Sentosa Store.

The use of e-commerce makes it easier to purchase goods. One of the features of e-commerce is the search for goods using text as input. This feature is very helpful in the process of using e-commerce applications. In the process of searching for goods, it is not uncommon for users to not know the name of the item but only have a picture of the product. This can be easily resolved in traditional markets by just showing a picture, and then the seller will provide a similar product.

The E-commerce Jaya Sentosa Store will make it easier for information about products to be obtained. This ease of information can allow buyers to see prices first without having to contact or visit the store. E-commerce Jaya Sentosa Store allows ordering goods before picking them up or sending them. This e-commerce also opens new markets and reaches customers from a wider area.

The E-commerce Jaya Sentosa Store also requires an item search feature using images as input. This is because e-commerce which generally only uses text searching cannot fulfill searches using patterns or shapes of goods. One implementation of text search that allows this is a filter, but the filter is not optimal because many parameters need to be filled in by the user.

In the development process of the e-commerce app, there has also been research previously conducted. Wijayakusuma et al. [5] conducted research related to the design of e-commerce websites for coffee products. In this research, the prototyping method is used, and the development uses the MySQL database. This research employs the Javascript (JS) programming language and the React JS framework in its development. This application has a menu on the website to manage data on goods sold. In the application, there is also a payment feature by uploading proof of payment. Meanwhile, the thesis research will be developed using the PostgreSQL database. The significant difference between that research and this research is the implementation of search with image input.

According to Addagarla et al. [6], many researchers have found various possibilities to deal with the problem. Most e-commerce uses text as the input for the search feature. This method has some limitations that have been felt by users. With this problem, the use of images as the input for search will be a solution that develops e-commerce further.

Implementation of image-based search can be done with various methods. Chen et al. [7] used CNN (convolutional neural network) with the VAL (Visiolinguistic Attention Learning) framework in the implementation of image-based search. The research shows the use of VAL in image searches on fashion product data. In another study, Addagarla et al. [6] applied the image-based search with the clustering method. In that research, many tests were used multiple clustering methods like K-means, K-medoids, Birch, GMM, agglomerative clustering, and Minibatch. That research also implemented the image-based feature into an e-commerce application that focuses on selling clothing.

The implementation of the image-based search in our research is going to be done using feature matching. Feature matching on an image can be called 2D object recognition. The most popular and commonly used algorithms in this field are Scale Invariant Feature Transform (SIFT), Speeded Up Robust Feature (SURF), and Oriented Fast and Rotated BRIEF (ORB) [8]. In the implementation, feature-matching algorithms like SIFT and others are still proven to be reliable and successful in many applications. Some of the implementations of this kind of algorithm include object detection, visual mapping, and image stitching [9].

The ORB algorithm is a result of the modification of some other algorithms. The ORB uses Fast Detector and BRIEF descriptors [10]. Rublee et al. [9] did multiple tests to test ORB capability in texture object detection and real-time feature tracking. The test came out with the ORB algorithm outrunning its predecessors, SIFT and SURF, with better results and a faster running time.

In previous research by Addagarla et al. [6] and Chen et al. [7], they used CNN, VAL, and clustering for the image-based search. The method used in our research is feature matching or 2D recognition. There are some options when it comes to feature matching and some that come to the top of mind are SURF, SIFT, and ORB. In our research, a test will be conducted to determine the best performance feature-matching algorithm to be implemented. This test will be based on runtime performance to ensure the user will get the result in a timely manner.

Our research aims to build a web-based sales application that can help Jaya Sentosa Store achieve a competitive advantage by expanding its market to the vast Internet. The implementation of the ORB algorithm aims to fulfill the e-commerce needs of the Jaya Sentosa. Store thus, it can function and serve customers, like coming directly to the store. This ORB algorithm will provide the output of several goods

that best match the input from customers. The result of this research is an e-commerce web that has been adjusted to the needs of the Jaya Sentosa Store and can be accessed from anywhere.

2. METHODS

2.1 Scrum Methodology

This study applies the Agile Scrum method in our study. Figure 1 shows stages in the Scrum method. Scrum methodology consists of 3 (three) stages, i.e., (i) planning, (ii) sprint, and (iii) closure [11]. There are 2 fixed stages: (i) planning and (ii) closure. Scrum has good flexibility because it assumes the analysis, design, and development process is unpredictable and changes can occur. Scrum is an Agile method; thus, it will implement sprints, backlogs, and standups [12].

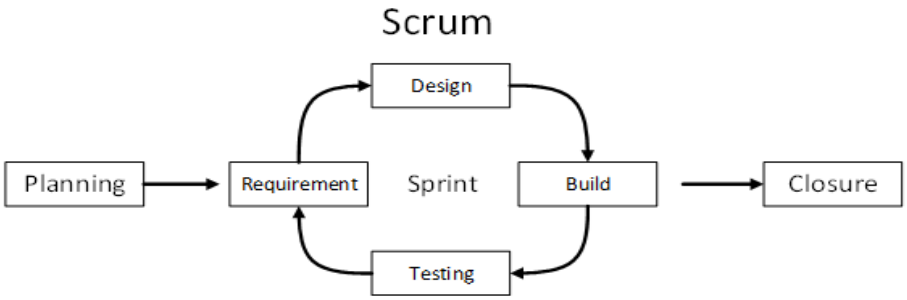

Figure 1. Scrum method

In the first stage, planning will be carried out. At this stage, an analysis of user needs is carried out. The information from this analysis will be used in creating a working design by creating detailed features and prioritizing all features in the form of a product backlog [13]. The data used for user requirement analysis is obtained by conducting direct interviews with the Jaya Sentosa Store and observing the Jaya Sentosa Store carry out the transaction process directly. The results of this analysis will be a Gantt chart that already contains a sprint plan that will be carried out. The analysis will also produce use case diagrams, class diagrams, and ERD (Entity Relationship Diagram) as the rough design of the entire application that can be changed along the development process.

We conduct interviews and observations in the planning stage. In the interview process, we communicate directly with people who are responsible for managing the Jaya Sentosa Store. We conducted the interview with several people, e.g., the owner, admin, and staff. The interview will focus on Mr. Alex as the owner to get to know the needs of the store and how it works. We obtain information on writing receipts, ordering goods, processing orders, payment, and recording sales

at Jaya Sentosa Store in the observation process. Observation will focus on the store's admin and staff in carrying out their activities. This observation is done by directly observing Jaya Sentosa Store in their everyday activities. In this planning stage, we design a use case diagram based on the feature list and the initial design of the logical database design.

The next step after the planning stage is the sprint stage. According to Heimrath [13], the sprint stage is a stage where the user requirements from before are analyzed more deeply. In this stage, we undergo more requirement analysis and then proceed to design. The design is based on the results of the analysis. At this stage, we made the design of the back-end and front-end according to the requirement analysis. The design is used as a reference in the development or build stage. In each sprint, we do unit testing on each feature and integration testing afterward. At the end of the sprint stage, we test the application with the user to get their feedback. After reaching user approval, we deploy the application.

The development or build stage in the sprint stage will be divided into 2 (two) tasks, i.e., (i) back-end or server-side and (ii) front-end or client-side. On the back end and front-end, the Javascript (JS) programming language and PostgreSQL database will be used. On the back end of the Jaya Sentosa Store's e-commerce application, the Express JS framework is used. On the front-end of the Jaya Sentosa Store's e-commerce application, the React JS framework is used. In an effort to fulfill the requirements of the Jaya Sentosa Store to search for a product using the product image, the ORB algorithm will be implemented. We implement the ORB algorithm using the Python programming language and the OpenCV library.

The last stage of the Scrum method is the closure stage. According to Heimrath [13], most of the tasks in the closure stage are formalities of the project. At this stage, there will be final testing and deployment. Furthermore, documentation will be submitted, and training will be carried out. This last stage includes the stage of performing maintenance on the application.

2.2 ORB Algorithm

Apart from the development of the whole web application, which uses the Scrum method, this research also utilizes the ORB algorithm. In the Jaya Sentosa Store, customers often come not knowing the name of the product but having the image of the product instead. The usage of the ORB algorithm itself enables image-based search to replicate the process of ordering products in the Jaya Sentosa Store. The image used in the development and implementation of the ORB algorithm will be from the Jaya Sentosa Store product.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

ORB (Oriented FAST and Rotated BRIEF) algorithm is a combination of two other algorithms. Both base algorithms are used because they have good performance at low cost [9]. The ORB algorithm used the Oriented FAST detector to get all the key points or features. The ORB algorithm specifically used FAST-9 (Circular radius of 9), which is stated to have good performance [9]. The FAST-9 detector detects a corner in an image. The FAST-9 works by dividing pixels into darker points, similar points, and lighter points. The FAST keeps the value of the maximum point as the final corner and deletes all neighborhood corners [14]. The FAST method used in the ORB algorithm does not produce a measure of cornerless, the solution implemented in the ORB algorithm is to use the Harris corner measure. The ORB algorithm modified the FAST detector threshold to get more keypoints. The keypoint is filtered using the Harris corner measure by ordering the keypoints and choosing the top keypoint [9].

The next step of the ORB algorithm is to measure the corner orientation by the intensity centroid. The centroid can be obtained by calculating the moment patch. The moments patch is

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \tag{1}$$

To get the centroid (C) from the moment patch as:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right) \tag{2}$$

From the centroid and moment patch operations, we can construct a vector from the center to the centroid. The orientation of the patch can be obtained using:

$$\theta = atan2(m_{01}, m_{10}) \tag{3}$$

The ORB algorithm used steered BRIEF to achieve the rotated BRIEF needed for the algorithm. The BRIEF has a n set of binary location features (x_i, y_i) in the form of a $2 \times n$ matrix.

$$S = \begin{pmatrix} x_1, \ldots, x_n \\ y_1, \ldots, y_n \end{pmatrix} \tag{4}$$

To get the steered version of the binary location features matrix S_θ, the algorithm will use the orientation θ and the rotation matrix R_θ:

$$S_\theta = R_\theta S \tag{5}$$

With all previous operators combined, the steered BRIEF operator became:

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

$$g_n(p,\theta) := f_n(p) \mid (x_i, y_i) \in S_\theta \tag{6}$$

The ORB algorithm processes the image by getting the keypoints using the FAST algorithm, filtering the best keypoints using Harris Corner Measure, calculating orientation, and extracting descriptors using steered BRIEF. Figure 2 shows the flowchart of the ORB algorithm.

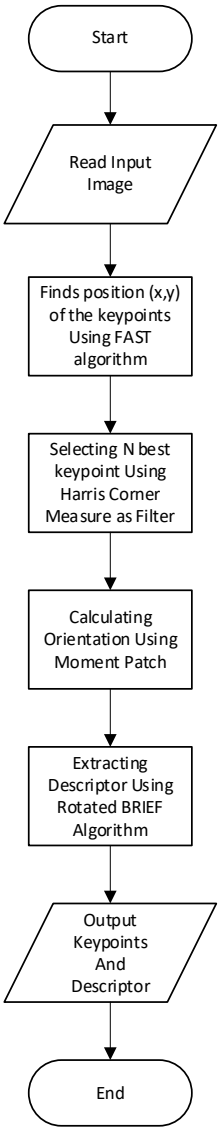

Figure 2. Flowchart of ORB algorithm

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

3. RESULTS AND DISCUSSION

3.1 ORB Algorithm Implementation

At this stage, the development of image-based search is carried out by implementing the ORB algorithm. To utilize the ORB algorithm, we use the OpenCV library in Python. Before fully utilizing the algorithm, several tests have been carried out using the algorithm for searching for goods with image input by calculating the image with the most similarity to the image input. Tests were conducted on the SIFT (Scale Invariant Feature Transform), KAZE, and ORB algorithms. The test was conducted to get the most suitable image from the input of 150 images. The test used an image as input as shown in Figure 3 and produced the same output in all three tests as shown in Figure 4.

Figure 3. Test image input

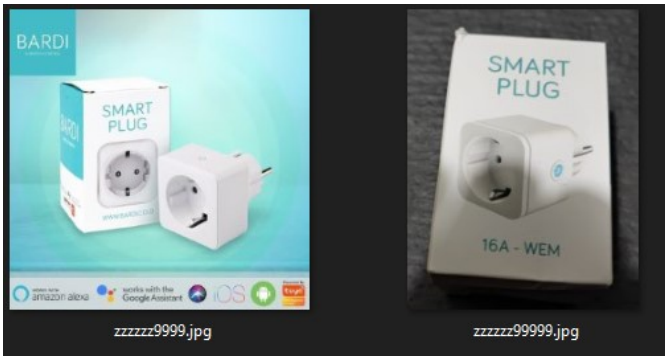

Figure 4. Test image output

All three algorithms' tests are done ten times using different images each time. Table 1 shows the run time performance test results of all three algorithms. In each test, all the algorithms resulted in the same output image result. In testing using the SIFT algorithm on average, it takes 42,539 seconds. The KAZE

algorithm takes 166,332 seconds on average. We also found out that the ORB algorithm takes 3,415 seconds. In testing these three algorithms, it shows that the ORB algorithm is the algorithm that came on top with execution in the shortest time. This execution duration also makes Jaya Sentosa e-commerce use this algorithm.

Table 1. Algorithm run time test

Run Time	Algorithm Run Result (s)		
	ORB	**SIFT**	**KAZE**
1	3,301	43,376	164,817
2	3,377	41,084	162,945
3	3,344	41,513	164,712
4	3,322	41,908	164,466
5	3,963	46,373	178,177
6	3,403	41,267	163,161
7	3,358	41,946	164,579
8	3,362	43,954	168,171
9	3,380	42,798	166,350
10	3,336	41,172	165,940
Average	**3,415**	**42,539**	**166,332**

The implementation of the image-based search feature algorithm which uses the ORB algorithm will be placed on the back end or server side. The user will later use the feature from the front-end or the clientside using the API (Application Programming Interface) endpoint. The back end itself is developed using JavaScript programming language and the ORB algorithm using Python programming language. The difference in the programming language of the algorithm library used and the back end will require another approach. One of the ways to make the feature come to work is to use a Node JS library to run the Python script. In this case, we use the childprocess library to run the Python script.

Figure 6 shows the flow of the image-based feature implemented in our research. The flow of the image-based feature in back end starts with receiving the input image from the front-end using the API POST method that sends files as data. Then, the file that back-end receives is saved temporarily in the server and the third step is sending it to the Python script as a parameter. After running the Python script, it will give an array of image paths as an output. The Python script will run a loop to check the similarity of the input image against the product's images in the server. For every loop action, the first step is to resize the image to a smaller size with a maximum dimension of 1024 according to its aspect ratio. In the process of resizing the image, we use a function from the OpenCV library.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

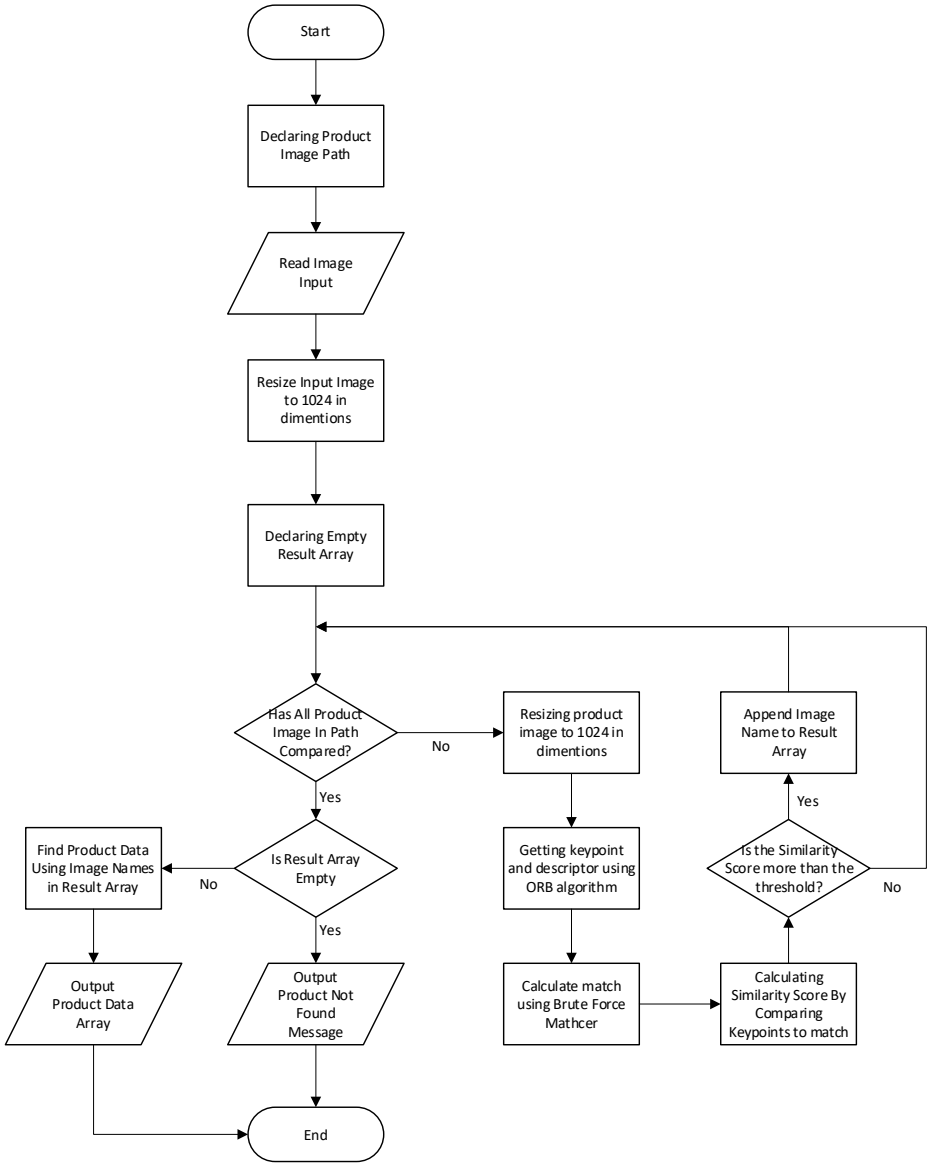

Figure 6. Flowchart of image-based search feature

The second step in the Python script is to compute the ORB algorithm and get the keypoint and descriptor from both the input image and the product image. In the process of getting the similarity of 2 (two) images matche keypoints will be calculated. To calculate the match, the descriptors of both images are used. In the OpenCV library, there is a matcher function, i.e., *knnmatch* that will return *k* amount of best match where *k* is specified. The function will take the query

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

descriptor and train the descriptor as a parameter. The function will return a list of matches and will have 2 variables in each list item. Then we need to loop through the entire list and check the distance of 2 variables to a threshold, if true then add the item to an array. This process will be done twice with descriptor 1 as the query descriptor and descriptor 2 as the train descriptor and vice versa. After getting both best matches to get the final best match or top result, a loop will run to look for the matches that appear in both lists. To check it we just need to check the query and train index of both matches and add the match to the top result if it is true. The top result will be returned as the output. To calculate the similarity the total keypoint in the matches is divided by the smallest total number between keypoint 1 and keypoint 2. The similarity score will determine if the image of the product is similar to the input image, if the similarity score surpasses a certain threshold the product image's path will be appended to the array of results. Lastly, the temporary image file will be deleted, and the array of results will be passed to the backend in JSON format. Figure 6 shows matches between 2 (two) images by using the ORB algorithm. The match keypoints are represented in the lines connecting 2 points in both images.

Figure 6. ORB algorithm matching result

The result from the ORB algorithm will be used as a parameter to find product data from the database. The process to get the product that corresponds to the image is to search the product ID list from the image table and use the "any" operator in the SQL query where condition. The database will search for all product IDs from the image table that have any of the image paths from the ORB result. After getting the list of product IDs, the next process will be simply finding all the product data from the list of product IDs and returning it as an API response to the front-end. Lastly, the front-end will display the result to the end user.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

3.2 Development of web-based sales applications

3.2.1 Planning

At this stage, interviews and direct observations were carried out at Jaya Sentosa Store. The interview will focus on Mr. Alex, the owner of Jaya Sentosa Store. Interviews were conducted regarding how things work in the Jaya Sentosa Store, such as how to receive orders and through what media to receive orders. We also obtained data on payment methods and sales records. In the interview process, supporting data such as receipts, records of goods lists, and some sample images of the goods sold by Jaya Sentosa Store were also obtained. Observation of the buying and selling activities of Jaya Sentosa Store is also carried out to get more data on how Jaya Sentosa Store works. Observations will get data on the stages of people making transactions, such as starting with searching for goods, ordering continuing with payment, and ending with the delivery of goods. The results of these interviews and observations will be developed into a feature list and managed into several diagrams, including use case diagrams, class diagrams, and Gantt charts.

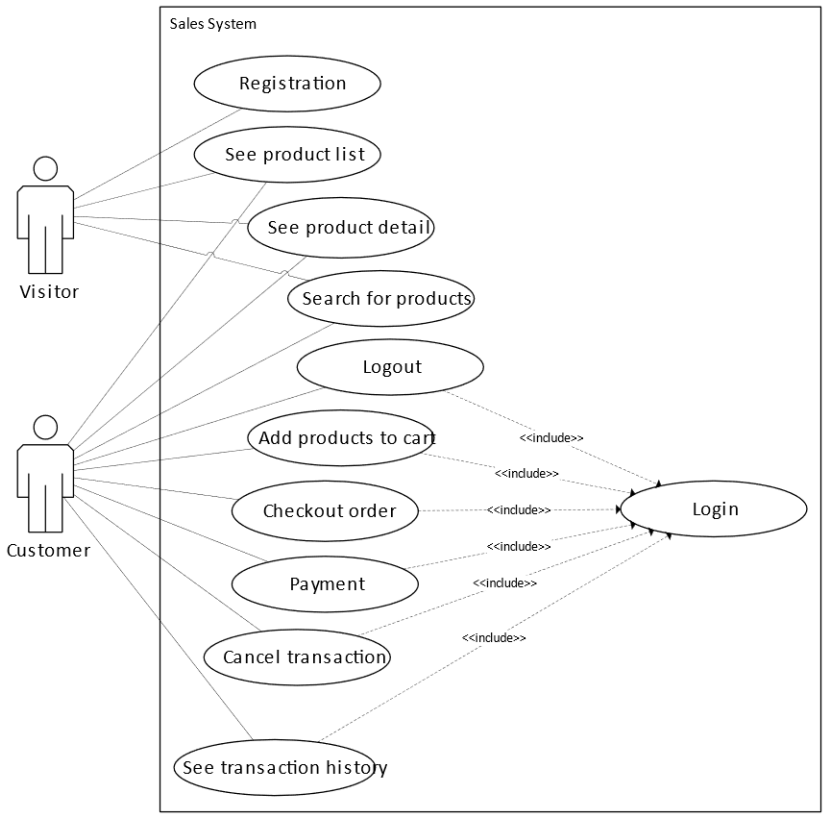

Figure 7. Use case diagram

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

A use case diagram is a diagram used to describe the interaction between external actors who interact with the system. Use cases are scenarios or functions performed by the system. This diagram describes what the system does from the point of view of an external observer [15]. Figure 7 shows the use case diagram of the Jaya Sentosa Store's e-commerce application. The use case diagram focuses on the e-commerce part of the buyer. There are 2 (two) actors in the use case: (i) visitors and (ii) customers. Both actors can search and see the product details. Visitors will be able to register, and then the account can be used to access transaction features in sales applications, such as adding items to the cart, check out, and making payments.

The use case diagram will be a reference in planning application development in the form of a sprint. This plan is in the form of a Gantt chart. Figure 8 shows the summary of the Gantt chart, which is also divided into 3 big groups, i.e., (i) planning, (ii) sprint, and (iii) closure. Figure 9 shows the details of Sprint 1.

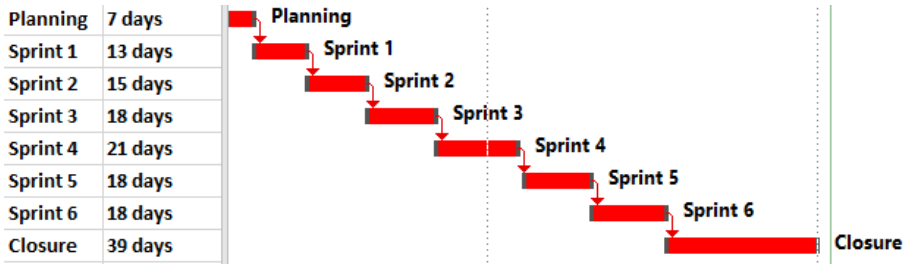

Figure 8. Gantt Chart

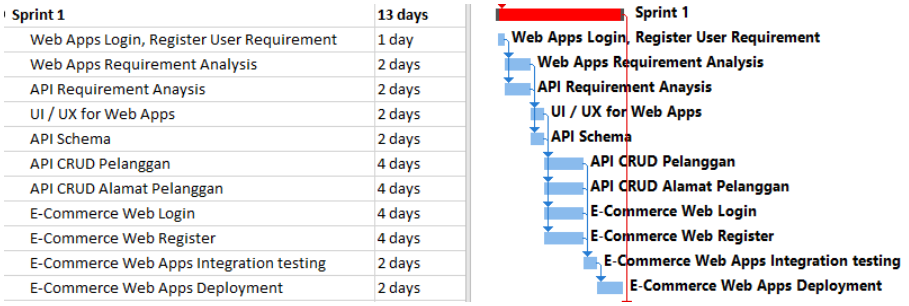

Figure 9. Sprint 1

In the next step, we designed a database to fulfill all the requirements and features. We use the use case diagram and feature list as a requirement reference. The Logical Database Design shows the details of the database. In this design, table normalization has also been carried out, hence there is minimal redundancy. Figure 10 shows the Logical Database Design.

Journal of Information Systems and Informatics
Vol. 6, No. 1, March 2024
p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

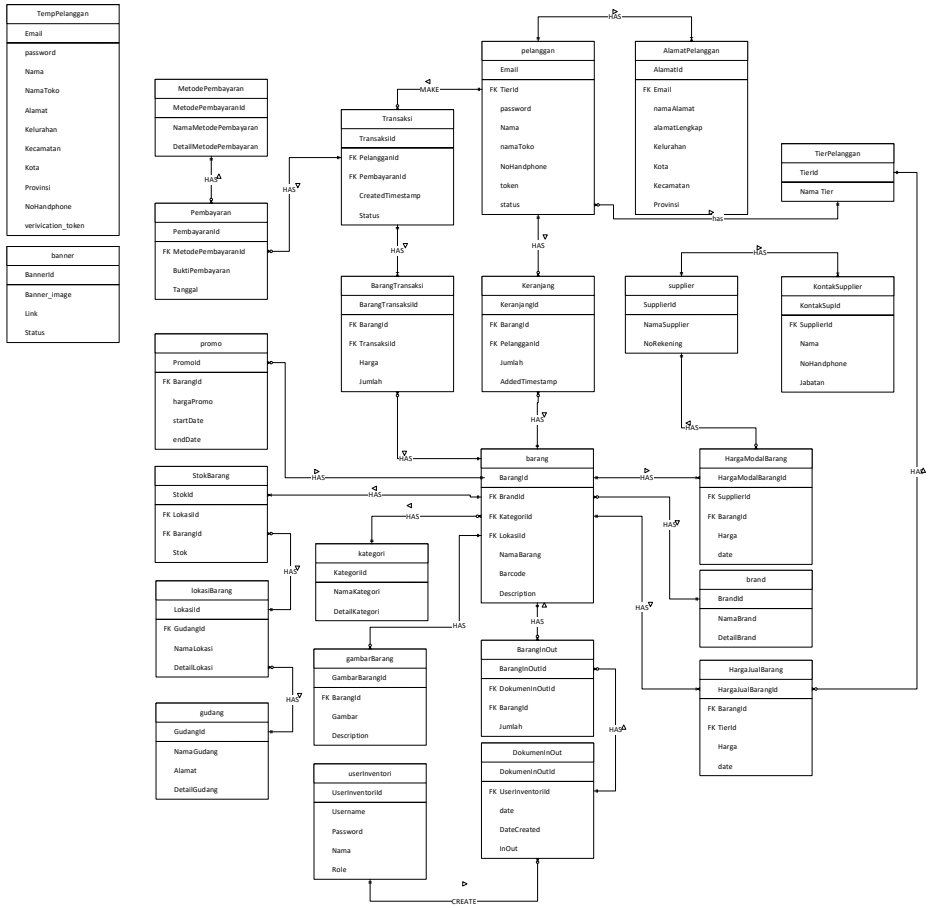

Figure 10. Entity relationship diagram

3.2.2 Sprint Stage

The sprint stage is where we developed the sales application. We do the development as planned in the planning stage. The sprint stage is divided into 4 (four) steps, i.e., (i) requirement analysis, (ii) design, (iii) build or development, and (iv) testing [11]. Each sprint will start by getting user requirements about the features that will be developed in more detail and in-depth. After the requirements are obtained, the design process continues, both front-end UI/UX design and back-end design based on the data needed by the front-end.

The development of the web applications will use the Javascript programming language. On the front-end, the React JS framework will be used. React JS is one of the Javascript libraries used to build user interfaces. It was developed by Facebook (now Meta). As the name implies, react has the ability to react to change

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

according to the state of the property and update the user interface [16]. The web application back-end will use the Express JS framework. Express JS is often considered the standard of the server framework in Node JS and is used to build online applications and API (Application Programming Interface). Express JS handles multiple tasks in the back-end development like routing and handling requests [17].

The build stage, or development stage, starts right after the design of the feature is finished. We develop both back-end and front-end at the same time as we have a design as guidance. Thus, the development can be done simultaneously without waiting for one or another. The front-end development will begin by creating the necessary components and then creating a page and hook to connect it to the server using the API. Figure 11 shows the home page as a result of the development stage.

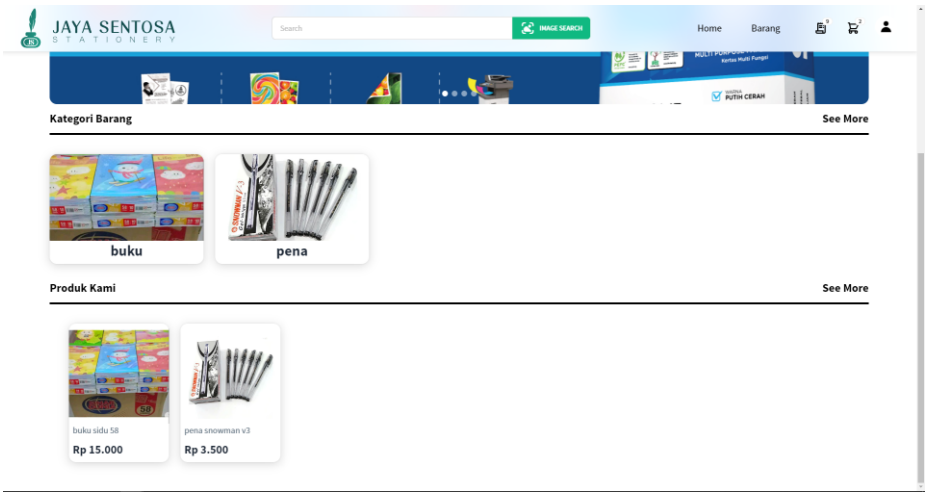

Figure 11. Application home page

The back-end development will start by creating database logic in the form of SQL queries. The next step is to create logic in the form of several services that will be used in the controller. We assign each controller to a specific endpoint. The front-end will access the function using those endpoints. We test every endpoint and request method using Postman. Postman is one of the tools that is usually used to test APIs. Figure 12 shows the results of calling the product endpoint using the Postman tool.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

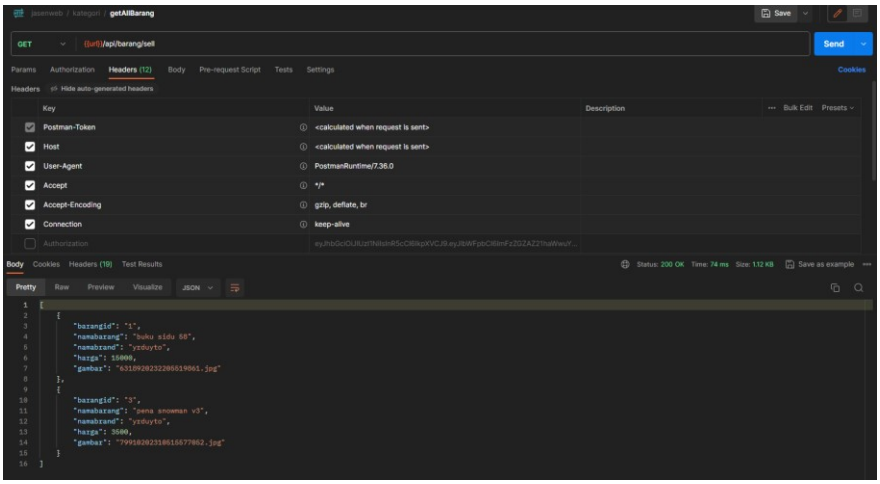

Figure 12. Product Endpoint in Postman

3.2.3 Closure

At the closure stage, final testing is carried out with the Jaya Sentosa shop. Final testing with Jaya Sentosa Store was carried out with black box testing. Black box testing was carried out with the owner of Jaya Sentosa Store (Mr. Alex) on December 22, 2023 in Jakarta. Table 2 shows the result of the black box testing.

Table 2. Black box testing results

Feature	Scenario	Expected Result	Test Result
Registration	Registration	Shows a successful registration message	Valid
	Email Verification	Send an email to the user containing the verification link and display a success message after opening the verification link.	Valid
Login	Login using correct credential	Redirect to home	Valid
	Login using wrong credential	Shows an error message	Valid
Navbar	Display a login button if the user has not logged in yet	Check the login and show the login button if the check fails.	Valid

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

Feature	Scenario	Expected Result	Test Result
	Display transactions, cart, and profile when logged in	Check the login and display the transaction, cart, and profile when the check is successful.	Valid
	Products Searching using keyword	Shows products according to the keyword inserted	Valid
	Image-based product searching	Shows product that has similarity to the input image	
Home	Shows Category	Shows product categories	Valid
	Show Products	Shows Products	Valid
Product	Show all product	Shows all product	Valid
	Shows product's detail	Shows product details like its description, price, and all of the images from the product	Valid
	Add product to cart	Add product to cart and show successful message afterward	Valid
Cart	Show all products in the cart	Show all products that user has added to the cart	Valid
	Update product in cart	Change data of product in the cart, like adding quantity of product or decreasing the quantity	Valid
	checkout	Create a transaction and show it in transaction history	Valid
Transaction	Shows a list of all transaction	shows all of the transactions that a specific user has ever made	Valid
	Shows transaction detail	Shows transaction details like all of the products in the transaction, the price of the products, and the total	Valid

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

Feature	Scenario	Expected Result	Test Result
	Cancel transaction	Cancel the transaction and update the transaction status to canceled	Valid
	Transaction payment	Make new payment and show payment window from payment gateway	Valid

3.3 Discussion

The utilization of the ORB algorithm in image-based search resulted in an application that can replicate the Jaya Sentosa Store ordering process. In the physical store, Jaya Sentosa receives orders from customers saying the product name is already replicated in almost all e-commerce with a search bar that takes a string as an input. The customers of Jaya Sentosa Store also often come not knowing the product but only having the product image or the example of the product itself, to replicate the order process, an image-based search feature is needed. The ORB algorithm in this case can be used to compare the input image from the customer to the product image from the server and output the most similar product to the user input. In the physical store, a customer who does not know the product's name will show the product or product's image to the staff. In our sales application, the customer who doesn't know the product name just needs to input the image that they have. To use the image-based feature, customers need to click image search on the right side of the search bar, and a form input to take an image file will appear. Customers just need to click "Choose file" and upload the product image that they want to search. Figure 13 shows the image-based search input process.

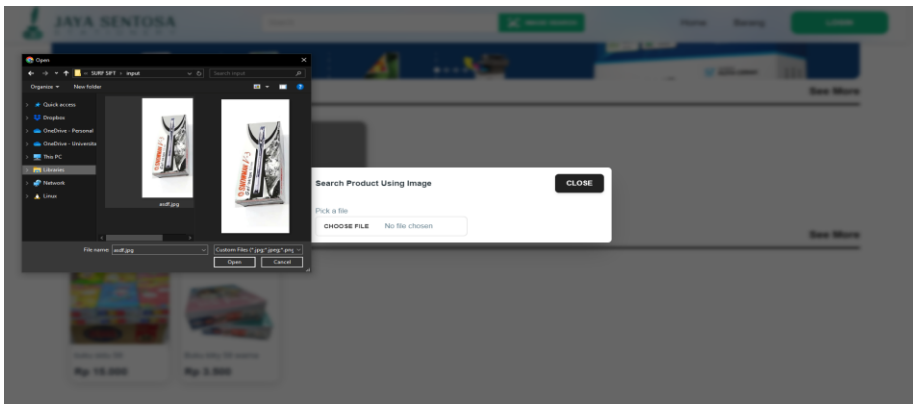

Figure 13. Image-based search input

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

Immediately, after inputting the image without clicking any more buttons, the searching process runs to search for the most similar products on the server. The list of similar products will appear in the same pop up as the form. Figure 14 shows image-based search output. This image-based search feature can replicate the ordering process at the Jaya Sentosa Store. This feature also sets apart this research web-based sales application from the majority of sales apps and gives a competitive advantage to the Jaya Sentosa Store.

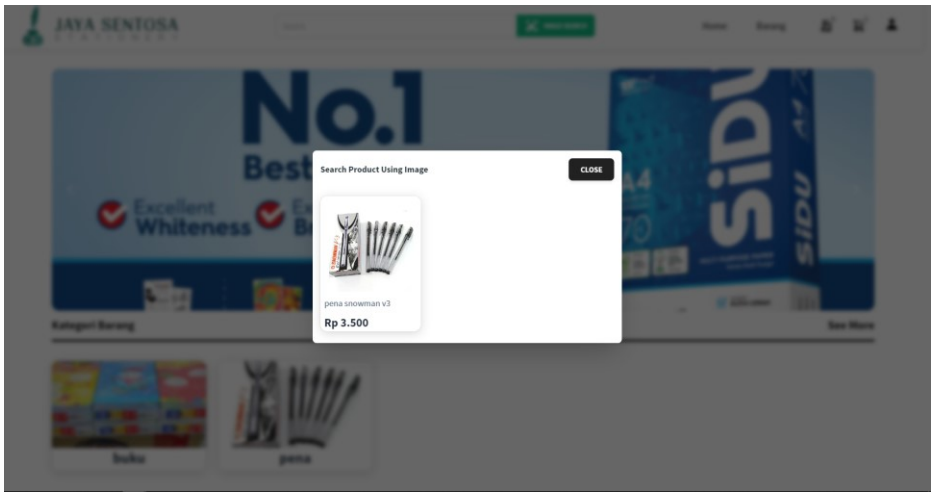

Figure 14. Image-based search output

4. CONCLUSION

In our research, a web-based sales application for the Jaya Sentosa Store was developed. This application employed an image-based feature to replicate the Jaya Sentosa Store order process. This research used the ORB algorithm to implement the image-based feature. The utilization of the ORB algorithm was in accordance with the tests carried out, with the results showing that ORB was the algorithm with the shortest duration. In this test of three feature matching algorithms, SIFT resulted in an average execution time of 42.539 seconds, KAZE in an average of 166.332 seconds, and ORB in 3.415 seconds. The development of this feature used Python scripts and the OpenCV library. This feature aimed to fulfill the needs of the store to handle customers who order goods based on sample items so that they could search for products based on images. This web-based sales application provided a competitive advantage to Jaya Sentosa Store and opened up sales opportunities for a new and larger market share.

REFERENCES

[1] N. Nurlela, "E-Commerce, Solusi di Tengah Pandemi COVID-19," *Jurnal Simki Economic*, vol. 4, no. 1, pp. 47–56, 2021.

[2] M. Ahmadar, P. Perwito, and C. Taufik, "Perancangan Sistem Informasi Penjualan Berbasis Web pada Rahayu Photocopy dengan Database MySQL," *Dharmakarya: Jurnal Aplikasi Ipteks Untuk Masyarakat*, vol. 10, no. 4, pp. 284–289, 2021.

[3] S. Gai, *Ecommerce Reimagined: Retail and Ecommerce in China*. Springer, 2022.

[4] David Brock, *Your Ecommerce Store: Discover How to Get Your Piece of The Multi-Million Dollar eCommerce Pie ...Even If You Have ZERO Online Experience!* Scribl, 2019.

[5] N. H. P. Wijayakusuma, Y. Saintika, and I. Susanto, "Perancangan Website E-commerce Produk Kopi Menggunakan Metode Prototyping (Studi Kasus: Kedai Kopi Kontekstual)," *Journal of Information Systems and Informatics*, vol. 3, no. 3, pp. 471–482, 2021.

[6] S. K. Addagarla and A. Amalanathan, "Probabilistic Unsupervised Machine Learning Approach for a Similar Image Recommender System for E-Commerce," *Symmetry (Basel)*, vol. 12, no. 11, p. 1783, Oct. 2020, doi: 10.3390/sym12111783.

[7] Y. Chen, S. Gong, and L. Bazzani, "Image Search with Text Feedback by Visiolinguistic Attention Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020.

[8] M. Bansal, M. Kumar, and M. Kumar, "2D object recognition: a comparative analysis of SIFT, SURF and ORB feature descriptors," *Multimed Tools Appl*, vol. 80, no. 12, pp. 18839–18857, May 2021, doi: 10.1007/s11042-021-10646-0.

[9] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 International Conference on Computer Vision*, IEEE, Nov. 2011, pp. 2564–2571. doi: 10.1109/ICCV.2011.6126544.

[10] M. Jahangir Alam, T. Chowdhury, and Md. Shahzahan Ali, "A smart login system using face detection and recognition by ORB algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 2, p. 1078, Nov. 2020, doi: 10.11591/ijeecs.v20.i2.pp1078-1087.

[11] A. Stellman and J. Greene, *learning agile: Understanding scrum, XP, lean, and kanban*. "O'Reilly Media, Inc.," 2014.

[12] J. C. Stanley and E. D. Gross, *Project Management Handbook: Simplified Agile, Scrum, and DevOps for Beginners*. Prosper Consulting, Incorporated, 2020.
[13] M. Heimrath, *Agile Project Management: SCRUM For Beginners*. 2023.
[14] H. Wu, W. Yang, and J. Liu, "Image Matching Algorithm for Remote Sensing based on FAST-9 and SURF," in *2015 2nd International Workshop on Materials Engineering and Computer Sciences*, 2015, pp. 331–334.
[15] R. Y. Lee, *Object-Oriented Software Engineering With UML: A Hands-on Approach*. Nova Science Publishers, Incorporated, 2019.
[16] G. Huang, *DYNAMIC TRIO: Building Web Applications with React, Next.js & Tailwind*. 2023.
[17] S. Smith, *Full Stack Web Development Guide: Everything Node JS, Express, APIs, EJS, React JS, Database Fundamentals, SQL Databases*. 2022.