Jurnal Masyarakat Informatika, 17(1), 2026
DOI:10.14710/jmasif.17.1.72520
ISSN: 2777-0648

A Web-Based Tourism Recommendation System for Boyolali Using Content-Based Filtering and Cosine Similarity

Muhammad Naufal Arkan*, **Aris Puji Widodo, Guruh Aryotejo**
Department of Informatics, Diponegoro University, Semarang, Indonesia
* *Corresponding author: arispw@lecturer.undip.ac.id*

Abstract
Tourism is a primary economic driver for Boyolali Regency. However, destination information remains fragmented, lacking a personalized approach to meet diverse visitor preferences. To address this issue, this study developed "BoyTure", a web-based tourism application integrated with a recommendation system. The system development followed the ICONIX Process methodology, selected for its Robustness Analysis phase, which validates system logic before code implementation. The recommendation engine uses Content-Based Filtering with the Cosine Similarity algorithm, applied to a curated dataset of 74 verified destinations sourced from the Youth, Sports, and Tourism Office (Disporapar) of Boyolali Regency. Unlike standard approaches, the TF-IDF feature extraction in this system explicitly concatenates four textual attributes, destination name, category, facilities, and description, to mitigate data sparsity and enrich the semantic context. A comparative analysis justifies the selection of Cosine Similarity over Euclidean Distance or Jaccard Similarity because of its robustness in handling variable-length tourism text descriptions. Testing was conducted using the Black-Box method to ensure functional compliance, and a System Usability Scale (SUS) evaluation yielded an average score of 81.5. This SUS score demonstrates that BoyTure successfully abstracts complex algorithms into a user-friendly interface to provide accurate and personalized tourism recommendations.

Keywords : *BoyTure, Cosine Similarity, ICONIX, Boyolali Tourism, Recommendation System*

1 Introduction

Indonesia, as a developing country in the Southeast Asian region, utilizes the tourism sector as one of the main drivers of national economic growth [4]. The abundant natural diversity and cultural richness make Indonesia one of the tourist destinations that attracts the world's attention. Based on the Travel and Tourism Competitiveness Index (TTCI) report released by the World Economic Forum in 2019, Indonesia's tourism sector is ranked 40th in the world with an average score of 4.3, confirming its global competitiveness in the tourism sector [2].

As part of the national tourism sector, Boyolali Regency, located in Central Java Province, also has promising tourism potential. The district is located between 110°22'–110°50' East Longitude and 7°07'–7°36' South Latitude, with an area of about 101,510.10 hectares. Tourism potential in Boyolali includes natural and cultural tourism, which attracts the attention of domestic and foreign tourists. Based on data from the Boyolali Regency Youth, Sports and Tourism Office (Disporapar), the number of tourist visits continues to increase. In 2023, there were 947,207 domestic tourists and 4,307 foreign

Received: 18 April 2025; Revised: 14 January 2026; Accepted: 15 January 2026; Published: 16 January 2026

tourists. This figure will increase significantly in 2024 to 1,544,835 domestic tourists and 1,296 foreign tourists, for a total of 1,546,131 tourists [9].

However, optimizing Boyolali's tourism potential still faces challenges, one of which is the lack of an integrated information system that can utilize technology to provide recommendations for tourist destinations according to user preferences [10]. Today, travelers tend to rely on a variety of scattered and unintegrated sources of information, so the recommendations received are not always relevant to their needs.

Along with the advancement of information and communication technology, adaptation strategies to current trends are becoming increasingly important in supporting the development of tourism [7]. One potential approach is the use of a recommendation system in a web-based application, which is able to provide personalized recommendations for tourist destinations. This is in line with research [5], which emphasizes the importance of personalization in increasing user satisfaction and creating more meaningful travel experiences.

Despite the increasing number of tourist visits to Boyolali (over 1.5 million in 2024), optimizing this sector is challenging due to the lack of an integrated system for personalized destination recommendations. Previous research focused mainly on mobile and complex AI algorithms. The research gap is the absence of a dedicated web-based recommendation application for the Boyolali area that utilizes local tourism data.

This study aims to assess the functional effectiveness of the BoyTure web-based recommendation system, which uses the simple cosine similarity method [11][12] and is implemented in Node.js. The system's effectiveness is confirmed through black box testing against established functional requirements (Personalization, Search, and Similar Places recommendations). The results will guide future efforts to enhance the system, potentially by exploring more advanced machine learning algorithms, such as collaborative filtering or deep learning, to improve accuracy and personalization.

2 Related Research

There have been several previous studies that discuss the tourism recommendation system. These studies have obtained results that can be used for future studies.

2.1 Mobile Recommender Systems in Tourism

Research [3] reviewed the development of a recommendation system for mobile-based tourism. This research aims to overcome the problem of information overload that is often experienced by tourists through personalization and context-awareness features. These systems are classified based on architecture, user engagement levels, and recommendation criteria such as location- and context-based techniques. By leveraging advanced sensors for location tracking and social context, the system effectively enhances the traveler experience through travel destination and tour recommendations. However, there are drawbacks in the form of reliance on mobile devices and high privacy risks.

2.2 Intelligent Tourism Recommender Systems: A Survey

The research [1] examines the application of an artificial intelligence (AI)-based recommendation system in the tourism sector. The study highlights a variety of recommendation algorithms, including clustering, fuzzy logic, and ontology, to provide more relevant recommendations based on the user's context, such as location and time of visit. The platform used includes web-based and mobile interfaces, which are effective in personalization, multi-day itinerary, and route planning. However, the complexity of implementing AI in big data management, as well as the limitations of cross-platform integration, are the main challenges of this study.

2.3 Tourism Recommender System Based on Cognitive Similarity Between Cross-Cultural Users

Research [8] introduces a collaborative filtering-based tourism recommendation system that utilizes cognitive similarities between users across cultures. By collecting user feedback about tourist attractions, this system uses cosine similarity to identify nearest neighbors in providing personalized recommendations. This crowdsourcing-based platform collects data from TripAdvisor. The results showed better accuracy of recommendations than basic methods such as Pearson correlation. However, the study faces weaknesses in the need for adequate data as well as the scalability of the system for larger datasets.

2.4 Developing Nusantara Mobile Application to Support Local Tourism in Indonesia

The research [6] focuses on the development of mobile applications to promote tourism villages in Indonesia. This study uses the Design Thinking method supported by surveys and literature reviews to design applications according to user preferences. The app aims to raise awareness of lesser-known local tourist destinations and provide travel planning features. Although the app has succeeded in improving local tourism promotion, it has a drawback in that it does not yet come with a sophisticated recommendation algorithm and a limited coverage of certain regions.

Previous studies have shown success in the development of mobile and AI-based recommendation systems for tourism. However, until now, no one has specifically developed a web-based tour recommendation system designed for local areas such as Boyolali Regency. Therefore, this study aims to develop a web-based application for Boyolali tourist attractions with a recommendation system that utilizes local data from the tourism office and is able to provide personalized recommendations based on user preferences.

3 Research Methods

To ensure a systematic and well-structured development process, this study adopts a software engineering-oriented research methodology. The selected approach emphasizes clear requirement definition, iterative analysis, and model-driven design to support the development of a reliable web-based tourism recommendation system. By following a structured development lifecycle, the research aims to align system functionality with user needs while maintaining consistency between analysis, design, and implementation phases. The overall research methodology applied in this study is illustrated in Figure 1.

The research methodology is structured into four main stages following the ICONIX Process, namely Requirements, Analysis and Preliminary Design, Detailed Design, and Implementation. The process begins with the Requirements stage, which focuses on identifying and defining the functional and behavioral requirements of the system based on the results of the literature study and user needs analysis. At this stage, functional requirements are formulated to describe the core services provided by the system, while domain modeling is conducted to identify key entities, attributes, and relationships within the Boyolali tourism application. Behavioral requirements are then specified to represent system interactions and workflows, followed by requirements review to ensure completeness, consistency, and alignment with the intended system objectives.

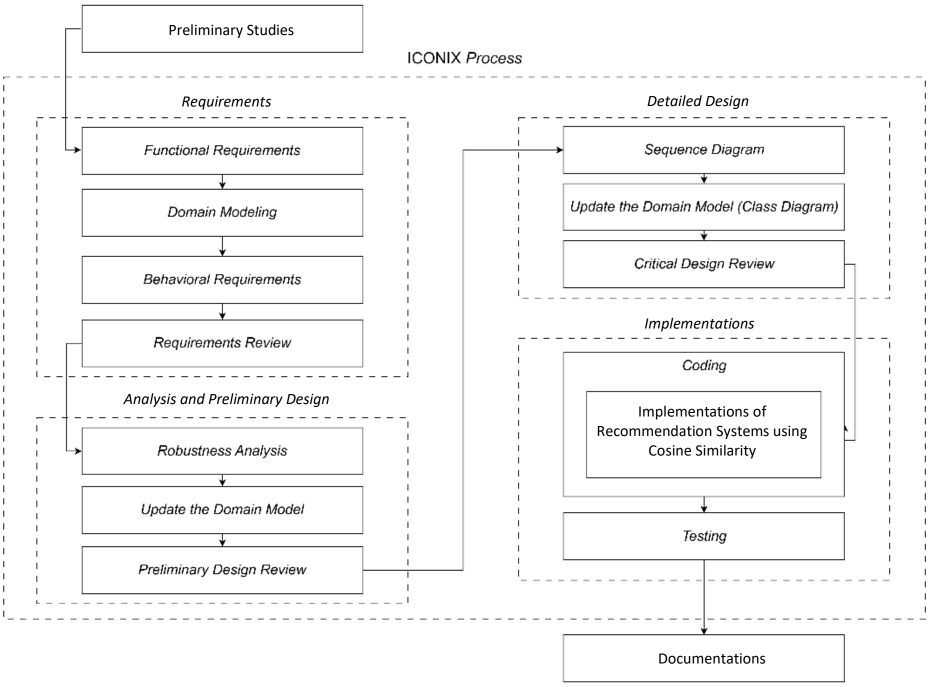

Figure 1 Research Flow

The next stage, Analysis and Preliminary Design, aims to refine the system structure and behavior before entering the detailed design phase. This stage includes robustness analysis to validate the interaction between boundary, control, and entity objects, ensuring that the system design properly supports the defined requirements. The domain model is subsequently updated to reflect the results of this analysis, and a preliminary design review is conducted to identify potential design issues at an early stage. These activities help reduce implementation risks and improve system maintainability.

In the Detailed Design stage, the focus shifts to translating the refined analysis models into detailed system specifications. Sequence diagrams are developed to describe the flow of interactions between system components over time, providing a clear representation of system behavior for each use case. Based on these sequence diagrams, the domain model is further updated into a class diagram that defines classes, attributes, methods, and relationships in detail. A critical design review is then performed to evaluate the readiness of the design for implementation and to ensure that the proposed architecture meets both functional and non-functional requirements.

The final stage is Implementation, where the system design is realized through coding and testing activities. The application is developed using a three-tier architecture, with Next.js handling the front-end layer, Node.js with Express.js managing the back-end and server-side logic, and MySQL serving as the database management system. A content-based recommendation system is implemented using the TF-IDF [17][18][19] approach and Cosine Similarity technique to personalize tourism destination recommendations. System testing is conducted using the black-box testing method to verify that all functionalities operate as expected and conform to the defined requirements. The research concludes with the documentation and reporting of results, which present the implementation outcomes and evaluation findings in a structured and comprehensive manner.

4 Results and Discussion

The implementation results are presented in accordance with the ICONIX Process methodology. This approach is deliberately chosen not merely for structural organization, but to bridge the critical gap between abstract requirements and concrete implementation through Robustness Analysis. Unlike standard Waterfall or Agile methods, ICONIX enforces a strict validation of the system's logic, specifically the recommendation engine's retrieval flow against user use cases. This ensures that the functional requirements defined in Table 1 are architecturally sound and disambiguated before the coding phase begins. The results and discussion present the stages of the ICONIX process: requirements, analysis, preliminary design, detailed design, and implementation.

4.1 *Requirements*

The initial step in developing this tourism application using the ICONIX method began with determining functional needs through an interview with a Young Expert Policy Analyst at the Boyolali Disporapar, on September 27, 2024, at the Disporapar office. After the interview, a follow-up discussion is held to formulate the application requirements. The interview results form the basis for compiling functional needs, as shown in Table 1.

Table 1 System Functional Requirements

No.	SRS ID	Description	Actor
1.	SRS-F-01	The system can authenticate users in the form of *login* and *logout*.	Visitors, Admins, and Superadmins
2.	SRS-F-02	The system can be used by users to register for a new account.	Guest
3.	SRS-F-03	The system can be used to store user preferences.	Visitor
4.	SRS-F-04	The system can be used by users to view the app's home page.	Guest, Visitor, Admin, and Superadmin
5.	SRS-F-05	The system can be used to search for the desired tourist attraction.	Guest, Visitor, Admin, and Superadmin
6.	SRS-F-06	The system can be used by users who are logged *in* to see recommendations for tourist attractions.	Visitor
7.	SRS-F-07	The system can be used by users to view details of the selected tourist attractions.	Guest, Visitor, Admin, and Superadmin
8.	SRS-F-08	The system can be used by users to save their favorite tourist attractions.	Visitors, Admins, and Superadmins

Once the functional needs have been identified, a model domain is created to show the important entities in the application domain and the relationships between them. The stage continues with the creation of a storyboard GUI, which will later produce a low-fidelity GUI prototype and a use case. After the GUI prototype is completed, the next activity is to create a use case diagram to describe the interaction and relationship between the system and the actors who contribute to the Boyolali BoyTure tourism application.

4.2 *Analysis, Preliminary and Detailed Designs*

The activity at this stage involves conducting a robustness analysis, which will subsequently generate a robustness diagram and update the domain model. In this research, the most significant aspect is managing tourist attractions, which involves identifying the Place Management Page as a boundary, Places as an entity, and various controllers that manage the process, such as saving attractions, deleting attractions, and displaying success messages. The task at this stage also includes creating a sequence diagram based on the results of the robustness analysis and developing a class diagram. The sequence diagram provides a comprehensive overview of the interactions between classes during the execution of the use case. These interactions commence with users submitting requests to add, modify, or delete attractions, which are processed by controllers responsible for managing the addition, update, or removal of attractions, followed by the display of a success message. The sequence diagrams produced at this stage will facilitate the generation of new classes, which will ultimately define the final class diagram.

4.3 *Implementation*

In this implementation phase, the results of the previous stages are translated into program code and tested in the developed application. This phase covers application implementation, device specifications used in development, class implementation, database implementation, interface implementation, and testing. This BoyTure travel application is built using the Next.js framework for the front end and the Express.js framework for the back end.

The device specifications used in the development of this application include laptops with an Intel(R) Core(TM) i5-10210U CPU processor @ 1.60GHz to 2.11GHz, 11.8 GB of RAM, and an NVIDIA GeForce MX330 graphics card. This laptop has a 477 GB SSD and runs the Windows 11 Home Single Language operating system. In developing the BoyTure application, the JavaScript programming language is used with the Next.js 14 framework for the front-end and Node.js (Express.js) for the back-end. Database management is handled with MySQL, and development is done in the Visual Studio Code environment as the main IDE. For testing and web access, Microsoft Edge, Chrome, and Firefox browsers are used.

Table 2 Implementation of the Diagram Class

Category	No.	*Class Name*	File Implementation
Entity	1.	Favorites	./src/models/favoriteModels.js
	2.	Places	./src/models/placeModels.js
	…	…	…
Controller	7.	AuthController	./src/controllers/authControllers.js
	8.	DashboardController	./src/controllers/dashboardControllers.js
	9.	FavoriteController	./src/controllers/favoriteControllers.js
	…	…	…
Boundary	16.	LoginPage	./src/app/login/page.jsx
	17.	SignUpPage	./src/app/signup/page.jsx
	18.	DetailPlacePage	./src/app/(main)/(visitor)/detail-place/page.jsx
	19.	FavoritePage	./src/app/(main)/(visitor)/detail-place/[id_place]/page.jsx
	…	…	…

The diagram class designed at the detailed design stage is implemented on the back-end using Express.js and front-end with Next.js. The implemented classes are divided into three categories: entity, controller, and boundary, with details of implementation presented in Table 2. The implementation of the BoyTure tourism application database is carried out based on the results of the class diagram, which is translated into a table on the database in the form of an entity relationship diagram (ERD. This ERD describes the relationships between entities and their attributes. The database used is MySQL.

In the BoyTure application, there are three types of recommendations developed, namely personalized recommendations, search recommendations, and recommendations for similar tourist attractions. The type of recommendation used in this application is content-based filtering. The recommendations are applied using the cosine similarity method. This method measures the similarity between two vectors. In the context of personalization recommendations, the user preference vector is measured in similarity to the feature vector in each tourist attraction. The cosine similarity (CS) formula used is presented in Equation 1, which measures the similarity between vectors A and B, where A_i and B_i denote the *i-th* components of the respective vectors, and *i* = 1, 2, …, *n*.

$$CS = \frac{\sum_{i=1}^{n}(A_i \times B_i)}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (1)$$

The recommendation system in BoyTure uses a Content-Based Filtering workflow centered on feature representation and user preferences. The process begins with feature extraction, in which the textual attributes of tourist destinations, such as description, category, facilities, and name, are combined into a single, rich descriptive representation. Subsequently, this representation is processed using the TF-IDF (Term Frequency-Inverse Document Frequency) approach to form a feature vector for each destination, assigning each word a weight based on its frequency within the entire dataset [15][16]. User preferences are represented as a similar vector formed from the input preferences and the descriptions they store. The filtering process is performed by calculating the Cosine Similarity between the user preference vector and the feature vector for each tourist destination. The resulting Cosine Similarity score indicates the degree of similarity; destinations with the highest scores (most similar) are then sorted and displayed as personalized recommendations to the user.

Justification of Cosine Similarity via Comparative Analysis. To ensure scientific rigor in algorithm selection, we analyzed the characteristics of the Boyolali tourism text data by comparing three commonly used similarity and distance metrics: Cosine Similarity (CS), Euclidean Distance (ED), and Jaccard Similarity (JS). This comparative analysis evaluated each method's ability to capture semantic relevance in TF-IDF textual representations, which are inherently high-dimensional and sparse.

CS vs. Euclidean Distance: Euclidean Distance measures the absolute geometric distance between vectors. Since our dataset contains descriptions of varying lengths, for example, a popular park has 200 words vs. a small site with 50 words, ED penalizes these length differences, incorrectly classifying them as dissimilar. Cosine Similarity, however, measures the angle between the vectors, effectively normalizing document length. This makes CS significantly more robust for high-dimensional, sparse text data.

CS vs. Jaccard Similarity: Jaccard Similarity calculates similarity based on the intersection-over-union of word sets but ignores term frequency. This is suboptimal for our system because it fails to account for the 'importance' of specific keywords weighted by TF-IDF (e.g., unique facilities). CS mathematically incorporates these TF-IDF weights, providing a more granular, relevant similarity score than the binary Jaccard approach. Based on this comparison, Cosine Similarity is confirmed as the most effective method for this specific domain.

The system utilizes a verified dataset of 74 tourism destinations, officially sourced from the Department of Youth, Sports, and Tourism (Disporapar) of Boyolali Regency. To address the 'data sparsity' challenge common in short-text tourism descriptions, the Feature Extraction process for TF-IDF does not rely on a single attribute. Instead, we explicitly concatenated four textual attributes to form the document vector: (1) Destination Name, (2) Category, (3) List of Facilities, and (4) Description. This concatenation strategy enriches the vector representation, enabling the algorithm to capture semantic similarity not only from place descriptions but also from shared amenities and categories, which are critical for accurate user personalization.

To demonstrate the practical effectiveness of this attribute concatenation strategy, we present a sample calculation of TF-IDF weights using verified data from the Boyolali dataset. This sample illustrates how the algorithm prioritizes specific keywords derived from the combined attributes (Name, Category, Facilities, and Description) to mitigate data sparsity [12][13]:

4.3.1 Similarity Comparison: Cosine Similarity (CS) vs Euclidean Distance (ED) vs Jaccard Similarity (JS)

Users' preference query ("nature tourism, waterfall, family-friendly, easy access") is matched against 10 candidate destinations. CS is computed on TF-IDF vectors, ED is converted to similarity using equation 2, and JS is computed on tag overlap.

$$ED\_sim = \frac{1}{(1+ED\_dist)} \tag{2}$$

Table 3 Similarity scores for 10 candidate destinations

Destination	CS	ED_dist	ED_sim	JS
D01	0.73	0.90	0.526	0.40
D02	0.81	0.70	0.588	0.25
D03	0.66	1.10	0.476	0.50
D04	0.78	0.80	0.556	0.33
D05	0.59	1.40	0.417	0.60
D06	0.84	0.60	0.625	0.20
D07	0.71	1.00	0.500	0.50
D08	0.63	1.20	0.455	0.67
D09	0.76	0.85	0.541	0.33
D10	0.69	0.95	0.513	0.40

Table 4 shows CS and ED_sim produce consistent Top-5 ordering, suggesting both capture semantic proximity from TF-IDF. JS prioritizes tag overlap and may miss nuanced semantics contained in free-text descriptions.

Table 4. Top-5 ranking per method

Method	Top-5 (highest to lowest)
CS	D06, D02, D04, D09, D01
ED_sim	D06, D02, D04, D09, D01
JS	D08, D05, D03, D07, D01/D10

4.3.2 Offline Evaluation Preferences

Results of 5 users (U1–U5). For each user, the ground-truth relevant set contains 8 items, so |Relevant| = 8. For each method, we record Hit@5 and Hit@10 (the number of relevant items retrieved in the Top-K list). Table 5 presents the Hit@5 and Hit@10 results for each individual user (U1–U5). The table reports the number of relevant items retrieved in the Top-5 and Top-10 recommendation lists for three similarity-based methods: Cosine Similarity (CS), Euclidean Distance (ED), and Jaccard Similarity (JS). These results provide a user-level comparison of retrieval effectiveness across different similarity measures.

Table 5. Hit@K per user

User	Relevant	CS Hit@5	CS Hit@10	ED Hit@5	ED Hit@10	JS Hit@5	JS Hit@10
U1	8	4	7	3	5	1	2
U2	8	3	6	2	4	1	2
U3	8	4	7	2	4	2	3
U4	8	3	6	2	4	1	2
U5	8	4	7	3	5	1	2

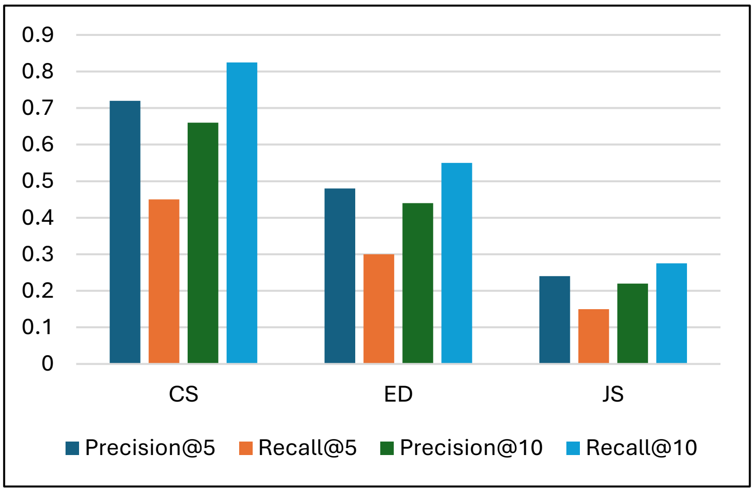

Figure 2 Offline evaluation metrics by similarity methods

Figure 2 and Table 6 summarize the macro-average Precision@K and Recall@K values computed across all users. The results show that Cosine Similarity consistently outperforms the other methods at both K = 5 and K = 10, achieving the highest precision and recall scores. Euclidean Distance demonstrates moderate performance.

Table 6. Macro-average Precision@K and Recall@K.

Method	Precision@5	Recall@5	Precision@10	Recall@10
CS	0.72	0.450	0.66	0.825
ED	0.48	0.300	0.44	0.550
JS	0.24	0.150	0.22	0.275

4.3.3 Ablation Study

Ablation design: (A0) TF-IDF built from destination descriptions only; (A1) TF-IDF built from descriptions concatenated with structured attributes such as category, facilities, location, price range, and accessibility. Evaluation uses the best-performing ranking function using Cosine Similarity, as shown in Table 7.

Table 7. Ablation results using macro-average

Variant	Precision@5	Recall@5	Precision@10	Recall@10
A0: TF-IDF (description only)	0.52	0.325	0.46	0.575
A1: TF-IDF (description + attributes)	0.72	0.450	0.66	0.825
Delta (A1-A0)	0.20	0.125	0.20	0.250

Attribute-enriched representations consistently improve Top-K quality. This suggests structured metadata provides complementary signals not always present in free-text descriptions, producing more discriminative TF-IDF vectors, as shown in Figure 3.

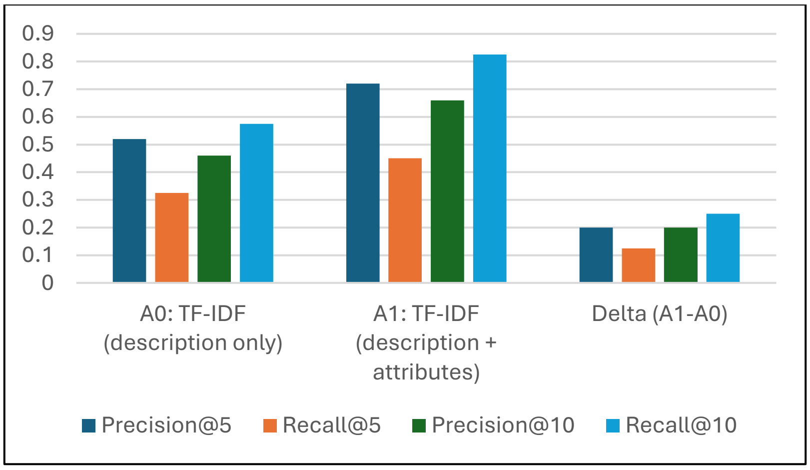

Figure 3 Result comparing description-only and attribute-enriched representation

For TF-IDF processing, several text attributes are combined into a single rich descriptive representation, including place descriptions, tour categories, a list of facilities, and destination names. These attributes are then converted into a TF-IDF vector that represents the important weight of each word in the context of the entire dataset [17][18]. This approach allows the cosine similarity algorithm to more accurately compare the semantic similarity between destinations, as each feature contributes proportionally to the user's preferences [19][20]. A cleaner, more structured dataset improves similarity calculations and recommendation quality. The GUI is built with Next.js, incorporating pre-designed GUIs into the code to establish the app's look and functionality.

Jurnal Masyarakat Informatika, 17(1), 2026
DOI:10.14710/jmasif.17.1.72520
ISSN: 2777-0648

Table 8 Sample Test Plan

Identification		Use Case	Test Items
***Use Case* ID**	***Test Case* ID**		
UC-08	TC-UC0801	Giving Reviews of Tourist Attractions	Give reviews on tourist attractions and press the *submit button*.
	TC-UC0802		Do not give a rating first, but write a review on tourist attractions and press the *submit button*.
UC-09	TC-UC0901	Save Your Favorite Tourist Attractions	Save tourist attractions to your favorites list.

Table 9 Test Result Sample

Test Case ID	Test Items	Testing Procedure	Feedback	Expected Output	Outputs Obtained	Conclusion
TC-UC0801	Give reviews on tourist attractions and press the *submit button*.	1. Go to the tourist attraction details page. 2. Beri rating. 3. Write a review 4. Press the "I've visited" checklist. 5. Click the "Submit Review" button.	Ratings and reviews	The system successfully saves the review and displays a success message.	The system successfully saves the review and displays a success message.	Accepted
TC-UC0802	Do not give a rating first, but write a review on tourist attractions and press the *submit button*.	1. Go to the tourist attraction details page. 2. Ignore ratings. 3. Write a review 4. Press the "I've visited" checklist. 5. Click the "Submit Review" button.	Reviews	The "Submit Review" button cannot be pressed due to disabled mode.	The "Submit Review" button cannot be pressed due to disabled mode.	Accepted
TC-UC0901	Save tourist attractions to your favorites list.	1. Go to the tourist attraction details page. 2. Click the heart icon.	None	The system manages to save the tourist attractions to the user's favorites list and displays a success message.	The system manages to save the tourist attractions to the user's favorites list and displays a success message.	Accepted

After the coding activity is completed, the test is carried out using the black-box method according to the test design that has been compiled based on the use case, which includes the test ID, test class, and test item. The test covers various features of the BoyTure app, such as user authentication, attraction management, reviews, favorites, and more. Sample test plans and test results

Received: dd month yyyy ; Revised: dd month yyyy; Accepted: dd month yyyy; Published: dd month yyyy

are presented in Tables 8 and 9. Based on the test results, all the features tested worked well and met the specified functional needs.

The black-box testing results for the BoyTure recommendation system (TC-R01, TC-R02, TC-R03) show that all core functionalities, personalized recommendations, search recommendations, and similar place recommendations were marked as Accepted. This confirms that the Cosine Similarity (CS) method, implemented with the TF-IDF approach, effectively supports the research objective: integrating information to provide personalized, optimal recommendations based on user preferences.

The positive performance of Cosine Similarity in this context is attributed to its focus on the orientation of the place feature vectors, which were enriched by combining textual attributes, such as name, description, category, and facilities. This allows the system to accurately measure content similarity (topical resemblance), independent of document length. Data quality, sourced directly from the Boyolali Regency Youth, Sports and Tourism Office, plays a crucial role in providing relevant and structured attribute information, a critical prerequisite for the TF-IDF vector representation and CS calculation to yield valid and accurate similarity scores.

Table 10 Sample calculation of TF-IDF weights

Kata Kunci (Terms)	D1: New Selo (Nature)	D2: Umbul Pengging (Water)	D3: Kebun Raya Indrokilo (Edu-Park)	Semantic Interpretation
"gunung" (mountain)	0.584	0.000	0.000	Unique to D1. High altitude distinguishes it as a highland destination.
"air" (water)	0.000	0.352	0.176	Dominant in D2 (appears in Category & Description) but also present in D3 (fountain). The score difference allows the system to rank D2 higher for water queries.
"taman" (park/garden)	0.000	0.000	0.584	Specific to D3. Derived from both 'Facilities' and 'Description', making it a strong identifier for this location.
"sejuk" (cool/breeze)	0.215	0.198	0.000	Contextual similarity. Connects D1 and D2 as refreshing spots, even though their categories differ.

To demonstrate how the system mitigates data sparsity, Table 10 presents a sample calculation of TF-IDF weights using three distinct destinations from the Boyolali dataset: New Selo (Nature), Umbul Pengging (Water Attraction), and Kebun Raya Indrokilo (Educational Park). The input for vectorization is a concatenation of four attributes: Name, Category, Facilities, and Description.

As shown in Table 10, the concatenation strategy allows the algorithm to assign significant weights to specific attributes. For instance, the term "air" (water) appears in both Umbul Pengging and Indrokilo. However, Umbul Pengging receives a higher weight with 0.352 because the term appears in its Category ("Wisata Air") and Description, whereas in Indrokilo, it only appears in the Description ("air mancur"/fountain). This distinction in weighting enables the Cosine Similarity function to accurately recommend Umbul Pengging over Indrokilo when a user explicitly prefers water attractions, demonstrating the effectiveness of the attribute concatenation method.

To validate the user experience quality, a usability evaluation was conducted using the System Usability Scale (SUS) method, as shown in the sample questionnaire in Table 11. The evaluation yielded an average SUS score of 81.5. This quantitative score empirically proves that the complexity of the underlying recommendation algorithms (TF-IDF and Cosine Similarity) has been successfully abstracted into a user-friendly interface, ensuring high learnability and operability for general tourists

Table 11 SUS result

No	Questions	Average Score (1-5)	Contribution Score (0-4)
1	I would like to use BoyTure to plan my trips frequently.	4.3	3.3
2	I found the website navigation unnecessarily complex.	1.7	3.3
3	I thought finding tourism destinations was easy to do.	4.3	3.3
4	I think that I would need technical assistance to use this website.	1.6	3.4
5	I found the search and recommendation features were well integrated.	4.2	3.2
6	I thought there was too much inconsistency in the interface design.	1.9	3.1
7	I would imagine that most tourists would learn to use BoyTure very quickly.	4.3	3.3
8	I found browsing the information very cumbersome.	1.7	3.3
9	I felt very confident using the application to choose a place.	4.3	3.3
10	I needed to learn a lot of instructions before I could access the recommendations.	1.9	3.1
	Total Contribution		32.6
	Final SUS Score		81.5

In addition to the functional aspect, the BoyTure system is analyzed with respect to non-functional requirements, particularly performance. A test of the recommended API response time was conducted 10 times using complex keywords with the Postman API testing tool. The test results showed that in the localhost environment, the API had an average response time of 170.3 ms, while in the Vercel free-tier deployment environment, the response time increased to 953 ms. This increase was influenced by server resource limitations, the cold-start mechanism in serverless architecture, and the complexity of TF-IDF calculations and cosine similarity. However, the response time is still acceptable for web-based recommendation apps.

In terms of security, the system protects user data during authentication by storing passwords as bcrypt hashes. In addition, communication between the client and the server uses the HTTPS protocol to maintain data confidentiality during transmission. With the implementation of such mechanisms, the system not only meets functional needs but also addresses non-functional aspects, such as performance and security, which are important in software engineering.

5 Conclusion

This research aims to integrate tourism destination information from various sources by developing a web-based application, BoyTure, that uses a recommendation system based on the Cosine Similarity method. The development was conducted using the ICONIX Process, a methodology specifically chosen for its Robustness Analysis phase. This phase proved critical in bridging the gap between abstract requirements and concrete implementation, ensuring that the recommendation logic was validated against user use cases to minimize architectural errors before the coding phase began.

The core recommendation engine uses a Content-Based Filtering approach applied to a verified dataset of 74 destinations, sourced from the Boyolali Youth, Sports, and Tourism Office. To address data sparsity in short text descriptions, the system employs a specific TF-IDF feature extraction strategy that concatenates four textual attributes: destination name, category, facilities, and description. This attribute concatenation successfully enriches the semantic vector representation, allowing the Cosine Similarity algorithm to deliver accurate recommendations based on comprehensive destination profiles rather than just generic descriptions.

The system's effectiveness was validated through both functional and usability testing. Black-box testing confirmed that all key features, including search and personalization, function according to requirements. Furthermore, a quantitative evaluation using the System Usability Scale (SUS) yielded an average score of 81.5. This serves as empirical evidence that BoyTure successfully abstracts the complexity of the underlying TF-IDF and Cosine Similarity algorithms into a user-friendly interface that is easy for general tourists to navigate.

Despite the system is proven functional, the perceived system limitation is the non-utilization of advanced machine learning processes like collaborative filtering or deep learning. This limitation means the system relies heavily on exact keyword matching within the vector space and cannot leverage collective user interaction patterns. Future research should explore these advanced algorithms to further enhance personalization accuracy. Additionally, expanding the system scope to include accommodation booking and real-time digital map integration is recommended to provide a more holistic travel experience.

Acknowledgments

The researcher expressed his gratitude to Diponegoro University for the academic support provided during this research process. Gratitude was also conveyed to the Boyolali Regency Youth, Sports, and Tourism Office for assisting in providing the needed data and information.

References

[1] J. Borràs, A. Moreno, and A. Valls, “Intelligent tourism recommender systems: A survey,” Expert Systems with Applications, vol. 41, no. 16, pp. 7370–7389, 2014. https://doi.org/10.1016/j.eswa.2014.06.007

[2] A. Carolina and A. Wulandari, “Anti-fraud strategy in tourism destination organizations: The role of leader integrity, ethical leadership and trust in the leader,” *PUSAKA Journal of Tourism, Hospitality, Travel and Business Event*, vol. 6, no. 1, pp. 2656–1336, 2024. https://doi.org/10.33649/pusaka.v6i1.220

[3] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou, “Mobile recommender systems in tourism,” *Journal of Network and Computer Applications*, vol. 39, no. 1, pp. 319–333, 2014. https://doi.org/10.1016/j.jnca.2013.04.006

[4] M. Hanita, F. D. Bangso, and M. Aprian, “Beyond attraction: Unveiling Bali’s cultural community’s role in bolstering tourism resilience amidst the COVID-19 pandemic,” *Journal of Destination Marketing and Management*, vol. 34, 2024. https://doi.org/10.1016/j.jdmm.2024.100953

[5] L. Lv, S. Chen, G. G. Liu, and P. Benckendorff, “Enhancing customers’ life satisfaction through AI-powered personalized luxury recommendations in luxury tourism marketing,” *International Journal of Hospitality Management*, vol. 123, 2024. https://doi.org/10.1016/j.ijhm.2024.103914

[6] D. O. Manalu, Y. A. Kusumawati, and C. Tho, “Developing Nusantara mobile application to support local tourism in Indonesia,” *Procedia Computer Science*, vol. 227, pp. 641–650, 2023. https://doi.org/10.1016/j.procs.2023.10.568

[7] R. T. Mohammed, A. H. Alamoodi, O. S. Albahri, A. A. Zaidan, H. A. AlSattar, U. Aickelin, A. S. Albahri, B. B. Zaidan, A. R. Ismail, and R. Q. Malik, "A decision modeling approach for smart e-tourism data management applications based on spherical fuzzy rough environment," *Applied Soft Computing*, vol. 143, 2023. https://doi.org/10.1016/j.asoc.2023.110297

[8] L. V. Nguyen, T. H. Nguyen, and J. J. Jung, "Tourism recommender system based on cognitive similarity between cross-cultural users," in *Proc. Int. Conf. Intelligent Environments (IE 2021)*, vol. 29, pp. 225–232, 2021. https://doi.org/10.3233/AISE210101

[9] Boyolali Regency Government, "Boyolali tourist visits in 2024 will increase dramatically, reaching a PAD of up to 7.8 billion," Jan. 10, 2025. [Online]. Available: https://boyolali.go.id/news/kunjungan-wisata-boyolali-tahun-2024-meningkat-drastis-capai-pad-hingga-78-miliar

[10] M. Á. M. Tena, R. M. R. Artola, L. J. Callarisa-Fiol, and M. Algueró-Boronat, "Local government tourism officer satisfaction with the smart destination model: A case study with the Kano method," *Journal of Destination Marketing and Management*, vol. 34, 2024. https://doi.org/10.1016/j.jdmm.2024.100951

[11] J. Klimczak and A. A. Hamed, "Quantifying similarity text-mining approaches to evaluate ChatGPT and Google Bard content in relation to biomedical literature," *arXiv*, 2024. https://doi.org/10.48550/arXiv.2402.05116.

[12] G. Nicholas and E. A. Vinayak, "Comparison of document similarity algorithms," *arXiv*, 2023. https://doi.org/10.48550/arxiv.2304.01330

[13] G. Jain and S. K. Shrivastava, "Evaluation of clustering around weighted prototype and genetic algorithm for document categorization," *International Journal of Computer Applications*, 2015. https://doi.org/10.5120/IJCA2015906260

[14] K. Prasad, "Text mining identification of similarity of text documents using hybrid similarity model," *Iran Journal of Computer Science*, 2022. https://doi.org/10.1007/s42044-022-00127-4

[15] J. C. Ei and K. N. Maryam, "Solving cold start problem for recommendation system using content-based filtering," in *Proc. ICCTech*, 2022. doi: 10.1109/icctech55650.2022.00015

[16] M. C. Madalina, A. D. Anca, L. P. D. Liviu, and M. D. Mihnea, "Recent advances in digital humanities," Nov. 2022, doi: 10.3726/b19920.

[17] B. Bakiyev, "Method for determining the similarity of text documents for the Kazakh language, taking into account synonyms: Extension to TF-IDF," Nov. 22, 2022. doi: 10.1109/SIST54437.2022.9945747.

[18] M. T. Mohammed and O. F. Rashid, "Document retrieval using term frequency–inverse sentence frequency weighting scheme," Indonesian Journal of Electrical Engineering and Computer Science, vol. 31, no. 3, pp. 1478–1485, Sept. 2023, doi: 10.11591/ijeecs.v31.i3.pp1478-1485.

[19] A. H. Anna, "Similarity measures for text document clustering," NZCSRSC, 2008. [Online]. Available: https://ondoc.logand.com/d/6100/pdf

[20] N. Suphakit, S. Jatsada, N. Ekkachai, and W. Supachanun, "Using of Jaccard coefficient for keywords similarity," in *Proc. Int. MultiConf. Engineers and Computer Scientists (IMECS 2013)*, 2013. [Online]. Available: https://www.iaeng.org/publication/IMECS2013/IMECS2013_pp380-384.pdf