

# IMPLEMENTASI *MACHINE LEARNING* DENGAN METODE *COLLABORATIVE FILTERING* DAN *CONTENT-BASED FILTERING* PADA APLIKASI MOBILE TRAVEL (BANGKIT ACADEMY)

*(Implementation of Machine Learning with Collaborative Filtering and Content-Based Filtering Methods in Mobile Travel Application (Bangkit Academy))*

Maulana Surya Negara, Ahmad Zafrullah M.  
Dept Informatics Engineering, Mataram University  
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA  
Email: [maulanasuryanegara123@gmail.com](mailto:maulanasuryanegara123@gmail.com), [zaf@unram.ac.id](mailto:zaf@unram.ac.id)

## Abstrak

Dalam menghadapi kekurangan talenta digital, Indonesia meluncurkan program *Bangkit Academy* dengan tujuan melatih peserta dalam berbagai jalur teknologi, termasuk *machine learning*, *cloud computing*, dan pengembangan *mobile*. Salah satu hasilnya adalah aplikasi "*DestiMate*" yang dikembangkan oleh tim CH2-PS452, sebuah platform cerdas berbasis Android untuk merekomendasikan tempat wisata dan resort menggunakan teknologi *machine learning*. Proyek ini merespons tantangan dalam memilih destinasi dan akomodasi yang sesuai dengan preferensi wisatawan, baik lokal maupun mancanegara, dengan menghadirkan solusi berupa sistem rekomendasi yang mempertimbangkan keberagaman preferensi pengguna dan kekayaan tempat wisata Indonesia. Diharapkan bahwa "*DestiMate*" dapat meningkatkan *travel experience*, mempromosikan Indonesia sebagai destinasi wisata unggulan, dan memberikan nilai tambah bagi industri pariwisata Indonesia.

**Keywords:** *Machine Learning, Neural Network, Recommender System, Content-Based Filtering, Collaborative Filtering*

## 1. PENDAHULUAN

Dalam menghadapi kekurangan talenta digital, Indonesia meluncurkan program *Bangkit Academy*, bekerja sama dengan Google, Kementerian Pendidikan, dan perusahaan teknologi. Program ini melatih peserta dalam berbagai jalur, termasuk *machine learning*, *cloud computing*, dan *mobile development*. Salah satu hasilnya adalah aplikasi "*DestiMate*" yang dikembangkan oleh tim CH2-PS452, sebuah platform cerdas berbasis Android untuk merekomendasikan tempat wisata dan resort menggunakan teknologi *machine learning*.

Pertumbuhan signifikan dalam industri pariwisata Indonesia selama beberapa tahun terakhir telah menunjukkan potensi besar untuk pengembangan inovasi dalam membantu wisatawan, termasuk wisatawan mancanegara, dalam merencanakan perjalanan mereka. Indonesia, sebagai destinasi wisata yang kaya dengan berbagai pilihan wisata alam dan budaya, menarik perhatian wisatawan dari seluruh dunia [1].

Tantangan yang dihadapi oleh wisatawan, baik lokal maupun mancanegara, adalah dalam memilih destinasi dan akomodasi yang sesuai dengan preferensi mereka. Dalam lingkungan pariwisata global seperti sekarang, wisatawan mancanegara menjadi faktor penting yang turut berkontribusi signifikan terhadap pertumbuhan industri pariwisata di Indonesia. Keberagaman preferensi dari wisatawan asing menambah kompleksitas dalam memberikan pengalaman perjalanan yang memuaskan untuk setiap individu.

Seiring dengan tren globalisasi dan meningkatnya mobilitas wisatawan, penting untuk memberikan solusi yang dapat memahami dan merespons keberagaman preferensi pengguna. Penggunaan teknologi untuk menganalisis data preferensi pengguna merupakan salah satu langkah penting yang diambil untuk mencapai hal tersebut.

Proyek pengembangan sistem rekomendasi wisata dan resort cerdas "*DestiMate*" muncul sebagai inisiatif yang merespons secara langsung kebutuhan ini. Kemampuan sistem ini untuk memberikan rekomendasi yang lebih menyeluruh dan personal, dengan mempertimbangkan preferensi individu dan kekayaan tempat wisata Indonesia, dapat menjadi instrumen penting dalam meningkatkan *travel experience*. Dalam konteks ini, proyek ini menghadirkan solusi yang tidak hanya menawarkan destinasi wisata yang sesuai dengan preferensi pengguna, tetapi juga menyertakan rekomendasi akomodasi yang secara geografis berdekatan. Pendekatan ini memberikan gambaran tentang perjalanan, memudahkan wisatawan untuk merencanakan dan menikmati pengalaman mereka tanpa perlu mencari informasi terpisah tentang akomodasi.

Dengan demikian, diharapkan bahwa proyek ini akan memberikan kontribusi positif dalam mempromosikan Indonesia sebagai destinasi wisata unggulan, meningkatkan kepuasan wisatawan, dan memberikan nilai tambah bagi industri pariwisata Indonesia.

### 1.1 Lingkup

Dalam program Magang Studi Independen Bersertifikat (MSIB) Bangkit, penulis mengambil *Machine Learning*. Dalam kegiatan ini, peserta melakukan beberapa kegiatan seperti belajar materi secara *asynchronous* melalui modul belajar di *Dicoding Academy* dan *Coursera*, mengikuti *Instructor-Led Training (ILT)*, mengikuti kegiatan konsultasi mingguan dengan mentor, mengerjakan *submission* kuis, dan tugas yang diberikan untuk setiap materi, serta menyelesaikan *capstone* proyek akhir yang mengimplementasikan *machine learning*.

### 1.2 Tujuan

Berikut merupakan tujuan dari diadakannya program *Bangkit Academy* yaitu:

- a. Menghasilkan talenta digital berkaliber tinggi untuk perusahaan teknologi dan rintisan Indonesia kelas dunia.
- b. Mengembangkan diri dengan *soft skills* yang perlu dimiliki seorang karyawan
- c. Mengerjakan proyek akhir yang diberikan pada program Bangkit 2023.
- d. Mengikuti semua pembelajaran individu yang telah diberikan oleh tim Bangkit Academy.

### 1.3 Deskripsi Pekerjaan

Aktivitas MSIB Bangkit Academy *Machine Learning* meliputi pembelajaran individu dan proyek akhir dalam bentuk tim. Pada pembelajaran individu, setiap peserta akan mengikuti kelas dalam bentuk *asynchronous (online)* melalui modul belajar di *Dicoding Academy and Coursera*) di mana peserta dapat berkonsultasi dengan *expert* terkait materi yang dipelajarinya melalui forum diskusi. Selain itu, setiap peserta akan memiliki mentor sebagai tempat konsultasi jika ditemui kesulitan non-akademik dalam mengikuti pembelajaran. Peserta akan memperoleh sertifikat kompetensi di setiap kelas di dalam *learning path machine learning developer* jika peserta berhasil lulus dari setiap ujian/penilaian yang diadakan untuk setiap kompetensi. Setelah mengikuti program ini, peserta juga dipersiapkan untuk mengikuti ujian sertifikasi global TensorFlow Developer dari Google yang dapat diambil setelah mengikuti kegiatan MSIB ini. Selain itu juga, terdapat proyek akhir yang dikerjakan secara berkelompok 6-7 orang dengan tema dan judul ditentukan oleh masing-masing kelompok.

## 2. TINJAUAN PUSTAKAN

### 2.1 Data Cleaning

Teknik pembersihan data merupakan suatu metode yang digunakan untuk mengatasi ketidaklengkapannya. Proses pembersihan data tidak hanya terbatas pada penghapusan duplikasi, pengecekan in-konsistensi, dan perbaikan kesalahan cetak, tetapi juga melibatkan proses *enrichment*. *Enrichment* merupakan tahap di mana data yang sudah ada diperkaya dengan tambahan data atau informasi yang relevan dan diperlukan untuk KDD (*Knowledge Discovery in Databases*), seperti data atau informasi eksternal. Langkah-langkah ini membantu memastikan integritas dan kualitas data serta meningkatkan manfaatnya dalam proses analisis. Seiring dengan itu, upaya perbaikan data ini menjadi esensial untuk menghasilkan hasil analisis yang akurat dan dapat diandalkan [2].

### 2.2 Machine Learning

*Machine learning* dapat didefinisikan sebagai penerapan komputer dan algoritma matematika yang menggunakan pembelajaran dari data untuk menghasilkan prediksi di masa depan. Proses pembelajaran ini melibatkan dua tahap utama, yaitu pelatihan (*training*) dan pengujian (*testing*) untuk mengembangkan kecerdasan [3].

### 2.3 Neural Network

*Neural Network* atau Jaringan Syaraf Tiruan (JST) adalah metode pembelajaran yang umumnya digunakan untuk menangani permasalahan diskrit, *real*, dan vektor. JST juga mencoba memodelkan sistem saraf manusia dalam menjalankan tugas tertentu. Proses pembelajaran pada jaringan syaraf tiruan terdiri dari dua jenis utama, yaitu belajar dengan pengawasan (*supervised learning*) dan belajar tanpa pengawasan (*unsupervised learning*) [4].

### 2.4 Recommender System

Sistem rekomendasi merupakan suatu perangkat lunak dan teknik yang memberikan rekomendasi *item* yang bermanfaat bagi pengguna. Rekomendasi ini didasarkan pada berbagai proses pengambilan keputusan, seperti menentukan *item* yang sebaiknya dibeli, musik yang sebaiknya didengar, makanan yang sebaiknya dikonsumsi, atau

berita *online* yang sebaiknya dibaca. Tujuan utama sistem rekomendasi adalah memberikan rekomendasi yang bersifat personal untuk setiap pengguna, dengan fokus pada *item* yang mungkin sesuai dengan minat mereka, bukan sekadar menyajikan daftar *item* yang populer secara umum. Dua pendekatan utama dalam pengembangan sistem rekomendasi adalah *content-based filtering* dan *collaborative filtering* [5].

## 2.5 Content-Based Filtering

*Content-based filtering* dalam sistem rekomendasi merupakan pendekatan yang mempertimbangkan pola perilaku pengguna dari masa lalu untuk memberikan rekomendasi yang sesuai. Metode ini melibatkan analisis preferensi perilaku pengguna untuk membangun sebuah model. Model tersebut kemudian dihubungkan dengan serangkaian karakteristik atribut dari barang yang akan direkomendasikan. Barang dengan tingkat kesesuaian tertinggi kemudian dihasilkan sebagai rekomendasi untuk pengguna. Pendekatan ini digunakan untuk memastikan bahwa rekomendasi yang diberikan sesuai dengan pola perilaku pengguna dan menciptakan pengalaman yang lebih personal [6].

## 2.6 Bag of Words

*Bag of Words* (BoW) merupakan representasi sederhana yang umum digunakan dalam *Natural Language Processing* (NLP) dan *Information Retrieval* (IR), juga dikenal sebagai model *vector space*. Dalam model ini, teks, baik berupa kalimat atau dokumen, diwakili sebagai kantong (*bag*) multiset dari kata-kata yang terdapat di dalamnya. Representasi ini tidak memperhatikan urutan kata atau tata bahasa, namun tetap mempertahankan keberagaman kata yang terdapat dalam teks tersebut. Definisi lain untuk BoW adalah suatu model yang mempelajari kosakata dari seluruh dokumen, kemudian memodelkan tiap dokumen dengan menghitung jumlah kemunculan setiap kata [7].

## 2.7 Cosine Similarity

*Cosine similarity* adalah metode perhitungan yang umum digunakan untuk mengukur kemiripan antara *item-item*. Secara umum, fungsi *similarity* adalah suatu fungsi yang mengambil dua objek berupa bilangan riil (0 hingga 1) dan menghasilkan nilai kemiripan (*similarity*) antara keduanya, juga berupa bilangan riil [5].

## 2.8 Collaborative Filtering

*Collaborative filtering* dalam sistem rekomendasi merupakan salah satu metode yang menggunakan informasi dari pengguna berupa nilai peringkat produk atau preferensi sesuai dengan keinginan mereka. Pendekatan kolaboratif menghasilkan rekomendasi pengguna untuk produk berdasarkan nilai peringkat atau pemakaian, seperti pembelian produk atau penilaian produk yang diberikan oleh pengguna [8].

## 2.9 Python

Python merupakan salah satu bahasa pemrograman yang populer digunakan oleh perusahaan besar dan para pengembang untuk membangun berbagai aplikasi, termasuk yang berbasis *desktop*, *web*, dan *mobile*. Guido van Rossum menciptakan Python di Belanda pada tahun 1990, dan nama bahasa ini diambil dari acara televisi favoritnya, yaitu Monty Python's Flying Circus. Awalnya, Van Rossum mengembangkan Python sebagai hobi, namun kemudian bahasa pemrograman ini menjadi sangat populer dalam industri dan dunia pendidikan karena kesederhanaannya, sintaksis yang ringkas, intuitif, dan memiliki pustaka yang kaya [9].

## 2.10 Pandas

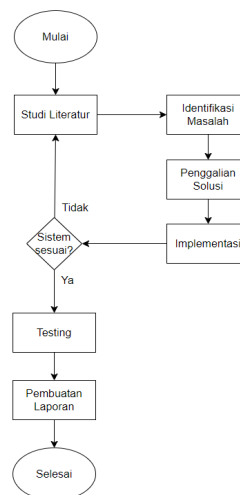
Pandas (*Python Data Analysis*) adalah pustaka pengolahan data dalam domain pemrograman Python yang menyediakan fitur *data frame* untuk manajemen data berbasis tabel atau *array* (larik). Pustaka ini merupakan bagian integral dari *Data Science Toolkit* dalam proses manajemen data [10].

## 2.11 Tensorflow

Dikembangkan oleh tim Google Brain, TensorFlow merupakan sebuah perpustakaan (*library*) sumber terbuka untuk komputasi numerik dan *machine learning* berskala besar. TensorFlow mengintegrasikan berbagai model, algoritma *machine learning*, dan algoritma *deep learning* (jaringan syaraf). Pengembangan TensorFlow melibatkan penggunaan Python sebagai API *front-end* untuk membangun aplikasi dengan *framework* tersebut, sambil menjalankan aplikasi tersebut dengan kinerja yang tinggi [11].

# 3. METODE PENGABDIAN MASYARAKAT

Pada Gambar 1 menunjukkan bagaimana proses pelaksanaan kegiatan pengabdian pada masyarakat. Yang di mana terdapat studi literatur, identifikasi masalah, penggalian solusi, implementasi, *testing* dan membuat laporan.



Gambar 1. Diagram Alir Pelaksanaan

#### a. Studi Literatur

Pada tahap ini, tim melakukan penelusuran literatur yang berkaitan langsung dengan pengembangan *recommender system* pada aplikasi rekomendasi tempat wisata. Analisis literatur ini bertujuan untuk memahami konsep, metode, dan *best practices* terkait sistem rekomendasi.

#### b. Identifikasi masalah

Setelah melakukan studi literatur, tim mengidentifikasi masalah yang dihadapi oleh pengguna dalam menentukan pilihan tempat wisata. Pemahaman mendalam terhadap permasalahan ini menjadi dasar untuk merumuskan solusi yang tepat.

#### c. Penggalian solusi

Tim melakukan penggalian solusi dengan mengeksplorasi opsi dan strategi dalam pengembangan aplikasi rekomendasi DestiMate. Pada tahap ini, dipertimbangkan berbagai metode rekomendasi, *dataset*, fitur-fitur pendukung, dan teknologi yang sesuai.

#### d. Implementasi

Tahapan implementasi melibatkan proses pembuatan dan pengembangan aplikasi rekomendasi tempat wisata DestiMate. Solusi yang dihasilkan dari tahap penggalian solusi diimplementasikan dalam bentuk aplikasi Android dengan metode rekomendasi *content-based filtering* dan *collaborative filtering* untuk memberikan rekomendasi yang personal dan relevan. Jika sistem sudah sesuai maka akan lanjut ke tahap berikutnya, tetapi jika tidak tim kembali akan *research* studi literatur.

#### e. Testing

Jika sistem sudah selesai, maka tahap selanjutnya yaitu melakukan *testing* aplikasi ke beberapa *user*. *User* akan mencoba untuk menggunakan sistem rekomendasi pada aplikasi, lalu memberikan *feedback* kesesuaian hasil rekomendasi dengan referensi mereka.

#### f. Pembuatan laporan

Pada tahapan ini dilakukan pembuatan laporan berdasarkan apa yang telah dibuat sebagai bentuk bahwa telah melakukan pengabdian kepada masyarakat.

## 4. HASIL DAN PEMBAHASAN

### 4.1 Collecting Dataset

Dalam pengembangan aplikasi rekomendasi tempat wisata "DestiMate," langkah awal yang harus dilakukan adalah pengumpulan data dari sumber yang kredibel dan relevan. Dua jenis *dataset* utama yang digunakan berasal dari *platform Kaggle*, sebuah sumber terkemuka dalam dunia *data science*. *Dataset* pertama adalah *dataset tourism*, yang terdiri dari 437 data tempat wisata, 10.000 data *rating*, dan 300 data *user*. Sementara itu, *dataset* kedua adalah *dataset resort* yang mencakup 626 data penginapan beserta informasi fasilitasnya.

## 4.2 Preprocessing Dataset

Langkah selanjutnya setelah mengumpulkan dataset yaitu *preprocessing*. Langkah ini bertujuan untuk mengubah format, menghilangkan *null value*, menghilangkan *duplicate* data, membuang kolom yang tidak digunakan, dll. Berikut merupakan tahap *preprocessing* dari masing-masing *dataset*.

### a. Tourism Dataset

Pertama, dilakukan pemeriksaan terhadap setiap kolom dalam *dataset* untuk mengidentifikasi dan mengatasi nilai *null*, dengan menghapus baris atau mengisi nilai *null* jika memungkinkan. Kemudian, untuk meningkatkan relevansi, dipilih kolom-kolom yang paling relevan untuk rekomendasi tempat wisata, dan beberapa kolom.

### b. Resort Dataset

Pertama, dilakukan penggabungan *dataset* berdasarkan wilayah menjadi satu *file* CSV. Selanjutnya, dilakukan pemeriksaan terhadap setiap kolom untuk menemukan dan mengatasi nilai *null*, khususnya pada kolom alamat dengan menghapus baris yang memiliki nilai *null*. Selain itu, dilakukan perbaikan format penulisan pada kolom *rating* dengan mengubah nilai "Belum ada *rating*" menjadi 0.

## 4.3 Implementasi Content-Based Filtering

Pada aplikasi rekomendasi tempat wisata "DestiMate" *content-based filtering* diimplementasikan pada *detail page* tempat wisata. Sehingga ketika pengguna memilih salah satu tempat wisata maka akan muncul tempat wisata yang serupa. Pada implementasi ini melibatkan dua metode yaitu *bag of words* dan *cosine similarity*. Berikut merupakan penjelasan implementasi dari masing-masing metode.

### a. Bag of Words (BoW)

Metode ini melibatkan representasi suatu tempat wisata berdasarkan kata-kata yang muncul dalam deskripsi atau informasi terkait lainnya. Setiap kata dianggap sebagai fitur, dan kemudian vektor fitur ini digunakan untuk menggambarkan profil unik dari suatu tempat wisata. Dengan menggunakan pendekatan *bag of words*, dapat diekstrak fitur-fitur penting dari deskripsi tempat wisata, kategori, dan kota. Proses ini memungkinkan pembentukan vektor representasi yang dapat digunakan untuk membandingkan kemiripan antar tempat wisata.

Tahap pertama dari metode ini yaitu melakukan penggabungan kolom deskripsi, kategori, dan kota menjadi kolom *tags*. Tahap selanjutnya yaitu menghilangkan kata-kata umum yang tidak memberikan kontribusi signifikan dalam membedakan tempat wisata, yang dikenal sebagai *stop words*. Oleh karena itu, dilakukan tahap penghapusan *stop words* untuk meningkatkan relevansi dan akurasi representasi vektor pada kolom *tags*.

```
stopwords = pd.read_csv('.././././Dataset/Stop Words/stopwordbahasa.csv')
stopwords_list = stopwords.iloc[:,0].tolist()
print(stopwords_list[:10])
```

✓ 0.0s

['adalah', 'adanya', 'adapun', 'agak', 'agaknyanya', 'agar', 'akan', 'akankah', 'akhir', 'akhirini']

Gambar 2. Proses Impor *Dataset Stopwords* Bahasa

Pada Gambar 2 merupakan proses mengimpor *dataset stopwords* bahasa dan terlihat juga contoh beberapa dari *stopwords* bahasa. Nantinya list *stopwords* ini akan digunakan sebagai acuan untuk penghapusan *stopwords* pada kolom *tags*. Setelah itu akan dilakukan ekstraksi fitur kata dengan objek *CountVectorizer* yang berasal dari *library* Scikit-Learn. Dengan objek tersebut didapatkan contoh fitur yang sudah diekstrak dari salah satu tempat wisata yaitu Monumen Nasional.

	Features	Frequency
2754	monumen	4
4393	tanggal	2
2752	monas	2
3351	perjuangan	2
2788	nasional	2
4731	tugu	2
2089	jakarta	2
2564	melambangkan	1
1389	didirikan	1
3618	pusat	1

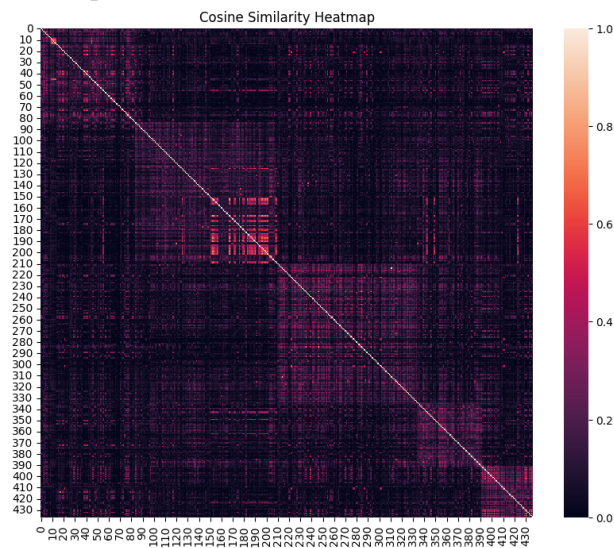
Gambar 3. Fitur Teratas pada Monumen Nasional

### b. Cosine Similarity

Setelah mendapatkan vektor fitur kata dari masing-masing tempat wisata, langkah selanjutnya adalah mengukur kemiripan antara mereka menggunakan metode *cosine similarity*. Metode ini membandingkan sudut antara dua vektor dalam ruang fitur. Semakin kecil sudut antara vektor-vektor tersebut, semakin mirip kedua tempat wisata tersebut. Proses perbandingan ini menggunakan *cosine similarity* yang berasal dari *library* Scikit-Learn, berikut merupakan *code-nya*.

```
similarity = cosine_similarity(vectors)
```

Code di atas berfungsi untuk menghitung *similarity* (kesamaan) antara vektor-vektor tempat wisata. Sehingga *output* dari “*cosine similarity*” yaitu berupa matriks dengan ukuran 437x437. Jika matriks tersebut divisualisasikan dalam bentuk *heatmap* maka akan terlihat seperti berikut.



Gambar 4. Diagram *Heatmap Cosine Similarity* Tempat Wisata

Pada Gambar 4 terlihat diagram *heatmap* dari *cosine similarity* dari tempat wisata. Semakin terang warna dari suatu titik maka semakin kuat *similarity* dari kedua tempat wisata tersebut

#### 4.4 Implementasi *Collaborative Filtering*

Pada aplikasi rekomendasi tempat wisata “DestiMate” *collaborative filtering* diimplementasikan pada *home page* tempat wisata. Sehingga ketika *user* berada pada *home page*, maka sistem akan memberikan rekomendasi tempat wisata berdasarkan riwayat *rating user* tersebut. *Dataset* yang diperlukan untuk implementasi ini yaitu data *rating*, berikut merupakan datanya.

User_Id	Place_Id	Place_Ratings
0	1	179
1	1	344
2	1	5
3	1	373
4	1	101
...	...	...
9995	300	425
9996	300	64
9997	300	311
9998	300	279
9999	300	163

10000 rows x 3 columns

Gambar 5. *Dataset Rating*

Langkah pertama dalam implementasi ini yaitu melakukan *encoding* pada *user\_id* dan *place\_id*. Karena kedua ID tersebut bertipe *integer* maka akan menggunakan *layer IntegerLookup* dari *library* Tensorflow. Langkah selanjutnya yaitu melakukan normalisasi pada kolom *place\_ratings*. Normalisasi pada kolom *place\_ratings* bertujuan untuk menyederhanakan model, terutama pada *layer dense* terakhir. Normalisasi ini dilakukan untuk mengubah rentang nilai *rating* tempat wisata (1-5) menjadi skala (0-1).

Selanjutnya *dataset* dibagi menjadi data latih (*training set*) dan data uji (*testing set*) menggunakan objek *train\_test\_split* yang berasal dari *library* Scikit-Learn. Berikut merupakan *code*-nya.

```
x_train, x_test, y_train, y_test = train_test_split(df_collaborative[['User_Id_Encode',
'Place_Id_Encode']].values, df_collaborative['Place_Ratings'].values, test_size=0.2,
random_state=42, shuffle=True)
```

Pemisahan *dataset* menjadi data latih dan data uji adalah langkah penting dalam evaluasi model. Dengan menggunakan 80% data sebagai data latih, model dapat belajar dari pola dan hubungan antara *user* dan tempat wisata. Sementara itu, 20% data uji digunakan untuk menguji sejauh mana model mampu memberikan prediksi yang akurat pada data yang belum pernah dilihat sebelumnya.

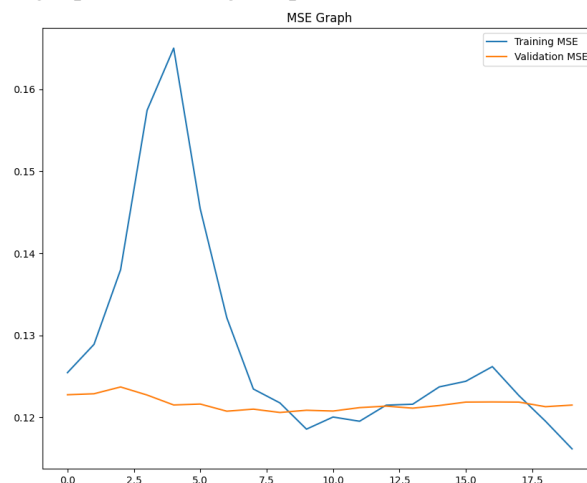
Setelah membagi *dataset* menjadi data *train* dan data *test* Langkah selanjutnya yaitu membuat model *recommender neural network* dengan menggunakan *library* Tensorflow. Berikut merupakan *code* model tersebut.

```
def build_recommender_net(num_users, num_place, embedding_size):
    input_layer = tf.keras.layers.Input(shape=(2,), name='input_layer')

    user_input = input_layer[:,0]
    place_input = input_layer[:,1]
    user_embedding = tf.keras.layers.Embedding(
        num_users,
        embedding_size,
        embeddings_initializer='he_normal',
        embeddings_regularizer=tf.keras.regularizers.l2(1e-6)
    )(user_input)
    user_bias = tf.keras.layers.Embedding(num_users, 1)(user_input)
    place_embedding = tf.keras.layers.Embedding(
        num_place,
        embedding_size,
        embeddings_initializer='he_normal',
        embeddings_regularizer=tf.keras.regularizers.l2(1e-6)
    )(place_input)
    place_bias = tf.keras.layers.Embedding(num_place, 1)(place_input)
    dot_user_place = tf.tensordot(user_embedding, place_embedding, 2)
    x = dot_user_place + user_bias + place_bias
    output = tf.nn.sigmoid(x)
    model = tf.keras.models.Model(inputs=input_layer, outputs=output)
    return model
```

Pada *code* di atas dua *embedding layers* digunakan untuk mengubah *input user* dan *place* menjadi representasi vektor dengan dimensi tertentu. Penambahan *layer bias* membantu dalam mengakomodasi perbedaan individu antar *user* dan tempat wisata. *Layer bias* memberikan model fleksibilitas untuk mengenali dan menyesuaikan preferensi serta kecenderungan unik dari setiap *user* dan tempat wisata. Dengan adanya *layer bias*, model dapat memperhitungkan karakteristik khusus dari setiap entitas, seperti apakah suatu tempat wisata memiliki *rating* yang biasanya lebih tinggi atau lebih rendah dari yang diperkirakan oleh fitur-fitur umum.

Operasi *dot product* antara vektor representasi *user* dan *place* diikuti dengan penambahan *user bias* dan *place bias* membentuk *output* model. Pada layer *output* menggunakan aktivasi *sigmoid* untuk menghasilkan *output* probabilitas yang dapat diinterpretasikan sebagai prediksi *rating* tempat wisata.



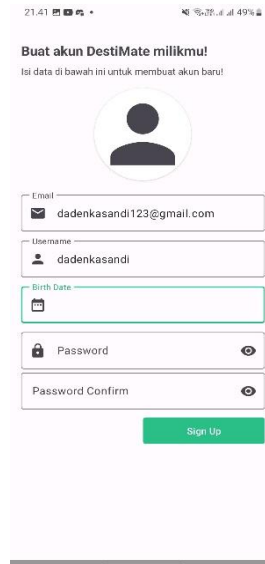
Gambar 6. Line Diagram Training MSE dan Validation MSE

Pada Gambar 6 hasil akhir dari pelatihan model *collaborative filtering* menunjukkan bahwa model memiliki kinerja yang baik dengan nilai *error* sebesar 0.1162 untuk data *train* dan 0.1215 untuk data *test*. Hal ini menunjukkan bahwa model memiliki kemampuan yang baik dalam mempelajari pola-pola yang terdapat dalam data tanpa terlalu menggeneralisasi atau terlalu beradaptasi dengan data *train*, sehingga tidak terjadi *overfitting* ataupun *underfitting*.

#### 4.5 Penggunaan Aplikasi dan Implementasi Sistem

Dalam sub-bab ini, akan dijelaskan *user experience* seputar penggunaan aplikasi "DestiMate" dan implementasi sistem rekomendasi di dalamnya. Aplikasi "DestiMate" memberikan pengguna akses yang mudah untuk menjelajahi tempat-tempat wisata berdasarkan preferensi pribadi mereka. Berikut merupakan penjelasan setiap *page*-nya dalam aplikasinya.

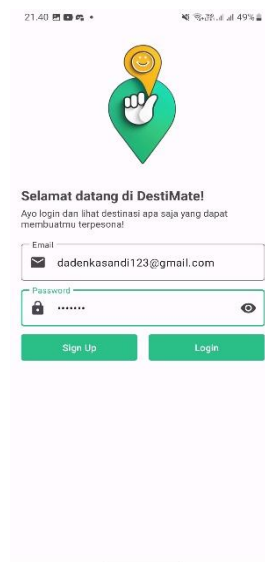
a. *Register Page*



Gambar 7. *Register Page* DestiMate

Pada Gambar 7, jika *user* belum memiliki akun maka *user* diharuskan untuk memasukkan data-data pribadi seperti *email*, *username*, *birthdate*, *password*, dan *profile picture*. Setelah itu *user* akan dialihkan ke *login page*.

b. *Login Page*

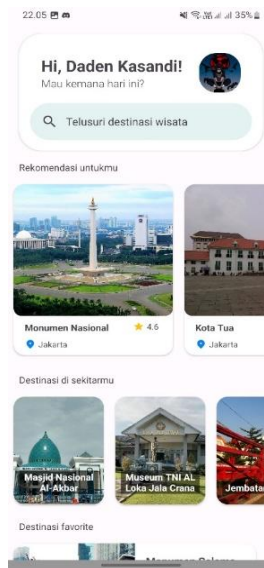


Gambar 8. *Login Page* DestiMate

Pada Gambar 8, *user* diharuskan untuk memasukkan *email* dan *password* agar dapat masuk ke dalam sistem aplikasi.



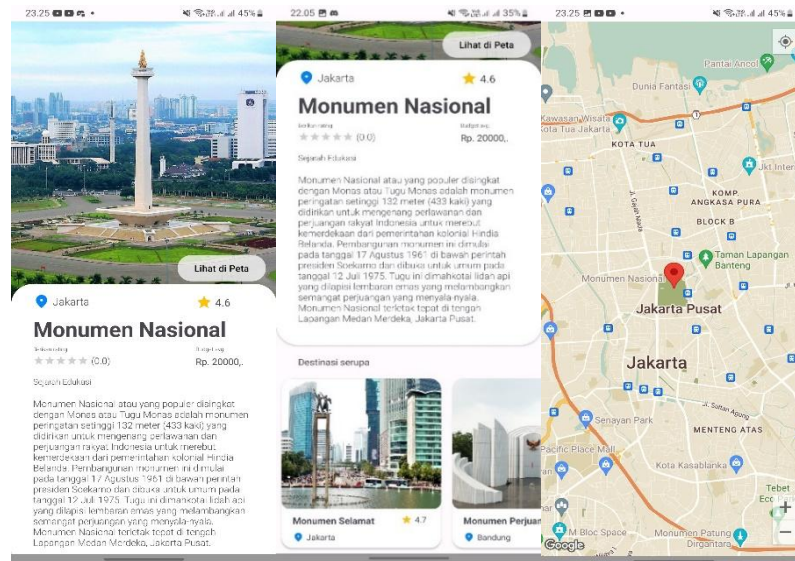
c. Home Page



Gambar 9. Home Page DestiMate

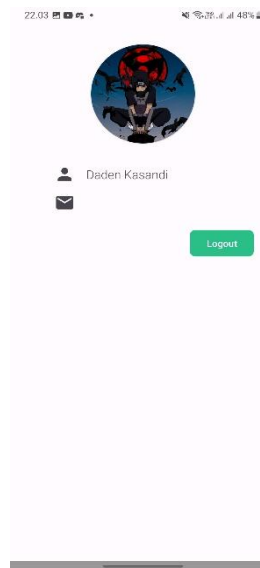
Pada Gambar 9 terlihat hasil implementasi *collaborative filtering* pada *home page*, *user* dapat melihat daftar rekomendasi tempat wisata yang mungkin menarik perhatian mereka. Selain itu juga terdapat wisata-wisata lainnya berdasarkan lokasi *user* dan wisata favorit.

d. Detail Page



Gambar 10. Detail Page DestiMate

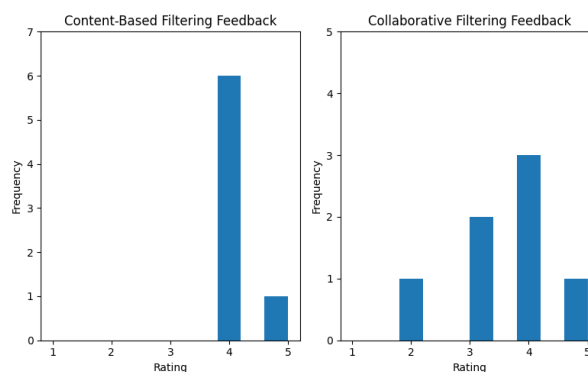
Pada Gambar 10 terdapat keterangan lebih lanjut terkait wisata yang dipilih oleh *user* seperti lokasi, *average rating*, *user rating*, kategori, dan deskripsi wisata. Terdapat juga fitur “Lihat di Peta”, fitur ini memungkinkan *user* dapat langsung mengakses *map*. Ketika *user* melakukan *scroll down*, terdapat implementasi *content-based filtering* pada *detail page*, *user* dapat melihat wisata-wisata serupa dengan wisata yang dipilih oleh *user*.

e. *Profile Page*Gambar 11. *Profile Page* DestiMate

Pada Gambar 11, *user* dapat melihat data pribadi mereka seperti *profile picture*, *username*, dan *email*. Ketika *user* menekan “Logout” maka *user* akan langsung dialihkan Kembali ke *login page*.

#### 4.6 System Testing

Setelah melakukan implementasi sistem rekomendasi pada aplikasi, selanjutnya akan dilakukan pengujian sistem tersebut kepada beberapa *user*. *User* nantinya akan melakukan *feedback* berdasarkan hasil rekomendasi dari riwayat *rating* mereka (*collaborative filtering*) dan juga *feedback* berdasarkan kemiripan satu tempat wisata dengan tempat wisata lainnya (*content-based filtering*). Berikut merupakan hasil dari *feedback user* dalam bentuk visualisasi diagram batang.

Gambar 12. Diagram Batang *Feedback User*

Pada Gambar 12 terlihat *feedback* untuk *content-based filtering* yang terdiri 6 orang yang memberi *rating* 4 dan 1 orang yang memberi *rating* 5. Hal ini menunjukkan bahwa sebagian besar *user* memberikan penilaian positif terhadap rekomendasi yang diberikan oleh sistem *content-based filtering*. *Rating* sebagian besar berada pada tingkat kepuasan yang tinggi, ditandai dengan penilaian 4 dari 5, yang mengindikasikan bahwa rekomendasi wisata serupa memiliki tingkat kemiripan yang tinggi.

Sedangkan pada *feedback* untuk *collaborative filtering* terdiri 1 orang yang memberi *rating* 2, 2 orang yang memberi *rating* 3, 3 orang yang memberi *rating* 4, dan 1 orang yang memberi *rating* 5. Hal ini menunjukkan adanya perbedaan persepsi dan tingkat kepuasan pengguna terhadap rekomendasi yang diberikan oleh sistem *collaborative filtering*. *Rating* yang bervariasi dapat mengindikasikan bahwa sistem ini mungkin memiliki tingkat akurasi dan konsistensi yang beragam dalam memberikan rekomendasi. Beberapa pengguna mungkin merasa puas dengan rekomendasi yang diberikan (*rating* 4 dan 5), sementara yang lain mungkin tidak sepenuhnya puas (*rating* 2 dan 3).

## 5. KESIMPULAN DAN SARAN

Dalam implementasi *content-based filtering* dengan teknik *bag of words* dan *cosine similarity* memberikan hasil yang memadai pada *detail page* tempat wisata. Dari implementasi *collaborative filtering* menunjukkan bahwa model telah berhasil memberikan hasil yang memuaskan, dengan nilai *error* yang rendah pada data *train* dan data *test*, yaitu 0.1162 dan 0.1215 secara berturut-turut. Hal ini menandakan bahwa model mampu mempelajari pola-pola preferensi pengguna terhadap tempat wisata tanpa mengalami *overfitting* atau *underfitting*. Hasil ini menegaskan bahwa model memiliki kemampuan yang baik dalam memberikan rekomendasi tempat wisata yang sesuai dengan preferensi pengguna pada aplikasi "DestiMate". Dengan demikian, implementasi *collaborative filtering* berhasil meningkatkan pengalaman pengguna dan memberikan nilai tambah yang signifikan pada aplikasi. Dari implementasi metode-metode sistem rekomendasi pada aplikasi "DestiMate" berikut merupakan beberapa saran yang dapat untuk pengembangan lebih lanjut.

- a. Melakukan pengembangan dataset dengan menambah jumlah data pengguna, tempat wisata, dan interaksi antara keduanya. Dengan *dataset* yang lebih besar, model dapat memperoleh pemahaman yang lebih baik tentang pola-pola preferensi pengguna.
- b. Melakukan evaluasi model dengan melibatkan lebih banyak *feedback* pengguna. Hal ini dapat membantu mendapatkan wawasan lebih mendalam tentang kualitas rekomendasi dan kebutuhan pengguna.
- c. Mempertimbangkan integrasi metode *hybrid collaborative filtering* dengan *content-based filtering* untuk memperkaya rekomendasi dengan mempertimbangkan konten atau fitur-fitur spesifik dari tempat wisata.
- d. Menyempurnakan antarmuka pengguna "DestiMate" untuk lebih responsif dan *user-friendly*, memberikan pengguna kemudahan dalam menavigasi, memberikan *feedback* terhadap rekomendasi, dan meningkatkan pengalaman pengguna secara keseluruhan.

## DAFTAR PUSTAKA

- [1] H. Jurnal, E. Purike, W. Kurniasih, F. W. Wulandari, A. Nirwani, and P. Perhotelan, "Jurnal Ilmiah Pariwisata Transaksi Digital Dan Perkembangan E-Tourism Indonesia," vol. 1, no. 2, 2022.
- [2] S. Jasmir, M. Kom, and Kom, "Implementasi Teknik *Data Cleaning* dan Teknik *Roughset* pada Data Tidak Lengkap dalam *Data Mining*," 2016.
- [3] J. Homepage, A. Roihan, P. Abas Sunarya, and A. S. Rafika, "IJCIT (*Indonesian Journal on Computer and Information Technology*) Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper," 2019.
- [4] Y. D. Pristanti and F. Windana, "Pengembangan Metode Neural Networks untuk Menentukan Karakter Seseorang," 2015.
- [5] R. H. Mondy and A. Wijayanto, "*Recommendation System with Content-Based Filtering Method for Culinary Tourism in Mangan Application*."
- [6] M. Fajriansyah, P. P. Adikara, and A. W. Widodo, "Sistem Rekomendasi Film Menggunakan *Content Based Filtering*," 2021.
- [7] J. Muara Sains, dan Ilmu Kesehatan, W. Trisari Harsanti Putri, and R. Hendrowati, "Penggalian Teks dengan Model *Bag of Words* Terhadap Data Twitter," vol. 2, no. 1, pp. 129–138, 2018.
- [8] A. H. Ritdrix and P. W. Wirawan, "Sistem Rekomendasi Buku Menggunakan Metode *Item-Based Collaborative Filtering*," 2016.
- [9] M. Romzi and B. Kurniawan, "Pembelajaran Pemrograman Python dengan Pendekatan Logika Algoritma," 2020.
- [10] I. Albanna, R. Tri Hadi Laksono, J. S. Informasi, T. Adhi, and T. Surabaya, "Implementasi *Pandas Data frame* sebagai Agregasi dan Tabulasi Penyajian Data Luaran Survei Kepuasan Pengguna Proses Pembelajaran dalam Pendidikan Tinggi."
- [11] R. F. Muharram *et al.*, "Implementasi *Artificial Intelligence* untuk Deteksi Masker Secara *Realtime* dengan Tensorflow dan *Ssdmobilenet* Berbasis Python," *Jurnal Widya*, vol. 3, no. 2, pp. 281–290, [Online]. Available: <https://jurnal.amikwidyaloka.ac.id/index.php/awl>