

Pembuatan Sistem Rekomendasi Pariwisata Yogyakarta Menggunakan Triangle Multiplaying Jaccard

Creating a Yogyakarta Tourism Recommendation System Using Triangle Multiplaying Jaccard

Eva Suhailah¹, Hartatik*²

^{1,2} Program Studi Informatika, Fakultas Ilmu Komputer, Universitas AMIKOM Yogyakarta
Correspondence author email: *hartatik@amikom.ac.id

Abstrak

Yogyakarta adalah satu-satunya daerah Istimewa di Indonesia yang dikenal karena sejarahnya, selain dikenal karena sejarahnya, Yogyakarta termasuk kota yang memiliki berbagai macam jenis dan tempat wisata yang menarik untuk dikunjungi. Yogyakarta memiliki berbagai macam obyek wisata dari berbagai macam jenis, mulai dari wisata budaya, wisata edukasi, wisata alam, wisata belanja dan lain sebagainya., untuk memudahkan wisatawan dalam mencari dan mengetahui rekomendasi tentang wisata yang paling sesuai dengan obyek wisata yang pernah dikunjungi sebelumnya maka Penelitian ini dilakukan untuk membuat suatu sistem rekomendasi obyek wisata menggunakan algoritma Triangle Multiplaying Jaccard, yaitu algoritma yang mencari kesamaan (similarity) untuk tujuan rekomendasi. Berdasarkan perhitungan secara MAE, RMSE dan Confusion Matrix maka didapatkan rata-rata akurasi yang dihasilkan adalah 70% dengan nilai presisi yang relative rendah yaitu 45% dengan recall 1% namun dengan hasil dari F1 Score sebesar 65% dari 20 data yang diolah.

Katakunci: Rekomendasi Pariwisata, Yogyakarta, Trianggle Multiplaying Jaccard, MSE, RMSE, Confusion Matriks

Abstrack

Yogyakarta is the only special region in Indonesia that is known for its history. Apart from being known for its history, Yogyakarta is a city that has various types and interesting tourist attractions to visit. Yogyakarta has a variety of tourist attractions of various types, ranging from cultural tourism, educational tourism, natural tourism, shopping tourism and so on, to make it easier for tourists to find and find out recommendations about the most suitable tourism according to the tourist attractions they have visited before. So this research was conducted to create a tourist attraction recommendation system using the Triangle Multiplaying Jaccard algorithm, namely an algorithm that looks for similarities for recommendation purposes. Based on MAE, RMSE and Confusion Matrix calculations, the resulting average accuracy is 70% with a relatively low precision value of 45% with a recall of 1% but with an F1 Score result of 65% from the 20 data processed.

Keyword: Tourism Recommendation, Yogyakarta, Trianggle Multiplaying Jaccard, MSE, RMSE, Confussion Matriks

1. PENDAHULUAN

Yogyakarta merupakan salah satu kota dengan berbagai macam keistimewaan, dikenal sebagai kota pelajar, dikarenakan Yogyakarta memiliki banyak pusat pendidikan. Dikenal sebagai kota pariwisata dikarenakan Yogyakarta memiliki banyak dan ragam wisata populer dari berbagai jenis wisata diantaranya wisata budaya, museum, wisata alam, pantai dan juga gunung berapi. Yogyakarta ramai dikunjungi wisatawan domestik maupun mancanegara. Banyaknya destinasi wisata yang ada di Yogyakarta sering membuat wisatawan bingung dalam memilih tujuan wisatanya. Berdasarkan masalah tersebut maka dibuatlah sebuah sistem rekomendasi agar dapat membantu wisatawan dalam memilih tujuan wisatanya. Rekomendasi Sistem sangat penting karena kemampuannya untuk memberikan referensi yang disukai oleh customer, membuat strategi marketing, hingga meningkatkan loyalitas pelanggan.

History of article:

Received: September, 2023 : Accepted: September, 2023

Secara umum sistem rekomendasi terbagi dalam 3 teknik yaitu: Content Based Filtering (Hybrid Filtering), Collaborative Filtering dan Knowledge-based Filtering[1]. Content-based Filtering digunakan untuk memberikan rekomendasi kepada masing-masing pengguna yang teregistrasi berdasarkan produk yang di sukai pengguna itu sendiri[2][3], Collaborative Filtering bekerja dengan membangun database (matriks item-pengguna) preferensi untuk item oleh pengguna yang kemudian mencocokkan pengguna dengan minat dan preferensi yang relevan dengan menghitung kemiripan antara profil mereka untuk membuat rekomendasi[4][5], serta Knowledge-based Filtering merekomendasikan item berdasarkan domain pengetahuan yang spesifik tentang bagaimana fitur-fitur yang ada pada suatu item dapat memenuhi kebutuhan pengguna dan berguna bagi pengguna[6]. Diantara ketiga metode tersebut, Collaborative filtering yang paling sering dan paling banyak digunakan karena bekerja sangat efektif dalam menghasilkan rekomendasi[7], memanfaatkan informasi dan preferensi dari user atau item lain untuk memberikan rekomendasi item[8]. Collaborative filtering dapat merekomendasikan item atau produk berdasarkan kumpulan pengguna yang memiliki kesukaan yang sama dengan pengguna tersebut[9] dimana collaborative filtering terdiri dari 2 metode yaitu Neighbourhood dan Model-based[10]. Dimana neighbourhood memiliki kelebihan dalam mengoptimalkan rute terpendek dan dapat meminimalisir penggunaan data dibandingkan Model-based yang masih tergolong baru. Untuk mengatasi kekurangan tersebut bisa digunakan metode *metaheuristik* seperti Tabu Search (TS) [11], *Genetic Algorithm* (GA)[12], *Particle Swarm* (PSO)[13], dan *Firefly Algorithm* (FA)[14].

Namun algoritma *Triangle Multiplying Jaccard* juga pernah dilakukan untuk memberikan rekomendasi beberapa produk dari marketplace yang sesuai dengan pembeli [15]. Selain itu penelitian lain yang juga pernah membahas mengenai penerapan rekomendasi wisata khususnya di Yogyakarta juga pernah dilakukan dengan menggunakan *content based and collaborative filtering* [16], wisata kuliner bagi para wisatawan di Yogyakarta dengan *Item Based Collaborative Filtering* [17], rekomendasi tempat untuk parkir di kota lama Semarang dengan *Collaborative Filtering* [18].

2. METODE PENELITIAN

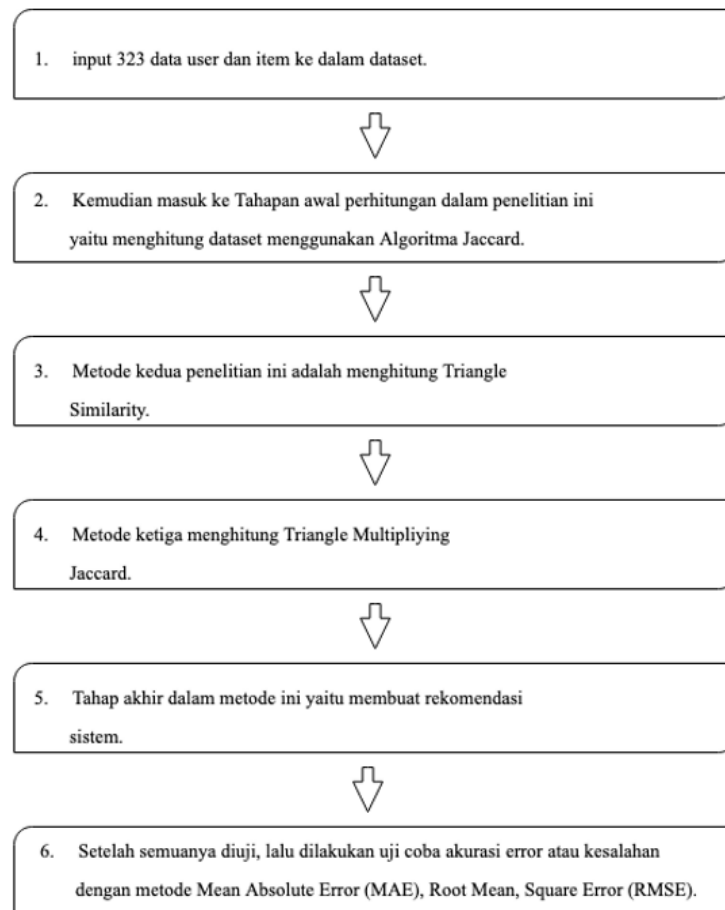
Alur Penelitian

Gambar 1 adalah tahapan penelitian yang dilakukan dimana pada tahap pertama peneliti akan melakukan inputan terhadap 323 data user dan item ke dataset, dataset diambil dari kuisioner data pengguna yang pernah melakukan kegiatan wisata di Yogyakarta dengan output angka dimana angka tersebut merupakan jumlah berapa kali pengguna mengunjungi wisata tersebut. Rangkuman mengenai dataset yang digunakan terdapat pada Tabel 1.

Tabel 1. Dataset yang digunakan

No	Nama	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	...	24	25
1	Yahya Sugandi	0	0	0	4	0	0	3	0	4	0	0	0	4	5	0	0	0	0	3	3	..	0	4
2	Nurul Hidayati	3	5	4	4	5	5	4	4	5	5	4	5	4	5	4	5	5	5	3	5	..	5	4
3	Fariz Aditama	3	4	4	0	0	4	4	0	0	0	3	0	3	3	1	0	0	4	2	2	..	0	0
4	Remu Tapitahrara	3	0	5	5	0	0	0	0	0	0	3	0	5	0	0	4	0	4	4	4	..	5	4
5	Fadeli Muhammad	3	3	2	3	4	4	3	3	3	5	4	5	4	4	4	0	3	5	4	3	..	5	4
..	...	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
323	Numa Dzulianingsih	2	4	3	0	0	0	0	3	0	0	5	0	4	3	0	0	3	0	0	4	..	5	3

Kemudian akan dilakukan proses perhitungan awal dengan menggunakan algoritma jaccard, selanjutnya akan dihitung perhitungan kedua dengan menghitung triangle similarity yang selanjutnya akan dilakukan perhitungan triangle multiplaying jaccard yang pada akhirnya akan dihasilkan rekomendasi beberapa tempat rekomendasi yang berada di Yogyakarta. Selanjutnya untuk mengetahui kinerja sistem baik dari sisi akurasi yang dihasilkan, ketepatan sistem dalam menghasilkan output serta recall yang didapatkan akan dilakukan perhitungan dengan menggunakan *MAE*, *RMSE* dan *Confusion Matrix*.



Gambar 1. Alur penelitian yang dilakukan

Algoritma Triangle Multiplaying Jaccard (TMJ)

Algoritma Triangle Multiplying Jaccard (TMJ) adalah penggabungan persamaan Segitiga dan Jaccard sebagai perkalian untuk membentuk satu ukuran tunggal[15]. Indeks Jaccard adalah ukuran kemiripan dan keragaman himpunan sampel[18]. Namun, keduanya juga secara umum mengambil rasio Irisan per Gabungan atau *Intersection over Union/IoU*. Adapun persamaan irisan ini terdapat pada persamaan (1) [18]

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad \dots \text{persamaan (1)}$$

Dimana:

A = vector A

B = vector B

Sedangkan persamaan yang digunakan pada algoritma *triangle similarity* terdapat pada persamaan (2) [15]

$$Triangle(i_j, i_q) = Triangle(\overrightarrow{OA}, \overrightarrow{OB}) = 1 - \frac{|AB|}{|OA| + |OB|} \quad \dots \text{persamaan (2)}$$

Dimana:

OA = vector ij OA

OB = vector iq OB

Triangle hanya mempertimbangkan pengguna *co-rating*. Untuk memberikan informasi lebih lanjut tentang pengguna *non-co-rating*, selanjutnya menggabungkan ukuran *Jaccard* untuk meningkatkan segitiga, sehingga mendapatkan persamaan (3)

$$TMJ(i_j, i_q) = Triangle(i_j, i_q) \times Jaccard(i_j, i_q) \quad \dots \text{persamaan (3)}$$

Adapun rangkuman mengenai tahapan perhitungan Algoritma *Triangle Multiplying Jaccard* terbagi menjadi 6 langkah yaitu [15]

1. Inisialisasi
2. Membaca Dataset
3. Penghitungan Nilai Jaccard
4. Triangle Similarity
5. Triangle Multiply Jaccard
6. Hasil.

3. HASIL DAN PEMBAHASAN

Implementasi Data

Berikut ini adalah hasil program yang dibuat dengan menggunakan algoritma triangle multiplying jaccard

- a. Inisialisasi library surprise dan integrasi dengan Google Drive
Gambar 2 merupakan installasi library surprise python yang berfungsi sebagai library untuk menghitung similarity



```

Installasi Libraries

[ ] !pip install --quiet surprise

Koneksi dengan Google Drive

[ ] from google.colab import drive
    drive.mount('/content/drive')
    %cd /content/drive/MyDrive/TMJ

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive/MyDrive/TMJ")
  
```

Gambar 2. Installasi library

- b. Import libraries yang diperlukan untuk melakukan perhitungan triangle multiplying jaccard
Selanjutnya akan dilakukan installasi libraries yang diperlukan untuk melakukan perhitungan terhadap persamaan-persamaan yang digunakan pada *Algoritma Triangle Multiplying Jaccard*. Gambar 3 merupakan perhitungan *Algoritma Triangle Multiplying Jaccard*

Import Libraries

```
[ ] import os, difflib, random
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from surprise import BaselineOnly, Dataset, accuracy, Reader, AlgoBase, NMF
from surprise.model_selection import train_test_split, cross_validate, GridSearchCV
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
from imutils import paths
```

Gambar 3. Perhitungan Algoritma Triangle Multiplying Jaccard

c. Import dataset

Tahap berikutnya adalah melakukan import terhadap dataset wisata, rating, dan user. Hasil dari import tersebut terdapat pada Gambar 4.

Import Dataset

```
[ ] rating = pd.read_csv('rating.csv')
wisata = pd.read_csv('wisata.csv')
user = pd.read_csv('user.csv')
rating.head(10)
```

	wisata_id	user_id	rating
0	1	1	0
1	1	2	3
2	1	3	3
3	1	4	3
4	1	5	3
5	1	6	3
6	1	7	2

Gambar 4. Import dataset

d. Memeriksa struktur data dari dataset

Selanjutnya untuk memastikan kelengkapan data dan kesesuaian type data maka diperlukan pengecekan dataset. Hasil pengecekan dataset ini terdapat pada Gambar 5.

```
[ ] #info data rating
rating.info()
#info data user
user.info()
#info data wisata
wisata.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8075 entries, 0 to 8074
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  -
0   wisata_id   8075 non-null   int64
1   user_id     8075 non-null   int64
2   rating      8075 non-null   int64
dtypes: int64(3)
memory usage: 189.4 KB
```

Gambar 5. Memeriksa struktur data dari dataset

- e. Convert data dari pandas dataframe ke surprise dataset
Untuk penyesuaian dataset antara library pandas dan surprise maka dilakukan konversi, selanjutnya dilakukan split dataset ke dalam trainset dan testset. Hasil dari split ini terdapat pada Gambar 6.

Convert DataFrame ke Surprise Dataset Kemudian Split Dataset

```
[ ] reader = Reader(rating_scale=(1, 5))
data = Dataset.load_from_df(rating[['user_id', 'wisata_id', 'rating']], reader)
trainset, testset = train_test_split(data, test_size=.25)
```

Gambar 6. Converting Dataset

- f. Membuat class perhitungan jaccard
Class Jaccard ini nantinya akan digunakan untuk melakukan penghitungan jaccard. Gambar 7 merupakan class penghitungan yang dilakukan.

Jaccard Similarity

```
[ ] class Jaccard(AlgoBase):
    def __init__(self):

        # Always call base method before doing anything.
        AlgoBase.__init__(self)

    def estimate(self, u, i):
        intersection = u & i
        jaccard = intersection/(u+i-intersection)
        return jaccard
```

Gambar 7. Class Penghitungan Jaccard

- g. Training jaccard similarity
Selanjutnya akan dilakukan training jaccard similarity dengan metode SGD, dimana nilai learning rate 0,00005 dan epoch 10. Hasil training ini terdapat pada Gambar 8.

Jaccard Similarity Model, Training, Prediksi dan dan RMSE,MSE,MAE

```
[ ] bsl_options = {'method': 'sgd', 'learning_rate': .00005, 'epoch': 10}
algo = BaselineOnly(bsl_options=bsl_options)

algo = Jaccard()
algo.fit(trainset)
predictions = algo.test(testset, verbose=False)
RMSE = accuracy.rmse(predictions, verbose=True)
MAE = accuracy.mae(predictions, verbose=True)
MSE = accuracy.mse(predictions, verbose=True)

RMSE: 2.7095
MAE: 2.4190
MSE: 7.3413
```

Gambar 8. Training jaccard similarity

- h. Perhitungan *Triangle Multiplying Jaccard*
Selanjutnya akan dilakukan perhitungan dengan menggunakan Algoritma *Triangle Multiplying Jaccard* seperti pada Gambar 9.

Triangle Multiplying Jaccard Model, Training, Prediksi dan RMSE,MSE,MAE

```
[ ] class TMJ(AlgoBase):
    def __init__(self, sim_options={}, bsl_options={}):

        AlgoBase.__init__(self, sim_options=sim_options,
                           bsl_options=bsl_options)

    def fit(self, trainset):

        AlgoBase.fit(self, trainset)

        # Compute baselines and similarities
        self.bu, self.bi = self.compute_baselines()
        self.sim = self.compute_similarities()

        return self
```

Gambar 9. Perhitungan Algoritma Triangle Multiplying Jaccard

- i. Pembuatan class *Triangle Multiplying Jaccard*, training, prediksi dan perhitungan RMSE

```
# Compute similarities between u and v, where v describes all other
# users that have also rated item i.
neighbors = [(v, self.sim[u, v]) for (v, r) in self.trainset.ir[i]]
# Sort these neighbors by similarity
neighbors = sorted(neighbors, key=lambda x: x[1], reverse=True)
triangle = (1 - (u - i)/(u+i))
intersection = u&i
jaccard = ((intersection)/(u + i - intersection))
tmj = triangle * jaccard
print('The 3 nearest neighbors of user', str(u), 'are:')
for v, sim_uv in neighbors[:3]:
    print('user {0:} with sim {1:1.2f}'.format(v, sim_uv))

bsl_options = {'method': 'als', 'n_epochs': 10, 'learning_rate': .00005}
algo = BaselineOnly(bsl_options=bsl_options)
algo.fit(trainset)
predictions = algo.test(testset)
accuracy.rmse(predictions, verbose=True)
accuracy.mae(predictions, verbose=True)
accuracy.mse(predictions, verbose=True)
```

Estimating biases using als...
 RMSE: 1.6740
 MAE: 1.4445
 MSE: 2.8024
 2.802394106119184

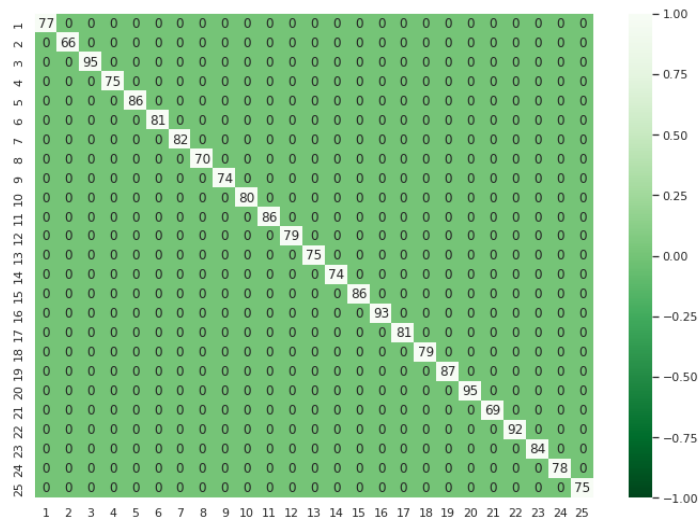
Gambar 10. Class Triangle Multiplying Jaccard

- j. Perhitungan confusion matrix
 Untuk mengetahui kinerja algoritma yang digunakan maka akan dilakukan pengujian secara confusion matriks terhadap data yang dipakai. Gambar 11 merupakan potongan inti perintah ini dengan menggunakan python.

```
[ ] obj_wisata=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25]
cm = confusion_matrix(Y_test, Y_pred)
matrix = pd.DataFrame(data=cm, columns=obj_wisata, index=obj_wisata)
sns.set(rc={'figure.figsize':(11.7,8.27)})
sns.heatmap(data=matrix, vmin=-1, vmax=1, annot=True, cmap=plt.cm.Greens_r)
plt.show()
```

Gambar 11. Confusion Matriks

Jika divisualisasikan maka hasil dari perhitungan *Confusion Matriks* dengan menggunakan python terdapat pada Gambar 12.



Gambar 12. Confussion matriks

- k. Perhitungan precision, recall, F1-Score dan support
Dari hasil pelatihan menggunakan *Triangle Multiplying Jaccard* maka akan dihitung juga nilai dari precision, recall, F-1 score dan support dari masing-masing data.

```
[ ] print(classification_report(Y_test, Y_pred))
```

	precision	recall	f1-score	support
1	1.00	1.00	1.00	77
2	1.00	1.00	1.00	66
3	1.00	1.00	1.00	95
4	1.00	1.00	1.00	75
5	1.00	1.00	1.00	86
6	1.00	1.00	1.00	81
7	1.00	1.00	1.00	82
8	1.00	1.00	1.00	70
9	1.00	1.00	1.00	74
10	1.00	1.00	1.00	80
11	1.00	1.00	1.00	86
12	1.00	1.00	1.00	79
13	1.00	1.00	1.00	75
14	1.00	1.00	1.00	74
15	1.00	1.00	1.00	86
16	1.00	1.00	1.00	93
17	1.00	1.00	1.00	81
18	1.00	1.00	1.00	79
19	1.00	1.00	1.00	87
20	1.00	1.00	1.00	95
21	1.00	1.00	1.00	69
22	1.00	1.00	1.00	92
23	1.00	1.00	1.00	84
24	1.00	1.00	1.00	78
25	1.00	1.00	1.00	75
accuracy			1.00	2019
macro avg	1.00	1.00	1.00	2019
weighted avg	1.00	1.00	1.00	2019

Gambar 13. Precision, Recall, F1-score dan Support

1. Pembuatan function untuk rekomendasi

Function Rekomendasi Wisata

```
[ ] def get_wisata_id(wisata_wisata, metadata):
    existing_wisatas = list(metadata['wisata'].values)
    closest_wisatas = difflib.get_close_matches(wisata_wisata, existing_wisatas)
    wisata_id = metadata[metadata['wisata'] == closest_wisatas[0]]['wisata_id'].values[0]
    return wisata_id

def get_wisata_info(wisata_id, metadata):
    wisata_info = metadata[metadata['wisata_id'] == wisata_id][['wisata_id', 'wisata']]
    return wisata_info.to_dict(orient='records')

def predict_review(user_id, wisata_wisata, model, metadata):
    wisata_id = get_wisata_id(wisata_wisata, metadata)
    review_prediction = model.predict(uid=user_id, iid=wisata_id)
    return review_prediction.est

def generate_recommendation(user_id, model, metadata, thresh=3):
    wisata_wisatas = list(metadata['wisata'].values)
    random.shuffle(wisata_wisatas)
    for wisata_wisata in wisata_wisatas:
        rating = predict_review(user_id, wisata_wisata, model, metadata)
        if rating >= thresh:
            wisata_id = get_wisata_id(wisata_wisata, metadata)
            return get_wisata_info(wisata_id, metadata)
    else:
        return 'No Rekomendasi'
```

Gambar 14. Pembuatan function untuk rekomendasi

m. Rekomendasi wisata masing-masing pengguna / user

REKOMENDASI

```
[ ] for t in range(1,3):
    print("=====")
    print("Threshold : ",t)
    print("=====")
    for i in range(1, len(user)):
        recom=generate_recommendation(i, algo, wisata,t)
        wst="Tidak Ada Rekomendasi"
        if recom[0]!="N":
            wst=recom[0]['wisata']
        print("User ",i,".",user['nama_lengkap'][i]," : ",wst)

=====
Threshold :  1
=====
User  1 . Nurul Hidayati : Taman Gunung Merapi
User  2 . Fariz Aditama : Pantai Drini
User  3 . Remu Tapitahrara : Pantai Parang Tritis
User  4 . Fadeli Muhammad : Candi Ijo
User  5 . Muhammad Mathla'il Fajri : Pantai Drini
User  6 . Dhanang Mughsen : Pantai Indrayanti
User  7 . Maie Istighosah : Pantai Parang Tritis
User  8 . Ninda Putri Hermawanti : Pantai Pok Tunggal
User  9 . Miladdiena Maharani : Puro Paku Alaman
User 10 . Hannan Asrawi : Pantai Nglambor
User 11 . Wahid fatahilah : Pantai Nglambor
User 12 . Aditya Himawan : Taman Kalirurang
User 13 . Beta Wulandari : Kraton Yogyakarta
```

Gambar 15. Hasil Rekomendasi

Hasil Rekomendasi

Berdasarkan hasil perhitungan diatas, maka rekomendasi yang dapat diberikan untuk rekomendasi tempat wisata di Yogyakarta adalah:

a. Hasil rekomendasi wisata Yogyakarta dengan threshold 1 adalah:

1. User 1. Nurul Hidayati : Taman Gunung Merapi
2. User 2. Fariz Aditama : Pantai Drini
3. User 3. Remu Tapitahrara : Pantai Parang Tritis
4. User 4. Fadeli Muhammad : Candi Ijo
5. User 5. Muhammad Mathla'il Fajri : Pantai Drini
6. User 6. Dhanang Mughsen : Pantai Indrayanti
7. User 7. Maie Istighosah : Pantai Parang Tritis
8. User 8. Ninda Putri Hermawanti : Pantai Pok Tunggal
9. User 9. Miladdiena Maharani : Puro Paku Alaman
10. User 10. Hannan Asrawi : Pantai Nglambor
11. User 11. Wahid fatahilah : Pantai Nglambor
12. User 12. Aditya Himawan : Taman Kalirurang
13. User 13. Beta Wulandari : Kraton Yogyakarta

b. Hasil rekomendasi wisata Yogyakarta dengan threshold 2 adalah:

1. User 1. Nurul Hidayati : Taman Sari
2. User 2. Fariz Aditama : Pantai Siung
3. User 3. Remu Tapitahrara : Kraton Yogyakarta
4. User 4. Fadeli Muhammad : Tidak Ada Rekomendasi
5. User 5. Muhammad Mathla'il Fajri : Kali Biru
6. User 6. Dhanang Mughsen : Pantai Indrayanti
7. User 7. Maie Istighosah : Pantai Indrayanti
8. User 8. Ninda Putri Hermawanti : Kraton Yogyakarta
9. User 9. Miladdiena Maharani : Gunung Purba Nglanggeran
10. User 10. Hannan Asrawi : Candi Ijo
11. User 11. Wahid fatahilah : Tebing Breksi
12. User 12. Aditya Himawan : Candi Ijo
13. User 13. Beta Wulandari : Candi Prambanan

Perhitungan Pengujian

a. Perhitungan MAE

Dalam menghitung ramalan F dengan 3 prediksi / pergerakan. Hasil perhitungan tersebut adalah:

$$MAE = \frac{x_1 + x_2 + x_3}{3} = \frac{14.88149695 + 11.68406571 + 6.61929526}{3} = 7.556399596$$

$$\text{Kemudian nilai MAE nya adalah} = \frac{261.6090103}{17} = 15.38876531$$

b. Perhitungan RMSE

Setelah nilai MAE dihitung maka dilanjutkan dengan menghitung RMSE. Data masih sama dengan data pada tabel data pengujian.

$$RMSE = \sqrt{\frac{6959.178658}{17}} = 20.232732$$

c. Perhitungan Confusion Matriks

Selanjutnya adalah menghitung Confusion matrix. Langkah awalnya adalah menghitung accuracy. Dari 20 data perhitungan KNN, dapat diambil nilai TP, FP, FN dan TN nya. Selanjutnya baru akan dilakukan perhitungan Accuracy yaitu:

$$\text{Akurasi} = \frac{5+9}{5+6+0+9} \times 100\% = 0,7 \times 100\% = 70\%$$

Setelah accuracy didapatkan, dilanjutkan dengan perhitungan precision, yaitu

$$\text{Presisi} = \frac{5}{5+6} \times 100\% = 0,45 \times 100\% = 45\%$$

Selanjutnya perhitungan Recall, yaitu

$$\text{Recall} = \frac{5}{5+0} \times 100\% = 1 \times 100\% = 1\%$$

d. Perhitungan F1-Score

Perhitungan akhir dari Confusion Matrix adalah F-1 Score

$$\text{F1-Score} = 2 \times \frac{(\text{recall} \times \text{precision})}{(\text{recall} + \text{precision})} = 2 \times \frac{(1 \times 0,45)}{(1 + 0,45)} = 0,620689655 \times 100\% = 62\%$$

Berdasarkan hasil diatas diketahui bahwa nilai recall masih cenderung sangat rendah bila dibandingkan dengan nilai akurasi, hal ini memungkinkan bahwa Algoritma *Triangle Multiplying Jaccard* masih belum dapat bekerja dengan maksimal. Dimana akurasi yang tinggi dengan kumpulan data yang tidak seimbang akan menjadikan *output* yang dihasilkan tidak akan berarti apa-apa karena hanya akan memberikan rekomendasi yang paling umum saja. Dengan kelas yang tidak seimbang (*un-balancing dataset*) maka dapat dengan mudah menghasilkan nilai akurasi yang tinggi tanpa benar-benar menghasilkan rekomendasi yang berguna.

4. KESIMPULAN

Berdasarkan hasil penelitian maka dapat diambil kesimpulan bahwa Algoritma *Triangle Multiplying Jaccard* mampu memberikan rekomendasi terhadap berbagai wisata yang berada di Yogyakarta sesuai dengan inputan masing-masing user namun tidak bisa maksimal. Hal ini dilihat dari perhitungan kinerja sistem berdasarkan MAE, RMSE dan *Confussion Matriks* yang menunjukkan nilai kurang dari 80% yaitu 7,5 untuk MAE, 20,23 untuk nilai RMSE, serta accuracy 70% precision 45% dan recall 1%.

5. SARAN

Adapun saran yang dapat diberikan dalam penelitian ini yaitu sebaiknya dilakukan data balancing untuk mengatasi imbalance dataset pada hasil rekomendasi wisata kuliner sistem.

DAFTAR PUSTAKA

- [1] M. Alkaff, H. Khatimi, and A. Eriadi, "Sistem Rekomendasi Buku pada Perpustakaan Daerah Provinsi Kalimantan Selatan Menggunakan Metode Content-Based Filtering," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 20, no. 1, pp. 193–202, 2020, doi: 10.30812/matrik.v20i1.617.
- [2] A. Muliawan, T. Badriyah, and I. Syarif, "Membangun Sistem Rekomendasi Hotel dengan Content Based Filtering Menggunakan K-Nearest Neighbor dan Haversine Formula," *Technomedia J.*, vol. 7, no. 2, pp. 231–247, 2022, doi: 10.33050/tmj.v7i2.1893.
- [3] L. Tommy, C. Kirana, and V. Lindawati, "Recommender System Dengan Kombinasi Apriori Dan Content-Based Filtering Pada Aplikasi Pemesanan Produk," *J. Teknoinfo*,

- vol. 13, no. 2, p. 84, 2019, doi: 10.33365/jti.v13i2.299.
- [4] B. Prasetyo, H. Haryanto, S. Astuti, E. Z. Astuti, and Y. Rahayu, "Implementasi Metode Item-Based Collaborative Filtering dalam Pemberian Rekomendasi Calon Pembeli Aksesoris Smartphone," *Eksplora Inform.*, vol. 9, no. 1, pp. 17–27, 2019, doi: 10.30864/eksplora.v9i1.244.
 - [5] A. Alim, A. Soluchin, and P. Painem, "Rekomendasi Keterampilan Teknologi Informasi Menggunakan Metode User-Based Collaborative Filtering dan Log-Likelihood Similarity," *Cogito Smart J.*, vol. 6, no. 2, 2020, doi: 10.31154/cogito.v6i2.234.141-154.
 - [6] M. Muhith, D. Hartanti, J. Maulindar, P. Pertama, P. Kedua, and P. Ketiga, "Sistem Rekomendasi Pemilihan Paket Instalasi CCTV menggunakan Metode Knowledge Based pada CCTV Center Delanggu," *Pros. Semin. Nas. Teknol. Inf. dan Bisnis*, pp. 222–227, 2022.
 - [7] F. W. Ardiyanto *et al.*, "Diversity Balancing pada Two-stage Collaborative Filtering untuk Mengatur Less Popular Item dalam Domain Film," in *eProceedings of Engineering*, 2023, vol. 10, no. 3, pp. 3455–3463.
 - [8] I. M. I. Subroto, S. Mulyono, R. Firmansyah, M. Qomaruddin, and E. N. B. Susila, "Sistem Rekomendasi pada Pembelajaran Mobile Menggunakan Metode Cosine Similarity dan Collaborative Filtering," *TRANSISTOR Elektro dan Inform.*, vol. 4, no. 1, pp. 21–28, 2022, doi: 10.30659/ei.4.1.21-28.
 - [9] V. L. Jaja, B. Susanto, and L. R. Sasongko, "Penerapan Metode Item-Based Collaborative Filtering Untuk Sistem Rekomendasi Data MovieLens," *d'CARTESIAN*, vol. 9, no. 2, p. 78, 2020, doi: 10.35799/dc.9.2.2020.28274.
 - [10] F. A. Prayoga and K. Kusnawi, "Smartphone Recommendation System Using Model-Based Collaborative Filtering Method," *J. Tek. Inform.*, vol. 3, no. 6, pp. 1613–1622, 2022, doi: 10.20884/1.jutif.2022.3.6.413.
 - [11] P. I. A. Purwadana, I. M. Candiasa, and I. N. Sukajaya, "Pengembangan Aplikasi Penentuan Rute Pengiriman Barang Berdasarkan Berat dan Time Windows Menggunakan Metode Nearest Neighbour dan Tabu Search," *Maj. Ilm. Teknol. Elektro*, vol. 20, no. 2, p. 299, 2021, doi: 10.24843/mite.2021.v20i02.p14.
 - [12] S. Yulianto, I. G. A. Widyadana, and N. Sepadyati, "Optimasi Rute Pengiriman pada CV. X Menggunakan Metode Evolutionary Algorithm," *J. Tirta*, vol. 10, no. 1, pp. 17–22, 2020.
 - [13] S. R. A. Robbani, "Penyelesaian Urban Transit Routing Problem Menggunakan Algoritma Hyper-heuristics berbasis Modified Particle Swarm Optimization based on Gravitational field Interactions," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 8, no. 4, pp. 1984–1997, 2021, doi: 10.35957/jatisi.v8i4.1233.
 - [14] S. S. Mansur, S. Widowati, and M. Imrona, "Terakreditasi SINTA Peringkat 2 Analisis Performansi Algoritma Firefly dan Tabu Search untuk Optimasi Rute Angkutan Kota," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 5, pp. 884–891, 2020, doi: 10.29207/resti.v4i5.2259.
 - [15] F. R. Hariri and L. W. Rochim, "Sistem Rekomendasi Produk Aplikasi Marketplace Berdasarkan Karakteristik Pembeli Menggunakan Metode User Based Collaborative Filtering," *Teknika*, vol. 11, no. 3, pp. 208–217, 2022, doi: 10.34148/teknika.v11i3.538.
 - [16] A. S. N. S. Ningrum, "Content Based Dan Collaborative Filtering Pada Rekomendasi Tujuan Pariwisata Di Daerah Yogyakarta," *Telematika*, vol. 16, no. 1, p. 44, 2019, doi: 10.31315/telematika.v16i1.3023.
 - [17] H. Hartatik, S. D. Nurhayati, and W. Widayani, "Sistem Rekomendasi Wisata Kuliner di Yogyakarta dengan Metode Item-Based Collaborative Filtering," *J. Autom. Comput. Inf. Syst.*, vol. 1, no. 2, pp. 55–63, 2021, doi: 10.47134/jacis.v1i2.8.
 - [18] A. S. Muarif and E. Winarno, "Sistem Rekomendasi Tempat Parkir di Kota Lama Semarang Menggunakan Collaborative Filtering," *J. Ilm. Univ. Batanghari Jambi*, vol. 22, no. 2, p. 906, 2022, doi: 10.33087/jiubj.v22i2.2066.