

Implementation of Cloud Computing on Android-Based Travel Agent Application Using User-Based Collaborative Filtering Method

Novi Aditia¹, Hari Toha Hidayat (2) (*), Fachri Yanuar Rudi F3

⁽¹⁾ ⁽²⁾ ⁽³⁾ Department of Information and Computer Technology, Politeknik Negeri Lhokseumawe, Indonesia

*Corresponding Author: haritoha@pnl.ac.id

Article info: Received 05/05/2024, Revised 07/06/2025, Accepted 12/07/2024

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Abstract

North Aceh has many natural tourist destinations, historical tourism, culinary tourism, and cultural tourism that can be visited, until now there are 26 tourist villages that have been inaugurated in 2022. The limited knowledge of a person about the destination area is an obstacle for visitors who are traveling to an area, thus the need for a travel agency that can guide visitors by providing the facilities and information needed by visitors. This travel agent application implements cloud computing and the User-Based Collaborative Filtering method which will provide recommendations to users based on rating results. The results of system testing using the blackbox method involving 50 respondents show that 95.4% system success rate and 86.25% user satisfaction level with the performance of the recommendation application. Then testing the performance of the server application in produces throughput using jmeter which shows that on average the application is able to produce a throughput of 14.65 requests per second with a response time in seconds on requests of 1, 10, 50, and 100 with 1 user. Furthermore, a comparison is made between manual calculations and the system against the User-based Collaborative Filtering method in displaying recommendations, obtained accurate and the same recommendation results.

Keywords: travel agent, cloud computing, user-based collaborative filtering, recommendation, north aceh district

1. Introduction

The tourism sector has a major influence on the economic development of a region, especially in driving the rotation of the economy. among others, because the role of tourism both directly and indirectly can create business opportunities, employment opportunities, regional income, and equitable development. Therefore, the development of tourism destinations is currently very important to improve the economy and also expand employment opportunities, so as to encourage the activities of other economic sectors such as the accommodation sector, transportation, home industry, travel agencies, restaurants, tourist information services, and many more.

E-Travel tries to provide travel services from the beginning to the end of the trip accompanied by detailed costs. By booking tickets on e-travel, consumers no longer need to think about traveling by what vehicle, where to stay, and what the purpose of the trip is. Because it is all already a package if we book tickets through e-travel services [1]. Travel packages prepared by travel agencies are basic guidelines for a tourist trip, from costs, time, supplies / preparations and others. in North Aceh Regency in general, this business is not yet professional and still shows conventional ways of working and has not responded perfectly to this business.

The author wants to design and build an android-based Travel Agent application that will recommend tour packages, where every visitor from outside the area can easily find attractive offers regarding facilities and information provided while in North Aceh. With the creation of the application, it is expected to support the development of regional tourism. In order for the application to run optimally, hardware (server) is needed so that the data in the application can run in realtime. The use of this hardware requires a lot of money. With the

development of current technology, the use of Cloud computing is one alternative that can be used as a substitute for the need for hardware in running applications in realtime. Cloud computing implementation is done with the aim of reducing the cost of making applications [2]. Based on these problems, the author wants to make a thesis report with the title "Implementation of Cloud computing in Android-Based Travel Agent Applications using the User-Based Collaborative Filtering Method". Testing of this application will be carried out using blackbox testing and Quality Of Service (QOS).

2. Methods

2.1. Data Collection

The data needed in this research is tourism data in North Aceh Regency which is the object to be used in the research. The research method is divided into several stages.

2.1.1. Literature study

At this stage, searches and studies are carried out on various kinds of literature such as books, scientific journals, and other references related to the title of this research.

2.1.2. Data collection

At this stage the data collection method was carried out using interview techniques conducted with the head of the North Aceh Youth, Sports and Tourism Office regarding the conditions and needs of tourism promotion in North Aceh Regency and observation techniques which are activities to directly observe an object.

2.1.3. Data analysis

At this stage, data analysis is carried out to design a system based on parameters that have been determined to be relevant.

2.2. System Design

System design is logically organized steps to design an application. The system design contains a series of processes that describe the entire application.

2.2.1. Flowchart user based collaborative filtering

The flow of the user-based method process starting from data sampling to providing recommendations can be seen in Figure 1.

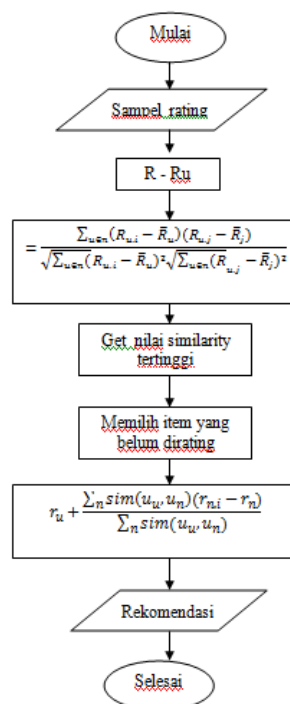


Figure 1. Flowchart User-Based Collaborative Filtering

In the first step, sampling of ratings that have been given by several users, then the rating that has been given by the user is calculated the average value and reduced by the rating per item to further find the *similarity* value. The *similarity* value is calculated based on the correlation of ratings between application users. After obtaining the *similarity* value, then sorted from the highest value. The user who has the highest *similarity* value will be taken all the rating data then will be compared with the user who wants to be recommended, items that have not been rated by the user will be recommended.

2.2.2. Cloud Computing Diagram

The following is a cloud computing design that will be integrated with the application as shown in Figure 2.

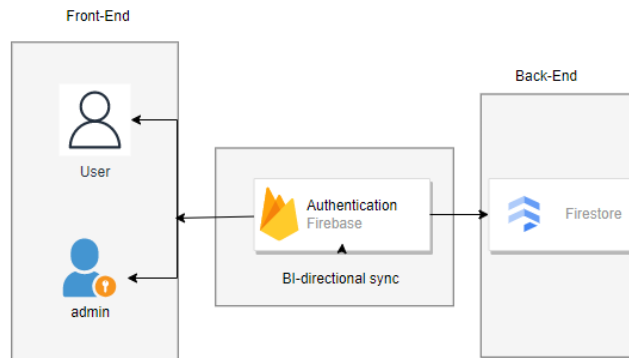


Figure 2. Cloud computing design

Users can synchronize two-way data directly to the *Cloud Firestore* database and enable various supporting capabilities, such as user authentication or identity validation process for users who want to access the system, *Authentication* in this application is used to validate users who want to enter the system with an Email account or Google account by receiving user data input during registration and will later verify the suitability of the data when the user wants to enter the system. All data inputted into the system will be stored in real time in the *cloud firestore database* so that users can synchronize data in *real time* with the *cloud firestore*.

2.2.3. Context Diagram

The following is a context diagram that describes the entire system as can be seen in Figure 3.

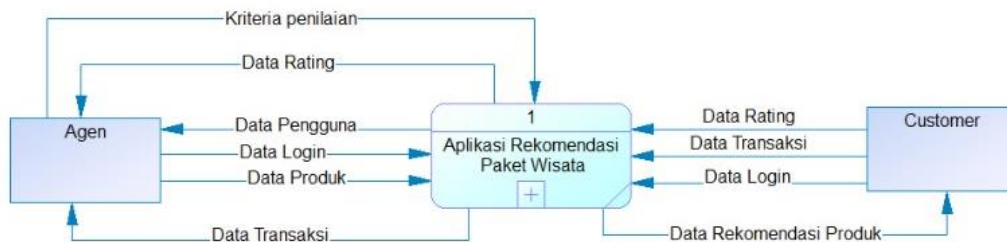


Figure 3. Application System Diagram

Based on Figure 3, the application system diagram describes the system and its environment and interactions. In the system there are two actors, namely Agent and Customer. The data flow that occurs in the Agent is login data, user data, product data, rating data and transaction data. While the data flow that occurs in the Customer is login data, transaction data, rating data and product recommendation data. Login data represents data such as email and keywords. Rating data represents data such as reviews and ratings. Product data represents data such as product names and others. Finally there is transaction data such as total purchases, item data, purchase time and others.

3. Result and Discussions

3.1. Interface Implementation

The implementation of the interface is a system display that has been made to make it easy to interact with the system, and is very helpful for users in understanding the processes carried out by the system.

3.1.1. Registration Page

Account registration page for users who do not have an account from the *Travel Agent* application, for users who do not have an account, they can register first using *Google* or email on the button provided. Users who already have an account are required to log in first. where on this page users and agents must log in first to verify and validate user data.

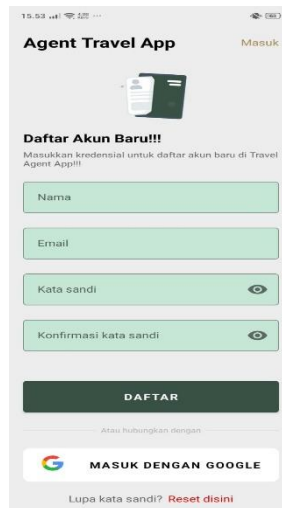


Figure 4. Registration Page Display

3.1.2. Account Page

The account page is a page that contains the account owner's profile, in the form of personal information such as name, email, profile photo that can be edited by the account owner. On the *Customer* account page there is the addition of a digital wallet containing a balance that can be used for shopping.



Figure 5. Account Page Display

3.1.3. Main Page

On this main page contains several menus, namely the travel category menus in which there are four packages offered from sub-categories of tours such as nature and religious tours, nature tours, and religious tours that can be selected by users to view and purchase.

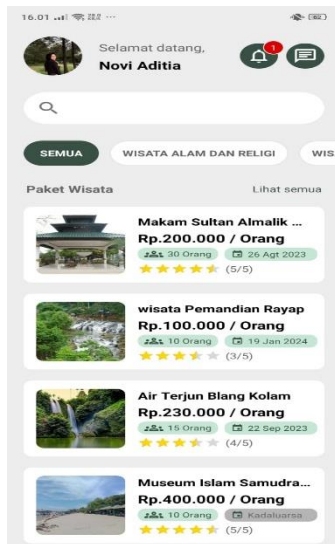


Figure 6. Main Page Display

3.1.4. Recommendation Page

In this recommendation menu, the user will be given a recommendation after rating the tour package using the *user-based collaborative filtering* method.

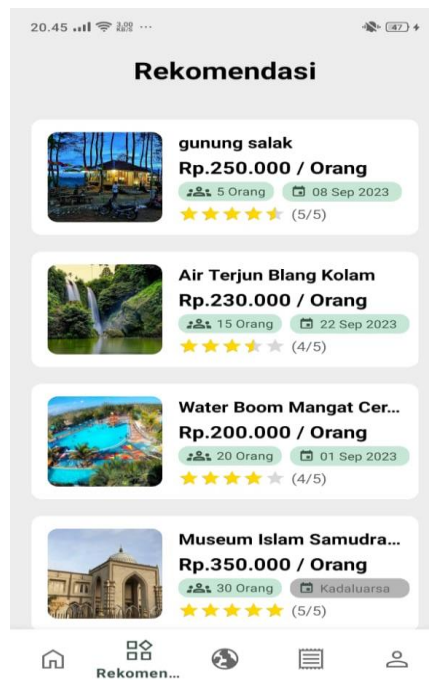


Figure 7. Display of Recommendation Menu

3.1.5. Tour Package Page

The tour package details page contains package information, such as the name of the tour package, tourist attractions and facilities to be obtained, departure date, number of quotas available, location of the meeting point, and location of the tour.



Figure 8. Tour Package Detail Page

3.1.6. Rating and Review Page

The *Review* and *Rating* page contains items that have been rated and reviewed by the user. Users can give stars according to the level of satisfaction with the package offered on a scale of one to five stars and can write reviews and comments about the selected tour package.

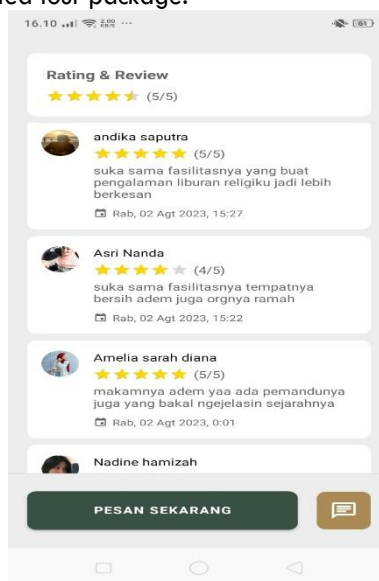


Figure 9. Assessment Page Display

3.2. Implementation of Cloud Computing

Implementation of *Cloud Computing* is the integration of the system with the *Cloud* as storage and the system communication path with the API on the server.

3.2.1. Manage Cloud with Firebase

There are three *firebase* features used in this application, namely *authentication*, used to validate users who want to enter the system with an Email account or Google account by receiving user data input during registration and will later verify the suitability of the data when the user wants to enter into the system.

Rating Page	47	3	$\frac{47}{50} \times 100\%$	94%
QR Code page	47	3	$\frac{47}{50} \times 100\%$	94%
Digital Wallet Page	48	2	$\frac{48}{50} \times 100\%$	96%
Average				95,4%

Based on table 1. it can be seen that testing the feasibility of applications using the *black box testing* method with 50 users the results obtained are 95.4% of successful users and unsuccessful users, the results of unsuccessful use in *black box testing* occur due to network interference or devices that do not support it. Then it can be concluded that this application is feasible to use as a tour package recommendation application.

3.3.2. Quality Of Service Testing

This *Quality of Service (QoS)* test aims to measure the performance of *cloud computing* on travel agent applications in providing recommendations by looking at the *throughput* generated per second using the Jmeter application. This test aims to determine the number of *bits* successfully sent on the network [3]. The results of *black box testing* on this application can be seen in Table 2.

Table 2. *Throughput* Testing

Number of users	Number of looping	Result obtained	
		Throughput	Success
1	1	10.2/sec	100%
1	10	12.8/sec	100%
1	50	17.8/sec	100%
1	100	17.8/sec	100%

3.3.3. Testing User-based collaborative filtering method

User-based collaborative filtering is a technique used to predict items that users may like based on ratings given on previous *items* [4]. Testing the *User-Based collaborative filtering* method on travel agent applications on sample data Risma, Putri, Farhan, and Hani where Putri, Farhan, and Hani have rated all tour packages while Risma has not rated one more tour, namely Lhok seulayang bathing tour as table 3. Rating Data Sample.

a. Sample Data

Rating data samples are used as data to be tested, rating data samples are rating data given by several users on tour packages.

Table 3. Rating Data Sample

User	Pemandin Krueng Tuan	Bahari Beach	Termite Waterfall	Peut Peut Tomb	Ploh	Lhok Seulayang Bathing Place
Risma	5	4	1	4		
Putri	3	1	2	3		3
Farhan	4	3	4	3		5
Hani	3	3	1	5		4

b. *Similarity Value*

Next, calculate the *Similarity* between the target user, namely Risma and other users, at first we will calculate the average rating of all users excluding the Lhok seulayang bathing attraction which is not rated by Risma as shown in Table 4. below.

Table 4. Calculating Similarity

User	Krueng Tuan Baths	Bahari Beach	Termite Waterfall	Peut Ploh Peut Tomb	Average
Risma	5	4	1	4	3,5
Putri	3	1	2	3	2,25
Farhan	4	3	4	3	3,5
Hani	3	3	1	5	3

Furthermore, the calculation of each *item rating* then subtracts the average value of each user to make it easier for us to calculate *similarity* as shown in Table 5.

Table 5. Item Less Average

User	Krueng Tuan Bathing Place	Bahari Beach	Termite Waterfall	Peut Ploh Peut Tomb
Risma	1,5	0,5	-2,5	0,5
Putri	0,75	-1,25	-0,25	0,75
Farhan	0,5	-0,5	0,5	-0,5
Hani	0	0	-2	2

And then calculate the *similarity* value with other users using the *Pearson Correlation* formula as in Table 6.

Table 6. Similarity Value

User	Risma	Princess	Farhan	Hani
Risma		0.3015113 4	- 0.3333333 3	0.7071067 8
Putri	0.301511 34		0.0039998 5	0.4264014 3
Farhan	- 0.3333333 33	0.3015113 4		0.7071067 8
Hani	0.707106 78	0.4161251 9	- 0.7071067 8	

From the calculation results, it can be seen that the highest *similarity* value obtained is between Risma and Hani, which is 0.70710678. The similarity of the two vectors from the *cosine similarity* value is seen from if the greater the value (close to 1), the more similar the two vectors are. Next is the prediction of the rating for the target user, namely Risma on item 5, namely the Lhok seulayang bathing tour. To determine this prediction, we can determine the limit of the similarity value of user similarity, for example we assume the similarity in this dataset if the value is more than 0.3, then we will calculate data from Risma, Putri and Hani. Farhan is not included in the calculation because the *similarity* value obtained by is below 0.3 (the limit of similarity value that we determine). There is no standard rule for determining this limit value.

c. *Prediction Value*

Furthermore, calculations are carried out to find predictions of items that will be recommended to target users by taking the similarity value with the *neighbor*. The results of the Risma user rating prediction value can be seen in Table 7.

Table 7. Rating prediction value

User	Bathroom Krueng Tuan	Beach Bahari	Waterfall Termite	Tomb Peut Ploh Peut	Krueng Seulayang Bathing Place
Risma	4.424	4.574	2.321	5.126	4.425

Based on the results of the *rating* prediction calculation above, the prediction values will be sorted, the highest prediction value is the most recommended for *user rating* by Risma, from the above results the rating prediction value is obtained. the following rating value table in Table 7.

4. Conclusion

Based on the implementation and testing that has been done on the travel agent application it can be concluded that:

1. Based on the results of *blackbox* testing involving 50 respondents showed that the system success rate reached 95.4% and the level of user satisfaction with application performance reached 86.25%.
2. Based on the results of testing the performance of the application server in produces throughput using jmeter which shows that the average application is able to produce a *throughput* of 14.65 requests per second with a response time in seconds on requests of 1, 10, 50, and 100 with 1 user.
3. Based on the results of testing the *user-based collaborative filtering* method that is calculated manually and implemented into the system, the results of the recommendations are accurate and the same.

REFERENCES

- [1] Eggy Ramadhani Djunaedi, "Concept of E-Traveling Application Design Based on Customer Relationship Management Using Rich Internet Application (Case Study: Bayu Buana Travel Services)," p. 8, 2011, [Online]. Available: [http://repository.uinjkt.ac.id/dspace/bitstream/123456789/5869/1/EGGY RAMADHANI DJUNAEDI-FST.pdf](http://repository.uinjkt.ac.id/dspace/bitstream/123456789/5869/1/EGGY_RAMADHANI_DJUNAEDI-FST.pdf)
- [2] M. R. Alfarizi and H. P. Permatasari, "Implementation of Cloud Computing for Parking Reservation Application in Jakarta City," vol. 16, pp. 37-49, 2019.
- [3] A. R. Maulana, H. Walidainy, M. Irhamsyah, F. Fathurrahman, and A. Bintang, "Analysis of Internet Network Quality of Service (Qos) on the E-Learning Website of Syiah Kuala University Based on Wireshark," J. Komputer, Inf. Technol. and Elektro, vol. 6, no. 2, pp. 27-30, 2021, doi: 10.24815/kitektro.v6i2.22284.
- [4] W. E. Pangesti, R. Suryadithia, M. Faisal, and ..., "Collaborative Filtering Based Recommender Systems For Marketplace Applications," Int. J. ..., pp. 1201-1209, 2021, [Online]. Available: <https://ijersc.org/index.php/go/article/view/184>