



Aplikasi Sistem Rekomendasi Lokasi Wisata Candi dan UMKM di Yogyakarta Berbasis Android dengan Metode Hybrid

Rifki Alfarro Zaidan, Tomi Tri Sujaka, I Nyoman Switrayana

Universitas Bumigora, Mataram, Indonesia

Abstrak

Yogyakarta, sebuah kota yang kaya akan warisan budaya dan keindahan alam, menarik banyak wisatawan setiap tahunnya. Namun, banyaknya pilihan destinasi wisata dan UMKM seringkali membuat wisatawan kesulitan dalam menentukan pilihan mereka. Penelitian ini bertujuan untuk mengembangkan sebuah aplikasi Android yang dapat memberikan rekomendasi lokasi wisata candi dan UMKM di Yogyakarta dengan memanfaatkan metode hybrid. Metode hybrid ini menggabungkan Content-Based Filtering dan Collaborative Filtering untuk memberikan rekomendasi candi yang sesuai dengan preferensi pengguna berdasarkan deskripsi candi dan riwayat kunjungan pengguna lain. Selain itu, metode Regresi Linear digunakan untuk memberikan rekomendasi UMKM terdekat berdasarkan lokasi pengguna. Hasil penelitian menunjukkan bahwa aplikasi ini berhasil memberikan rekomendasi candi dan UMKM yang relevan dan bermanfaat bagi pengguna. Aplikasi ini juga dilengkapi dengan informasi detail tentang candi, termasuk sejarah, tradisi, dan tata cara berkunjung yang santun, serta fitur navigasi yang memudahkan pengguna mencapai lokasi yang direkomendasikan. Dengan demikian, aplikasi ini diharapkan dapat meningkatkan pengalaman wisata di Yogyakarta, mempromosikan UMKM lokal, dan melestarikan warisan budaya.

Abstract

Yogyakarta, a city rich in cultural heritage and natural beauty, attracts many tourists every year. However, the abundance of tourist destinations and MSMEs often makes it challenging for tourists to decide on their options. This study aims to develop an Android application that provides recommendations for temple tourism and MSME locations in Yogyakarta using a hybrid method. This hybrid method combines Content-Based Filtering and Collaborative Filtering to recommend temples that align with user preferences based on temple descriptions and the visitation history of other users. Additionally, Linear Regression is employed to recommend nearby MSMEs based on the user's location. The study results indicate that the application successfully provides relevant and useful recommendations for temples and MSMEs. The application is also equipped with detailed information about the temples, including history, traditions, and proper visiting etiquette, as well as navigation features that make it easier for users to reach recommended locations. Therefore, this application is expected to enhance the tourism experience in Yogyakarta, promote local MSMEs, and preserve cultural heritage.

Corresponding Author:

Rifki Alfarro Zaidan,
Email: 2001010052@universitasbumigora.ac.id

Vol. 1, no. 1, hlm. 23-34, Mei 2025

DOI: [10.30812/juteks.v1i1.5129](https://doi.org/10.30812/juteks.v1i1.5129)

How to cite:

R. A. Zaidan, T. T. Sujaka, dan I. N. Switrayana, "Aplikasi Sistem Rekomendasi Lokasi Wisata Candi dan UMKM di Yogyakarta Berbasis Android dengan Metode Hybrid," *Jurnal Teknologi, Kesehatan, dan Sosial (JUTEKS)*, vol.1, no.1, pp. 23-34, Mei, 2025.

1. PENDAHULUAN

Berwisata merupakan kegiatan berpindah dari suatu tempat ke tempat lainnya dengan tujuan menikmati objek dan daya tarik wisata dimana setiap wisatawan yang berkunjung ke suatu tempat wisata. Wisatawan memiliki tujuan tertentu baik ingin menghabiskan waktu bersama keluarga, melakukan wisata budaya, memanjakan diri, bersantai, mengeksplorasi tempat wisata dan lain sebagainya [1]. Pariwisata adalah salah satu sektor penting bagi perekonomian Indonesia. Keindahan alam dan keanekaragaman budaya merupakan nilai lebih yang dianggap dapat menarik para wisatawan [2].

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. Menurut data statistik kepariwisataan DIY tahun 2017 sebanyak 25.950.793 wisatawan datang ke Yogyakarta [2] Yogyakarta ramai dikunjungi wisatawan domestik maupun mancanegara. Banyaknya destinasi wisata yang ada di Yogyakarta sering membuat wisatawan bingung dalam memilih tujuan wisatanya [3]. Selain lokasi wisatanya, Jogja juga menyimpan segudang aneka kuliner yang bisa didapat dari UMKM yang tersedia dilokasi wisata. Wisata kuliner menjadi salah satu tren masa kini didalam dunia kuliner. Beragam informasi yang tersedia di ininternet membuat banyak kalangan masyarakat semakin sulit dalam memilih kebutuhan yang diinginkan [4].

Sistem rekomendasi sering digunakan oleh sebagian besar area bisnis dalam memberi rekomendasi kepada konsumen dari informasi yang telah didapatkan. Area pariwisata merupakan salah satu contoh bisnis area yang dapat menerapkan sistem rekomendasi untuk membantu para wisatawan dalam memberi rekomendasi wisata untuk mereka. Metode collaborative dan content-based adalah metode yang sering digunakan pada sistem rekomendasi karena teknik ini menyaring informasi berdasarkan keinginan pengguna dan berdasarkan content yang disediakan Namun terdapat beberapa kekurangan dalam kedua metode tersebut. Maka untuk menutupi kelemahan pada metode-metode tersebut dapat dilakukan dengan menggunakan metode hybrid yaitu menggabungkan beberapa metode yang terdapat pada sistem rekomendasi untuk menghasilkan item rekomendasi yang sesuai dengan keinginan pengguna. Untuk mendukung cara kerja metode tersebut maka diperlukan algoritma yang akan mendukung sistem rekomendasi dalam memberi informasi yang sesuai. Salah satunya adalah cosine similarity. Metode cosine similarity merupakan salah satu cara yang dapat digunakan untuk mencari data dalam data mining dan sering digunakan untuk menemukan layanan yang mirip. Cosine similarity akan menghitung tingkat kemiripan dengan menggunakan kata kunci.

Oleh Karena itu tujuan dari perancangan sistem rekomendasi ini adalah untuk merekomendasikan lokasi wisata dan produk UMKM yang berada di sekitaran lokasi candi di Yogyakarta menggunakan metode hybrid guna meningkatkan potensi pariwisata budaya dan Usaha Mikro, Kecil, dan Menengah (UMKM) di Yogyakarta yang masih belum terlirik baik oleh wisatawan maupun masyarakat lokal.

2. METODE PENELITIAN

Metode perancangan sistem yang digunakan dalam penelitian ini adalah Waterfall Software Development Life Cycle atau yang disingkat Waterfall SDLC. Metode ini merupakan metode yang umumnya digunakan dalam perancangan sebuah aplikasi dan mendefinisikan proses siklus pembangunan dan pengembangan sistem aplikasi. Menurut [5], metode Waterfall terdiri dari 6 (enam) tahapan yaitu Requirement Analysis, System Design, Implementation, Integration and Testing, employment of system dan Maintenance.

3. HASIL DAN PEMBAHASAN

1. Requirement Analyst

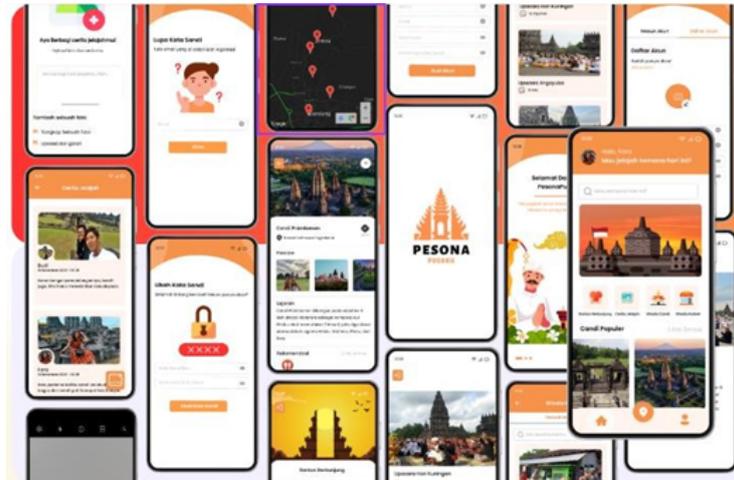
Dalam tahapan ini penulis melakukan analisa dan pemilihan data yang akan digunakan dan dimasukkan kedalam sistem sesuai dengan kebutuhan yang akan ditampilkan dalam sistem rekomendasi.

2. Perancangan (Design)

Tahapan ini merupakan perancangan pembuatan sistem rekomendasi berdasarkan hasil dari analisis. Langkah dalam perancangan pengembangan sistem rekomendasi wisata Candi dan UMKM di Yogyakarta adalah sebagai berikut:

Berikut adalah rancangan desain aplikasi yang telah dibuat :

a. Desain Aplikasi

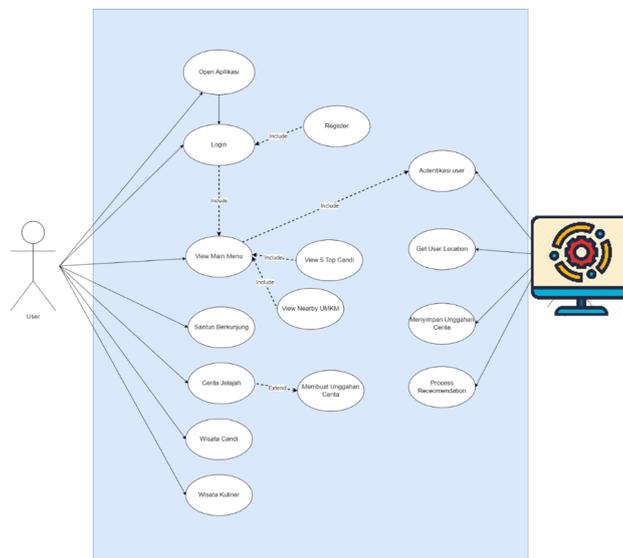


Gambar 1. Rancangan Desain Aplikasi

3. Implementation

Pada tahapan ini, penulis merancang diagram use case dan activity diagram sebagai bagian dari proses perancangan sistem. Diagram use case digunakan untuk menggambarkan interaksi antara aktor (pengguna) dengan sistem. Sementara itu, activity diagram digunakan untuk menggambarkan alur kerja atau proses dalam sistem secara visual, termasuk langkah-langkah yang harus dilakukan dan keputusan yang mungkin diambil dalam proses tersebut. Berikut adalah bentuk dari diagram yang telah dirancang bisa dilihat pada *Gambar 1* dan *Gambar 2*:

4. Use Case Diagram



Gambar 2. Rancangan Use Case Diagram

User :

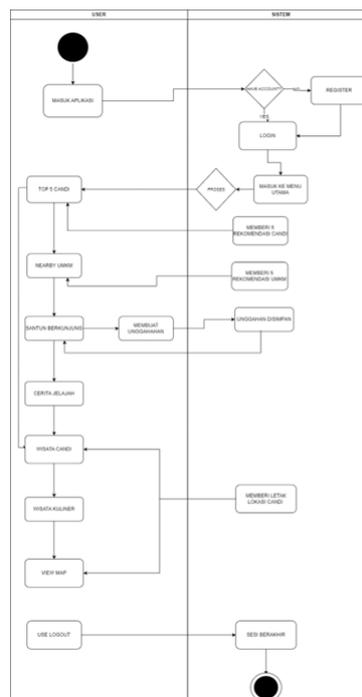
- (a) User akan malakukan register bila tidak memiliki akun
- (b) User malakukan login ke aplikasi
- (c) Sebelum melakukan login user diharapkan mendaftarkan terlebih dahulu pada halaman register

- (d) Setelah user melakukan login user secara otomatis diarahkan pada menu utama aplikasi
- (e) User dapat melihat rekomendasi 5 candi yang telah diberikan sistem
- (f) User dapat melihat rekomedasi UMKM berdasarkan lokasi user
- (g) User dapat melohat rekomendasi 5 Candi terbaik berdasarkan rating yang diberikan sistem
- (h) User dapat mengakses fitur "Santun Berkunjung"
- (i) User dapat mengkses fitur "Cerita Jelajah"
- (j) User dapat membuat unggahan cerita di dalam fitur "Cerita Jelajah"
- (k) User dapat Mengakses fitur "Wisata Candi"
- (l) User dapat Mengakses fitur "Wisata Kuliner"
- (m) User dapat Mengakses Map

System :

- (a) System memverifikasi akun dari pengguna jika benar maka user akan diarahkan ke menu utama aplikasi
- (b) System mengambil kordinat lokasi user dengan GPS
- (c) System menyimpan unggahan cerita dari user
- (d) System memberika rekomendasi Candi dan UMKM kepada user

5. Activity Diagram



Gambar 3. Rancangan Activity Diagram

User

- (a) User akan masuk kedalam aplikasi dan kemudian melakukan login, bila user tidak memiliki akun maka user akan melakukan register dan kemudian login kembali
- (b) User dapat mengakses berbagai fitur yang tersedia didalam aplikasi
- (c) User dapat mengunggah cerita atau pengalaman pada fitur cerita jelajah
- (d) Bila sudah tidak ada aktivitas yang dilakukan maka user akan melakukan logout System
- (e) System memverifikasi apakah user memiliki akun atau tidak, bila tidak maka sistem akan memberitahu user untuk melakukan registrasi
- (f) Pada halaman utama sistem akan memberikan 5 rekomendasi candi berdasarkan model rekomendasi yang telah terintegrasi
- (g) Sistem juga memberikan rekomendasi UMKM terdekat dengan user berdasarkan model rekomendasi yang telah terintegrasi
- (h) Sistem akan menyimpan hasil unggahan cerita user
- (i) Sistem akan mengakhiri sesi aplikasi bila user melakukan logout

Deployment

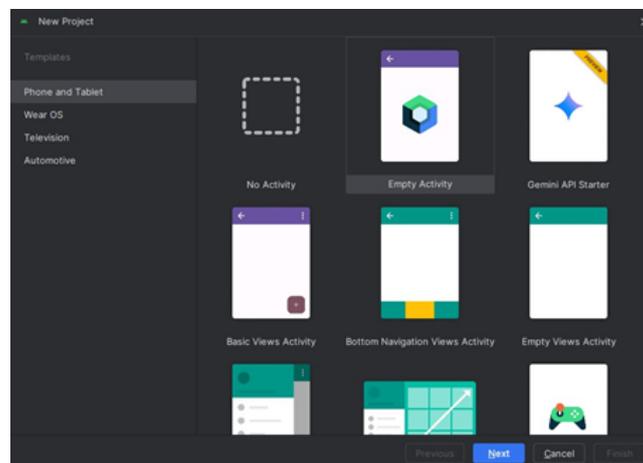
Tahap deployment merupakan kulminasi dari proses pengembangan aplikasi, di mana rancangan konseptual dari tahap desain dan implementasi teknis dari tahap sebelumnya diwujudkan menjadi produk fungsional. Dalam konteks pengembangan aplikasi Android dengan integrasi model machine learning, deployment melibatkan dua jalur utama. Pertama, aplikasi Android yang dibangun di Android Studio dikompilasi menjadi berkas APK. Kedua, model machine learning yang dikembangkan di Google Colab dikonversi ke format yang efisien, lalu diintegrasikan ke dalam aplikasi Android. Pengujian akhir dilakukan untuk memvalidasi interaksi antara aplikasi dan model, memastikan pengalaman pengguna yang optimal.

Pembuatan Aplikasi

Bagian ini mencakup pembuatan aplikasi menggunakan Android Studio sesuai dengan rancangan desain yang sebelumnya telah dijelaskan.

(a) Pemilihan Template Project

Pada tahap ini, dilakukan pemilihan template project sesuai dengan aplikasi yang akan dirancang.



Gambar 4. Template Project

Pada *Gambar 4* ditampilkan menu pemilihan template project untuk memulai pembuatan

(c) Perancangan Model Sistem Cerdas

Pada tahap ini, dilakukan perancangan model sistem cerdas yang akan diimplementasikan pada aplikasi sebagai alat untuk merekomendasikan candi dan UMKM kepada pengguna (user).

```
# Content-based Filtering
def get_content_based_recommendations(candi_id, cosine_sim, candi_filtered_df, num_recommendations=5):
    idx = candi_filtered_df.index[candi_filtered_df['place_id'] == candi_id].tolist()[0]
    sim_scores = list(enumerate(cosine_sim[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:num_recommendations+1]
    candi_indices = [i[0] for i in sim_scores]
    recommendations = candi_filtered_df.iloc[candi_indices][['Nama_Candi', 'Gambar_Candi', 'place_id']]
    recommendations['content_score'] = [1] for i in sim_scores]
    return recommendations
```

Gambar 8. Perancangan Fungsi Content-Based Filtering

Pada **Gambar 8**, merupakan fungsi content-based yang dimana fungsi ini digunakan untuk memproses rekomendasi candi dengan mencari indeks candi referensi dalam dataframe, dilanjutkan dengan mengambil skor kemiripan antara candi referensi dan candi-candi lainnya.

```
# Collaborative Filtering
def get_user_based_collaborative_recommendations(user_id, user_item_matrix, candi_filtered_df, num_recommendations=5):
    user_similarity = cosine_similarity(user_item_matrix)
    user_index = user_item_matrix.index.get_loc(user_id)
    similar_users = user_similarity[user_index].argsort()[::-1][1:]
    top_similar_users = similar_users[:10]
    similar_user_ratings = user_item_matrix.iloc[top_similar_users]
    avg_ratings = similar_user_ratings.mean()
    top_candi_indices = avg_ratings.argsort()[::-1][1:num_recommendations]
    recommended_candi_idx = user_item_matrix.columns[top_candi_indices]
    recommendations = candi_filtered_df[candi_filtered_df['place_id'].isin(recommended_candi_idx)]
    recommendations['collaborative_score'] = avg_ratings[top_candi_indices].values
    return recommendations[['Nama_Candi', 'Gambar_Candi', 'place_id', 'collaborative_score']]
```

Gambar 9. Perancangan Fungsi Collaborative Filtering

Pada **Gambar 9**, merupakan sebuah fungsi collaborative filtering yang dimana fungsi ini digunakan untuk memproses metode collaborative filtering yang dimana akan menghitung kemiripan antar pengguna berdasarkan rating mereka.

```
# Hybrid Recommendations
def get_hybrid_recommendations(user_id, user_item_matrix, cosine_sim, candi_filtered_df, user_filtered_df, num_recommendations=5):
    last_rated_candi = user_filtered_df[user_filtered_df['user_id'] == user_id]['place_id'].iloc[-1]
    content_recommendations = get_content_based_recommendations(last_rated_candi, cosine_sim, candi_filtered_df, num_recommendations)
    collaborative_recommendations = get_user_based_collaborative_recommendations(user_id, user_item_matrix, candi_filtered_df, num_recommendations)

    # Find Intersection
    intersection = pd.merge(content_recommendations, collaborative_recommendations, on='place_id')
    intersection['hybrid_score'] = (intersection['content_score'] + intersection['collaborative_score']) / 2
    intersection = intersection.sort_values('hybrid_score', ascending=False)

    # Get remaining recommendations
    remaining = pd.concat([content_recommendations, collaborative_recommendations]).drop_duplicates(subset='place_id')
    remaining = remaining[remaining['place_id'].isin(intersection['place_id'])]
    remaining['hybrid_score'] = remaining[['content_score', 'collaborative_score']].max(axis=1)
    remaining = remaining.sort_values('hybrid_score', ascending=False)

    # Combine results
    final_recommendations = pd.concat([intersection, remaining]).head(num_recommendations)
    return final_recommendations[['Nama_Candi', 'hybrid_score', 'Gambar_Candi', 'place_id']]
```

Gambar 10. Perancangan Fungsi Hybrid Recommendation

Pada **Gambar 10**, merupakan fungsi metode hybrid yang dimana fungsi ini akan menggabungkan content based dan kolaboratif untuk memberikan rekomendasi yang lebih beragam dan relevan. Proses dimulai dengan menemukan irisan dari kedua set rekomendasi ini dan menghitung skor hibryd untuk irisan tersebut dengan mengambil rata-rata dari skor konten dan kolaboratif.

(d) Evaluasi Kinerja Model

Evaluasi kinerja model dilakukan untuk membandingkan kinerja dari semua metode yang digunakan, evaluasi akan dibuat menggunakan beberapa metrik kinerja. Cara kerja evaluasi ini adalah membagi data menjadi 2 yaitu data latih dan data test dengan rasio 8:2 (80% latih dan 20% test). Data test yang diperoleh dapat dilihat pada tabel 1.

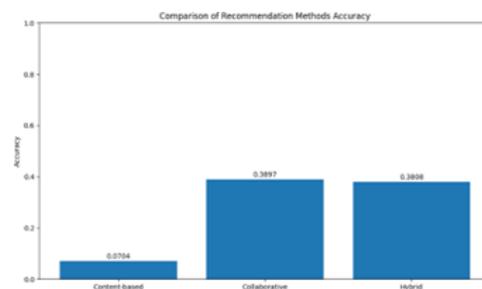
Tabel 1. Data Testing

User_ID	Place_ID
1191	17. 1
619	2. 22
488	16. 6
1292	1. 14
165	4. 21. 23
...	...
1854	11. 21
1470	16. 8
516	17. 23
1394	8. 4
1211	13. 23

Berdasarkan *Tabel 1*, terdapat 426 user yang akan digunakan sebagai evaluasi. Pada *Tabel 1* menampilkan 5 (lima) user pertama dan 5 (lima) user terakhir. Kemudian diambil sebuah sample data yang digunakan sebagai perhitungan evaluasi :

User 488 : [16, 6]
CBF : 1, 3, 6, 15, 19
Accuracy : $\frac{1}{2} = 0,5$
CF : 5, 6, 13, 20, 22
Accuracy: $\frac{1}{2} = 0,5$
Hybrid : CF + CBF
Accuracy: $\frac{1}{2} = 0,5$

Kemudian proses ini diulang untuk setiap metode rekomendasi dan evaluasi akan menampilkan hasil akhir menampilkan rata-rata dari setiap metrik untuk ketiga metode untuk mendapatkan hasil kinerja dari keseluruhan metode.



Gambar II. Hasil Evaluasi Kinerja Metode

Pada *Gambar II*, Merupakan grafik hasil evaluasi, evaluasi menunjukkan bahwa metode content-based memiliki kinerja yang kurang memuaskan, dengan nilai yang rendah, yang dapat menyebabkan ketidakpuasan pengguna karena banyaknya rekomendasi yang tidak relevan. Di sisi lain, metode collaborative filtering dan hybrid menunjukkan performa yang jauh lebih baik yang menandakan efektivitas mereka dalam memberikan rekomendasi yang relevan dan berguna. Secara keseluruhan, pendekatan hybrid muncul sebagai pilihan yang paling diinginkan.

```
# Fungsi untuk menghitung jarak antara dua titik koordinat
def haversine_distance(lat1, lon1, lat2, lon2):
    R = 6371 # Radius bumi dalam kilometer

    lat1, lon1, lat2, lon2 = map(radians, [lat1, lon1, lat2, lon2])
    dlat = lat2 - lat1
    dlon = lon2 - lon1

    a = sin(dlat/2)**2 + cos(lat1) * cos(lat2) * sin(dlon/2)**2
    c = 2 * atan2(sqrt(a), sqrt(1-a))
    distance = R * c

    return distance
```

Gambar 12. Hasil Evaluasi Kinerja Metode

Pada **Gambar 12**, Merupakan perancangan model rekomendasi untuk lokasi UMKM yang berada di sekitar wisata candi. Metode yang digunakan pada model ini adalah regresi linear dengan pendekatan haversine sebagai pengukur jarak lokasi.

(e) Pembuatan Endpoint

Pembuatan endpoint ini dilakukan agar fitur aplikasi dapat digunakan sesuai dengan yang diharapkan. Endpoint API merupakan sebuah titik akhir dari komunikasi aplikasi dengan sebuah layanan atau server, ini merupakan sebuah URL (alamat) spesifik yang akan dituju untuk mengakses sumber daya atau fungsi tertentu dari API tersebut.

```
import express from 'express';
import { Low } from 'lowdb';
import { JSONFile } from 'lowdb/node';
import { fileURLToPath } from 'url';
import { dirname, join } from 'path';
import path from 'path';
import cors from 'cors';
import fs from 'fs';
import { getStoriesFromDatabase, getStoriesById, createStory, updateStory, deleteStory } from './stories.js';
import { getUserData, getUserById, updateUser } from './user.js';
import multer from 'multer';
```

Gambar 13. Import Library

Pada **Gambar 13**, terdapat beberapa library (layanan) yang dibutuhkan untuk membangun sebuah endpoint server.

```
// Middleware
app.use(cors({
  origin: '*', // Ganti dengan domain aplikasi Android Anda jika diperlukan
  methods: ['GET', 'POST', 'PUT', 'DELETE'],
  allowedHeaders: ['Content-Type', 'Authorization']
}));
app.use(express.json());
app.use(express.urlencoded({ extended: true })); // Middleware untuk form-data

// Make db accessible to our router
app.use((req, res, next) => {
  req.app.db = db;
  next();
});

// Serve static files from the 'uploads' directory
app.use('/Backend/uploads', express.static(uploadDir));
app.use('/Backend/avatar', express.static/avatarDir)); // Serve avatar from 'avatar' folder
```

Gambar 14. Konfigurasi Middleware

Pada **Gambar 14**, merupakan konfigurasi keamanan CORS (Cross-Origin Resource Sharing) untuk mengontrol aplikasi dalam mengakses sumber daya dari server seperti metode GET, POST, PUT, DELETE dan beberapa header khusus.

```
// routes
const registerRoute = await import('./register.js');
app.post('/register', registerRoute.default); // Hapus middleware upload.single('avatar')

const loginRoute = await import('./login.js');
app.post('/login', (req, res) => loginRoute.default(req, res));

const forgotRoute = await import('./forgot-password.js');
app.post('/forgot-password', (req, res) => forgotRoute.default(req, res));

const candiRoute = await import('./candi.js');
app.get('/candi/:id', (req, res) => {
  req.query.id = req.params.id;
  candiRoute.default(req, res);
});

app.get('/candi', candiRoute.default);

const umkmRoute = await import('./wisata-kuliner.js');
app.get('/wisata-kuliner', (req, res) => umkmRoute.default(req, res));

app.get('/stories', async (req, res) => {
  try {
    const storiesResponse = await getStoriesFromDatabase();
    res.json(storiesResponse);
  } catch (err) {
    res.status(500).json({ status: 'E', message: 'Internal Server Error', data: {} });
  }
});
```

Gambar 15. Konfigurasi Rute Sumber Daya

Pada Gambar **Gambar 15**, dilakukan inisialisasi ruter sumber daya yang dibutuhkan untuk mengakses fitur pada aplikasi. Rute ini digunakan untuk mengambil atau memanggil fungsi yang berada di dalam skrip kode yang lain.

(f) Implementasi Endpoint Model Rekomendasi Implementasi ini dilakukan untuk memungkinkan model rekomendasi yang telah dirancang sebelumnya dapat digunakan pada aplikasi dalam bentuk endpoint.

```

# 4. flask endpoint (diperbaiki)
@app.route('/api/recommendations', methods=['GET'])
def get_recommendations():
    user_id_str = request.args.get('user_id')

    try:
        user_id = int(user_id_str) if user_id_str else 1 # Gunakan user_id 1 sebagai default jika tidak diberikan
    except ValueError:
        return jsonify({"message": "Error: 'user_id' must be an integer."}), 400 # Bad Request

    recommendations = get_hybrid_recommendations(user_id, matrix_factorized, cosine_sim, candi_filtered_df)

    if not recommendations.empty:
        # Ubah nama kolom 'gambar.Candi' menjadi 'thumbnail' agar sesuai dengan CandiModel di Kotlin
        recommendations = recommendations.rename(columns={'gambar.Candi': 'thumbnail'})

        # Bungkus rekomendasi dalam objek CandiResponse
        response_data = {
            "data": recommendations.to_dict(orient='records'),
            "status": "S",
            "message": "Success"
        }
        return jsonify(response_data)
    else:
        return jsonify({"message": "No recommendations found for this user."}), 404

if __name__ == '__main__':
    app.run(debug=True)

```

Gambar 16. Endpoint Model Rekomendasi Candi

Pada **Gambar 16**, Model rekomendasi candi yang sebelumnya telah dibangun dan dievaluasi kemudian diimplementasikan kedalam aplikasi android yang akan dibangun. Model akan digunakan dalam endpoint API yang dimana model akan memberikan prediksi 5 candi teratas yang kemudian hasil prediksi ini akan dikirim ke aplikasi sebagai rekomendasi candi yang mungkin akan disukai oleh pengguna.

```

@app.route('/recommendation-wisata-kuliner', methods=['GET'])
def get_recommendations_api():
    try:
        user_lat = request.args.get('latitude')
        user_lon = request.args.get('longitude')

        if user_lat and user_lon:
            user_lat = float(user_lat)
            user_lon = float(user_lon)
            recommendations = get_umkm_recommendations(user_lat, user_lon)
        else:
            # Berikan rekomendasi default jika tidak ada input lokasi
            recommendations = umkm_data.sort_values('rating_umkm', ascending=False).head(5) # Atau logika rekomendasi default lainnya

        if not recommendations.empty:
            # Ubah nama kolom 'gambar.umkm' menjadi 'thumbnail'
            recommendations = recommendations.rename(columns={
                'gambar.umkm': 'thumbnail',
                'nama.umkm': 'name',
                'provinsi': 'location'
            })

            # Bungkus rekomendasi dalam objek CandiResponse
            response_data = {
                "data": recommendations.to_dict(orient='records'),
                "status": "S",
                "message": "Success"
            }
            return jsonify(response_data)
        else:
            return jsonify({"message": "Tidak ada rekomendasi yang ditemukan", "status": "F"})

    except ValueError:
        return jsonify({"message": "Invalid latitude or longitude parameters", "status": "F"}), 400
    except Exception as e:
        return jsonify({"message": "An error occurred", "status": "F"}), 500

if __name__ == '__main__':
    app.run(debug=True)

```

Gambar 17. Endpoint Model Rekomendasi UMKM

Pada **Gambar 17**, model rekomendasi UMKM yang sebelumnya telah dibangun kemudian akan diimplementasikan kedalam aplikasi android. Model akan digunakan sebagai endpoint API untuk memberikan rekomendasi kepada pengguna, bila endpoint tidak menerima koordinat user maka akan diberikan rekomendasi default.

(g) Testing

Tahap testing merupakan tahap dimana program yang sudah dibuat akan melalui tahap uji coba, agar program yang telah dibuat berjalan dengan baik dan optimal. Untuk pengujian program dilakukan dengan cara uji Blackbox dan Whitebox dimana pengujian berfokus kepada fungsional dari program yang telah dibuat.

i. Blackbox Testing

Untuk memastikan bahwa sistem yang dikembangkan telah sesuai yang diharapkan, maka dilakukan pengujian dengan teknik Blackbox. Pengujian dilakukan dengan memeriksa fungsionalitas dari fitur perangkat lunak.

Tabel 2. Hasil Pengujian Blackbox

No	Skenario	Test Case	Hasil yang Diharapkan	Hasil Uji	Status
1	Mengosongkan form register	Avatar:- User:- E-mail:- Password:-	Akan muncul pesan error berupa "Harap isi semua field" dan sistem menolak request register	Sesuai	Valid

No	Skenario	Test Case	Hasil yang Diharapkan	Hasil Uji	Status
2	Mengisi form register dengan avatar, username, e-mail dan password yang belum terdaftar	Username: rifk e-mail: anonymous-pako225@gmail.com Password: openrc1234	Registration successful	Sesuai	Valid
3	Mengisi form register dengan akun yang telah terdaftar	Username: rifk e-mail: anonymous-pako225@gmail.com Password: openrc1234	Email is already exist/ Bad Request	Sesuai	Valid
4	Mengosongkan form login	E-mail: - Password: -	Please fill all field	Sesuai	Valid
5	Memasukan e-mail dan password yang tidak terdaftar ke form login	E-mail: test-ing@gmail.com	Invalid email or password/ Unauthorized	Sesuai	Valid
6	Memasukan e-mail dan password yang terdaftar ke form login	E-mail: anonymous-pako225@gmail.com Password: openrc1234	Login successful	Sesuai	Valid
7	Mencoba search bar pada halaman utama	-	Tidak menampilkan apapun	Sesuai	Valid
8	Mengakses Halaman santun berkunjung	-	Mengakses Halaman santun berkunjung	Sesuai	Valid
9	Mengakses halaman Cerita Jelajah	-	Menampilkan list cerita jelajah	Sesuai	Valid
10	Mencoba unggah cerita	Image : upload image Description : ini adalah percobaan	Unggahan berhasil tersimpan	Sesuai	Valid
11	Mengakses halaman Wisata Candi	-	Menampilkan list Candi	Sesuai	Valid
12	Mencoba search bar pada halaman wisata candi	-	Tidak menampilkan apapun	Sesuai	Valid
13	Mengakses halaman Wisata Kuliner	-	Menampilkan list kuliner/UMKM	Sesuai	Valid

(h) Whitebox Testing

Whitebox testing merupakan skenario pengujian kode aplikasi secara langsung, skenarion ini digunakan untuk menemukan bugs (masalah) pada proses pengembangan aplikasi. Skenario uji coba white box dilakukan dengan pengujian statis (Static Testing).

Tabel 3. Hasil Skenario Uji Whitebox

Bugs	Rincian Masalah
Kompatibilitas versi android	Masalah ini dapat menyebabkan aplikasi berhenti dengan sendirinya (crash) ketika dijalankan pada versi android yang tidak didukung
Library Version	Fitur library yang digunakan perlu diperbarui untuk menghindari masalah kompatibilitas untuk perangkat baru
Perfomance Issue	masalah kinerja (performa), pada layout (halaman) dapat memengaruhi kinerja dari aplikasi
API Stability	API yang digunakan pada aplikasi masih dalam tahap tidak stabil dan berpotensi mengalami beberapa masalah bila terjadi pembaruan API

4. KESIMPULAN

Berdasarkan penelitian di atas, dapat disimpulkan bahwa penelitian ini berhasil mengembangkan sebuah aplikasi rekomendasi lokasi wisata candi dan UMKM di Yogyakarta. Aplikasi ini tidak hanya berfungsi sebagai alat bantu dalam menemukan destinasi wisata yang menarik, tetapi juga memberikan informasi mendalam tentang candi-candi tersebut, termasuk sejarah, tradisi, dan tata cara berkunjung yang santun. Penerapan metode hybrid yang menggabungkan Content-Based Filtering dan Collaborative Filtering dalam aplikasi ini dapat menghasilkan rekomendasi candi. Selain itu, integrasi metode Regresi Linear memungkinkan aplikasi untuk memberikan rekomendasi UMKM terdekat

berdasarkan lokasi pengguna, sehingga dapat meningkatkan visibilitas dan potensi ekonomi UMKM lokal. Berdasarkan hasil pengujian menunjukkan bahwa model rekomendasi masih jauh dari sempurna namun model sudah berhasil diimplementasikan dan berjalan dengan baik.

DAFTAR PUSTAKA

- [1] "(PDF) Utilization of CollaboratPemanfaatan Metode Based Collaborative Filtering Untuk Rekomendasi Wisata Di Kabupaten Malangive Filtering Method for Tourism Recommendations in Malang Regency," *ResearchGate*, Apr. 2025. <https://doi.org/10.32815/jitika.v13i2.70>.
- [2] A. S. Ningrum, H. C. Rustamaji, and Y. Fauziah, "Content based dan collaborative filtering pada rekomendasi tujuan pariwisata di daerah yogyakarta," *Telematika : Jurnal Informatika dan Teknologi Informasi*, vol. 16, pp. 44–51, Apr. 2019. <https://doi.org/10.31315/telematika.v16i1.3023>.
- [3] "(PDF) Pembuatan Sistem Rekomendasi Pariwisata Yogyakarta Menggunakan Triangle Multiplaying Jaccard," *ResearchGate*. <https://doi.org/10.47134/jacis.v3i2.62>.
- [4] "Sistem rekomendasi paket wisata se-malang raya menggunakan metode hybrid content based dan collaborative."
- [5] M. S. Irwinsyah and J. W. Kuswinardi, "Rancang bangun sistem rekomendasi pemilihan destinasi wisata di kota batu berbasis android," *Semnas SENASTEK Unikama 2019*, vol. 2, Dec. 2019.