

# SISTEM REKOMENDASI LAYANAN PEMESANAN MAKANAN “EatAja” MENGUNAKAN ALGORITMA COLLABORATIVE FILTERING

## RECOMMENDATION SYSTEM FOR FOOD ORDERING SERVICE “EatAja” USING COLLABORATIVE FILTERING

Muhamad Naufal Syaiful Bahri<sup>1</sup>, I Putu Yuda Danan Jaya<sup>2</sup>, Burhanuddin Dirgantoro<sup>3</sup>, Istikmal<sup>4</sup>, Umar Ali Ahmad<sup>5</sup>, Reza Rendian Septiawan<sup>6</sup>

<sup>1,2,3,4,5,6</sup> Universitas Telkom, Bandung

<sup>1</sup>mnaufalbahri@student.telkomuniversity.ac.id, <sup>2</sup>yudadananjaya@student.telkomuniversity.ac.id,  
<sup>3</sup>burhanuddin@telkomuniversity.ac.id, <sup>4</sup>istikmal@telkomuniversity.ac.id, <sup>5</sup>umar@telkomuniversity.ac.id,  
<sup>6</sup>zaseptiawan@telkomuniversity.ac.id,

---

### Abstrak

EatAja adalah perusahaan *startup* di Indonesia yang memberikan solusi dalam pemesanan makanan pada restoran berbasis aplikasi *mobile*. Variasi menu dari berbagai restoran sering kali membuat bingung pengguna dalam memilih. Dari masalah tersebut, restoran biasanya memberikan rekomendasi menu terlaris. Namun, hal tersebut tidak menjadi solusi bagi pengguna yang memilih makanan sesuai selera mereka. Untuk restoran, mempromosikan menu tertentu berdasarkan selera pengguna cukup menantang karena pengguna memiliki preferensi masing-masing dan tidak tersedianya informasi tersebut. Dari banyaknya pengguna yang mungkin memiliki preferensi makanan yang sama, sistem rekomendasi menjadi fitur yang wajib diimplementasikan dalam aplikasi semacam itu yang melibatkan data dari banyak pengguna. Dalam penelitian ini digunakan metode *memory-based collaborative filtering* untuk mencari nilai kesamaan antara pesanan tiap pengguna. Dengan menggunakan data riwayat pesanan pada database EatAja dan dikombinasikan dengan data bantuan terhadap penilaian pengguna berdasarkan menu yang telah mereka pesan, sistem rekomendasi menghasilkan nilai *Mean Absolute Error* (MAE) 0.96823 dengan akurasi 99.03%. Hasil dari rekomendasi dapat diimplementasikan pada aplikasi untuk meningkatkan penjualan sebagai menu yang disarankan kepada pengguna.

Kata Kunci: Recommendation system, memory-based collaborative filtering, food ordering service

---

### Abstract

EatAja is a startup company in Indonesia that provides solutions in order in ordering food at mobile app-based restaurants. The variety of menu from various restaurants will make it confusing for users to choosing. Due to this problem, restaurants would be displaying the best sellers' menu. However, there is not solution for users who eat according to their taste. For restaurants, promoting specific menus based on users' taste is quite challenging because of users have preferences by themselves and unavailability about that information. Since many users may have similar food preferences, recommender system is a must-have feature to be implemented in such applications that involve data from many users. In this research are using memory-based collaborative filtering method to check a similarity between users' orders. By using real order history data from EatAja combined with generated auxiliary data to implicitly find customers' ratings towards menu they have been ordered, the recommender system obtains Mean Absolute Error (MAE) 0.96823 with the best accuracy is 99.03%. The result of the recommendation system can be applied to the application to be able to increase sales to the restaurant as a suggested menu.

Keywords: Recommendation system, memory-based collaborative filtering, food ordering service

### 1. Pendahuluan

Restoran merupakan tempat menjual berbagai macam makanan, minuman, dll. Selain untuk menghilangkan rasa lapar, restoran juga bisa menjadi tempat untuk diskusi atau bersenda gurau bersama teman. Namun, ada kalanya kita bingung dalam memilih restoran yang memiliki selera yang sama dengan diri kita. Sulitnya mencari restoran yang sesuai dengan selera kita menjadi masalah yang harus dipecahkan.

Dengan bantuan aplikasi yang mengimplementasikan sistem rekomendasi kita dapat dengan mudah menemukan restoran yang memiliki makanan dengan selera yang sama dengan kita [1]. Aplikasi akan menampilkan data yang sudah di proses berdasarkan riwayat pesanan pengguna untuk membantu menemukan makanan pada restoran yang

sesuai dengan selera pengguna. Tentu hal tersebut dapat membantu pengguna dalam memilih restoran dengan kriteria, karena setiap orang memiliki selera yang berbeda-beda.

Penelitian ini menjelaskan studi yang dilakukan dengan menerapkan teknik sistem rekomendasi *Collaborative Filtering* pada aplikasi Android bernama EatAja. EatAja adalah perusahaan *startup* di Indonesia yang bergerak dalam pemesanan makanan pada restoran.

*Collaborative Filtering* adalah salah satu algoritma yang paling umum digunakan dengan menggunakan data penilaian pengguna sebagai datasetnya [2]. Hal yang paling logis dalam menemukan selera mereka adalah dengan menganalisis perilaku mereka dan merekomendasikan pengguna dengan melihat makanan yang mirip dengan yang dipesan pengguna lain [3].

Dengan menggunakan metode *memory-based collaborative filtering* dengan pendekatan *user-based* pengguna dapat dengan mudah menemukan makanan berdasarkan pengguna lain yang memiliki selera yang sama. Dengan metode *memory-based collaborative filtering*, algoritma akan menemukan hubungan antara pengguna lain berdasarkan tabel penilaian (*rating*) dan memberikan rekomendasi menu pengguna yang mirip terhadap pengguna yang ingin diberikan rekomendasi [4]. Dataset tabel penilaian yang digunakan pada penelitian ini menggunakan data riwayat pesanan pengguna dan data bantuan yang telah disesuaikan setiap iterasinya.

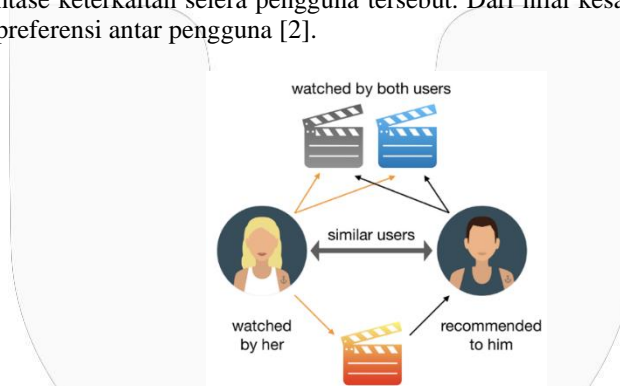
Tujuan dari penelitian ini adalah memberikan inovasi baru pada aplikasi EatAja dengan menampilkan fitur rekomendasi berdasarkan selera pengguna dan mengukur ketepatan akurasi dari algoritma *collaborative filtering*. Pada penelitian ini juga mengkaji pengujian keseluruhan aplikasi EatAja beserta fitur rekomendasi yang diujikan kepada calon pengguna atau pengguna saat ini.

Manfaat yang diharapkan dari penelitian ini adalah membantu mitra EatAja untuk memasarkan produk lainnya yang tidak masuk pada kategori *best seller*. Tentunya hal ini tidak hanya berguna untuk mitra EatAja, namun dapat berguna untuk pengguna yang memilih restoran atau menu makanan berdasarkan selera mereka.

## 2. Dasar Teori

### 2.1 Collaborative Filtering

Seperti yang dikatakan sebelumnya, algoritma *collaborative filtering* menggunakan data penilaian (*rating*) untuk menemukan pola kesamaan antar pengguna berdasarkan selera mereka. Semakin tinggi nilai kesamaan maka semakin tinggi pula persentase keterkaitan selera pengguna tersebut. Dari nilai kesamaan tersebut, didapatkan data rekomendasi berdasarkan preferensi antar pengguna [2].



Gambar 1. Ilustrasi proses algoritma *collaborative filtering*

Algoritma *collaborative filtering* dituntut untuk memiliki kemampuan untuk menangani sebaran data yang beragam untuk memberikan hasil dalam waktu singkat dan meningkatkan skala kualitas dari pengguna dan item [5]. Terdapat dua metode dalam menggunakan algoritma *collaborative filtering* yaitu dengan menggunakan metode *memory-based* dan metode *model-based*. Pada penelitian ini digunakan *memory-based* sebagai pendekatan dalam membuat sistem rekomendasi. Dalam proses mencari hasil rekomendasi, *memory-based* akan menghitung kesamaan antara pengguna dan item. Hasil yang diperoleh berdasarkan penilaian terhadap suatu item pengguna lain yang serupa oleh pengguna yang ingin diberikan rekomendasi. Algoritma tersebut menghitung dua pengguna atau item kemudian menghasilkan prediksi dengan mengambil rata-rata dari semua penilaian item [6].

*Memory-based collaborative filtering* memiliki kelebihan mudah diimplementasikan sebagai sistem rekomendasi dikarenakan prosesnya yang cukup mudah untuk dilakukan. Dengan menggunakan teknik *memory-based* pengguna juga dapat menambahkan data baru secara bertahap [7]. Namun, terdapat pula tantangan dalam algoritma *collaborative filtering* yaitu [8]: (1) *Cold Start*. Hal ini terjadi ketika terdapat pengguna baru yang terdaftar kepada aplikasi, karena tidak adanya informasi tentang pengguna tersebut; (2) Skalabilitas. *Collaborative filtering* terkadang membutuhkan data yang banyak yang telah dikumpulkan sebelumnya. Hal ini tentunya membutuhkan perhitungan komputasi yang baik pula; (3) Data yang beragam. Hal ini sering kali terjadi pada kasus belanja *online*, jumlah barang yang dijual sangat banyak. Karena hal tersebut memungkinkan produk yang terjual tidak dilakukan penilaian oleh pengguna. Sehingga memungkinkan produk yang dibutuhkan oleh pengguna lain, namun skala penilaian produk yang kecil ataupun bahkan tidak adanya penilaian sama sekali.

### 2.2 Cosine Similarity

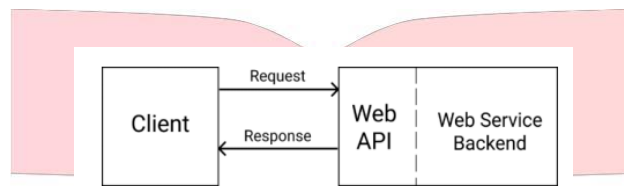
*Cosine Similarity* merupakan salah satu metode dalam mencari nilai kesamaan pada pendekatan *memory-based collaborative filtering*. Pendekatan *memory-based* menggunakan matriks interaksi antar pengguna dan seluruh menu yang ada dengan nilai *rating* sebagai *value*-nya untuk menghasilkan rekomendasi. Dari nilai matriks tersebut, sistem akan menghitung nilai kesamaan pengguna dengan pengguna lainnya. Hasil perhitungan tersebut akan menjadi matriks dengan *value*-nya adalah nilai kesamaan antar pengguna (*similarity matrix*) [9].

*Cosine similarity* secara teori adalah matriks yang membantu dalam menentukan seberapa mirip data terlepas dari ukurannya [10]. Pada *cosine similarity*, dataset akan dibuat menjadi vektor. Secara matematis, *cosine similarity* akan menghitung nilai sudut kosinus antara dua vektor yang diproyeksikan menjadi matriks *multi-dimensional* [11]. Berikut rumus untuk menentukan kesamaan kosinus antara dua buah sudut vektor:

$$\cos(x, y) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \|\vec{y}\|} = \frac{\sum_1^n x_i y_i}{\sqrt{\sum_1^n a_i^2} \sqrt{\sum_1^n b_i^2}} \dots \dots \dots (1)$$

### 2.3 REST API

REST API merupakan singkatan dari *Representational State Transfer Application Programming Interface*, sebuah jembatan untuk mengkomunikasikan antara sisi klien dan *web-service* [12]. Data yang akan diberikan oleh *web-service* dapat berupa JSON, *text*, dan XML. Pada penelitian ini digunakan JSON sebagai data keluaran dari *web-service*.



Gambar 2. Web API

Secara umum, REST API memberikan fasilitas yang memungkinkan pertukaran informasi antara sisi klien (aplikasi) dan sisi server. Dapat dilihat pada gambar 2, klien akan memberikan *request* pada Web API yang akan dikerjakan perintah tersebut oleh server menggunakan metode HTTP *request* dan Web API akan memberikan hasil *response* kepada klien [12].

### 2.4 JSON

JSON atau *JavaScript Object Notation* adalah sebuah pesan balikan objek (dimulai dengan kurung kurawal buka { dan diakhiri dengan kurung kurawal akhir }) yang mudah dibaca dan ditulis oleh manusia sekaligus mudah ditulis dan diurai oleh mesin. JSON merupakan salah satu pesan balikan (*response*) yang digunakan pada konsep REST API. JSON terdiri dari 2 struktur yaitu gabungan *name/value* dan sebuah list *value* yang biasa dikenal sebagai *array* dalam bahasa pemrograman [13].

```

    {
      "data": [
        - {
          user_id: [REDACTED],
          order_id: [REDACTED],
          nama_pesanan: "Madiem Makarim",
          payment_method: "CASH",
          status: "WAITING",
          nomor_meja: 24,
          point: 10,
          estimate: "15-30 Menit",
          waktu_order: "20/05/2021 07:49 PM",
          jumlah_pesanan: 3,
          total_price: 19000,
          rating: null,
          note: null,
          + menu_order: [-]
        },
        + (-),
        + (-),
        + (-),
        + (-),
        + (-),
        + (-),
        + (-),
        + (-),
        + (-),
        + (-),
        + (-),
        + (-),
      ]
    }
  
```

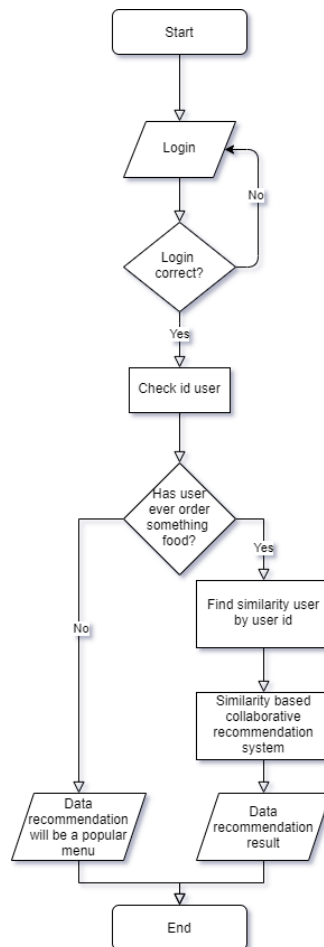
Gambar 3. Contoh JSON pada data *response order* EatAja

## 3. Perancangan

### 3.1 Gambaran Sistem *Collaborative Filtering*

Proses pembuatan sistem rekomendasi dimulai dengan membuat cara kerja alur dari sistem. Fitur rekomendasi terletak pada menu utama aplikasi EatAja. Untuk mengakses menu utama, pengguna diwajibkan memiliki akun dan melakukan *login* terlebih dahulu. Untuk memiliki akun pengguna dapat menggunakan *login* menggunakan Google atau mendaftarkan akun terlebih dahulu pada menu register. Setiap akun akan memiliki UUID masing-masing. UUID ini akan menjadi nilai *unique* setiap pengguna dan sistem akan mengecek apabila pengguna sudah pernah melakukan

pemesanan atau belum. Jika sudah, sistem akan memproses dan menghasilkan data rekomendasi, sebaliknya jika pengguna belum pernah melakukan pemesanan sebelumnya akan menghasilkan nilai rekomendasi berupa menu paling populer yang telah dipesan oleh seluruh pengguna. Pengguna baru artinya pengguna yang tidak pernah memesan di restoran mana pun. Untuk mendapatkan rekomendasi berdasarkan selera, pengguna harus memesan setidaknya sekali. Proses di atas digambarkan pada diagram *flowchart* yang ditunjukkan pada Gambar 4.



Gambar 4. *Flowchart* aplikasi EatAja.

### 3.2 Data yang dibutuhkan

Untuk membuat sistem rekomendasi, diperlukan data riwayat seperti *user\_id*, *menu\_id*, dan *rating*. Nilai *user\_id* dan *menu\_id* adalah *unique* dan *rating* memiliki nilai skala 1 – 5. Semua data memiliki total sejumlah 10,286 yang terdiri dari 10,000 data bantuan dan 286 data pengguna asli aplikasi EatAja dengan mengambil dari nilai *response* API EatAja. EatAja merupakan *startup* yang baru dan belum memiliki data yang cukup banyak pemesanan dalam pembuatan sistem rekomendasi, maka dari itu data bantuan diperuntukkan karena algoritma *collaborative filtering* membutuhkan data yang banyak dalam *rating* suatu menu dan untuk menghindari keragaman data. EatAja dapat ditemukan pada Google Play Store, namun tidak terdapat fitur rekomendasi karena fitur rekomendasi masih aplikasi percobaan dan sebagai wadah penelitian. Berikut tahapan dalam pembuatan data bantuan:

1. Sistem akan memberikan nilai angka acak dari 1 sampai 3000 sebagai nilai untuk kolom *user\_id* dengan asumsi 10,000 data akan diisi oleh 3,000 macam pengguna.
2. Sistem akan memberikan nilai acak dari keseluruhan *menu\_id* yang terdaftar pada database EatAja untuk nilai dataset pada kolom *menu\_id*.
3. Sistem akan melakukan 10,000 iterasi dan nilai *rating* akan diberikan nilai acak dari skala 1 – 5. Namun, jika pada proses iterasi tertentu memiliki *user\_id* dan *menu\_id* yang sama dengan iterasi sebelumnya, akan diberikan nilai yang sama dengan iterasi sebelumnya untuk menghindari nilai *error* yang tinggi pada dataset dikarenakan data yang sangat beragam pada nilai *rating*.

Dengan asumsi bahwa rata-rata pengguna akan memberikan nilai *rating* yang tinggi, sehingga *rating* pada skala 5 memiliki 3,989 data dan *rating* skala 4 memiliki 4,009 data. Untuk nilai *rating* lainnya seperti skala 1 memiliki 104 data, skala 2 memiliki 527 data, dan skala 3 memiliki 1,371 data.

### 3.3 Membuat *Interaction Matrix* (Matriks Interaksi)

Proses ini merupakan langkah awal penelitian ini untuk membuat sistem rekomendasi dengan algoritma *collaborative filtering*. Dari dataset awal akan dibuat menjadi data matriks interaksi antara pengguna dan seluruh menu EatAja dengan nilai *rating* yang dirata-ratakan sebagai isi setiap interaksinya. Nilai *rating* yang dirata-ratakan dengan asumsi terdapat pengguna yang memesan menu yang sama namun berbeda waktu.

### 3.4 Menghitung nilai *User Similarity*

Setelah membuat *interaction matrix*, langkah selanjutnya adalah menghitung nilai kesamaan antar pengguna (*user similarity*). Untuk mendapatkannya pada penelitian ini digunakan rumus dari *cosine similarity*. Nilai yang dihasilkan pada rumus ini akan memiliki rentang 0 – 1. Perbandingan suatu matriks menggunakan *cosine similarity* dinilai cukup baik karena menghasilkan nilai dari rentang 0 – 1 untuk menghindari hasil dimensi matriks yang buruk. Nilai ini menunjukkan bahwa hubungan semakin kuat apabila nilainya mendekati 1 dan semakin lemah apabila nilainya mendekati 0. Berikut rumus untuk mencari nilai kesamaan dengan menggunakan rumus *cosine similarity*:

$$\cos(x, y) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \|\vec{y}\|} = \frac{\sum_1^n x_i y_i}{\sqrt{\sum_1^n a_i^2} \sqrt{\sum_1^n b_i^2}} \dots \dots \dots (1)$$

### 3.5 Menentukan Hasil Rekomendasi Menu

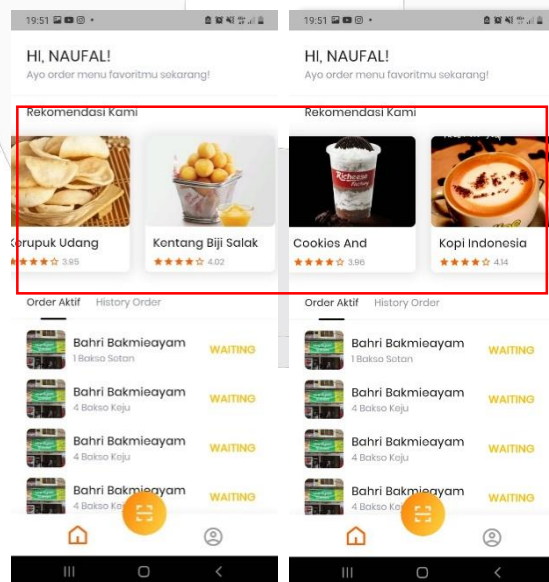
Pada proses ini adalah mengurutkan pengguna yang memiliki kesamaan yang paling tinggi dengan pengguna yang ingin diberikan rekomendasi. Kemudian dijabarkan menu apa saja yang pernah dipesan oleh pengguna yang memiliki nilai kesamaan yang tinggi dengan pengguna yang ingin diberikan rekomendasi. Menu yang akan keluar pada fitur rekomendasi ini adalah menu yang belum pernah dipesan oleh pengguna yang ingin diberikan rekomendasi.

### 3.6 Memberikan Data *Response* dan Implementasi

Pada tahap ini merupakan tahap saat data rekomendasi akan dikembalikan kepada *front-end* dari aplikasi *mobile*. Proses pengembalian menggunakan konsep REST API GET yang diminta oleh *front-end* aplikasi dan *back-end* mengembalikan dengan format JSON. Hasil *response* yang diterima *front-end* akan berupa list sebanyak 5 posisi teratas menu rekomendasi dan ditampilkan pada halaman utama aplikasi EatAja.

## 4. Hasil dan Pembahasan

Setelah selesai melakukan tahapan maka akan menghasilkan sebuah aplikasi EatAja demo yang sudah terdapat fitur rekomendasi. Aplikasi saat ini dapat digunakan pada perangkat Android saja. Berikut tampilan dari fitur rekomendasi yang terdapat pada aplikasi EatAja.



Gambar 5. Tampilan fitur rekomendasi aplikasi EatAja.

Pembuatan aplikasi dibuat menggunakan framework Flutter yang merupakan framework multi-platform yang dibuat oleh Google [14]. Untuk pembuatan sistem rekomendasi dibuat menggunakan bahasa pemrograman Python dan menggunakan FastAPI untuk konsep REST API yang dikembalikan dengan format JSON kepada *front-end* aplikasi.

Aplikasi siap diujikan kepada masyarakat umum, terutama calon pengguna dan pengguna saat ini. Proses pengujian aplikasi dilakukan dengan pengujian Alfa dan pengujian Beta. Sedangkan

## 1. Pengujian Alfa

Pengujian Alfa dilakukan dengan metode pengujian black box dengan menguji fitur rekomendasi berjalan setiap section-nya. Hasil pengujian dapat dilihat pada Tabel 1 yang menunjukkan bahwa fitur rekomendasi yang telah dibuat dan diuji didapatkan bahwa semua telah berfungsi dengan baik dan sesuai dengan harapan walaupun proses penampilan loading fitur rekomendasi sedikit lama dikarenakan proses pembelajaran yang cukup memakan waktu. Hal tersebut akan dijelaskan lebih lanjut pada bagian pengujian sistem rekomendasi.

Tabel 1. Pengujian Alfa

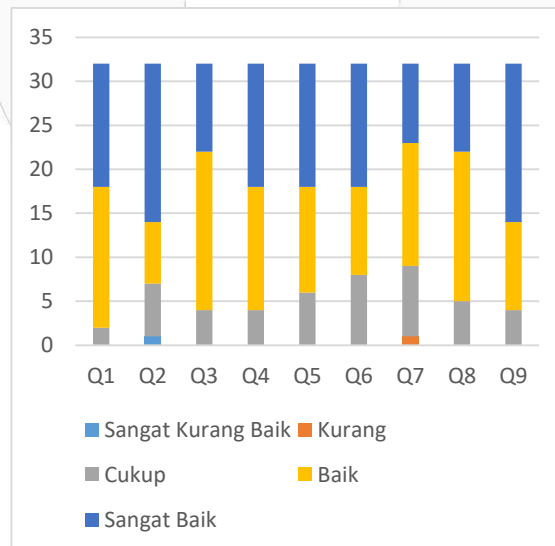
Data Masukan	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
<b>Klik menu utama</b>	Tampilan menu utama terdapat list rekomendasi sebanyak 5 buah	Berhasil menampilkan halaman detail menu rekomendasi. Namun proses <i>loading</i> rekomendasi sedikit lama.	Berhasil
<b>Klik salah satu menu rekomendasi</b>	Menampilkan halaman detail menu rekomendasi	Berhasil menampilkan halaman detail menu rekomendasi.	Berhasil

## 2. Pengujian Beta

Pengujian beta bertujuan untuk menguji aplikasi secara objektif dan dilakukan dengan media Google Form yang diujikan secara langsung kepada pengguna maupun calon pengguna aplikasi EatAja dalam bentuk kuesioner. Kuesioner secara teori adalah sekumpulan pertanyaan yang digunakan untuk memperoleh informasi dari responden terkait dengan pribadinya maupun hal-hal lain yang terkait dengan materi penelitian [15]. Pada pengujian kali ini juga digunakan skala Likert yang secara teori adalah suatu skala psikometrika yang umum digunakan dalam kuesioner, dan merupakan skala yang paling banyak digunakan dalam riset berupa survei [16].

Skema pengujian beta terdapat 10 kuesioner (9 kuesioner skala Likert, 1 kuesioner saran) dan terdapat link aplikasi yang tersedia pada Google Drive atau jika responden keberatan untuk meng-unduh disediakan video demo aplikasi EatAja beserta fitur rekomendasi. Kuesionernya yaitu:

1. Apakah aplikasi nyaman digunakan?
2. Apakah aplikasi mudah dioperasikan walau tanpa *user guide*?
3. Apakah aplikasi ini sudah membantu Anda dalam melakukan pemesanan makanan?
4. Apakah aplikasi ini cukup bermanfaat menurut Anda?
5. Apakah aplikasi memiliki tampilan yang menarik?
6. Apakah tampilan detail restoran dan detail menu makanan sudah cukup informatif?
7. Apakah aplikasi memiliki fitur yang lengkap?
8. Apakah fitur rekomendasi memiliki fitur yang lengkap?
9. Menurut Anda, secara keseluruhan aplikasi ini cukup memuaskan?



Gambar 6. Hasil pengujian beta.

Dari hasil pada Gambar 6 menunjukkan skor tertinggi terdapat pada kuesioner 2 dan kuesioner 9 yaitu 18 responden setuju bahwa aplikasi ini memuaskan dan mudah dioperasikan. Dan untuk kuesioner 2 ada 1 responden yang memilih bahwa aplikasi ini sulit digunakan. Hasil dari kuesioner 8 sehubungan dengan kuesioner fitur rekomendasi diperoleh bahwa fitur rekomendasi berfungsi dengan baik.

Selain pengujian aplikasi terdapat pengujian pada sistem rekomendasi itu sendiri. Pengujian sistem rekomendasi dilakukan dengan melihat hasil lamanya waktu eksekusi untuk menghasilkan rekomendasi dan pengujian akurasi menggunakan perhitungan Mean Absolute Error (MAE).

### 1. Pengujian *time execution* sistem rekomendasi

Pengujian ini melihat alur dari proses rekomendasi itu terbentuk. Proses tersebut meliputi: (1) pembuatan dataset; (2) pembuatan interaction matrix; (3) pembuatan; (3) proses rekomendasi. Hasil dari proses alur tersebut menggunakan 5 user\_id yang berbeda-beda dapat dilihat pada Tabel 2.

Tabel 2. Pengujian *time execution* sistem rekomendasi

No.	Jumlah Order	(1)	(2)	(3)	Total waktu (s)
User1	0	4.99	-	0.004	4.994
User2	28	5.35	0.04	1.33	6.72
User3	7	5.54	0.03	1.35	6.92
User4	3	5.56	0.03	1.35	6.94
User5	0	4.68	-	0.003	4.683

Dapat dicermati pada Tabel 2 bahwa proses banyak memakan waktu pada pembuatan dataset (1). Hal tersebut dikarenakan terjadinya proses REST API request data seluruh pesanan pada database EatAja. Proses lambat karena bergantung pada kecepatan request yang dilakukan kepada server. Pada proses rekomendasi juga bergantung pada pengguna tersebut adalah pengguna baru atau pengguna lama. Dapat dilihat pada proses rekomendasi (3) yang dihasilkan sangat singkat dikarenakan pengguna akan mendapatkan rekomendasi menu teratas yang paling banyak dipesan (best seller). Sedangkan pengguna lama perlu melakukan perhitungan pada proses user similarity berdasarkan pengguna tersebut. Pengguna baru juga tidak terdapat pembuatan interaction matrix karena tidak adanya data yang perlu dianalisis oleh sistem.

### 2. Pengujian akurasi dengan MAE

Pengujian ini dilakukan untuk mengetahui keakuratan rekomendasi yang dihasilkan dengan menggunakan metode k-fold cross validation. Teknik ini membagi jumlah dataset menjadi sebuah kumpulan data secara acak sebanyak k bagian. Dalam pengujian ini digunakan k sebanyak 5. Data akan dibagi secara acak sebanyak 5 bagian, lalu pada setiap iterasi data ke-i akan berkesempatan menjadi data uji dan data sisanya akan menjadi data latih. Dari hasil pengujian yang telah dilakukan didapatkan hasil MAE (Mean Absolute Error) sebagai berikut:

Tabel 3. Pengujian akurasi dengan MAE

Fold	MAE	Akurasi (%)
1	0.96632	99.03%
2	0.96812	99.03%
3	0.96726	99.03%
4	0.96842	99.03%
5	0.97101	99.03%
<b>Rata-rata</b>	0.96823	99.03%

Berdasarkan pada Tabel 3 diperoleh nilai rata-rata akurasi dan MAE masing-masing adalah 99.03% dan 0.96823. Semakin besar nilai MAE, maka akan semakin besar nilai error dari model tersebut, begitu pun sebaliknya. Sehingga dapat disimpulkan bahwa sistem rekomendasi dengan menggunakan memory-based collaborative filtering yang digunakan dapat dipakai dan akurat untuk diimplementasikan.

### 5. Kesimpulan dan Saran

Kinerja rekomendasi mempengaruhi penilaian pelanggan terhadap aplikasi dan dapat membawa dampak baik bagi restoran tersebut. Kinerja aplikasi memiliki pengaruh terhadap kenyamanan pelanggan dalam menggunakan aplikasi. Oleh karena itu, dibutuhkan kecepatan dalam pengolahan data dan keakuratan hasil rekomendasi. Berdasarkan jumlah 10,286 dataset yang digunakan untuk membentuk sistem rekomendasi menggunakan algoritma *collaborative filtering* dapat diambil kesimpulan yaitu: (1) Fitur rekomendasi dapat berjalan dengan baik berdasarkan penilaian objektif yang disimpulkan dari pengujian beta; (2) Waktu eksekusi sistem rekomendasi sangat acak karena bergantung pada lamanya *request* ketika pengambilan seluruh data pesanan pengguna pada database EatAja; (3) Dalam pengujian akurasi menghasilkan nilai *Mean Absolute Error* (MAE) sebesar 0.96823 dan nilai akurasi sebesar 99.03%. Dengan akurasi tersebut dapat dikatakan bahwa hasil yang diperoleh akurat dan dapat diimplementasikan pada aplikasi EatAja.

Saran untuk penelitian selanjutnya diharapkan mencampurkan algoritma *collaborative filtering* dengan algoritma yang membandingkan berdasarkan isi ataupun jenis dari restoran seperti algoritma *content-based recommendation*. Selain itu dapat mencoba menggunakan pendekatan *model-based collaborative filtering* seperti algoritma *Funk Singular Value Decomposition* (Funk SVD) atau dapat menggabungkannya dengan algoritma K-Means dalam mengategorikan pesanan.

## REFERENSI

- [1] B. Shapira, F. Ricci dan L. Rokach, *Recommender Systems Handbook*, Springer US, 2015.
- [2] S. P. Sahu, A. Nautiyal dan M. Prasad, "Machine Learning Algorithms for Recommender System - a comparative analysis," *International Journal of Computer Applications Technology and Research*, vol. 6, no. 2, pp. 97-100, 2017.
- [3] N. K. Kusumawardhani, M. Nasrun dan C. Setianingsih, "Web Recommended System Library Book Selection Using Item Based Collaborative Filtering Method," pp. 1-8, 2019.
- [4] S. Ghazarian dan M. A. Nematbakhsh, "Enhancing memory-based collaborative filtering for group recommender systems," *Expert Systems with Applications*, vol. 42, no. 7, pp. 3801-3812, 2015.
- [5] S. Natarajan, S. Vairavasundaram, S. Natarajan dan A. H. Gandomi, "Resolving data sparsity and cold start problem in collaborative filtering recommender system using Linked Open Data," *Expert Systems with Applications*, vol. 2020, no. 0957-4174, p. 113248, 2020.
- [6] C. R. Aberger, "Recommender : An Analysis of Collaborative Filtering Techniques," 2014.
- [7] P. B. Thorat, R. M. Goudar dan S. Barve, "Survey on Collaborative Filtering, Content-based," *International Journal of Computer Applications (0975 – 8887)*, vol. 110, pp. 31-36, 2015.
- [8] G. M. Dakhel dan M. Mahdavi, "A New Collaborative Filtering Algorithm Using K-means Clustering and Neighbors' Voting," dalam *2011 11th International Conference on Hybrid Intelligent Systems (HIS)*, Melacca, 2011.
- [9] M. Al-Ghamdi, H. Elazhary dan A. Mojahed, "Evaluation of Collaborative Filtering for Recommender Systems," (*IJACSA*) *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 3, pp. 559-565, 2021.
- [10] S. R., "Cosine Similarity," GeeksforGeeks, 2020 October 2020. [Online]. Available: <https://www.geeksforgeeks.org/cosine-similarity/>. [Diakses 2 June 2021].
- [11] S. Prabhakaran, "Cosine Similarity – Understanding the math and how it works (with python codes)," *Machine Learning +*, 22 October 2018. [Online]. Available: <https://www.machinelearningplus.com/nlp/cosine-similarity/>. [Diakses 2 June 2021].
- [12] L. Richardson, M. Amundsen, M. Amundsen dan S. Ruby, *RESTful Web APIs: Services for a Changing World*, O'Reilly Media, Inc., 2013.
- [13] M. I. Perkasa dan E. B. Setiawan, "Pembangunan Web Service Data Masyarakat Menggunakan REST API dengan Access Token," *Ultima Computing: Jurnal Sistem Komputer*, vol. X, pp. 19-26, 2018.
- [14] "Flutter," [Online]. Available: <https://flutter.dev/>. [Diakses 19 May 2021].
- [15] E. Nugroho, *Prinsip-prinsip Menyusun Kuesioner*, Universitas Brawijaya Press, 2018.
- [16] S. Sofyan, T. Setyaningsih dan N. Syamsiah, "Otomatisasi metode penelitian skala likert berbasis web," *Prosiding Semnastek*, 2015.