ILKOM Jurnal Ilmiah Vol. 17, No. 3, December 2025, pp.241-251
Accredited 2nd by KEMDIKTISAINTEK No. 10/C/C3/DT.05.00/2025; E-ISSN 2548-7779 | P-ISSN 2087-1716

Research Article Open Access (CC–BY-SA)

A Deep Learning Approach for Tourism Destination Recommendation Using IndoBERT and TF-IDF

Widya Silfianti[a,1,*]; Rama Dian Syah[a,2]; Adang Suhendra[a,3]; Ali Isra[a,4]; Astie Darmayantie[a,5]; Noviawan Rasyid Ohorella[a,6]

[a] *Universitas Gunadarma, Jalan Margonda Raya 100, Depok, Indonesia*
[1] *wsilfi@staff.gunadarma.ac.id;* [2] *rama_ds@staff.gunadarma.ac.id;* [3] *adang@staff.gunadarma.ac.id;* [4] *ali.isra.id@gmail.com;*
[5] *astie@staff.gunadarma.ac.id;* [6] *noviawanrasyid@staff.gunadarma.ac.id*
** Corresponding author*

Article history: Received August 24, 2025; Revised September 01, 2025; Accepted October 16, 2025; Available online October 29, 2025.

Abstract

The rapid development of information technology has transformed various sectors, including tourism, where recommendation systems play a vital role in providing personalized services. Tourists are often faced with a wide range of destination choices, making decision-making increasingly complex. To address this, Artificial Intelligence (AI) and Natural Language Processing (NLP) can be leveraged to enhance recommendation accuracy through deeper analysis of destination descriptions. This study proposes a tourism destination recommendation system combining IndoBERT, SimCSE, and TF-IDF methods. IndoBERT was applied to capture semantic and contextual meaning in the Indonesian language, SimCSE improved sentence-level embeddings, and TF-IDF extracted essential keywords from descriptions. The system was implemented on a website to generate personalized recommendations based on user input. Evaluation results demonstrated that the composition of IndoBERT and TF-IDF achieved strong performance, with precision, recall, and F1-score values of 1.0 at a similarity threshold of 0.20. However, higher thresholds reduced recall and F1-score, indicating that a lower threshold provided a better balance between accuracy and coverage. The recommendation outputs matched user preferences, and functional testing showed that all website features performed successfully. These findings highlight the effectiveness of combining semantic and keyword-based methods for tourism recommendation. Future work could expand the dataset, integrate user feedback, and benchmark against other state-of-the-art models to further enhance system performance.

Keywords: Indonesia Tourism; Deep Learning; IndoBERT; TF-IDF; Recommendation System; NLP.

Introduction

Technological advancements have significantly transformed the tourism industry, changing how people plan and experience their trips. In the past, travelers relied on guidebooks or personal recommendations, but today they face an overwhelming volume of information from various online platforms [1]. This abundance of choices often causes confusion and makes it difficult for tourists to identify destinations that truly match their preferences. In Indonesia is an archipelagic nation rich in natural and cultural attractions [2]. These challenges become even more evident due to the wide variety of destinations available.

Recommendation systems play a crucial role in addressing this issue by simplifying the search process and supporting personalized travel planning [3]. Personalization is particularly important to ensure that recommendations are relevant to individual needs and characteristics, thereby improving efficiency and user satisfaction [4]. However, existing approaches often rely on literal keyword matching or conventional similarity measures, which are limited in capturing the deeper semantic context of user queries and destination descriptions. As a result, many current systems struggle to deliver accurate and personalized recommendations.

Previous studies have attempted to enhance recommendation systems by incorporating semantic aspects. For example, Türkel applied user-weighted cosine similarity to improve tour package recommendations [5], while Chalkiadakis introduced a hybrid approach that combined hierarchical and non-hierarchical semantic similarities [6]. Although these studies achieved promising results, they are not fully optimized for the Indonesian language and cultural context. This reveals a research gap, as existing methods have not effectively integrated semantic and lexical understanding for tourism destination recommendations in Indonesia.

To address this gap, this study proposes a tourism destination recommendation system that integrates the IndoBERT model trained with the SimCSE (Simple Contrastive Sentence Embedding) method and the TF-IDF (Term Frequency-Inverse Document Frequency) approach. IndoBERT ensures accurate comprehension of Indonesian linguistic and cultural context, while SimCSE enhances semantic sentence representation. At the same time, TF-IDF captures literal keyword relevance, enabling a balance between semantic and lexical similarity. The novelty of this research lies in combining these methods into a hybrid model, while its contribution is to provide more precise and personalized tourism destination recommendations tailored to the Indonesian context.

Method

The method used in this study was CRISP-DM, which was a standard framework for developing data mining models [7], [8]. CRISP-DM had several stages for developing data mining models, namely Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment [9], [10]. The purpose of this study was to implement tourism recommendation system in Indonesia based on deep learning using a composition between IndoBERT and TF-IDF, where **Figure 1** showed the study stages.

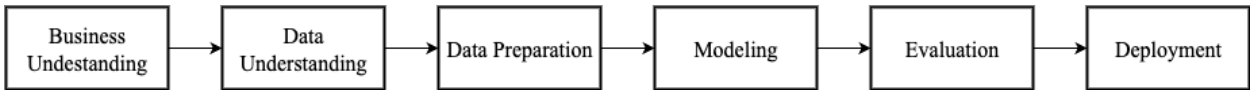

Figure 1. Methodology of Study

A. Business Understanding

The business understanding stage was used to explain the business objectives to be achieved from this study [11], [12]. The common problem faced by tourists was the difficulty in finding destination in Indonesia that matched personal preferences [13], [14]. Based on this problem, the analysis developed a tourism destination recommendation system in the country that was relevant to user input. Recommendation provided need to be accurate and relevant to the input given by the user, helping user in selecting tourism destination in Indonesia.

B. Data Understanding

The data understanding stage was used to identify and collect datasets [15], [16], [17]. This study used a dataset obtained from Kaggle, called Indonesia Tourism Destination. The available data included 437 tourism destination in Indonesia, as shown in **Figure 2**.

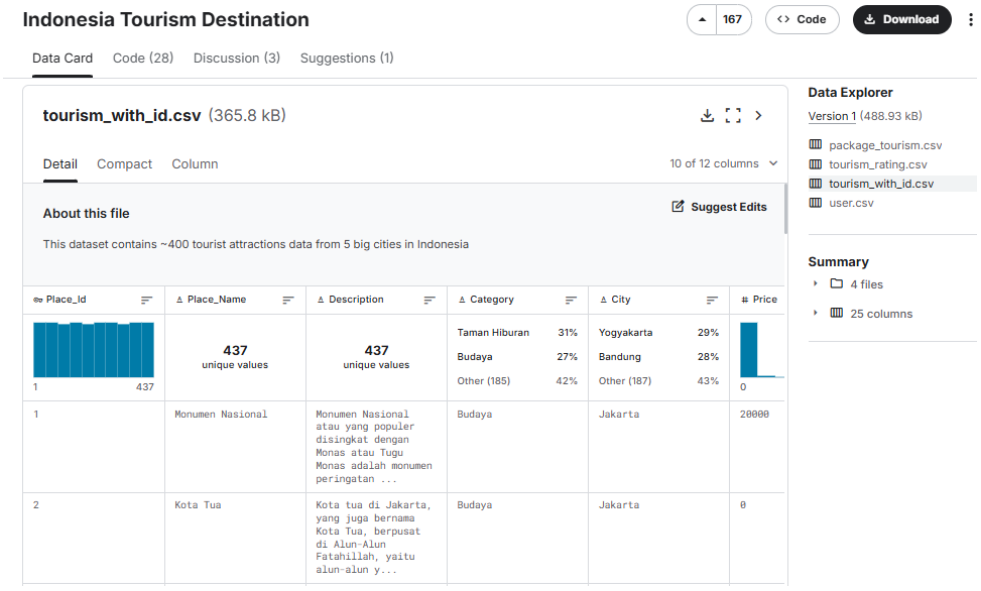

Figure 2. Dataset

The dataset used included five tourism destination cities, namely, Yogyakarta, Jakarta, Bandung, Semarang, and Surabaya. These destinations were also categorized six tourism categories, such as amusement parks, cultural attractions, nature reserves, marine attractions, places of worship, and shopping centers. **Figure 3** showed the distribution of tourism destination by city and tourism category.

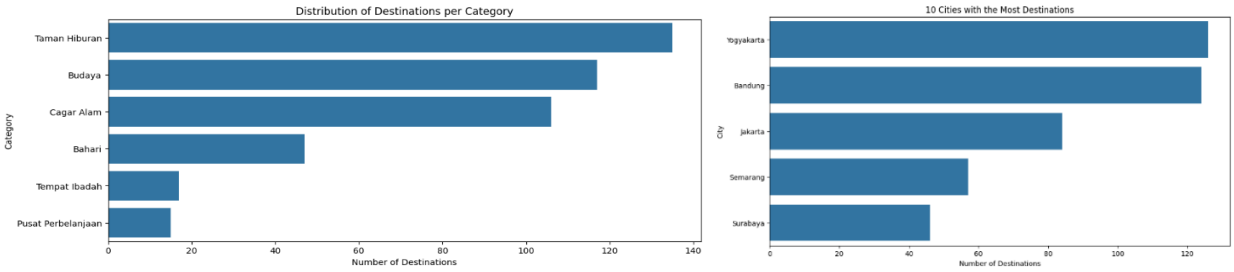

Figure 3. Destination Distribution by City and Tourism Category

Figure 3 showed the distribution of tourism destinations in Indonesia based on category and city. Amusement parks, cultural tourism, and nature reserves dominate the dataset, while maritime tourism, places of worship, and shopping centers appear less frequently. Yogyakarta and Bandung have the highest number of destinations, followed by Jakarta, Semarang, and Surabaya. This study focuses on these five cities and six main tourism categories to ensure a representative and diverse dataset for the recommendation system.

C. Data Preparation

The data preparation stage was used to prepare the dataset, making it ready for use by machine learning algorithms [18], [19], [20]. The dataset consisted of 13 columns, namely, Place_Id, Place_Name, Description, Category, City, Price, Rating, Time_Minutes, Coordinate, Lat, Long, Unnamed:11, and Unnamed:12, as shown in Table 1. In this data preparation stage, unused columns were deleted, namely Time_Minutes, Coordinate, Lat, Long, Unnamed:11, and Unnamed:12.

Table 1. Dataset Feature

Column	Data Type	Explanation
Place_Id	Int64	Unique ID for each tourism destination
Place_Name	Object	Name of tourism destination
Description	Object	Text description of tourism destination
Category	Object	Category of tourism
City	Object	City of tourism destination
Price	Int64	Entrance ticket price
Rating	Float64	Rating value from tourists

Table 1 showed the features of the dataset, which consisted of only seven columns, as the categorical features were Place_Name, Description, Category, and City. Additionally, the numeric features were Price as well as Rating, and the unique ID feature was Place_ID

D. Modeling

The data modeling stage was used to select, train, and evaluate machine learning algorithms [21], [22]. The training process used the IndoBERT model to generate word embedding that was used as sentence embedding. Figure 4 showed the modeling stages conducted during the process of this study.

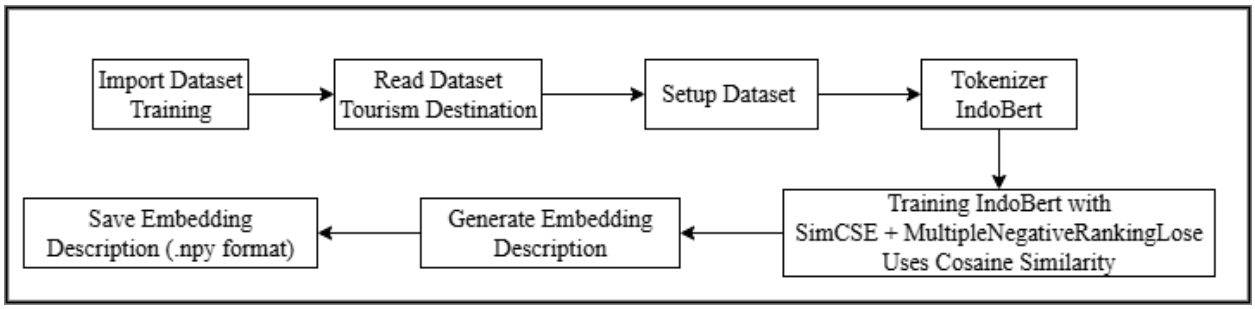

Figure 4. Flow of Modeling

Figure 4 illustrates the workflow of the IndoBERT modeling process for generating embeddings of tourism destination descriptions. The process begins with importing the training dataset and reading the tourism destination

dataset. Next, the dataset is set up and preprocessed into the required format. During the training stage, the indobenchmark/IndoBERT-base-p1 model was fine-tuned using the Sentence Transformers framework. The optimization applied the MultipleNegativeRankingLoss, where positive sentence pairs are encouraged to have higher similarity, while negative pairs are pushed further apart. The similarity between sentence embeddings was measured using cosine similarity, defined as:

$$Cosine\ Similarity(u, v) = \frac{u.v}{\|u\|\ \|v\|} \quad (1)$$

where u and v are embedding vectors of two sentences. After the training process, the fine-tuned IndoBERT model generated vector representations of the tourism destination descriptions. The sentences were first tokenized, processed through IndoBERT, and then pooled into fixed-length embeddings. Finally, the embeddings were stored in a NumPy array (.npy format), which was later used in the recommendation phase to compute similarity between user preferences and tourism destination descriptions.

E. Evaluation

The evaluation stage was used to determine the performance and suitability of the model that had been built in the modeling stage [23], [24], [25]. The evaluation conducted in this study aimed to determine the similarity value using cosine similarity, precision, recall, and F1-score. The calculation of the similarity value was performed on the results of the numerical representation of the user input and the encoded description [26], [27]. Furthermore, the description encoding methods used were fine-tuned IndoBERT, TF-DF, and the composition of IndoBERT with TF-IDF. **Figure 5** showed the evaluation stages that was conducted in this study.

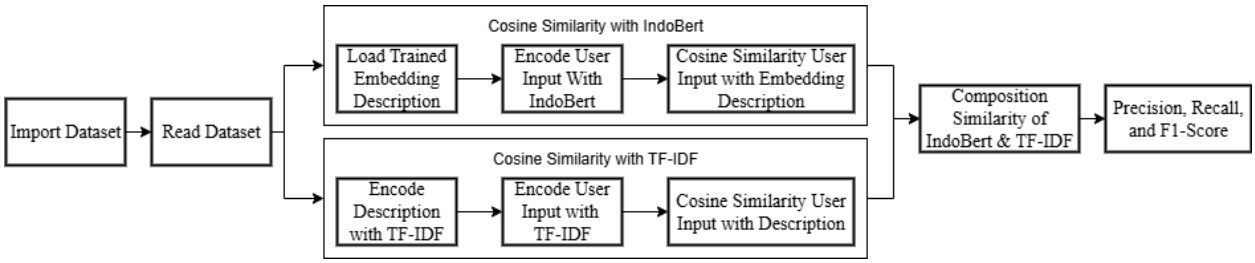

Figure 5. Flow of Evaluation

The evaluation process began with importing and reading the tourism destination dataset. For the IndoBERT pathway, the system loaded the trained embedding descriptions, then encoded user input with IndoBERT, and calculated cosine similarity between user input and embedding descriptions to capture semantic and contextual meaning. In the TF-IDF pathway, both destination descriptions and user input were encoded into numeric vectors using TF-IDF, and cosine similarity was calculated to identify literal keyword-based similarities.

After both similarity values were obtained, the results from IndoBERT and TF-IDF were combined through a weighted composition, allowing the model to integrate semantic context with keyword-level matching. The combined similarity values were then evaluated using standard performance metrics, including precision, recall, and F1-score, to measure the accuracy and effectiveness of the recommendation results. This approach ensured that the system was capable of producing recommendations that were both contextually relevant and quantitatively validated.

F. Deployment

The deployment stage was used to implement the model that had been built and conduct trials [7], [28]. Recommendation system implementation was conducted on a website-based application. During the process, the deployment results were tested using Black Box Testing. The website design was presented using a wireframe, as shown in **Figure 6**.

Figure 6 showed the website layout, which was divided into the home page, namely tourism destination, and travel personalization. The home page showed a list of tourism destination options as it pages allowed user to search for various destination. The travel personalization page enabled user to personalize tourism destination selections. In addition, user input text on the travel personalization page to receive recommended tourism destination modified to personal preferences.

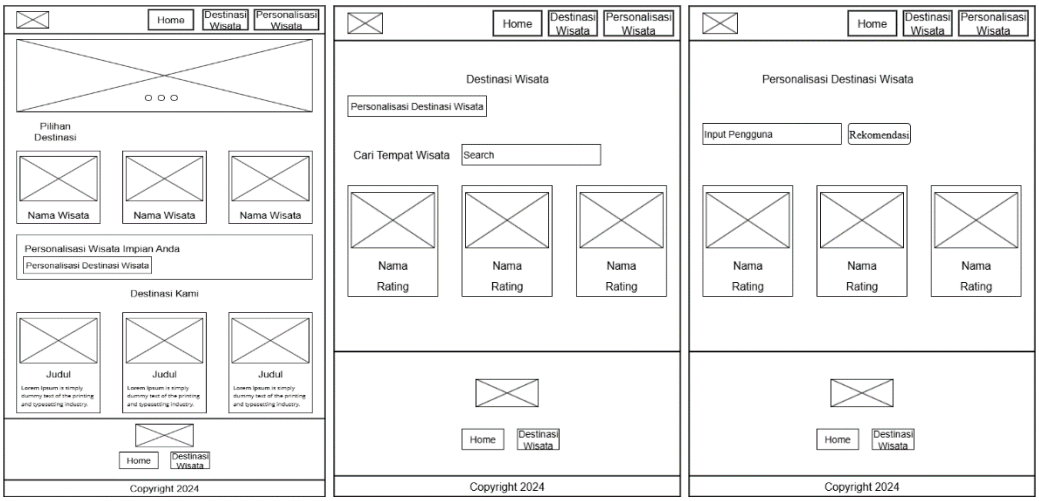

Figure 6. Website Design

Results and Discussion

The result was recommendation system website that used a combination of the IndoBERT and TF-IDF methods. The formation of recommendation website went through the modeling, evaluation, and deployment stages. The outcomes of the modeling stage included training loss values and description embedding storage. The evaluation stage produced cosine similarity values of the IndoBERT method, TF-IDF, and the composition of IndoBERT with TF-IDF. The deployment stage led to the appearance of tourism destination recommendation system website in Indonesia and the outcomes of the website trial.

A. Modeling Results

During the analysis, the IndoBERT model was trained over 25 epochs, with the learning rate gradually increased from 0 to a maximum value. To define the training objective, a tuple list was formed, where the train_data loader supplied the training data batches, train_loss served as the loss function, and the batch_size was set to 8. The results of the model training obtained during the process were shown in **Figure 7**.

```
# Membuat DataLoader
train_dataloader = DataLoader(train_examples, shuffle=True, batch_size=8)

# Menggunakan MultipleNegativesRankingLoss untuk SimCSE
train_loss = losses.MultipleNegativesRankingLoss(model=model)

# Pengaturan parameter pelatihan
num_epochs = 25
```

```
# Melatih model
model.fit(train_objectives=[(train_dataloader, train_loss)],
          epochs=num_epochs,
          output_path='/content/output/sentence-transformer-indobert')
```

[1475/1475 17:33, Epoch 25/25]

Step	Training Loss
500	0.005500
1000	0.000200

Figure 7. Result of Modelling Training

Figure 7 showed the training results at step 500 and step 1000. At step 500, the training loss value was 0.005500, and at step 1000, the training loss value was 0.000200. This loss value was relatively small, signifying that the model had begun to learn from the training data. At step 1000, the loss value dropped significantly, as this decrease in the loss value showed that the model was improving at predicting the training data, and the resulting error was getting minimal. During the analysis, the encoding process was performed after the model training procedure, and the SentenceTransformer library was used to generate sentence embedding. The results of the descriptive sentence embedding were shown in **Figure 8**, and then the sentence embedding results were saved.

Figure 8 showed the embedding results of descriptive sentences consisting of 470 descriptions and having an embedding dimension of 768, as all the results were stored in the description_embedding.npy file. The description embedding file would be used to generate the semantic embedding of travel descriptions.

```
[ ] import numpy as np

    description_embeddings = model.encode(descriptions)
    print(description_embedding.shape)

Shape of description embeddings: (470, 768)

First embedding vector (full):
[-0.8524349   0.04381678 -0.18910727 -0.20119072 -0.23659652 -0.21754566
  0.15196317  0.10610091  0.18474892  0.7006383   0.07894683  0.454226
  0.3130609  -0.35596114 -0.563652   -0.2905335   0.16165836 -0.12220635
 -0.69273514 -0.07750231  0.7059071  -0.45581222 -0.5084014   0.50435835
 -0.5512896  -0.26621917 -0.30842444  0.19285373  0.50888026 -0.52892363
 -1.1106861   0.16994613 -0.3706863   0.4786598  -0.3170965  -0.02025642
  0.29930118  0.58115065  0.19588475 -0.14168955 -0.2821087   0.16611217
  0.488354    0.2218935   0.17965649  0.3980641   0.22920948 -0.1818057
 -0.02058106  0.11428124  0.4713986   0.44849586  0.2504323   0.1655404

np.save('/content/description_embeddings.npy', description_embeddings)
```

Figure 8. Result of Description Embedding

B. Evaluation Results

The evaluation was conducted by examining the cosine similarity results of the IndoBERT, TF-IDF, and IndoBERT combined with TF-IDF methods. The steps taken to determine the IndoBERT cosine value included encoding user input using the IndoBERT method, then finding the semantic similarity value of sentences by applying cosine similarity from the descriptive sentence embedding results in the dataset with the descriptive embedding. The IndoBERT cosine similarity values were shown in **Table 2**.

Table 2. The Highest Cosine Similarity Value of IndoBERT

Place_Name	Rating	Description	Cosine Similarity
Kyotoku Floating Market	4.5	*Kyoutoku Floating Market Bandung, sebuah wisata ...*	0.407059
Kawasan Wisata Sosrowijayan	4.0	*Tempat wisata Sosrowijayan memang selama ini ....*	0.361653
Pantai Greweng	4.6	*Di Kabupater Gunungkidul, tidak sulit memilih ....*	0.354918
The Great Asia Afrika	4.4	*Kota Bandung tampaknya tidak pernah kehabisan ....*	0.354304
Chingu Café Little Seoul	4.5	*Selain popular karena memiliki pemandangan yang ...*	0.345106

Table 2 showed the highest cosine similarity value, 0.407059, for tourism destination Kyotoku Floating Market. The high cosine similarity value signified that the embedding of the user input sentence and the description were similar. The IndoBERT method had a high cosine similarity because it was able to capture syntactic and semantic information, including nuances of meaning as well as relationships between words. Furthermore, the process steps to determine the TF-IDF cosine value included encoding the description as well as user input with the TF-IDF method, and then calculating the cosine similarity. The TF-IDF cosine similarity value analyzed during the process was shown in **Table 3**.

Table 3. The Highest Cosine Similarity Value of TF-IDF

Place_Name	Rating	Description	Cosine Similarity
Puncak Kebun Buah Mangunan	4.6	*Berlibur di pegunungan memang menyenangkan ...*	0.202009
Pantai Nglambor	4.4	*Pantai Nglambor adalah sebuah pantai eksotis ...*	0.187289
Museum Mpu Tantular	4.4	*Museum Negeri Mpu Tantular adalah sebuah museum ....*	0.180255
Museum Tengah Kebun	4.6	*Museum di Tengah Kebun adalah sebuah museum ....*	0.178223
Kyotoku Floating Market	4.5	*Kyoutoku Floating Market Bandung, sebuah wisata ...*	0.170649

Table 3 showed the highest cosine similarity value, 0.202009, with the name Puncak Kebun Buah Mangunan as tourism destination. The cosine similarity value of the TF-IDF method was lower compared to the IndoBERT method because the TF-IDF method only captured literal similarities based on the words that appeared, without grasping the semantic meaning and context of the sentence. Subsequently, a composition of the IndoBERT and TF-IDF methods was conducted by combining the two methods. The results of the cosine similarity values from the IndoBERT and TF-IDF methods were shown in **Table 4**.

Table 4. The Highest Cosine Similarity Value of IndoBERT & TF-IDF Composition

Place_Name	Rating	Description	Cosine Similarity
Kyotoku Floating Market	4.5	*Kyoutoku Floating Market Bandung, sebuah wisata ...*	0.288854
Puncak Kebun Buah Mangunan	4.6	*Berlibur di pegunungan memang menyenangkan ...*	0.261163
Chingu Café Little Seoul	4.5	*Selain popular karena memiliki pemandangan yang ...*	0.256293
Wisata Kaliurang	4.4	*Jogja selalu menarik untuk dikulik, terlebih ....*	0.235503
The Great Asia Africa	4.4	*Kota Bandung tampaknya tidak pernah kehabisan ....*	0.232910

Table 4 showed the highest cosine similarity value, 0.288854, for tourism destination Kyotoku Floating Market. The average cosine similarity value was between the IndoBERT and TF-IDF cosine similarity results. This was because the IndoBERT and TF-IDF composition methods captured semantic meaning as well as context in sentences, and also captured literal similarities based on the words that appeared. To ensure a more rigorous evaluation, quantitative metrics were calculated based on cosine similarity thresholds of IndoBert & TF-IDF Composition. To ensure a more rigorous evaluation, quantitative metrics were calculated based on cosine similarity thresholds of IndoBERT & TF-IDF composition. The threshold values could be determined and varied to control the trade-off between precision and recall, allowing the system to balance accuracy and coverage in recommendations. Previous studies have also applied different similarity thresholds to optimize retrieval and classification tasks depending on the dataset characteristics and evaluation goals [29], [30].

Table 5. The Precision, Recall, and F1-Score of IndoBERT & TF-IDF Composition

Threshold	Precision	Recall	F1-Score
0.20	1.0	1.0	1.0
0.25	1.0	0.6	0.75
0.27	1.0	0.2	0.333
0.30	0	0	0

Table 5 show the threshold of 0.20, the system achieved precision, recall, and F1-score values of 1.0, indicating that all recommended destinations were relevant and correctly retrieved. When the threshold was increased to 0.25, the precision remained at 1.0, but recall dropped to 0.6 with an F1-score of 0.75, as fewer destinations met the stricter similarity criterion. At a threshold of 0.27, only one destination was retrieved, resulting in a recall of 0.2 and an F1-score of 0.333, while at 0.30 or higher, no recommendations were produced. These results highlight that a threshold of 0.20 provides the best balance between recall and precision, ensuring that the recommendation system remains both accurate and comprehensive.

After obtaining the cosine similarity values from the IndoBERT and TF-IDF methods, the highest value was used to generate recommendation, which were then returned in JSON format. An API was created using the POST method to obtain user input in the form of a user description. During the process, postman was used as an application to perform hits on the created API URL. **Figure 9** showed an example of the results of recommendation API using Postman.

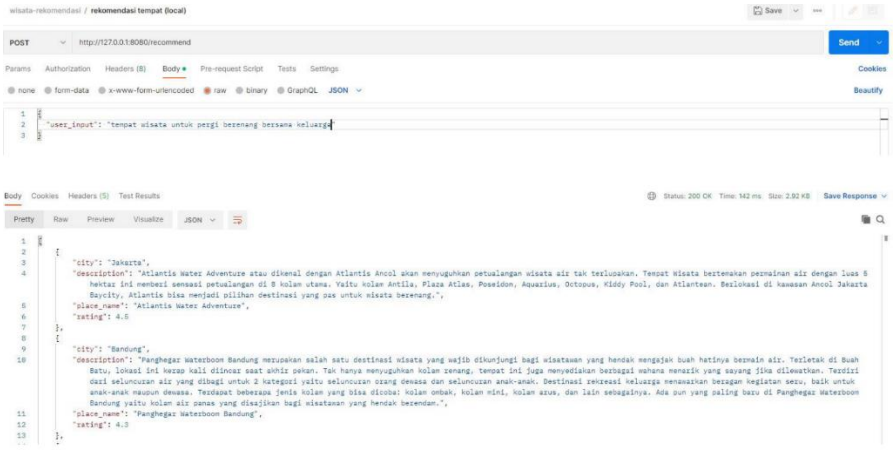

Figure 9. Example of Tourism Destination Recommendation Result

Figure 9 showed an example of tourism destination recommendation in JSON format as the results were references that matched user input. The JSON recommendation included the city name, tourism destination description, name, and rating.

C. Deployment Results

The deployment results represented the implementation of the user interface designed in **Figure 6**. The website consisted of a home page, tourism destination, and travel personalization. The home page contained tourism destination information, such as the total number of destination, travel categories, and the locations of tourism destination that could be visited, shown in **Figure 10**.

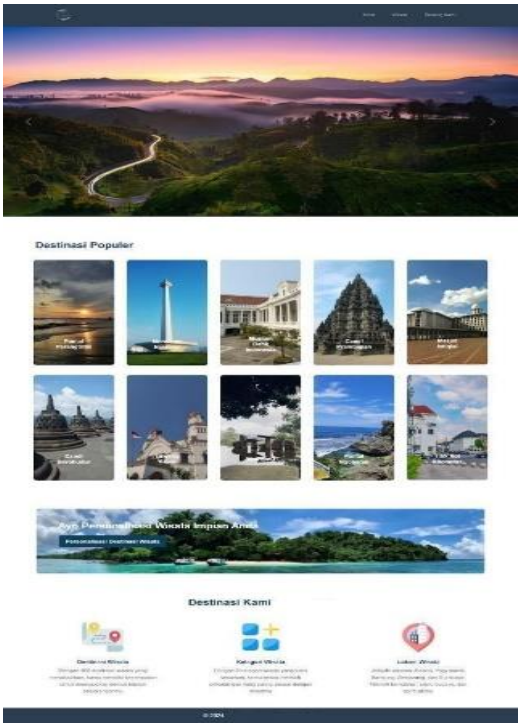

Figure 10. Home Page of Website

Figure 10 showed a home page where user could view popular tourism destination and images of the places. User was also able to search for tourism destination based on personal preferences using tourism destination page in **Figure 11**.

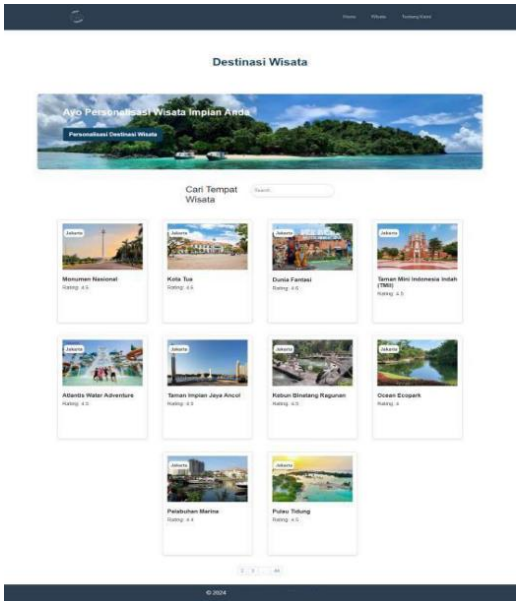

Figure 11. Page of Tourism Destination

In Figure 11, there was tourism destination page that user could use to search for destination based on personal preferences. When the user inputted the name as well as description of tourism destination, the rating appeared based on the input of user. In addition, user could also receive tourism destination recommendation from the travel personalization page in Figure 12.

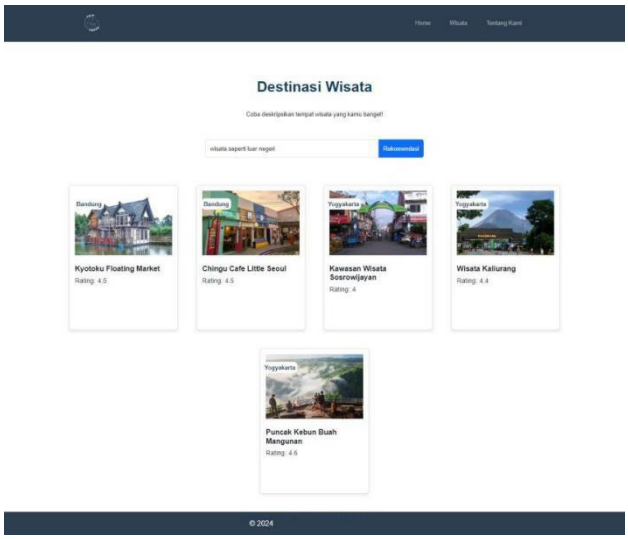

Figure 12. Page of Travel Personalization

Figure 12 showed a travel personalization page that the user could use to get tourism destination recommendation based on words or phrases inputted. The words or phrases were not names of tourist destination, as recommendation system matched the inputted sentence with the corresponding destination. This recommendation system implemented the IndoBERT and TF-IDF composition methods during the process.

D. Testing Results

The website was tested using the Black Box Testing method as it was applied to determine how all features were functioning properly. The test results included user testing, expected results, and a description of the results. Table 5 showed the results of the website test using the Black Box Testing method.

Table 6. Website Testing Result using Black Box Testing

Features	Testing Result	Expected results	Description
Home Page	User opens the home page	The home page displays destination options, the total number of destination, destination categories, and destination locations.	Success
Tourism Destination Page	User opens tourism destination page	Tourism destination page displays a list of available destination.	Success
Travel Personalization Page	User inputs tourism characteristics and clicks recommendation button	The travel personalization page displays input fields, recommendation buttons, and a list of processed travel recommendation.	Success

The website testing conducted in Table 6 showed that all page's function properly. Recommendation provided on the personalization page also successfully signified recommendation based on the travel characteristics entered by the user.

Conclusion

In conclusion, this study demonstrated that the composition of IndoBERT and TF-IDF methods was effective in generating tourism recommendations. The evaluation results showed that at a cosine similarity threshold of 0.20, the system achieved precision, recall, and F1-score values of 1.0, indicating optimal performance in retrieving relevant destinations. At higher thresholds (e.g., 0.25 and 0.27), recall and F1-score values decreased, suggesting that a lower threshold provides a better balance between accuracy and coverage. Furthermore, the developed website-based recommendation system successfully delivered outputs consistent with user input, and all features functioned properly as confirmed through Black Box Testing. For future work, this system could be expanded by incorporating larger and

more diverse datasets, integrating user feedback for adaptive learning, and comparing performance with other state-of-the-art models to further enhance recommendation quality

References

[1] C. Farinha, T. Borges-Tiago, S. Avelar, and A. Baldé, "Technological Adoption in Tourism: Benefits and Barriers Impacting Customer Satisfaction," *Smart Innov. Syst. Technol.*, vol. 439, pp. 363 – 378, 2025, doi: 10.1007/978-981-96-3081-3_23.

[2] N. L. M. Mahendrawati, "Policy on protection of cultural heritage through communal copyright in supporting sustainable tourism," *J. Environ. Manag. Tour.*, vol. 11, no. 4, pp. 920 – 924, 2020, doi: 10.14505//jemt.11.4(44).16.

[3] R. Sharma, "Genetic Algorithm Based Personalized Travel Recommendation System," in *2nd International Conference on Intelligent Data Communication Technologies and Internet of Things, IDCIoT 2024*, 2024, pp. 867 – 874. doi: 10.1109/IDCIoT59759.2024.10467470.

[4] P. Chaparala, P. Nagabhushan, L. T. Ponduri, and P. Kummari, "Mapping Item-wise Rating into Attribute-wise Ratings for Enhanced Personalized Recommendations," in *2024 15th International Conference on Computing Communication and Networking Technologies, ICCCNT 2024*, 2024. doi: 10.1109/ICCCNT61001.2024.10724325.

[5] E. Türkel and A. Alpkoçak, "A New Similarity Method for Tourism Recommendation Systems," *Artif. Intell. Theory Appl.*, vol. 3, no. 2, pp. 77–91, 2023.

[6] G. Chalkiadakis, I. Ziogas, M. Koutsmanis, E. Streviniotis, C. Panagiotakis, and H. Papadakis, "A Novel Hybrid Recommender System for the Tourism Domain," *Algorithms*, vol. 16, no. 4, p. 215, 2023, doi: 10.3390/a16040215.

[7] C. Schröer, F. Kruse, and J. M. Gómez, "A Systematic Literature Review on Applying CRISP-DM Process Model," *Procedia Comput. Sci.*, vol. 181, pp. 526–534, 2021, doi: 10.1016/j.procs.2021.01.199.

[8] M. J. Nodeh, M. H. Calp, and İ. Şahin, "Analyzing and Processing of Supplier Database Based on the Cross-Industry Standard Process for Data Mining (CRISP-DM) Algorithm," *Lect. Notes Data Eng. Commun. Technol.*, vol. 43, pp. 544 – 558, 2020, doi: 10.1007/978-3-030-36178-5_44.

[9] L. Rahmadi, Hadiyanto, R. Sanjaya, and A. Prambayun, "Crop Prediction Using Machine Learning with CRISP-DM Approach BT - Proceedings of Data Analytics and Management," A. Swaroop, Z. Polkowski, S. D. Correia, and B. Virdee, Eds., Singapore: Springer Nature Singapore, 2023, pp. 399–421.

[10] J. Schneider, M. Basalla, and S. Seidel, "Principles of green data mining," in *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2019, pp. 2065 – 2074.

[11] S. Staudinger, C. G. Schuetz, and M. Schrefl, "A Reference Process for Assessing the Reliability of Predictive Analytics Results," *SN Comput. Sci.*, vol. 5, no. 5, p. 563, 2024, doi: 10.1007/s42979-024-02892-4.

[12] M. Pohl, C. Haertel, D. Staegemann, and K. Turowski, "The Linkage to Business Goals in Data Science Projects," in *Australasian Conference on Information Systems, ACIS 2023*, 2023.

[13] D. S. Haryani, O. Abriyoso, and S. Kurnia, "Study of the Effects of Marketing Mix and Service Quality on Tourists' Decisions," *J. Environ. Manag. Tour.*, vol. 13, no. 2, pp. 486 – 496, 2022, doi: 10.14505/jemt.v13.2(58).18.

[14] D. O. Manalu, Y. A. Kusumawati, and C. Tho, "Developing Nusantara Mobile Application to Support Local Tourism in Indonesia," in *Procedia Computer Science*, 2023, pp. 641 – 650. doi: 10.1016/j.procs.2023.10.568.

[15] Z. Ma, B. N. Jørgensen, and Z. G. Ma, "A systematic data characteristic understanding framework towards physical-sensor big data challenges," *J. Big Data*, vol. 11, no. 1, p. 84, 2024, doi: 10.1186/s40537-024-00942-5.

[16] Z. Ma, B. N. Jørgensen, and Z. G. Ma, "DataPro – A Standardized Data Understanding and Processing Procedure: A Case Study of an Eco-Driving Project," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 15271 LNCS, pp. 136 – 154, 2025, doi: 10.1007/978-3-031-74738-0_10.

[17] R. Nisbet, K. McCormick, and G. Miner, *Data Understanding*. 2024. doi: 10.1016/B978-0-443-15845-2.00006-2.

[18] F. Hochkamp, A. A. Scheidler, and M. Rabe, "Review of Maturity Models for Data Mining and Proposal of a Data Preparation Maturity Model Prototype for Data Mining," *Computers*, vol. 14, no. 4, 2025, doi: 10.3390/computers14040146.

[19] J. Jiao and L. Yuan, "GA-PRE: A Genetic Algorithm-Based Automatic Data Preprocessing Algorithm," in *GECCO 2025 - Proceedings of the 2025 Genetic and Evolutionary Computation Conference*, 2025, pp. 1371 – 1378. doi: 10.1145/3712256.3726312.

[20] R. Jevsejev, D. Mazeika, and M. Bereisa, "An Approach for Building IT Support Dataset for Machine Learning Models," *Proc. Open Conf. Electr. Electron. Inf. Sci. eStream*, no. 2025, 2025, doi: 10.1109/eStream66938.2025.11016852.

[21] V. Plotnikova, M. Dumas, and F. P. Milani, "Applying the CRISP-DM data mining process in the financial services industry: Elicitation of adaptation requirements," *Data Knowl. Eng.*, vol. 139, p. 102013, 2022, doi: 10.1016/j.datak.2022.102013.

[22] D. La Torre, D. Liuzzi, M. Repetto, and M. Rocca, "Enhancing deep learning algorithm accuracy and stability using multicriteria optimization: an application to distributed learning with MNIST digits," *Ann. Oper. Res.*, vol. 339, no. 1–2, pp. 455–475, 2024, doi: 10.1007/s10479-022-04833-x.

[23] M. Elkabalawy, A. Al-Sakkaf, E. Mohammed Abdelkader, and G. Alfalah, "CRISP-DM-Based Data-Driven Approach for Building Energy Prediction Utilizing Indoor and Environmental Factors," *Sustainability*, vol. 16, no. 17, 2024, doi: 10.3390/su16177249.

[24] V. J. Raja, M. Dhanamalar, G. Solaimalai, D. L. Rani, P. Deepa, and R. G. Vidhya, "Machine Learning Revolutionizing Performance Evaluation: Recent Developments and Breakthroughs," in *2nd International Conference on Sustainable Computing and Smart Systems, ICSCSS 2024 - Proceedings*, 2024, pp. 780 – 785. doi: 10.1109/ICSCSS60660.2024.10625103.

[25] M. Muntean and F.-D. Militaru, "Metrics for Evaluating Classification Algorithms," *Smart Innov. Syst. Technol.*, vol. 321, pp. 307 – 317, 2023, doi: 10.1007/978-981-19-6755-9_24.

[26] W. Wei, X. Ding, M. Guo, Z. Yang, and H. Liu, "Review of Text Similarity Calculation Methods," *Jisuanji Gongcheng/Computer Eng.*, vol. 50, no. 9, pp. 18 – 32, 2024, doi: 10.19678/j.issn.1000-3428.0068086.

[27] Z. Dai, C. Fu, J. Liu, and Q. Long, "Performance Comparison of Multiple Similarity Algorithms in Meteorological Similarity Assessment," in *Procedia Computer Science*, 2025, pp. 1021 – 1028. doi: 10.1016/j.procs.2025.05.137.

[28] S. Singh, N. Singh, and V. Singh, "Comparative Analysis of Open Standards for Machine Learning Model Deployments," *Lect. Notes Networks Syst.*, vol. 321, pp. 499 – 507, 2022, doi: 10.1007/978-981-16-5987-4_51.

[29] V. Nuipian and J. Chuaykhun, "Book Recommendation System based on Course Descriptions using Cosine Similarity," in *ACM International Conference Proceeding Series*, 2023, pp. 273 – 277. doi: 10.1145/3639233.3639335.

[30] F. V Krasnov, "Embedding-based retrieval: measures of threshold recall and precision to evaluate product search," *Bus. Informatics*, vol. 18, no. 2, pp. 22 – 34, 2024, doi: 10.17323/2587-814X.2024.2.22.34.