Journal of Applied Informatics and Computing (JAIC)
Vol.8, No.2, December 2024, pp. 499~508
e-ISSN: 2548-6861

Optimization of Tourism Destination Recommendations in Batang Regency Using Content-Based Filtering

Ilmira Yulfihani [1]*, Muhammad Zakariyah [2]**
* Informatics, Universitas Teknologi Yogyakarta
** Informatics, Universitas Teknologi Yogyakarta
ilmirayulfihani1@gmail.com [1], muhammad.zakariyah@staff.uty.ac.id[2]

Article Info

Article history:

Received 2024-10-02
Revised 2024-11-06
Accepted 2024-11-08

Keyword:

Content-Based Filtering,
Mobile Application,
Recommendation System,
Tourism,
Haversine Formula.

ABSTRACT

In an era where tourism plays a pivotal role in economic development, the need for effective navigation through diverse attractions has never been more critical. This research presents a cutting-edge tourism recommendation system tailored for Batang Regency, leveraging Content-Based Filtering (CBF) to deliver personalized suggestions that enhance the tourist experience. By categorizing tourist attractions into Culinary, Culture, Accommodation, Nature, and Leisure, and employing the Haversine formula for precise geographical calculations, our system prioritizes recommendations based on user preferences and proximity. Recommendation testing yielded an impressive average F1 Score of 0.965, underscoring the system's accuracy and relevance, particularly in straightforward user scenarios. However, the research also identifies challenges in more complex cases, suggesting the need for future enhancements through hybrid models and the integration of user feedback. This innovative approach not only streamlines the decision-making process for tourists but also aims to boost local tourism, making it an invaluable tool for both visitors and the Batang Regency community. Join us in exploring how technology can transform the way we experience travel, ensuring that every journey is tailored to individual desires and needs.

I. INTRODUCTION

Tourism has become one of the essential needs of people today. Essentially, tourism is a journey undertaken for pleasure or vacation, accompanied by the planning involved in the trip [1]. Tourism is an important sector in enhancing the economic growth of a region [2]. As one of the country's foreign exchange sources, it holds great potential to drive economic growth. This sector can be developed through improvements in infrastructure, security, and good management to attract both domestic and international tourists with high satisfaction levels [3]. Batang Regency is one of the regions located on the northern coast of Java Island [4]. Despite its coastal location, the area of Batang Regency is vast, extending to highland areas. This geographical diversity results in an abundance of tourism options. The variety of tourism in Batang Regency consists of several types, such as nature tourism, agro-tourism, recreational tourism, religious tourism, cultural heritage tourism, and culinary tourism. However, the general public is still unaware of several aspects of Batang Regency's tourism industry, despite its vast area and significant potential [5]. The diversity of tourism in Batang Regency presents a challenge for tourists in finding destinations they wish to visit. The high level of information diversity creates its own challenge in the process of searching for relevant information [6]. Tourists have many available options but limited time to make a decision about where to visit [7]. Therefore, a recommendation system is needed as a solution to help tourists determine destinations that are relevant to their preferences.

Recommendation systems have become crucial in the era of abundant information, as they assist users in filtering and organizing content in a more personalized manner. These systems are essential in helping consumers find information, products, or services that match their preferences due to the vast amount of data and content accessible online.

Recommendation systems provide relevant suggestions using algorithms and data analysis, making it easier for consumers to navigate the digital world [8].

Various algorithms have been used in recommendation systems, but users often need more personalized solutions [9]. Collaborative Filtering, which relies on data from other users, has been widely applied in several researches. For example, Saifur Rohman Cholil et al. [10] used this method to recommend tourist destinations based on user ratings in Semarang, while Aprilia Sispianygala et al. [11] applied it for tourist spots in Jakarta, achieving a Mean Absolute Error (MAE) of 0.7561 and a Root Mean Square Error (RMSE) of 1.0634. Achmad Sidik et al. [12] employed Item-Based Collaborative Filtering to improve sales ratings in BSD's culinary tourism sector, reaching 83% accuracy with six neighbors. Despite its advantages, Collaborative Filtering faces challenges such as the cold start problem, where insufficient data on new users or items hampers accurate recommendations, and data sparsity, which limits the system's ability to identify relevant patterns. Additionally, user bias and reliance on the number of ratings can reduce accuracy, especially in tourism environments with many items. Due to these limitations, Content-Based Filtering is often more suitable as it provides personalized recommendations based on individual preferences, effectively addressing cold start and data sparsity issues.

Content-Based Filtering (CBF), which generates recommendations based on the attributes of items and user preferences, has been widely applied in several researches. For example, Laili Cahyani et al. [13] used this method to develop a recommendation system for Madura's culinary tourism, achieving a 97.2% accuracy rate using a keyword-based approach and a confusion matrix. Another research, by Alkaff et al. [14] also applied CBF for culinary recommendations, reporting similarly high accuracy rates, further demonstrating CBF's effectiveness in aligning recommendations with user preferences. Lastly, Putri et al. [15] emphasized precision metrics in recommendation systems, highlighting the importance of accuracy in enhancing user satisfaction. The study supports CBF's strength in offering reliable, user-centric recommendations. Despite the advantages of other approaches like Collaborative Filtering, CBF avoids issues like the cold start problem and data sparsity, making it a more effective method for environments with dynamic items, such as tourism.

Based on this background, the implementation of the Content-Based Filtering method is needed to optimize tourist destination recommendations in Batang Regency. This recommendation system is expected to assist local governments in providing information in the form of tourist recommendations and promoting tourist destinations in Batang Regency through a mobile application. This research is expected to make it easier for users to find tourist destinations that match their preferences, thereby indirectly supporting the development of the tourism sector in Batang Regency.

II. METHOD

A. *Recommendation System*

A recommendation system is a software that provides suggestions or support for products to users. It identifies relevant or desired items or products by using inputs given by the users (e.g., preferences or interests) and appropriate algorithms [15]. The goal of a recommendation system is to assist users in making decisions by presenting information that may be of interest to them. This information estimate is personal and is based on the user profile within the system. Most of the time, the user profile is created using the user's rating evaluation [16].

The system leverages information such as past preferences, search history, and other related data to generate suitable recommendations [17]. A recommendation system should be able to predict a user's decision to select an item based on their preferences, interests, user behavior, or the choices of other users. It helps in making objective decisions when faced with a large and complex amount of information [18].

In general, there are two main types of recommendation systems: personalized and non-personalized systems. However, research tends to focus on developing recommendation systems that can offer more personalized and relevant experiences for users, as the demand for recommendations tailored to individual preferences increases [19]. Recommendation systems can be categorized into several types: collaborative filtering, content-based filtering, demographic filtering, and hybrid filtering [13].

B. *Content-Based Filtering*

Content-Based Filtering in recommendation systems is a method that uses a technique called filtering, which examines a user's previous activities to identify behavioral patterns and then suggests items that align with those patterns [20]. In Content-Based Filtering, users receive recommendations based on their own preferences [14]. This method builds a model by analyzing the user's past behavioral preferences, which are then compared to a set of attributes from the recommended items. The items most similar to the user's previous interactions will be those with the highest matching score [18]. The Content-Based Filtering method works by analyzing the similarity of new items to previously rated items [14]. User preferences are aligned with the content or description of the item [13]. In the context of tourism, the Content-Based Filtering method offers advantages by focusing on the content features of tourist destinations, such as category and location, which can increase the relevance of recommendations. These recommendations are selected based on item similarity, do not depend on other users, and do not require user rating information to make recommendations to a specific user [15].

Content-Based Filtering (CBF) is used in this system to analyze user preferences by examining the attributes of tourist attractions in the user's wishlist. The system compares these attractions with other available destinations based on two primary factors: category similarity and geographical proximity. Category similarity is given a higher weight (70%) because it reflects the user's specific interests in categories of tourism, such as nature, cultural, or culinary tourism. Geographical proximity is calculated using the Haversine formula, contributing 30% to the overall similarity score. This ensures that the system not only recommends destinations based on the user’s interests but also suggests attractions that are physically close to their previously selected destinations.

C. System Development

The development of the tourism recommendation system follows a well-structured methodology consisting of system architecture, flow, and design. The system architecture (Figure 1) is designed to facilitate seamless interaction between two primary clients: the mobile application for tourists and a web-based admin interface for administrators, with a centralized server handling all operations. The web admin interface allows administrators to perform Create, Read, Update, and Delete (CRUD) operations on tourism-related data via RESTful API requests. These operations, such as adding or updating tourist destinations, are immediately reflected across the system through real-time data synchronization with the mobile app. The mobile application, developed for tourists, retrieves up-to-date tourism data, such as destination recommendations, categories, and pricing, via API requests. Tourists can also manage their personal preferences, like wishlists, which are synchronized with the backend database to ensure consistency and real-time updates across the system.

The backend server, developed using the Laravel framework, acts as the central hub that processes all incoming requests from both the web admin and mobile app. It manages authentication for users, processes travel recommendations based on user preferences, and handles data synchronization between the clients and the server. This server is responsible for ensuring that any modifications made by administrators are reflected immediately in the mobile app. The MySQL database stores all essential information, including tourist destination details, user preferences, and administrator actions. The database ensures the integrity of the system by managing CRUD operations from both the API and the web admin, making sure the data is always current and accurate.

When administrators make changes through the web admin interface, these trigger API requests to the backend server, which then updates the MySQL database. Upon successful updates, the backend sends API responses back to the web admin, confirming the operations. Similarly, tourists interact with the mobile app, which sends API requests to the backend server to retrieve or update information. The server processes these requests, accesses the relevant data from the database, and sends the response back to the mobile app, ensuring real-time data updates for the end-user.

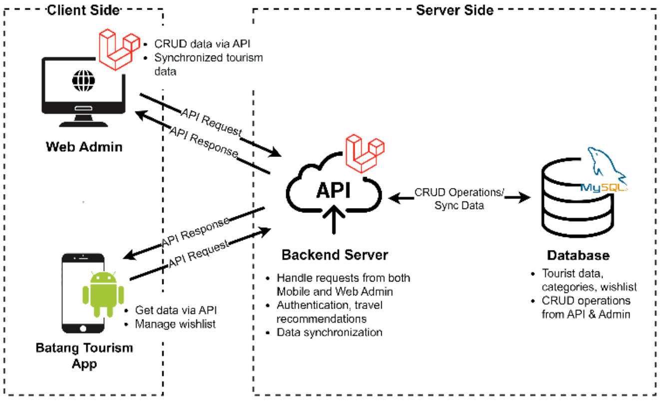

Figure 1. System Architecture

The mobile flow (Figure 2) starts with user authentication, allowing tourists to either log in or create a new account through the mobile application. Upon successful login, the system retrieves the user’s preferences from the database and generates personalized recommendations. These recommendations are based on the user's wishlist and take into account both the category similarity and geographical proximity of tourist attractions. Tourists can then add or remove destinations from their wishlist, which dynamically updates future recommendations. The admin interface enables administrators to manage tourism data, ensuring that tourist destinations, categories, and other relevant data are up-to-date, which directly impacts the recommendations provided to users.

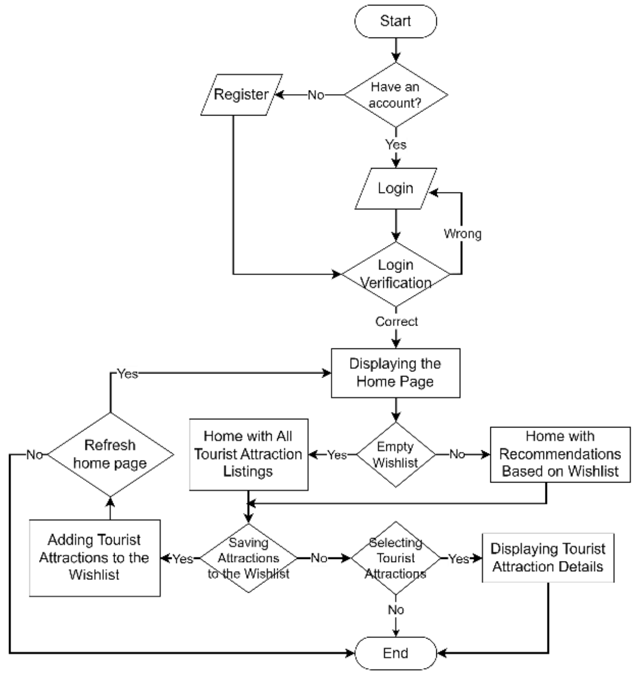

Figure 2. Application Mobile Flowchart

The Entity-Relationship Diagram (ERD) (Figure 3) provides a clear representation of the database schema used in the tourism recommendation system, which is managed through MySQL. The key entities within the system include Tourist (Wisatawan), Wishlist, Tourist Destination (Objek Wisata), Admin, and Tourism Category (Kategori Wisata). These entities are linked through relationships that define how data is organized and managed across the platform.

The Tourist entity stores user information, tourist attraction entity holds detailed information about each destination. Administrators, managed via the Admin entity, are responsible for data management operations within the system. The Tourism Category entity contains the classification of destinations, with each category being identified by id_kategori and a descriptive name (nama_kategori). This categorization allows for more efficient data filtering and enhanced user experiences within the mobile application.

This relational model within MySQL facilitates the efficient management of both tourist and destination data, ensuring quick access and updates. The interaction between Wishlist and Tourist Destination entities enhances the recommendation system, allowing the platform to provide tailored suggestions based on tourist preferences. Admins ensure that destination data is always up to date, and the use of MySQL optimizes the querying process, particularly for recommendation generation and user data management.

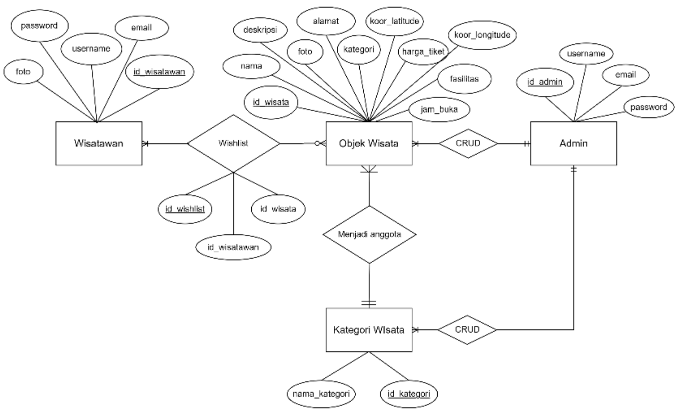

Figure 3. System ERD

The description of the database structure of each table is as follows:

TABLE I
ADMIN TABLE STRUCTURE

Attribute	Data type	Description
id_admin	INT	Primary key, unique ID for each user
nama	VARCHAR	User name
email	VARCHAR	User email address
kata_sandi	VARCHAR	User password (stored in hash format)

TABLE II
TOURIST TABLE STRUCTURE

Attribute	Data type	Description
id_pengguna	INT	unique ID for each user
username	VARCHAR	User name
email	VARCHAR	User email address
kata_sandi	VARCHAR	User password (stored in hash format)
foto	VARCHAR	URL or path to user photo

TABLE III
ATTRACTION TABLE STRUCTURE

Attribute	Data type	Description
id_objek_wisata	INT	Unique ID for each tourist attraction
nama_objek_wisata	VARCHAR	Name of tourist attraction
deskripsi	VARCHAR	Full description of tourist attraction
foto	VARCHAR	URL or path to tourist attraction photo
alamat	VARCHAR	Full address of tourist attraction
koordinat_latitude	VARCHAR	Latitude coordinates tourist attraction
koordinat_longitude	VARCHAR	Longitude of tourist attraction
kategori_wisata	VARCHAR	Type of tourism (nature, culture, history, religion, etc.)
harga_tiket	VARCHAR	Entrance ticket price of tourist attraction (can be a price range)
jam_buka	VARCHAR	Opening hours of tourist attraction
fasilitas	VARCHAR	Facilities available at tourist attraction

TABLE IV
CATEGORY TABLE STRUCTURE

Attribute	Data type	Description
id_kategori	INT	rimary key, unique ID for each tourist category
nama_kategori	VARCHAR	Name of tourist category

TABLE V
WISHLIST TABLE STRUCTURE

Attribute	Data type	Description
id_wishlist	INT	ID of each wishlist
id_wisatawan	INT	ID of the tourist who added
id_wisata	INT	ID of the tourist object that was added

D. System Implementation

Following the design phase, the system implementation was executed through several key components, starting with the web admin interface. This interface, developed using the Laravel framework, enables administrators to manage critical data such as tourist categories, attractions, and user wishlists. The admin panel supports CRUD operations, allowing authorized personnel to add, update, or delete tourist information. The web admin ensures data integrity and real-time synchronization with the backend, which is crucial for the accuracy of the recommendation system.

The backend implementation integrates all data managed via the web admin, allowing seamless interaction between the mobile app and the server through API endpoints. Developed using Laravel and PHP, the backend handles various functionalities, including tourist authentication, wishlist management, and real-time recommendation generation.

To determine recommendations, this system collects tourist data selected by users through the wishlist feature and calculates similarities with other tourist attractions based on category features and geographical distance. The recommendation system employs a content-based filtering algorithm that suggests tourist attractions based on user preferences. It begins by collecting user preference data from their wishlist and calculating similarity scores between tourist attractions based on two key factors: category similarity and location proximity.

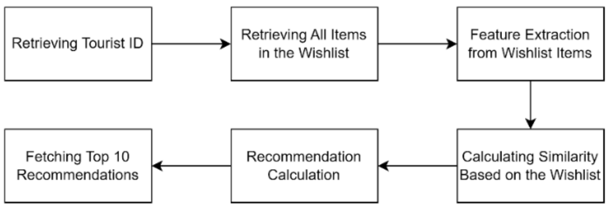

Figure 4. Tourist Attraction Recommendation Determination Process

The recommendation system uses a content-based filtering algorithm to suggest tourist attractions that align with user preferences. This system gathers user preference data primarily from their wishlist and computes similarity between tourist attractions based on two key factors: category similarity and geographical proximity. Category similarity is assigned a weight of 70%, making it the dominant factor in the recommendation process. This is determined by comparing the categories of tourist attractions in the user's wishlist with those of other available attractions. When a category match is identified, the system assigns the maximum similarity score to the respective tourist attraction.

In addition to category similarity, the geographical proximity between the user and tourist attractions is calculated using the Haversine formula. The Haversine formula, widely recognized for its accuracy in calculating distances between two points on the Earth's surface [21], is particularly suited for applications in tourism recommendation systems that involve geospatial data. It computes the arc distance between two locations using their longitude and latitude coordinates. The formula is critical in ensuring that the recommendations are not only relevant to user preferences but also practical in terms of accessibility.

In the algorithm, the geographical distance, expressed in kilometers, is incorporated as a secondary factor (30% weight) to refine the recommendations further. Tourist attractions that are closer to the user are ranked higher, provided they also meet the user's category preferences. The Haversine formula, which enhances the precision of distance calculations, can be represented by the following equation [22]:

$$\Delta latitude = latitude2 - latitude1 \qquad (1)$$
$$\Delta longitude = longitude2 - longitude1$$
$$distance = 2.R.arcsin\left(\sqrt{sin^2\left(\frac{\Delta latitude}{2}\right) + \cos(latitude2).\cos(latitude1).sin^2\left(\frac{\Delta longitude}{2}\right)}\right)$$

This formula is implemented as follows:

```
private function calculateHaversine($lat1,
$lon1, $lat2, $lon2)
{$earthRadius = 6371;

$dLat = deg2rad($lat2 - $lat1);
$dLon = deg2rad($lon2 - $lon1);

$a = sin($dLat / 2) * sin($dLat / 2) +
cos(deg2rad($lat1)) * cos(deg2rad($lat2)) *
sin($dLon / 2) * sin($dLon / 2);

$c = 2 * atan2(sqrt($a), sqrt(1 - $a));

return $earthRadius * $c;}
```

Figure 5. Haversine Formula Code

After calculating the similarity for category and location, the system combines these two values using formula (2):

$$totalSimilarity = (categorySimilatiry \times 0{,}7) + (locationSimilarity \times 0{,}3) \qquad (2)$$

The formula for combining these two values is applied as follows:

```
$totalSimilarity = $categorySimilarity * 0.7
+ $locationSimilarity * 0.3;
return $totalSimilarity;
```

Figure 6. Weighting Combination Code

The final similarity score is calculated by combining these two factors, prioritizing category similarity. The system then ranks the tourist attractions based on their total similarity score and provides the user with the top 10 most relevant recommendations, as shown in the following code:

```
$topRecommendations = $recommendations-
>take(10);
return $topRecommendations;
```

Figure 7. Recommendation Result Sorting Code

This method ensures that users receive personalized suggestions that align with their interests, while also considering the distance to the attractions. The algorithm is implemented in API Recommendations controller.

The mobile application, built using Android Studio and Kotlin, offers an interactive platform for tourists. It supports several key features such as user authentication, wishlist management, and personalized recommendations. Through API integration, the app retrieves data in real-time, allowing users to explore and filter tourist attractions based on categories and proximity. The API integration is facilitated by Retrofit, which handles requests to the server and ensures that data, including tourist attraction names, photos, and addresses, are properly displayed in the UI through RecyclerView adapters.

E. System Testing

To evaluate the effectiveness of the recommendation system, a comprehensive testing approach was employed, focusing on the performance of the Content-Based Filtering (CBF) algorithm. The testing involved multiple scenarios designed to assess how effectively the system generates relevant recommendations based on user inputs. Specifically, these scenarios examined various conditions for item placement in the wishlist:

- *The first scenario:* involved a single item oh wishlist,
- *The second scenario:* included two items from different categories within the same location,
- *The third scenario:* consisted of two items from the same category but in different locations, and
- *The final scenario:* incorporated a mix of categories and locations, with each containing three items.

The testing used a sample size of 10 trials, involving 32 tourist attractions categorized into 7 different categories across various regions in Batang Regency. In each instance, the system's output was compared with expected results to ensure that the recommendations met user preferences. Additionally, manual calculations were performed to validate the accuracy of the CBF algorithm’s recommendation scores. By comparing these manual results with the system-generated scores, the reliability and precision of the recommendation system were assessed. To evaluate the effectiveness of the recommendation system, the F1 Score is used as an evaluation metric. The F1 Score is the harmonic mean of precision and recall, providing a balanced measure between the relevance (precision) and the completeness (recall) of the recommendations produced by the system. Precision measures how many of the recommended items are relevant. It is calculated as:

$$Precision = \frac{Total\ number\ of\ recommendations}{Number\ of\ relevant\ items\ in\ recommendations} \quad (3)$$

Recall measures how many relevant items from the wishlist were successfully recommended. It is calculated as:

$$Recall = \frac{Total\ number\ of\ relevant\ items\ in\ the\ wishlist}{Number\ of\ relevant\ items\ in\ recommendations} \quad (4)$$

The F1 Score is the harmonic mean of precision and recall, calculated as:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

III. RESULT AND DISCUSSION

A. Mobile Implementation Results

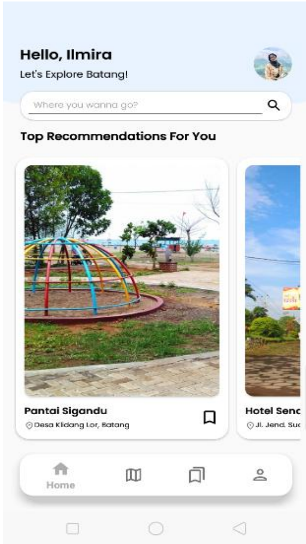

Figure 8. Tourist Home Page

Figure 8. shows the home page for tourists in the mobile recommendation system application. This page contains the results of tourist recommendations based on calculations using content-based filtering.

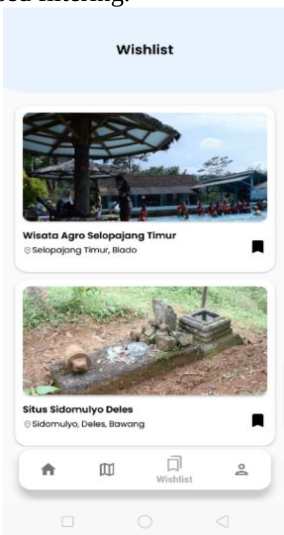

Figure 9. Tourist Wishlist Page

Figure 9. shows the tourist wishlist page in the mobile recommendation system application. This wishlist page contains tourist attractions that serve as a reference for the recommendation system calculations.

B. Recommendation Testing Results

The testing results, based on various wishlist scenarios, are presented in Table VI, showcasing the top three recommendation scores derived from the system's similarity calculation.

TABLE VI
RESULT OF RECOMMENDATION PAGE

Test Case	Wishlist Item(s)	Category	Location	Top Recommendations					
				Rank	Tourist Attraction	Category	Location	Score	Relevance
1	Agrowisata Pagilaran	Agrotourism	Blado	1	Wisata Agro Selopajang Timur	Agrotourism	Blado	0.7519	Same category and location
				2	Teh Kampoeng	Culinary	Blado	0.0973	Same location
				3	Makam Auliya Wonobodro	Culture	Blado	0.0767	Same location
2	Pantai Celong	Nature	Banyuputih	1	Pantai Jodo	Nature	Gringsing	0.7358	Same category
				2	Pantai Ujungnegoro	Nature	Batang	0.7185	Same category
				3	Pantai Sigandu	Nature	Batang	0.7140	Same category
3	Bandar Ecopark	Leisure	Bandar	1	Tubing Pandansari	Leisure	Bandar	0.7250	Same category and location
				2	THR Kramat	Leisure	Batang	0.7194	Same category
				3	Lomba Dayung Tradisional	Culture	Batang	0.7168	Near location
4	Serabi Kalibeluk	Culinary	Warungasem	1	Nasi Megono	Culinary	Batang	0.7437	Same category
				2	Lontong Lemprak	Culinary	Batang	0.7437	Same category
				3	Teh Kampoeng	Culinary	Blado	0.7107	Same category
5	Prasasti Sojomerto	Culture	Reban	1	Situs Sibebek	Culture	Bawang	0.7522	Same category
				2	Situs Pejanten	Culture	Tersono	0.7427	Same category
				3	Batu Gamelan	Culture	Bandar	0.7348	Same category
6	Hotel Sendangsari	Accomodation	Batang	1	Hotel Yudhistira	Accomodation	Batang	0.8248	Same category and location
				2	Batik Rizky	Culture	Batang	0.2317	Same location
				3	Nasi Megono	Culinary	Batang	0.1872	Same location
7	Pantai Sigandu, Safari Beach Batang	Nature, Leisure	Batang	1	Lomba Dayung Tradisional	Culture	Batang	0.4502	Same location
				2	Pantai Ujungnegoro	Nature	Batang	0.4005	Same category and location
				3	THR Kramat	Leisure	Batang	0.3980	Same category and location
8	Bandar EcoPark, Situs Balekambang	Leisure, Culture	Bandar, Gringsing	1	Batu Gamelan	Culture	Bandar	0.3916	Same category and location
				2	Situs Kepokoh	Culture	Blado	0.3800	Same category
				3	Makam Auliya Wonobodro	Culture	Blado	0.3738	Same category and location
9	Pantai Celong, Prasasti Sojomerto, Lontong Lemprak	Nature, Culture, Culinary	Banyuputih, Reban, Batang	1	Nasi Megono	Culture	Batang	0.3416	Same category and location
				2	Penjara Kolonial Belanda	Culture	Batang	0.3322	Same category and location
				3	Stasiun Batang	Culture	Batang	0.3187	Same category and location
10	Tubing Pandansari, Batik Rizky, Serabi Kalibeluk, Hotel Yudhistira	Leisure, Culture, Culinary, Accomodation	Bandar, Batang, Warungasem, Batang	1	Hotel Sendang Sari	Accomodation	Batang	0.2837	Same category and location
				2	Nasi Megono	Culinary	Batang	0.2764	Same category and location
				3	Lontong Lemprak	Culinary	Batang	0.2764	Same category and location

Journal of Applied Informatics and Computing (JAIC)
Vol.8, No.2, December 2024, pp. 499~508
e-ISSN: 2548-6861

The results in Table VI display the top three recommendations generated for various wishlist items, categorized by type and location. Each test case focuses on a different tourist attraction in Batang Regency, with recommendations ranked by their similarity scores. The system generally prioritizes attractions that share the same category and location as the wishlist item. For example, in Test 1, the recommendation for "Agrowisata Pagilaran" ranks "Wisata Agro Selopajang Timur" at the top of the list, with a similarity score of 0.7519, due to its matching category (Agrotourism) and location (Blado). In other tests, such as Test 2 with "Pantai Celong," the top recommendations share the same category (Nature), with "Pantai Ujungnegoro" achieving a similarity score of 0.7185. The notes section further highlights the relevance of these recommendations, indicating whether they match the wishlist item in terms of both category and location or just one of these criteria.

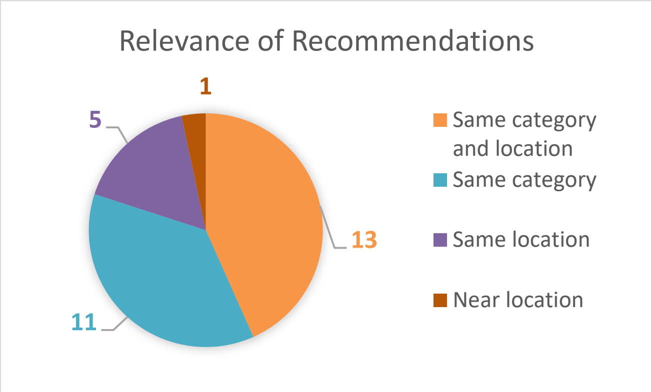

Figure 10. Relevance of Recommendations Pie Chart

We calculate the precision, recall, and F1 Score for each test case based on the number of wishlist items and the relevance of the recommendations. The results are summarized in Table VII, which displays the precision, recall, and F1 Score for each test case. It is important to note that a recommended item will be considered relevant only if it matches the same category and location, the same category, or the same location; otherwise, it will not be deemed relevant.

TABLE VII
RESULT OF EVALUATION METRIC

Test Case	Number of wishlist item	Precision	Recall	F1 Score
1	1	$\frac{3}{3} = 1$	$\frac{1}{1} = 1$	$2 \times \frac{1 \times 1}{1+1} = 1$
2	1	$\frac{3}{3} = 1$	$\frac{1}{1} = 1$	$2 \times \frac{1 \times 1}{1+1} = 1$
3	1	$\frac{2}{3} = 0.67$	$\frac{1}{1} = 1$	$2 \times \frac{0.67 \times 1}{0.67+1} = 0.8$
4	1	$\frac{3}{3} = 1$	$\frac{1}{1} = 1$	$2 \times \frac{1 \times 1}{1+1} = 1$
5	1	$\frac{3}{3} = 1$	$\frac{1}{1} = 1$	$2 \times \frac{1 \times 1}{1+1} = 1$
6	1	$\frac{3}{3} = 1$	$\frac{1}{1} = 1$	$2 \times \frac{1 \times 1}{1+1} = 1$
7	2	$\frac{3}{3} = 1$	$\frac{2}{2} = 1$	$2 \times \frac{1 \times 1}{1+1} = 1$
8	2	$\frac{3}{3} = 1$	$\frac{2}{2} = 1$	$2 \times \frac{1 \times 1}{1+1} = 1$
9	3	$\frac{3}{3} = 1$	$\frac{3}{3} = 1$	$2 \times \frac{1 \times 1}{1+1} = 1$
10	4	$\frac{3}{3} = 1$	$\frac{3}{4} = 0{,}75$	$2 \times \frac{1 \times 0{,}75}{1+0{,}75} = 0{,}85$
Average of F1 Score				$\frac{9{,}65}{10} = 0{,}965$

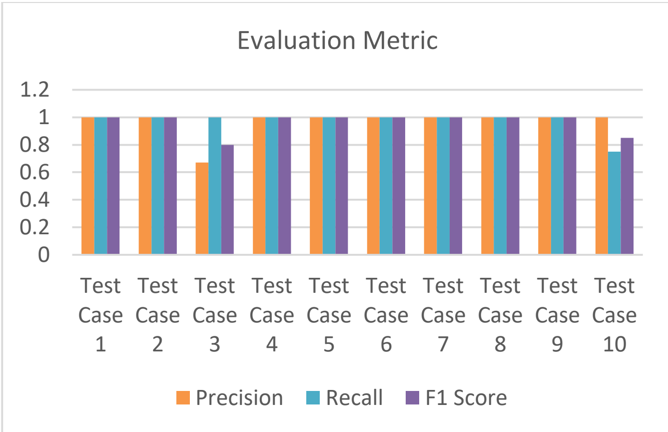

Figure 11. Evaluation Metric Chart

C. Testing of Content-Based Filtering

In this stage, a comparison is made between the scores obtained from manual calculations using the Haversine formula and the weighting recommendations, alongside the scores generated by the tourism recommendation system. This testing aims to ensure the accuracy of the algorithm in determining the distance between different tourist locations based on geographic coordinate data. The implementation of this calculation will be applied to a case study of two tourist attractions: Agrowisata Pagilaran as the item stored in the Wishlist and Agrowisata Selopajang Timur as one of the recommended attractions, to test the results provided by the system.

Next, the manual calculation of the scores between these two tourist attractions will be presented, followed by the results obtained through the recommendation system. The process of calculating recommendation scores is as follows:

1) Coordinates Data for the Case Study: The coordinates data for the two tourist attractions in the case study is presented in the following table:

TABLE VIII
CASE STUDY TOURIST ATTRACTION DATA

Name of Tourist Attraction	Category	Latitude	Longitude
Agrowisata Pagilaran	Agrotourism	−7.1105930	109.8549540
Agrowisata Selopajang Timur	Agrotourism	−7.0676910	109.8545620

2) Calculation of Category Similarity: If Category1 = Category2, then Similarity = 1

3) Calculation of Latitude and Longitude Differences:

$$\Delta lat = lat2 - lat1$$
$$\Delta lat = 7.0676910 - (-7.1105930)$$
$$\Delta lat = 0.04290$$
$$\Delta long = long2 - long1$$
$$\Delta long = 109.8545620 - 109.8549540$$
$$\Delta long = -0.000392$$

4) Convert Delta Latitude and Delta Longitude to Radians:

$$\Delta lat\ (rad) = deg2rad(0.042902)$$
$$\Delta lat\ (rad) = 0.0007486 rad$$
$$\Delta long\ (rad) = deg2rad(-0.000392)$$
$$\Delta long\ (rad) = -6.842 \times 10^{-6}\ rad$$

5) Calculations Using the Haversine Formula: The calculation process consists of three main stages: calculating a, c, and distance. The first step in the Haversine formula is to compute component aaa, which uses the changes (delta) in latitude and longitude between the two points.

$$a = sin^2\left(\frac{\Delta lat}{2}\right) + cos(lat1) \times cos(lat2) \times sin^2\left(\frac{\Delta long}{2}\right) \quad (3)$$
$$a = sin^2(0.0003743) + cos(deg2rad(-7.1105930)) \times cos(deg2rad(-7.0676910)) \times sin^2(-3.421 \times 10 - 6)$$
$$a = 1.400 \times 10 - 7 + (0.9921 \times 0.9922 \times 1.17 \times 10 - 11)$$
$$a = 1.400 \times 10 - 7 + 1.151 \times 10 - 11$$
$$a \approx 1.4001 \times 10 - 7$$

After obtaining the value of aaa, the next step is to calculate component c. Component c is known as the angular distance or central angle, representing the distance in radians between the two points on the surface of the Earth.

$$c = 2 \times atan2\ (\sqrt{a}, \sqrt{(1-a)})$$
$$c = 2 \times atan2(\sqrt{(1.4001 \times 10 - 7)}, \sqrt{(1 - 1.4001 \times 10 - 7)})$$
$$c = 2 \times atan2(0.0003743, 0.9999999)$$
$$c \approx 0.0007486$$

The final step is to convert the angular distance value (c) into a linear distance, such as kilometers or miles. To do this, we multiply the value of c by the radius of the Earth (commonly considered to be 6371 km).

$$distance = R \times 0.0007486$$
$$distance = 6371 \times 0.0007486$$
$$distance \approx 4.77 km$$

6) Calculation of Location Similarity:

$$Location\ Similarity = \frac{1}{1 + 4.77}$$
$$Location\ Similarity = 0.1736$$

7) Total Similarity Score Calculation:

$$Total\ Similarity = (1 \times 0.7) + (0.1736 \times 0.3)$$
$$Total\ Similarity = 0.7 + 0.052$$
$$Total\ Similarity = 0.7521$$

After completing the manual score calculation using the Haversine formula and weighting, the next step is to test the implementation of the algorithm within the system. This testing is conducted using the Postman tool to access the developed tourism recommendation API. The results of this testing will be compared with the manual calculation results to ensure the alignment between the algorithms applied in the system and the theory used.

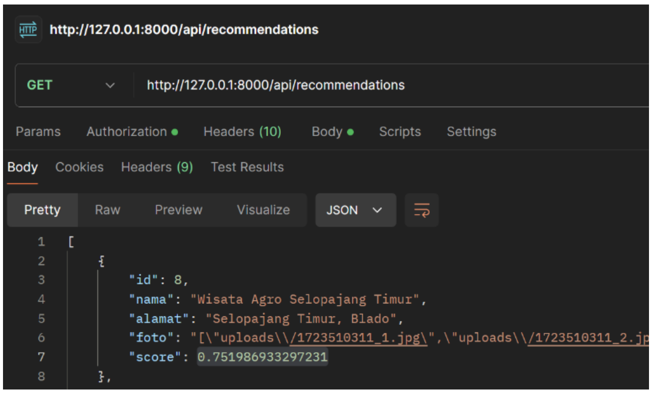

Figure 12. Postman Testing Results on the Recommendation API

From the results of the system testing using the Haversine formula, a recommendation score of 0.751986933297231 was obtained. This result was then compared with the manual calculation, which yielded a value of 0.7521. The very small difference between the two results, which is 0.0001, indicates that the algorithm implemented in the program is consistent with the manual calculation. Thus, it can be concluded that the implementation of the Haversine formula or algorithm in the system has been executed accurately and is consistent with the theoretical calculation method.

IV. CONCLUSION

. Based on the results and discussions presented earlier, it can be concluded that the recommendation system for tourism destinations in Batang Regency demonstrates high accuracy and relevance, particularly in scenarios with simpler user preferences, achieving an average F1 Score of 0.965 across

10 test cases. The system effectively matches user preferences with available tourist attractions, showcasing its strength in delivering personalized recommendations. However, its performance declines slightly when handling more complex cases involving mixed categories and locations. Furthermore, the limited sample size and number of categories may not fully reflect the diversity of potential user preferences. Future research should focus on expanding the dataset and integrating hybrid recommendation models that combine Content-Based Filtering with collaborative filtering, to enhance flexibility and accuracy. Additionally, incorporating user feedback mechanisms could further improve the recommendations, increasing user satisfaction and engagement in the tourism sector.

REFERENCES

[1] E. Puspita, E. Putri, and H. Februariyanti, "Sistem Rekomendasi Tempat Wisata Kota Padang Dengan Haversine," in *Proceeding SENDIU*, 2020.

[2] Y. I. Lubis, D. J. Napitupulu, and A. S. Dharma, "Implementation of Hybrid Filtering (Collaborative and Content-based) Methods for the Tourism Recommendation System," in *12th Conf. Inf. Technol. Electr. Eng.*, 2020, pp. 6–8.

[3] A. N. Millatina, F. Hakimi, I. Zaki, and I. Yuningsih, "Peran Pemerintah Untuk Menumbuhkan Potensi Pembangunan Pariwisata Halal Di Indonesia," *Jurnal Manajemen Dan Bisnis Indonesia*, vol. 5, no. 1, pp. 96–109, 2019.

[4] M. Nurulhuda and Alamsyah, "Pengembangan Tata Ruang dan Pengaruhnya Terhadap Pendapatan Asli Daerah Pariwisata Kabupaten Batang: 2011-2019," *Sabda : Jurnal Kajian Kebudayaan*, vol. 16, no. 2, pp. 88–96, Dec. 2021, Accessed: Sep. 03, 2024. [Online]. Available: https://ejournal.undip.ac.id/index.php/sabda/article/view/62566/25598

[5] M. N. Zulmi, I. Nugroho, and D. D. Salsabila, "Reaktualisasi Citra Wisata Kabupaten Batang Melalui Pengembangan Pop-Up Visit Batang Heaven Of Asia Sebagai Media Edumowisata (Edukasi Promosi Wisata) Kabupaten Batang," *RISTEK: Jurnal Riset, Inovasi dan Teknologi*, 2019.

[6] A. Akbarul Huda and L. Dwi Farida, "Kajian Sistem Rekomendasi Pada Keanekaragaman Podcast," *Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi*, 2021.

[7] R. M. Ariyani, V. Dhameria, and S. Suyitno, "Pengaruh Daya Tarik Wisata dan Promosi Media Sosial terhadap Minat Berkunjung Kembali Wisatawan di Waduk Setu Patok Kabupaten Cirebon," *JPEKA: Jurnal Pendidikan Ekonomi, Manajemen dan Keuangan*, vol. 6, no. 2, pp. 133–147, Nov. 2022, doi: 10.26740/jpeka.v6n2.p133-147.

[8] G. R. Putra, "Sistem Rekomendasi Pemilihan Smartphone Gaming Menggunakan Metode Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS)," *Jurnal Ilmiah Computer Science*, vol. 1, no. 1, pp. 41–48, Jul. 2022, doi: 10.58602/jics.v1i1.5.

[9] P. S. Pratiwi, "Perancangan Sistem Rekomendasi Berbasis Model Ontologi Untuk Rekomendasi Tempat Magang Mahasiswa," *Syntax Literate: Jurnal Ilmiah Indonesia*, vol. 7, no. 6, 2022, Accessed: Sep. 03, 2024. [Online]. Available: https://doi.org/10.36418/syntaxliterate.v7i6.7498

[10] S. R. Cholil, N. A. Rizki, and T. F. Hanifah, "Sistem Rekomendasi Tempat Wisata Di Kota Semarang Menggunakan Metode Collaborative Filtering," *JIKO (Jurnal Informatika dan Komputer)*, vol. 7, no. 1, p. 118, Feb. 2023, doi: 10.26798/jiko.v7i1.727.

[11] A. Sispianygala, S. Sandino Berutu, and J. Jatmitka, "Pengembangan Aplikasi Sistem Rekomendasi Tempat Wisata Dengan Collaborative Filtering," *Progresif: Jurnal Ilmiah Komputer*, vol. 20, pp. 828–838, Aug. 2024, doi: http://dx.doi.org/10.35889/progresif.v20i2.2044.

[12] A. Sidik, N. Nurmaesah, and Y. Carsoyo, "Sistem Rekomendasi Wisata Kuliner di BSD untuk Menaikan Rating Penjualan dengan Metode Item-Based Collaborative Filtering," *AJCSR [Academic Journal of Computer Science Research*, 2023.

[13] L. Cahyani, N. Sephiana, M. Tahir, and J. Aisyiah, "Sistem Rekomendasi Wisata Kuliner Madura Menggunakan Content Based Filtering," *EXPLORE-IT: Jurnal Keilmuan dan Aplikasi Teknik Informatika*, vol. 16, no. 1, Jun. 2024, doi: 10.35891/explorit.

[14] M. Alkaff, H. Khatimi, and A. Eriadi, "Sistem Rekomendasi Buku pada Perpustakaan Daerah Provinsi Kalimantan Selatan Menggunakan Metode Content-Based Filtering," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 20, no. 1, pp. 193–202, Sep. 2020, doi: 10.30812/matrik.v20i1.617.

[15] S. Z. Putri, M. Hariadi, and R. F. Rachmadi, "Rekomendasi Barang pada E-Commerce Menggunakan Content Based Recommender System Berbasis Ethereum Blockchain," *JURNAL TEKNIK ITS*, vol. 12, no. 2, 2023, Accessed: Sep. 03, 2024. [Online]. Available: 10.12962/j23373539.v12i2.115263

[16] R. Akbar, D. Richasdy, and R. Dharayani, "Sistem Rekomendasi Buku Dengan Collaborative Filtering Menggunakan Metode Singular Value Decomposition (SVD)," in *e-Proceeding of Engineering*, 2023, pp. 5013–5019.

[17] A. Anisatuzzumara, "Implementasi Latent Dirichlet Allocation (LDA) Dan K-Nearest Neighbors (KNN) Pada Sistem Rekomendasi," Universitas Islam Sultan Agung, 2024.

[18] M. Fajriansyah, P. P. Adikara, and A. W. Widodo, "Sistem Rekomendasi Film Menggunakan Content Based Filtering," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 6, pp. 2188–2199, May 2021, [Online]. Available: http://j-ptiik.ub.ac.id

[19] Z. Munawar, Y. Herdiana, N. I. Putri, and R. Rustiyana, "Dampak Intelijen Bisnis Pada Kualitas Pengambilan Keputusan," *Infotronik : Jurnal Teknologi Informasi dan Elektronika*, vol. 6, no. 1, p. 32, Jun. 2021, doi: 10.32897/infotronik.2021.6.1.661.

[20] S. Reddy, S. Nalluri, S. Kunisetti, S. Ashok, and B. Venkatesh, "Content-based movie recommendation system using genre correlation," in *Smart Innovation, Systems and Technologies*, Springer Science and Business Media Deutschland GmbH, 2019, pp. 391–397. doi: 10.1007/978-981-13-1927-3_42.

[21] M. Rofiq and R. F. Uzzy, "Penentuan Jalur Terpendek Menuju Cafe Di Kota Malang Menggunakan Metode Bellman-Ford Dengan Location Based Service Berbasis Android," *Jurnal Ilmiah Teknologi dan Informasi ASIA*, vol. 8, no. 2, 2014, Accessed: Sep. 10, 2024. [Online]. Available: https://jurnal.stmikasia.ac.id/index.php/jitika/article/view/176/131

[22] R. A. Azdy and F. Darnis, "Use of Haversine Formula in Finding Distance between Temporary Shelter and Waste End Processing Sites," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, May 2020. doi: 10.1088/1742-6596/1500/1/012104.