

# Rekomendasi Restoran Dengan Collaborative Filtering

Diana Rahmawati

Program Studi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Sebelas Maret

Email: [dianarahmawati01@student.uns.ac.id](mailto:dianarahmawati01@student.uns.ac.id)

## Abstrak

Restoran adalah bisnis di mana pelanggan dapat menikmati makanan mereka. Banyak restoran yang beragam menjadi faktor penting seseorang untuk memilih restoran. Membutuhkan waktu lama untuk memilih karena faktor banyaknya restoran yang tersedia. Hal ini karena kurangnya informasi yang tidak merata membuat pelanggan sulit untuk memilih restoran sehingga seseorang dapat mengambil keputusan yang tidak akurat. Dengan adanya rekomendasi restoran untuk pelanggan, maka akan membantu pelanggan dalam keputusan dalam memilih restoran atau tempat makan. Collaborative Filtering (CF) adalah salah satu teknik rekomendasi yang populer yang prediksi dan rekomendasinya berbasis pada nilai rating atau tingkah laku dari pengguna lain dalam sistem tersebut. Anggapan mendasar pada metode ini adalah opini pengguna lain dapat dipilih dan diagregasikan untuk memberikan prediksi dari preferensi pengguna aktif. Untuk menyelesaikan masalah diatas maka, dalam artikel ini akan membahas rekomendasi restoran dengan metode Collaborative Filtering dari data yang diambil dari kaggle.

**Kata kunci:** collaborative filtering, rekomendasi restoran

## 1. Pendahuluan

Restoran adalah bisnis di mana pelanggan dapat menikmati makanan mereka. Restoran memiliki berbagai cara penyajian seperti *take away*, *delivery*, dan *dine in*. Terdapat banyak menu dalam restoran yang berbeda. Banyak restoran yang beragam menjadi faktor penting seseorang untuk memilih restoran. Adapun restoran yang ada saat ini berusaha untuk memberikan yang terbaik dalam menarik minat pelanggan agar mengunjungi restorannya. Begitupun bisnis restoran hingga saat ini masih diyakini sebagai salah satu bisnis yang memiliki prospek yang menjanjikan dan mendapatkan keuntungan cukup menggiurkan (Farid & Fitriana, 2021). Membutuhkan waktu lama untuk memilih karena faktor banyaknya restoran yang tersedia. Hal ini karena kurangnya informasi yang tidak merata membuat pelanggan sulit untuk memilih restoran sehingga seseorang dapat mengambil keputusan yang tidak akurat.

Untuk mengatasi hal tersebut, dalam artikel ini dirancang dan dibuat sistem rekomendasi Restoran. Saat membuat sistem rekomendasi ini, dalam sistem rekomendasi paling sering digunakan adalah *collaborative filtering*. Cara *collaborative filtering* adalah rekomendasi untuk pengunjung baru didasarkan pada perkiraan selera yang diketahui oleh pengunjung lama untuk dijadikan prediksi selera yang tidak diketahui dari pengunjung baru. *Collaborative filtering* dapat dibagi menjadi dua metode utama: berbasis memori (*user based*) dan berbasis model (*item based*). Memori atau metode berbasis pengguna menggunakan riwayat untuk memilih pengunjung. Misalkan dalam kasus pengunjung restoran, jika ada pengunjung A yang menyukai restoran X, sedangkan ada pengunjung B yang mirip dengan selera pengunjung A dan pengunjung B belum pernah ke restoran X, maka pengunjung B akan direkomendasikan ke restoran X. Kelebihan dari pendekatan *user based* ini adalah dapat memberikan rekomendasi yang berkualitas tinggi, namun memiliki kelemahan yaitu meningkatnya kompleksitas komputasi dengan jumlah pengguna dan item.

Untuk metode berbasis model atau *item based*, gunakan kesamaan historis untuk nilai *rating* item. Misalkan jika terdapat pengunjung A yang menyukai restoran X, sedangkan *rating* restoran X mirip dengan restoran Y dan pengunjung A belum pernah ke restoran Y maka restoran Y dapat direkomendasikan kepada pengunjung A. Metode ini mengasumsikan kesamaan antara dua elemen yang terfokus pada satu nilai untuk memudahkan menemukan *neighbor* dalam sistem *offline*. Oleh karena itu, metode *item based* atau berbasis model dapat membuat rekomendasi dengan cepat, namun sayangnya kualitas rekomendasi yang dihasilkan rendah.

Dalam artikel ini, menggunakan *collaborative filtering* dengan *user based* dan data restoran dari Kaggle untuk memberikan rekomendasi restoran. Metode ini di gunakan karena telah banyak digunakan dalam mengambil keputusan karena memiliki banyak kelebihan dan dapat digunakan untuk mendefinisikan rekomendasi menggunakan data yang ada. Oleh karena itu, tujuan dari penelitian ini adalah untuk memberikan rekomendasi pemilihan restoran menggunakan *collaborative filtering* dengan *user based*.

## 2. Metode Penelitian

Terdapat beberapa tahapan metode dalam penelitian ini,

### 2.1. Pemilihan Data

Data yang digunakan dalam penelitian ini adalah data restoran yang ada di website kaggle. Data diambil dari website kaggle. Total kumpulan data yang digunakan berisi 10000 data pengunjung restoran. Data restoran meliputi *customer\_id*, *gender*, *latitude\_x*, *longitude\_x*, *latitude\_y*, *longitude\_y*, *vendor\_category\_en*,

delivery\_charge, serving\_distance, commission, delivery\_available, discount\_percentage, language, rank, restaurent\_rating, restaurent\_tag\_name, restaurant\_id.

## 2.2. Pemilihan Data

Data yang digunakan dalam penelitian ini adalah data restoran yang ada di website kaggle. Data diambil dari website kaggle. Total kumpulan data yang digunakan berisi 10000 data pengunjung restoran. Data restoran meliputi customer\_id, gender, latitude\_x, longitude\_x, latitude\_y, longitude\_y, vendor\_category\_en, delivery\_charge, serving\_distance, commission, delivery\_available, discount\_percentage, language, rank, restaurent\_rating, restaurent\_tag\_name, restaurant\_id.

## 2.3. Preprocessing Data

*Preprocessing* data adalah proses meningkatkan kualitas data dengan menghapus data yang tidak perlu dari data asli. Dalam penelitian ini, akan dilakukan *preprocessing* data antara lain:

### a. Data Cleaning

Dalam Pengolahan data *Data Cleaning* merupakan proses terpenting sehingga data benar-benar memenuhi persyaratan, konsisten, dan tidak ada redundansi. Dalam artikel ini pembersihan data untuk menghilangkan noise pada data, seperti data duplikat dan data kosong. Kemudian semua properti diperiksa dan dipilih untuk mendapatkan properti yang mengandung nilai paling konsisten dan relevan.

### b. Data Transformation

Langkah ini mengubah data menjadi nilai dalam format yang sesuai untuk diproses dalam pengembangan data.

## 2.4. Analisis Data

Analisis data adalah proses pengolahan data untuk menemukan informasi yang berguna yang dapat digunakan sebagai dasar pengambilan keputusan untuk memecahkan masalah.

## 2.5. Evaluasi Hasil

Penilaian untuk melihat hasil dari suatu proses atau program yang telah diselesaikan dan dicapai sebagai dasar pengambilan keputusan akhir.

## 3. Hasil dan Pembahasan

Penulisan artikel ini adalah studi kasus pada sistem rekomendasi untuk pemilihan restoran dengan data yang diambil dari website Kaggle. Berikut langkah – langkahnya :

### 3.1. Data Set

Data tahun 2020 diambil dari website kaggle. Total kumpulan data yang digunakan berisi 10000 data pengunjung restoran. 5 data teratas dari data restoran dapat dilihat pada Gambar 1.

```
data_resto = pd.read_csv("/content/resto.zip")
data_resto.head()
```

customer_id	gender	latitude_x	longitude_x	latitude_y	longitude_y	vendor_category_en	delivery_charge	serving_distance	commission	delivery_available	disco
SZ5J7X	male	-0.5050	0.0950	0.6187	0.5273	Restaurants	0.0	15.0	0.0	Yes	
WKYG878	male	0.2096	0.2810	-0.6010	0.0960	Restaurants	0.0	15.0	0.0	Yes	
UXCWXNG	male	0.1357	-78.6000	0.6187	0.5273	Restaurants	0.0	15.0	0.0	Yes	
B9HSJBN	male	-0.8800	0.0755	-0.6010	0.0960	Restaurants	0.0	15.0	0.0	Yes	
K3RGL6T	male	0.3879	0.5815	-0.1150	0.5460	Restaurants	0.7	15.0	0.0	Yes	

Gambar 1. Lima data dari data restoran

Data tersebut mempunyai 17 kolom yaitu customer\_id, gender, latitude\_x, longitude\_x, latitude\_y, longitude\_y, vendor\_category\_en, delivery\_charge, serving\_distance, commission, delivery\_available, discount\_percentage, language, rank, restaurent\_rating, restaurent\_tag\_name, restaurant\_id.

### 3.2. Preprocessing Data

#### a. Data Cleaning

Cek *missing value* yang terdapat pada data restoran. Syntax python dapat dilihat pada Gambar 2

```
data_resto.isnull().sum()

customer_id      0
gender           0
latitude_x       0
longitude_x      0
latitude_y       0
longitude_y      0
vendor_category_en 0
delivery_charge  0
serving_distance 0
commission       0
delivery_available 0
discount_percentage 0
language         0
rank             0
restaurent_rating 0
restaurent_tag_name 0
restaurant_id    0
dtype: int64
```

**Gambar 2. Missing Value**

Dari Gambar 2 tidak terdapat *missing value* maka lanjut ke langkah selanjutnya.

**b. Data Transformation**

Dalam data tersebut terlihat bahwa id *customer* adalah jenis objek dan karena tidak dapat menangani data tipe objek. Oleh karena itu selanjutnya akan memproses data dan menghasilkan nomor id unik untuk *customer*. Proses nomor id unik untuk *customer* terlihat dari Gambar 3.

```
list_idrestoran = list(data_resto['restaurant_id'].unique())
list_idrestoran[:10]

[106, 90, 43, 82, 189, 4, 191, 192, 157, 33]

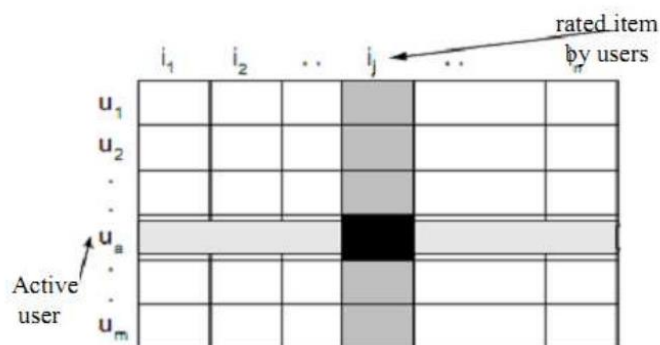
list_pelanggan = list(data_resto['id_customer'].unique())
list_pelanggan[:10]

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

**Gambar 3. Syntax id customer unik**

### 3.3. User based Collaborative Filtering

Metode ini adalah salah satu metode Collaborative Filtering dengan merekomendasikan item untuk suatu produk dengan menemukan pengguna yang mirip dengan pengguna lain. Rekomendasi didasarkan pada item-item yang sering dibeli/disukai oleh berbagai anggota kelompok. Metode yang paling umum digunakan adalah nearest neighbors method. Berdasar pada item-item yang telah dipilih oleh tetangga terdekat seorang user, item-item yang kemungkinan akan dipilih oleh user tersebut di masa yang akan datang diprediksi (Dzumiroh & Saptono, 2016). Pada metode ini dihasilkan sebuah matriks oleh pengguna  $u$  yang memberi rating pada item  $i$  (Hadi Ati et al., 2016). Dapat dilihat pada Gambar 4.



**Gambar 4. Matriks Interaksi user-item.**

Akan di cari matriks interaksi yang akan digunakan untuk rekomendasi *Collaborative Filtering*. Seperti terlihat pada Gambar 5 dan Gambar 6.

```
def generate_id(df, column_name, id_column):
    value_list = df[column_name].unique().tolist()
    int_value_list = [i for i in range(len(value_list))]
    id_list = []
    for i in df[column_name]:
        id_list.append(int_value_list[ value_list.index(i) ])
    df[id_column] = id_list

def similar_user_recs(user, interaction_mat, user_sim):
    if user not in interaction_mat.columns:
        return 'No data available on user {}'.format(user)
    sim_users = user_sim.sort_values(by=user, ascending=False).index[1:11]
    best = []
    most_common = {}

    for i in sim_users:
        max_score = interaction_mat.loc[:, i].max()
        best.append(interaction_mat[interaction_mat.loc[:, i]==max_score].index.tolist())
    for i in range(len(best)):
        for j in best[i]:
            if j in most_common:
                most_common[j] += 1
            else:
                most_common[j] = 1
    sorted_list = sorted(most_common.items(), key=operator.itemgetter(1), reverse=True)
    return sorted_list[:5]
```

Gambar 5. Syntax Fungsi Matrix

```
interaction_dict = dict()
for id_name in list_idrestoran:
    temp_user_order = []
    for customer in list_pelanggan:
        temp_user_order.append( data_resto[ (data_resto['id_customer']==customer) & (data_resto['restaurant_id']==id_name) ].shape[0] )
    interaction_dict[str(id_name)] = temp_user_order

interaction_dataset = pd.DataFrame(interaction_dict)
interaction_dataset
```

	106	90	43	82	189	4	191	192	157	33	28	44	105	289	13	265	104	148	110
0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	0	0	1
2	2	2	2	1	0	2	1	1	0	0	1	1	0	0	0	1	0	1	1
3	1	1	1	0	0	0	0	0	1	1	0	1	1	0	1	0	0	1	0
4	1	0	1	2	1	2	2	1	2	0	1	1	1	0	0	0	1	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
15262	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
15263	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
15264	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
15265	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15266	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Gambar 6. Syntax dan Hasil

### 3.4. Cosine Similarity

Cosine Similarity digunakan untuk melakukan perhitungan kesamaan dari Restoran. Metode pengklasifikasian yang digunakan pada sistem ini adalah dengan cara membandingkan kesamaan atau similaritas antara Restoran dengan pelanggan pertama, kemudian cara membandingkan kesamaan atau similaritas antara Restoran dengan kata pelanggan kedua, begitu seterusnya. Kemudian dicari jumlah similaritas yang tertinggi (Wahyuni et al., 2017). Rumus yang digunakan oleh cosine similarity adalah:

$$\text{Cosine Similarity} = \frac{A^T \cdot B}{|A| \cdot |B|}$$

Dari penjelasan diatas langkah yang pertama yaitu Transpose Matriks interaksi seperti pada Gambar 7.

```
interaction_dataset = interaction_dataset.T
interaction_dataset
```

	0	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	26	27	28	29	30	31	32	33	34	35	36	37	38	39	...	15227	1
106	1	0	2	1	1	0	1	0	0	0	0	1	1	0	0	1	2	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0	1	0	1	4	0	1	...	0	
90	0	1	2	1	0	2	2	0	0	0	0	0	1	0	1	1	0	0	0	0	1	1	3	0	0	0	1	0	1	0	0	0	1	0	1	1	0	1	0	0	...	0	
43	0	1	2	1	1	1	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	1	0	2	0	0	2	0	0	...	0	
82	1	1	1	0	2	2	0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	1	0	1	1	0	1	0	1	1	1	0	0	0	0	3	1	1	...	0	
189	0	1	0	0	1	0	1	0	1	0	1	0	0	0	1	2	0	0	0	1	0	1	2	0	0	1	0	0	3	0	1	1	0	0	1	0	1	3	1	2	...	0	
4	0	1	2	0	2	2	1	0	0	1	0	0	0	0	0	2	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	1	2	0	3	0	0	...	0	
191	0	1	1	0	2	2	1	1	1	0	1	1	1	0	1	1	1	1	0	1	1	1	1	0	0	0	0	0	2	1	2	1	1	1	0	1	0	5	0	0	...	0	
192	0	0	1	0	1	2	0	2	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0	1	1	1	2	...	0	
157	1	0	0	1	2	1	0	1	0	0	0	0	0	0	0	2	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	2	1	1	0	0	2	1	0	...	0
33	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	0	2	0	1	...	0	
28	0	0	1	0	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	1	1	0	0	0	0	1	2	1	0	2	1	1	...	0
44	0	0	1	1	1	0	1	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	1	0	0	1	0	1	2	0	2	...	0
105	0	0	0	1	1	1	1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	0	0	1	0	1	1	0	1	...	0	
289	0	0	0	0	0	0	1	0	1	0	0	0	1	0	1	2	2	1	0	0	0	0	1	1	0	1	0	0	1	1	0	1	0	0	2	1	1	1	0	1	...	0	

**Gambar 7. Matriks transpose interaksi**

Akan diubah dataset menjadi array yang jarang karena akan sangat membantu bagi sistem untuk bekerja pada matriks interaksi secara efisien. Dapat dilihat pada Gambar 8.

```
inter_sparse_data = sp.sparse.csr_matrix(interaction_dataset.values)
inter_sparse_data
```

<19x15267 sparse matrix of type '<class 'numpy.longlong'>'  
with 90005 stored elements in Compressed Sparse Row format>

**Gambar 8. Syntax ubah array**

Didapatkan Cosine Similarity Seperti Gambar 9.

```
user_similarity = cosine_similarity(inter_sparse_data.T)
user_sim = pd.DataFrame(user_similarity, index = interaction_dataset.columns, columns = interaction_dataset.columns)
user_sim
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1.000000	0.166667	0.306186	0.333333	0.588348	0.283473	0.129099	0.433013	0.447214	0.000000	0.500000	0.250000	0.176777	0.223607	0.158114	0.464238
1	0.166667	1.000000	0.680414	0.333333	0.588348	0.692935	0.688530	0.384900	0.447214	0.500000	0.500000	0.166667	0.353553	0.447214	0.632456	0.618984
2	0.306186	0.680414	1.000000	0.544331	0.680545	0.771517	0.737865	0.412479	0.182574	0.510310	0.204124	0.510310	0.649519	0.456435	0.516398	0.492764
3	0.333333	0.333333	0.544331	1.000000	0.457604	0.440959	0.602464	0.192450	0.000000	0.166667	0.000000	0.333333	0.353553	0.298142	0.316228	0.495188
4	0.588348	0.588348	0.680545	0.457604	1.000000	0.778312	0.557007	0.679366	0.526235	0.392232	0.588348	0.490290	0.416025	0.438529	0.558156	0.691939
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
15262	0.000000	0.333333	0.408248	0.000000	0.392232	0.377964	0.258199	0.000000	0.000000	0.500000	0.000000	0.000000	0.000000	0.000000	0.000000	0.371391
15263	0.000000	0.000000	0.204124	0.333333	0.196116	0.188982	0.000000	0.000000	0.000000	0.500000	0.000000	0.000000	0.353553	0.000000	0.316228	0.000000
15264	0.000000	0.000000	0.000000	0.333333	0.196116	0.188982	0.258199	0.288675	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.316228	0.000000
15265	0.000000	0.333333	0.000000	0.000000	0.196116	0.000000	0.258199	0.000000	0.447214	0.000000	0.500000	0.000000	0.000000	0.000000	0.316228	0.371391
15266	0.000000	0.333333	0.000000	0.000000	0.196116	0.000000	0.258199	0.000000	0.447214	0.000000	0.500000	0.000000	0.000000	0.000000	0.316228	0.371391

15267 rows x 15267 columns

**Gambar 9. Cosine Similarity**

### 3.5. Evaluasi Hasil

Step terakhir yaitu membuat rekomendasi sesuai dengan hasil Cosine Similarity Gambar 9. Dapat dilihat pada Gambar 10.

```
[ ] similar_user_recs(0, interaction_dataset, user_sim)
```

[('82', 10), ('104', 10), ('157', 8), ('106', 7), ('105', 1)]

**Gambar 10. Hasil Rekomendasi**

Dari output Gambar 10 pertama mewakili id restoran dan nilai kedua mewakili pengguna yang paling mirip dalam hal interaksi. Maka dapat disimpulkan bahwa tiga rekomendasi restoran tertinggi yaitu yang memiliki id restoran 82, 104, dan 157

#### 4. Kesimpulan

Teknik yang dapat menentukan rekomendasi pada *Recommendation System* untuk pelanggan baru dengan menggunakan riwayat pelanggan lama. Sistem rekomendasi untuk pelanggan baru dengan menemukan pengguna yang mirip dengan pengguna lain.

Pada tes rekomendasi restoran ini dapat disimpulkan bahwa tiga rekomendasi restoran tertinggi yaitu yang memiliki id restoran 82, 104, dan 157. Tes pada sistem rekomendasi pada dataset restoran dapat ditingkatkan dengan menerapkan berbagai kondisi dan meningkatkan berat pada matriks interaksi.

#### Daftar Pustaka

- Dzumiroh, L., & Saptono, R. (2016). Penerapan Metode Collaborative Filtering Menggunakan Rating Implisit pada Sistem Rekomendasi Pemilihan Film di Rental VCD. *Jurnal Teknologi & Informasi ITSmart*, 1(2), 54. <https://doi.org/10.20961/its.v1i2.590>
- Farid, M., & Fitriana, D. (2021). Rekomendasi Pemilihan Restoran Berdasarkan Rating Online Menggunakan Algoritma C4.5. *Jurnal Telekomunikasi Dan Komputer*, 11(1), 9. <https://doi.org/10.22441/incomtech.v11i1.9791>
- Hadi Ati, S., Saptono, R., & Salamah, U. (2016). Peningkatan Efektivitas Metode User-item based Collaborative Filtering pada Sistem Rekomendasi Wisata Kuliner Kota Solo. *Jurnal Teknologi & Informasi ITSmart*, 1(1), 01. <https://doi.org/10.20961/its.v1i1.574>
- Wahyuni, R. T., Prastiyanto, D., & Suprpto, E. (2017). Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi. *Jurnal Teknik Elektro*, 9(1), 18–23. <https://doi.org/10.15294/jte.v9i1.10955>

#### Data Kaggle

[Collaborative Filtering on Restaurant Data | Kaggle](#)