

Implementasi Sistem Rekomendasi Makanan pada Aplikasi EatAja Menggunakan Algoritma Collaborative Filtering

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

^{1,2,3,5,6}Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom
Bandung, Jawa Barat

⁴Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom
Bandung, Jawa Barat

mnaufalbahri@student.telkomuniversity.ac.id, yudadananjaya@student.telkomuniversity.ac.id,
burhanuddin@telkomuniversity.ac.id, istikmal@telkomuniversity.ac.id, umar@telkomuniversity.ac.id,
zaseptiawan@telkomuniversity.ac.id

Diterima: 24 Agustus 2021. Disetujui: 27 Maret 2022. Dipublikasikan: 30 Maret 2022.

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 80.63%. 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

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 80,63%. 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

I. PENDAHULUAN

Restoran merupakan tempat menjual berbagai macam makanan dan minuman. Selain untuk menghilangkan rasa lapar, restoran juga bisa menjadi tempat untuk diskusi atau bersenda gurau bersama teman. Namun, ada kalanya merasa bingung dalam memilih restoran yang memiliki

selera sesuai dengan keinginan. Sulitnya mencari restoran yang sesuai dengan selera menjadi masalah yang harus dipecahkan.

Dengan bantuan aplikasi yang mengimplementasikan sistem rekomendasi, secara mudah dapat ditemukan restoran yang memiliki makanan sesuai selera dan keinginan [1]. Aplikasi

akan menampilkan data yang sudah diproses berdasarkan riwayat pesanan pengguna untuk membantu menemukan makanan pada restoran yang sesuai dengan selera pengguna. Tentunya hal tersebut dapat membantu pengguna dalam memilih restoran dengan kriteria tertentu, karena setiap orang memiliki selera yang berbeda-beda.

EatAja adalah perusahaan *startup* di Indonesia yang bergerak dalam hal pemesanan makanan di sebuah restoran. EatAja memiliki layanan yang mempermudah proses pemesanan makanan pada restoran. Pelanggan dapat dengan mudah mengakses menu restoran hanya dengan melakukan *scanning QR Code*, dilanjutkan dengan melakukan pemesanan, dan pembayaran pada aplikasi EatAja. Dengan terus berkembangnya inovasi teknologi digital, EatAja ingin menghadirkan fitur baru yaitu sistem rekomendasi, namun dengan teknik yang berbeda dari yang sudah ada sebelumnya. Sistem rekomendasi biasanya hadir dengan menampilkan sejumlah menu yang angka penjualannya lebih tinggi atau diurut berdasarkan angka yang penjualannya tinggi dan harga yang sedang diskon. EatAja ingin menghadirkan sistem rekomendasi yang memiliki keterkaitan terhadap ketertarikan/selera setiap pengguna. Pada penelitian ini diterapkan sistem rekomendasi tersebut dengan menggunakan Algoritma *Collaborative filtering* kepada aplikasi EatAja.

Collaborative Filtering adalah salah satu algoritma yang paling umum digunakan, dimana memerlukan data penilaian pengguna sebagai datasetnya [2]. Hal yang paling logis dalam menemukan selera pengguna adalah dengan menganalisis perilaku pengguna dan merekomendasikannya berdasarkan makanan yang mirip dengan pesanan pengguna lain [3].

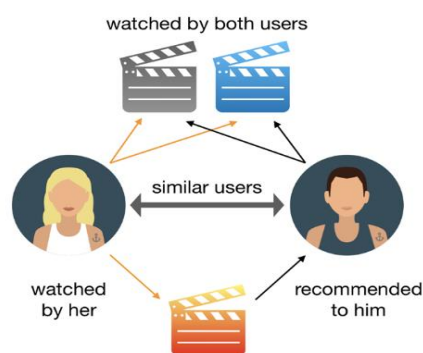
Menggunakan metode *memory-based Collaborative Filtering* dengan pendekatan *user-based*, pengguna dapat dengan mudah menemukan makanan berdasarkan pengguna lain dengan selera yang sama. Penerapan metode *memory-based Collaborative Filtering*, akan menemukan hubungan antara pengguna lain berdasarkan 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 berdasarkan 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 dilakukan 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 termasuk pada kategori *best seller*. Tentunya hal ini tidak hanya berguna untuk mitra EatAja, namun dapat berguna juga untuk pengguna yang memilih restoran atau menu makanan berdasarkan selera mereka.

II. TINJAUAN PUSTAKA



Gambar 1. Ilustrasi Proses Algoritma *Collaborative Filtering*

A. Collaborative Filtering

Seperti yang sudah dibahas 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].

Algoritma *Collaborative Filtering* dituntut untuk memiliki kemampuan dalam menangani sebaran data yang beragam dan memberikan hasil dalam waktu singkat serta meningkatkan skala kualitas dari pengguna dan item [5]. Terdapat dua metode dalam menggunakan algoritma *Collaborative Filtering* yaitu metode *memory-based* dan metode *model-based*. Pada penelitian ini digunakan metode *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 atas

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, namun belum ada informasi tentang pengguna tersebut; (2) Skalabilitas. *Collaborative filtering* terkadang membutuhkan banyak data 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. Oleh karena itu mungkin terdapat produk yang terjual tetapi tidak memiliki penilaian dari pengguna, sehingga memungkinkan terdapat produk yang dibutuhkan oleh pengguna lain, namun skala penilaian produk yang kecil ataupun bahkan tidak adanya penilaian sama sekali.

B. Cosine Similarity

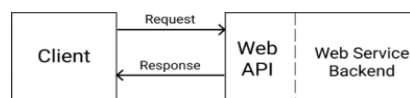
Cosine similarity merupakan salah satu metode dalam mencari nilai kesamaan pada pendekatan *memory-based collaborative filtering*. Pendekatan *memory-based* menggunakan matriks interaksi antar pengguna dan seluruh menu yang ada dengan nilai *rating* sebagai *value*-nya untuk menghasilkan rekomendasi. 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 kemiripan 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].

C. REST API

REST API merupakan singkatan dari *Representational State Transfer Application Programming Interface*, yaitu 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*.

Secara umum, REST API memberikan fasilitas yang memungkinkan pertukaran informasi antara sisi klien (aplikasi) dan sisi *server*. Dapat dilihat pada Gambar 2, klien memberikan *request*



Gambar 2. Web API

pada Web API, perintah tersebut akan dikerjakan oleh *server* menggunakan metode HTTP *request* dan Web API kemudian memberikan hasil *response* kepada klien [13].

D. JSON

```
{
  "data": [
    {
      "user_id": "XXXXXXXXXXXXXXXXXXXX",
      "order_id": "XXXXXXXXXXXXXXXXXXXX",
      "nama_pesanan": "Nadiem 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": [
        {
          "id": "XXXXXXXXXXXXXXXXXXXX",
          "nama": "XXXXXXXXXXXXXXXXXXXX",
          "harga": "XXXXXXXXXXXXXXXXXXXX",
          "jumlah": "XXXXXXXXXXXXXXXXXXXX",
          "keterangan": "XXXXXXXXXXXXXXXXXXXX"
        },
        {
          "id": "XXXXXXXXXXXXXXXXXXXX",
          "nama": "XXXXXXXXXXXXXXXXXXXX",
          "harga": "XXXXXXXXXXXXXXXXXXXX",
          "jumlah": "XXXXXXXXXXXXXXXXXXXX",
          "keterangan": "XXXXXXXXXXXXXXXXXXXX"
        },
        {
          "id": "XXXXXXXXXXXXXXXXXXXX",
          "nama": "XXXXXXXXXXXXXXXXXXXX",
          "harga": "XXXXXXXXXXXXXXXXXXXX",
          "jumlah": "XXXXXXXXXXXXXXXXXXXX",
          "keterangan": "XXXXXXXXXXXXXXXXXXXX"
        }
      ]
    }
  ]
}
```

Gambar 3. Contoh JSON pada Data Response Order EatAja

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 [14].

E. Penelitian Sebelumnya

Di dalam penelitian sebelumnya, tersedia banyak penelitian yang berhubungan dengan topik

sistem rekomendasi dengan menggunakan algoritma *Collaborative filtering*. Dari penelitian yang telah ada, terdapat beberapa tolak ukur untuk menyusun penelitian ini. Salah satu referensi berdasarkan penelitian oleh Liu Gang yang berjudul “*Personalized Recommendation of Tourist Attractions Based on Collaborative filtering*”. Pada penelitian ini dilakukan survei kuesioner yang bertujuan untuk mengumpulkan preferensi dari beberapa pengguna, dan menggunakan AHP dalam penentuan bobot atribut setiap pengguna. Hasil dari penelitian menunjukkan kesalahan dari perhitungan sistem kecil dan rekomendasi memiliki efek yang baik serta memungkinkan untuk pengembangan yang lebih luas [15].

Selanjutnya penelitian oleh Ashish Pal, Prateek Parhi dan Manuj Aggarwal yang berjudul “*An Improved Content Based Collaborative filtering Algorithm for Movie Recommendations*”. Pada penelitian ini digunakan algoritma *collaborative filtering* dengan gabungan metode *content based*. Hasil penelitian menunjukkan bahwa rekomendasi menggunakan konten *hybrid* menghasilkan MAE yang baik yaitu 0,90207 [16].

Terakhir adalah penelitian yang dilakukan oleh Ling Li, Ya Zhou, Han Xiong, Cailin Hu dan Xiafei Wei dengan judul “*Collaborative filtering based on user attributes and user ratings for restaurant recommendation*”. Di penelitian ini diterapkan algoritma *collaborative filtering* dengan tipe *user-based* untuk sistem rekomendasi restoran. Pada penelitian ini dilakukan 3 peningkatan yaitu peningkatan skor rata-rata untuk perhitungan *similarity*, ditambahkan faktor modifikasi untuk melemahkan kesalahan dan digunakan informasi pribadi pengguna untuk menghitung *similarity* berdasarkan atribut *user*. Hasilnya menunjukkan bahwa modifikasi algoritma tersebut dapat meningkatkan akurasi *similarity* dan memberikan rekomendasi yang lebih akurat untuk pengguna [17].

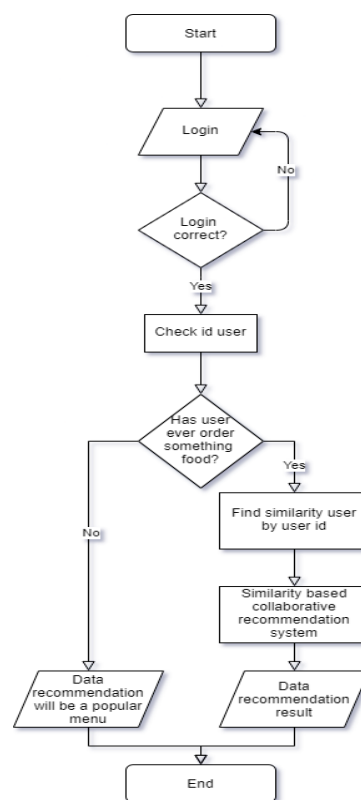
III. METODE PENELITIAN

A. Gambaran Sistem Collaborative Filtering

Proses pembuatan sistem rekomendasi dimulai dengan membuat cara kerja 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-

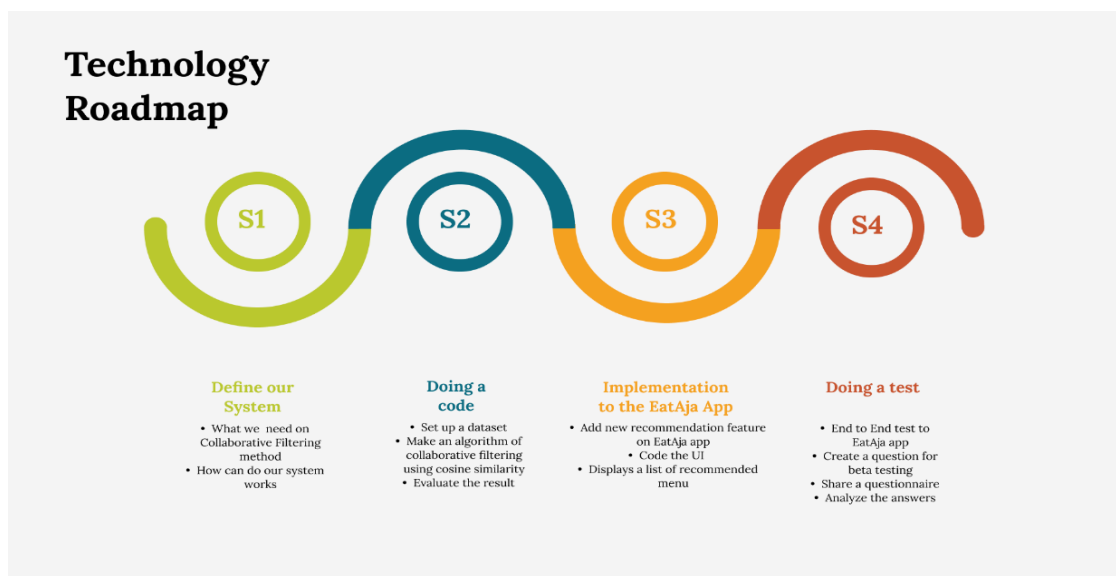
masing. UUID dapat dikatakan sebagai penanda ID dari setiap akun. UUID akan menjadi nilai *unique* setiap pengguna dan digunakan oleh sistem untuk mengecek apakah pengguna sudah pernah melakukan pemesanan atau belum. Jika sudah, sistem akan memproses dan menghasilkan data rekomendasi, sebaliknya jika 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.

B. Peta Jalan Teknologi



Gambar 4. Flowchart Aplikasi EatAja

Langkah-langkah yang kami lakukan dalam pengerjaan penelitian ini digambarkan pada Peta Jalan yang ditunjukkan di Gambar 5.



Gambar 5. Peta Jalan Teknologi

C. 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. Nilai *user_id* dan *menu_id* digambarkan sebagai hubungan antar pengguna dan menu. *Rating* digambarkan sebagai penilaian pengguna terhadap menu tersebut. Hal tersebut dibutuhkan karena acuan dasar dataset terhadap metode *Collaborative Filtering* yang digunakan untuk mengetahui pola relasi yang terjadi antar masing-masing pengguna dan setiap menu. Semua data memiliki total sejumlah 10286 yang terdiri dari 10000 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 3000 macam pengguna. Pembuatan 10000 data bantuan dengan jumlah kurang lebih sebanyak 3000 pengguna, dipilih dikarenakan alasan *Collaborative Filtering* membutuhkan

kumpulan data yang sangat besar untuk bisa mendapatkan nilai kesamaan yang dekat dan juga masalah *sparsity* data yang dapat menghasilkan jumlah *sparse matrix* yang bernilai 0.

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*.

Berdasarkan aplikasi pemesanan makanan lainnya yang memiliki penilaian berupa *rating*, rata-rata pengguna akan memberikan nilai *rating* yang tinggi, sehingga dengan *rating* pada skala 5 memiliki 3989 data dan *rating* skala 4 memiliki 4009 data. Untuk nilai *rating* lainnya seperti skala 1 memiliki 104 data, skala 2 memiliki 527 data, dan skala 3 memiliki 1371 data.

D. Membuat Interaction Matrix (Matriks Interaksi)

Proses ini merupakan langkah awal penelitian 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

bertujuan apabila terdapat pengguna yang memesan menu yang sama namun dalam waktu berbeda.

E. Menghitung nilai User Similarity

Setelah membuat *interaction matrix*, langkah selanjutnya adalah menghitung nilai kesamaan antar pengguna (*user similarity*). Untuk itu 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 akan semakin kuat apabila nilainya mendekati 1 dan semakin lemah apabila nilainya mendekati 0. Persamaan (1) digunakan untuk mencari nilai kesamaan dengan menggunakan rumus *cosine similarity*:

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

$$\cos(U1, U2) = \frac{\sum_1^n U1_i U2_i}{\sqrt{\sum_1^n U1_i^2} \sqrt{\sum_1^n U2_i^2}}$$

Dimana U1 dan U2 merupakan matriks 1x5 yang berisi data rating hasil dari proses sebelumnya.

F. Menentukan Hasil Rekomendasi Menu

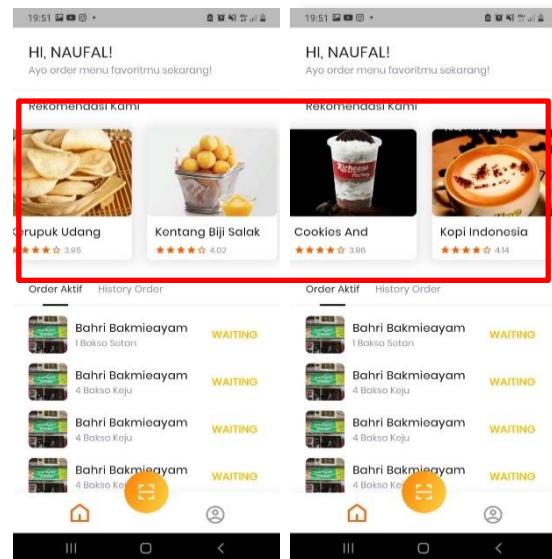
Pada proses ini adalah mengurutkan pengguna yang memiliki kesamaan paling tinggi dengan pengguna yang ingin diberikan rekomendasi. Kemudian dijabarkan menu apa saja yang pernah dipesan oleh pengguna yang memiliki nilai kesamaan tinggi dengan pengguna yang ingin diberikan rekomendasi. Menu yang akan keluar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

G. 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.

IV. 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 6. Tampilan Fitur Rekomendasi Aplikasi EatAja

Pembuatan aplikasi menggunakan *framework* Flutter yang merupakan *framework multi-platform* yang dibuat oleh Google [18]. 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.

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 I. HASIL UJI FITUR REKOMENDASI

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

2. Pengujian Beta

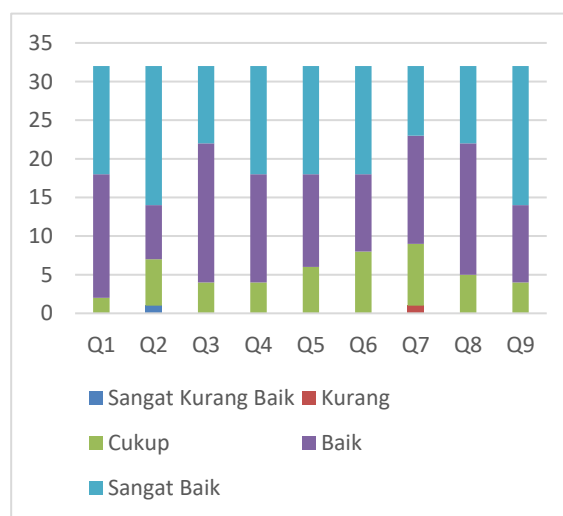
Pengujian beta bertujuan untuk menguji aplikasi secara objektif dan dilakukan dengan media Google Form yang diujikan secara langsung kepada pengguna maupun calon pengguna aplikasi EatAja dalam bentuk kuesioner. Kuesioner secara teori adalah sekumpulan pertanyaan yang digunakan untuk memperoleh informasi dari responden terkait dengan pribadinya maupun hal-hal lain yang terkait dengan materi penelitian [19]. Pada pengujian kali ini 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 [20].

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 mengunduh disediakan video demo aplikasi EatAja beserta fitur rekomendasi. Kuesionernya berisi:

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 pada tampilan utama berjalan dengan baik?
9. Menurut Anda, secara keseluruhan aplikasi ini cukup memuaskan?

Dari hasil pada Gambar 7 menunjukkan skor tertinggi terdapat pada kuesioner 2 dan kuesioner 9 yaitu 18 responden setuju bahwa aplikasi ini memuaskan dan mudah dioperasikan. 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.



Gambar 7. Hasil Pengujian Beta

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 proses rekomendasi; Hasil dari proses alur tersebut menggunakan 5 *user_id* yang berbeda-beda dapat dilihat pada Tabel 2.

TABEL II. HASIL PENGUJIAN *TIME EXECUTION*

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

No.	Jumlah Order	(1)	(2)	(3)	Total waktu (s)
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. Server EatAja memiliki spesifikasi memory 1 GB, 1 Intel CPU, 25 GB NVMe SSD, dan transfer rate 1000 GB. Pada proses rekomendasi juga bergantung pada pengguna, termasuk pengguna baru atau pengguna lama. Dapat dilihat pada proses rekomendasi (3) untuk user 1, waktu 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 tidak terdapat pembuatan interaction matrix karena tidak adanya data yang perlu dianalisis oleh sistem.

2. Pengujian akurasi dengan MAE

$$MAE = \frac{1}{N} \sum_i^N |y_{i,pred} - y_{i,true}| \quad (2)$$

MAE (*Mean Absolute Error*) pada persamaan (2) menyatakan nilai kesalahan prediksi dalam suatu model. Nilai dari MAE merupakan nilai metrik dengan skor berorientasi negatif, yang memiliki arti nilai yang lebih rendah lebih baik.

$$Accuracy = \frac{X - MAE}{X} * 100 \quad (3)$$

Setelah mendapatkan nilai MAE, dilakukan penghitungan untuk mencari nilai akurasi model tersebut. Dalam mencari akurasi dapat digunakan seperti penjabaran persamaan (3). X merupakan nilai maksimum dari MAE, yaitu 5 karena rating dari aplikasi EatAja maksimum bernilai 5.

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*) yang dapat dilihat pada Tabel 3.

TABEL III. HASIL PENGUJIAN MAE

Fold	MAE	Akurasi (%)
1	0,96632	80,67%
2	0,96812	80,63%
3	0,96726	80,65%
4	0,96842	80,63%
5	0,97101	80,57%
Rata-rata	0,96823	80,63%

Berdasarkan pada Tabel 3 diperoleh nilai rata-rata akurasi dan MAE masing-masing adalah 80,63% dan 0,96823. Semakin kecil nilai MAE, maka akan semakin kecil 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 cukup akurat untuk diimplementasikan.

V. 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 10286 dataset yang digunakan untuk membentuk sistem rekomendasi menggunakan algoritma *Collaborative Filtering* dapat diambil kesimpulan yaitu: (1) Pengembangan aplikasi EatAja dibangun menggunakan teknologi Flutter, FastAPI Python, dan Cloud VM dengan spesifikasi memory 1 GB, 1 Intel CPU, 25 GB SSD NVMe, dan transfer rate 1000 GB; (2) Fitur rekomendasi dapat berjalan dengan baik berdasarkan penilaian objektif aplikasi yang disimpulkan dari pengujian beta; (3) Waktu eksekusi sistem rekomendasi sangat acak karena bergantung pada lamanya request ketika pengambilan seluruh data pesanan pengguna pada database EatAja; (4) Dalam pengujian akurasi menghasilkan nilai *Mean Absolute Error* (MAE) sebesar 0,96823 dan nilai akurasi sebesar 80,63%. Dengan akurasi tersebut dapat dikatakan bahwa hasil

yang diperoleh akurat dan dapat diimplementasikan pada aplikasi EatAja.

Saran untuk penelitian selanjutnya diharapkan dilakukan pengujian secara objektif kecocokan rekomendasi dengan data menu restoran yang memiliki banyak variasi dan 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] M. Jalili, S. Ahmadian, M. Izadi, P. Moradi and M. Salehi, "Evaluating *Collaborative filtering* Recommender Algorithms: A Survey," *IEEE Access*, vol. 6, pp. 74003-74024, 2018.
- [2] S. P. Sahu, A. Nautiyal and 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 and C. Setianingsih, "Web Recommended System Library Book Selection Using Item Based *Collaborative filtering* Method," pp. 1-8, 2019.
- [4] D. Valcarce, A. Landin, J. Parapar and Á. Barreiro, "*Collaborative filtering* embeddings for memory-based recommender systems," in *Engineering Applications of Artificial Intelligence*, 2019.
- [5] S. Natarajan, S. Vairavasundaram, S. Natarajan and 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] M. R. Zarei and M. R. Moosavi, "A Memory-Based *Collaborative filtering* Recommender System Using Social Ties," in *2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA)*, 6 and, Iran, 2019.
- [7] P. B. Thorat, R. M. Goudar and S. Barve, "Survey on *Collaborative filtering*, Content-based," *International Journal of Computer Applications (0975 – 8887)*, vol. 110, pp. 31-36, 2015.
- [8] X. Wang, Y. Li, Y. Li, H. Zhang and B. Li, "*Collaborative filtering* and Data Privacy Protection: Overview and Challenges," in *2018 IEEE International Conference on Smart Cloud (SmartCloud)*, 2018.
- [9] M. Al-Ghamdi, H. Elazhary and 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/>. [Accessed 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/>. [Accessed 2 June 2021].
- [12] X. J. Hong, H. S. Yang and Y. H. Kim, "Performance Analysis of RESTful API and RabbitMQ for Microservice Web Application," in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, 2018.
- [13] L. Richardson, M. Amundsen, M. Amundsen and S. Ruby, *RESTful Web APIs: Services for a Changing World*, O'Reilly Media, Inc., 2013.
- [14] M. I. Perkasa and 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.
- [15] L. Gang, "Personalized Recommendation of Tourist Attractions Based on *Collaborative filtering*," in *13th International Conference on Intelligent Computation Technology and Automation (ICICTA)*, 2020.
- [16] A. Pal, P. Parhi and M. Aggarwal, "An improved content based *collaborative filtering* algorithm for movie recommendations," in *2017 Tenth International Conference on Contemporary Computing (IC3)*, 2017.
- [17] L. Li, Y. Zhou, H. Xiong, C. Hu and X. Wei, "*Collaborative filtering* based on user attributes and user ratings for restaurant recommendation," in *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2017.
- [18] "Flutter," [Online]. Available: <https://flutter.dev/>. [Accessed 19 May 2021].
- [19] E. Nugroho, *Prinsip-prinsip Menyusun Kuesioner*, Universitas Brawijaya Press, 2018.
- [20] U. Rahardja, N. Lutfiani and R. Rahmawati, "Persepsi Mahasiswa Terhadap Berita Pada Website APTISI," *Jurnal Ilmiah SISFOTENIKA*, vol. 8, p. 2, 2018.