

Rancang Bangun Sistem Rekomendasi Tempat Makan Menggunakan Algoritma Typicality Based Collaborative Filtering

Engineering of a Dining Place Recommendation System Using Typicality Based Collaborative Filtering Algorithm

Kevin Hartarto Muliadi¹, Caecilia Citra Lestari²

^{1,2}Jurusan Teknik Informatika Fakultas Teknologi Informasi, Universitas Ciputra
E-mail: ¹k.hartarto9@gmail.com, ²caecilia.citra@ciputra.ac.id

Abstrak

Makanan merupakan bagian penting bagi manusia baik sebagai kebutuhan primer maupun sebagai bagian dari gaya hidup seseorang. Tempat makan yang semakin banyak dan tawaran makanan yang beragam, membuat seseorang memiliki lebih banyak pilihan akan tempat makan yang dapat dikunjungi. Dengan dibantu oleh teknologi, sekarang seseorang bisa mencari rekomendasi dengan menggunakan algoritma sistem rekomendasi. Saat ini ada tiga algoritma sistem rekomendasi yang populer yaitu Content Based Filtering (CBF), Collaborative Filtering (CF) dan metode *hybrid*. Metode CF merekomendasikan sebuah *item* kepada pengguna dengan memprediksi preferensi dari pengguna aktif terhadap sejumlah *item* berdasarkan preferensi dari pengguna atau *item* lain yang mirip. Ada dua jenis metode dalam CF yaitu User Based CF dan Item Based CF. Terdapat sebuah metode baru yang dikembangkan dari metode User Based CF, metode ini adalah Typicality Based CF. Typicality Based CF (TyCo) memiliki kelebihan yang tidak dimiliki metode CF lainnya dapat memberikan prediksi yang akurat walau data terbatas, dapat melakukan clustering tanpa algoritma tambahan dan dapat mengatasi masalah cold-start yang biasa dialami metode CF. Berdasarkan pengujian yang telah dilakukan diketahui bahwa aplikasi ini memiliki nilai rata-rata Mean Absolute Error (MAE) sebesar 1.366 yang disebabkan karena kurangnya data training.

Kata kunci: penyaringan kolaboratif, sistem rekomendasi, tempat makan, tipikalitas

Abstract

Food is an important part of human both as primary need and as a part of a person's lifestyle. With many dining places to be choose a person can become more hesitant when choosing a dining place. The development of technology makes it easier for someone to find recommendations, one of them is by using a system recommendation algorithm. With the help of technology, now someone can look for recommendations using a system recommendation algorithm. Currently there are three popular system recommendation algorithms, that is Content Based Filtering (CB), Collaborative Filtering (CF) and hybrid methods. CF method recommends an item to user by predicting the preferences of an active user on a number of items based on preferences of other similar users or items. There are two types of methods in CF, User Based CF and Item Based CF. There's a new method developed from User Based CF this method is called Typicality Based CF. Typicality Based CF (TyCo) has advantages that other CF methods do not have that can provide accurate predictions even with sparse data, can do clustering without additional algorithms and can overcome cold-start which commonly happens on CF methods. Based on the tests that have been done it is known that this application has average Mean Absolute Error (MAE) of 1.366, which caused by sparse dataset.

Keywords: collaborative filtering, dining place, recommendation system, typicality

1. PENDAHULUAN

Makanan sangat penting bagi manusia baik sebagai kebutuhan primer maupun sebagai bagian dari gaya hidup seseorang. Sekarang tempat makan baru semakin banyak dan hal ini menyebabkan seseorang semakin sering makan diluar. Dalam memilih tempat makan baru biasanya seseorang akan mencari rekomendasi. Rekomendasi dicari agar tempat makan baru yang akan dikunjungi sesuai dengan preferensi dan tidak mengecewakan. Rekomendasi tempat makan bisa didapatkan dari berbagai sumber.

Berdasarkan hasil survei yang dilakukan, 82.1% responden mendapatkan rekomendasi tempat makan dari teman mereka dan 66.1% responden mendapatkan rekomendasi tempat makan dari sosial media. Berdasarkan hasil survei diketahui bahwa terdapat tiga faktor yang menurut responden masih tidak sesuai dengan preferensi. Ketiga faktor tersebut adalah rasa, porsi dan harga. Persentase responden yang menyatakan rasa makanan dari rekomendasi tempat makan tidak sesuai sebanyak 60.7%, faktor porsi sebanyak 42.9% dan faktor harga sebanyak 28.6%.

Sudah ada penelitian terdahulu mengenai aplikasi rekomendasi tempat makan seperti penelitian tentang aplikasi rekomendasi tempat makan dengan menggunakan algoritma slope one yang merekomendasikan tempat makan dengan memprediksikan rating yang akan diberikan pencari ke setiap tempat makan [1]. Penelitian dengan menggunakan algoritma squeezer dan term frequency ranking yang mencari tempat makan berdasarkan kemiripan nama tempat makan dengan kata pencarian yang dimasukkan oleh pengguna [2]. Dan penelitian dengan menggunakan algoritma collaborative filtering (CF) yang memberikan rekomendasi berdasarkan tempat makan yang disukai oleh pengguna lain yang memiliki pola kesukaan yang sama dengan pencari [3].

Terdapat tiga metode sistem rekomendasi yang populer yaitu Content Based (CB), Collaborative Filtering (CF) dan metode *hybrid*. Metode CF adalah metode yang memberikan rekomendasi dengan memprediksikan preferensi pengguna berdasarkan preferensi dari pengguna atau item lain yang mirip. Metode CF dibagi menjadi dua jenis metode yaitu User Based CF dan Item Based CF. User Based CF merupakan metode CF yang memprediksikan preferensi pengguna terhadap *item* berdasarkan preferensi pengguna lain yang mirip terhadap *item*. Terdapat sebuah metode baru pengembangan dari metode User Based CF yaitu Typicality Based Collaborative Filtering.

Typicality Based Collaborative Filtering (TyCo) adalah pengembangan dari metode *user based* CF. Perbedaan dari metode ini dengan metode CF lainnya adalah pada TyCo terjadi pengelompokan (*clustering*) terhadap *item* dan pengguna. Pengelompokan ini dilakukan tanpa menggunakan algoritma tambahan. Kelebihan dari metode ini adalah metode ini dapat mengatasi masalah utama pada metode CF yaitu awalan dingin (*cold-start*) dan lebih akurat 6.35% daripada metode CF lainnya [4].

Dalam penelitian terdahulu yang digunakan algoritma CF dalam sistem rekomendasinya digunakan *rating* tempat makan sebagai kriteria rekomendasi. Dalam penelitian ini menggunakan empat buah *rating* yang lebih spesifik yaitu rasa dan porsi makanan serta kebersihan dan pelayanan tempat makan. Oleh karena jumlah *rating* yang banyak maka digunakan algoritma TyCo dalam membangun aplikasi ini.

TyCo digunakan karena TyCo mencari *neighbor* dari pencari berdasarkan tipikalitas dan memprediksi *rating* berdasarkan *rating* sebenarnya dari *neighbor*. Selain itu penggunaan TyCo didasari bahwa dalam penelitian sebelumnya dijelaskan bahwa algoritma ini dapat mengatasi masalah *cold-start*. Diharapkan dengan digunakannya algoritma TyCo, sistem rekomendasi dalam aplikasi ini dapat memberikan hasil rekomendasi yang lebih akurat dan juga dapat memberi tahu seberapa efektif dan cocok algoritma ini dalam memberikan rekomendasi tempat makan.

2. METODE PENELITIAN

2.1 Sistem Rekomendasi

Sistem rekomendasi adalah sistem yang dibuat dengan tujuan untuk membantu pengguna untuk mengetahui *item* yang mungkin mereka tertarik. Sistem rekomendasi banyak diimplementasikan di *marketplace*, *social media* dan lainnya. Sekarang terdapat tiga metode sistem rekomendasi yang populer digunakan yaitu *collaborative filtering*, *content based* dan metode *hybrid*.

2.1.1 Content Based Filtering

Content based filtering merupakan algoritma yang cukup populer digunakan untuk sistem rekomendasi. Algoritma ini populer karena mudah digunakan dan kerjanya juga cepat. Algoritma penyaringan berbasis konten merekomendasikan sebuah item kepada pengguna berdasarkan kemiripan antara deskripsi item dan profil dari minat pengguna [5].

2.1.2 Collaborative Filtering

Collaborative filtering atau penyaringan kolaboratif merupakan salah satu algoritma sistem rekomendasi yang populer. Metode ini memprediksi preferensi dari pengguna aktif terhadap sejumlah item berdasarkan preferensi dari pengguna lain yang mirip atau item. Ada dua jenis metode *collaborative filtering* yaitu *user based collaborative filtering* dan *item based collaborative filtering*.

2.1.3 Hybrid

Hybrid collaborative filtering merupakan algoritma yang menggabungkan algoritma lain dengan algoritma *collaborative filtering*. Biasanya pendekatan hybrid menggunakan algoritma penyaringan berbasis konten dan penyaringan kolaboratif dimana algoritma penyaringan berbasis konten digunakan untuk mengisi matriks *rating* dan algoritma penyaringan kolaboratif digunakan untuk rekomendasi [6].

2.2 Typicality Based Collaborative Filtering

Typicality based collaborative filtering atau penyaringan kolaboratif berbasis tipikalitas merupakan pengembangan dari *user based collaborative filtering*. *User based collaborative filtering* adalah salah satu jenis metode *collaborative filtering*. Cara kerja dari *user based collaborative filtering* adalah mencari pengguna-pengguna yang memiliki pola kesukaan yang sama dengan pengguna aktif (*neighbor* dari pengguna aktif) dan merekomendasikan *item* yang disukai oleh *neighbor* kepada pengguna aktif.

2.3 Cosine Similarity

Cosine similarity merupakan metode yang digunakan untuk mencari kemiripan antara dua titik dengan menghitung derajat relevansi. Semakin relevan atau mirip, nilai *similarity* akan bernilai satu, dan sebaliknya semakin tidak relevan atau tidak mirip, nilai *similarity* akan bernilai nol. Metode ini akan memberikan nilai berupa angka diantara nol sampai dengan satu. *Cosine Similarity* dihitung dengan menggunakan persamaan (1).

$$Sim(A, B) = Cos(\theta) = \frac{A * B}{||A|| ||B||} \quad (1)$$

2.4 Mean Absolute Error

Mean absolute error (MAE) adalah metode yang umum digunakan untuk pengujian sistem rekomendasi. MAE digunakan untuk menghitung perbedaan antara nilai prediksi *rating*

dengan nilai *rating* pengguna sesungguhnya [7]. Semakin besar nilai MAE maka semakin tidak akurat prediksi *rating* dari sistem rekomendasi. Sebaliknya kecil nilai MAE maka semakin akurat prediksi *rating* dari sistem rekomendasi MAE dihitung dengan menggunakan persamaan (2).

$$MAE = \frac{\sum_{n=1}^N |\hat{r}_n - r_n|}{N} \quad (2)$$

\hat{r}_n merupakan nilai prediksi *rating*. r_n merupakan nilai *rating* pengguna sesungguhnya. Dan N merupakan jumlah pasangan *rating* antara hasil prediksi sistem rekomendasi dengan *rating* pengguna sesungguhnya.

2.5 Google Maps API

Google Maps API adalah layanan yang diberikan oleh Google untuk dapat menggunakan data pemetaan mereka. Ada beberapa layanan yang ada di *Google Maps API* seperti *Places SDK for Android*, *Maps SDK for Android*, *Direction API* dan *Distance Matrix API*. *Places SDK for Android* adalah layanan *Google Maps API* untuk *Android* yang memungkinkan dibangunnya aplikasi yang dapat mengetahui bisnis lokal dan tempat lain yang berada dekat dengan perangkat [8].

2.5.1 Distance Matrix API

Distance Matrix API adalah layanan yang menyediakan jarak dan waktu tempuh perjalanan berdasarkan matiks tempat asal dan destinasi [9]. Untuk mengirimkan *request* ke *Distance Matrix API* dibutuhkan jenis *output* yang diinginkan dan parameter utama. Jenis *output* dari *request* yaitu *JSON* dan *XML*. Parameter utama yang dibutuhkan *request* yaitu kunci API, data asal dan data tujuan. Data asal dan tujuan dapat berupa alamat, posisi lintang bujur pengguna dan id lokasi. Data yang dikembalikan oleh *request* meliputi *status code* dari *request*, alamat asal, alamat tujuan, dan baris yang berisi status, durasi dan jarak.

2.5.2 Places API

Places API adalah layanan yang menyediakan data mengenai suatu lokasi seperti perkantoran, *mall* hingga tempat makan. Untuk mengirimkan *request* ke *Places API* dibutuhkan jenis *output* yang diinginkan dan parameter utama. Jenis *output* dari *request* yaitu *JSON* dan *XML*. Parameter utama yang dibutuhkan *request* yaitu kunci API, id lokasi. Data yang dikembalikan oleh *request* meliputi *status code* dari *request*, dan *result* yang berisi informasi detail mengenai tempat yang diminta.

2.5.3 Geocoding API

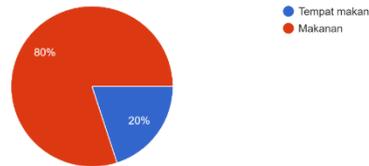
Geocoding API adalah layanan yang menyediakan data geografis dari suatu lokasi. Untuk mengirimkan *request* ke *Geocoding API* dibutuhkan jenis *output* yang diinginkan dan parameter utama. Jenis *output* dari *request* yaitu *JSON* dan *XML*. Parameter utama yang dibutuhkan *request* yaitu kunci API, alamat lokasi. Data yang dikembalikan oleh *request* meliputi *status code* dari *request* dan *results* yang menyimpan informasi geografis dan geometri alamat dalam bentuk *array*.

2.6 Analisa

Pada tahap analisa, dilakukan survei untuk mengetahui berdasarkan apa pengguna mencari tempat makan, data apa saja yang ingin diketahui oleh pengguna dari tempat makan dan persepsi pengguna terhadap lima faktor penilaian dengan menggunakan skala *Likert*. Kelima faktor ini antara lain rasa, porsi, harga, kebersihan dan pelayanan. Sampel yang diambil adalah masyarakat kota Surabaya dan didapatkan responden untuk survei sebanyak 100 responden. Dari Gambar 1, diketahui bahwa pengguna ketika berencana makan diluar akan

mencari berdasarkan makanan dibandingkan mencari berdasarkan tempat makan. Sehingga *item* dalam sistem rekomendasi adalah menu atau makanan dan yang akan diprediksi adalah *rating* dari tempat makan.

Ketika anda berencana makan diluar, berdasarkan apa yang anda mencarinya?
100 responses



Gambar 1 Hasil Survei Pencarian Pengguna Ketika Akan Makan Diluar

Kemudian pada Gambar 2, diketahui persepsi pengguna terhadap rasa, porsi, rasa, kebersihan dan pelayanan. Hasil survei mengenai persepsi pengguna lalu diolah lagi dengan menggunakan perhitungan skala Likert agar data tersebut dapat digunakan dalam sistem rekomendasi.

Apa yang dipertimbangkan anda ketika memilih tempat makan?



Gambar 2 Hasil Survei Persepsi Pengguna Untuk Setiap Faktor

Pada Tabel I, diperlihatkan perhitungan persepsi pengguna dari hasil survei. Nilai pada baris “Penyelesaian Akhir (%)” akan digunakan sebagai konstanta beban dalam sistem rekomendasi. Untuk data yang ingin diketahui oleh pengguna dapat dilihat pada Gambar 3.

Tabel 1 Perhitungan Persepsi Pengguna

Persepsi	Rasa	Porsi	Harga	Kebersihan	Pelayanan
Sangat Penting (5)	220	45	115	310	135
Penting (4)	128	88	116	120	224
Biasa (3)	66	165	120	18	45
Tidak Penting (2)	2	26	14	0	0
Sangat Tidak Penting (1)	1	1	1	2	2
Total	417	325	366	450	406
Hasil Akhir (%)	83,4	65	73,2	90	81,2

Dari Gambar 3, diketahui bahwa pengguna ingin mengetahui jam buka, kondisi buka atau tutup, jarak antara tempat makan dengan tempat makan, jangkauan harga makanan, jenis

makanan, penilaian kebersihan, penilaian pelayanan karena responden yang menjawab bahwa mereka membutuhkan data-data tersebut berjumlah lebih dari 50% responden. Sedangkan untuk jawaban lainnya tidak digunakan karena jumlah responden kurang dari 50%.

Setelah melakukan analisa, dilakukan desain terhadap sistem rekomendasi dan juga aplikasi. Dalam mendesain sistem rekomendasi dilakukan dua proses pendesainan yaitu desain algoritma dan desain pengumpulan. Dikarenakan *rating* dalam aplikasi ada empat yaitu *rating* rasa, porsi, kebersihan dan pelayanan maka dilakukan modifikasi pada algoritma TyCo.



Gambar 3 Hasil Survei Data yang Ingin Diketahui Pengguna

3. HASIL DAN PEMBAHASAN

Hasil implementasi dari algoritma *Typicality Based Collaborative Filtering* dalam sistem rekomendasi aplikasi ini dijabarkan dalam langkah-langkah sebagai berikut.

1. Membuat kelompok menu (k_j) dari tempat makan yang akan diprediksi. Dalam aplikasi ini, jumlah kelompok menu yang akan dibentuk berjumlah lima. Jumlah tersebut didapatkan dari membagi *range* harga. Harga termurah dalam aplikasi sebesar 20 ribu rupiah dan harga termahal sebesar 120 ribu rupiah. Pembagian untuk kelompok menu yaitu kelompok pertama (≥ 20000 dan ≤ 40000), kelompok kedua (> 40000 dan ≤ 60000), kelompok ketiga (> 60000 dan ≤ 80000), kelompok keempat (> 80000 dan ≤ 100000), dan kelompok kelima (> 100000 dan ≤ 120000). Setelah dibentuk kelompok menu lalu dibuat profil menu yang merepresentasikan kelompok menu. profil yang dibuat dirumuskan dalam persamaan (3).

$$\vec{t}_{k_j} = (p_{k_jk}:r_{k_jk}, p_{k_jh}:r_{k_jh}, p_{k_jj}:r_{k_jj}) \quad (3)$$

\vec{t}_{k_j} merupakan vektor profil kelompok menu, k_j merupakan kelompok menu, p_{k_jk} merupakan properti kategori dari profil menu, p_{k_jh} merupakan properti harga dari profil menu, p_{k_jj} merupakan properti jarak dari profil menu, r_{k_jk} merupakan nilai properti kategori dari *prototype* tempat makan, r_{k_jh} merupakan nilai properti harga dari *prototype* tempat makan, r_{k_jj} merupakan nilai properti jarak dari *prototype* tempat makan.

2. Nilai dari r_{k_jk} didapatkan dari menghitung kemiripan antara kategori yang dimasukkan pengguna dengan kategori dari menu pada *database*. Kemiripan ini dihitung dengan menggunakan metode Jaccard.
3. Nilai dari r_{k_jh} didapatkan dari hasil pengurangan nilai harga menu pada *database* dengan nilai harga menu yang dimasukkan pengguna yang dibagi dengan nilai harga menu pada *database*. Sehingga didapatkan nilai antara nol sampai dengan satu.
4. Nilai dari r_{k_jj} didapatkan dari menghitung jarak antara posisi pengguna dengan lokasi tempat makan. Nilai jarak ini didapatkan dengan membandingkan jarak pengguna dengan tempat makan dengan kriteria perbandingan. Diketahui bahwa jarak antara Surabaya Barat dengan Surabaya Timur sebesar 24 km. Dan jarak antara Surabaya Utara dengan Surabaya

Selatan sebesar 25 km. Perbandingan jarak yang digunakan untuk menghitung nilai jarak dan membentuk kelompok menu yaitu jarak lebih dari nol km sampai dengan lima km bernilai satu. Jarak antara lebih dari lima km sampai dengan 10 km bernilai 0.8. Jarak antara lebih dari 10 km sampai dengan 15 km bernilai 0.6. Jarak antara lebih dari 15 km sampai dengan 20 km bernilai 0.4. Jarak antara lebih dari 20 km sampai dengan 25 km bernilai 0.2.

5. Menghitung nilai tipikalitas setiap menu dalam kelompok menu terhadap profil menu dari kelompok menu tersebut. Nilai tipikalitas tiap menu didapatkan dengan menggunakan persamaan (4).

$$w_{j,y} = Sim(\vec{t}_{k_j}, \vec{p}_{O_y}) \quad (4)$$

$w_{j,y}$ merupakan nilai tipikalitas menu (O_y) terhadap profil dari kelompok menu t_{k_j} , \vec{t}_{k_j} merupakan vektor profil menu dari kelompok menu, \vec{p}_{O_y} merupakan vektor properti menu (O_y).

6. Membuat kelompok pengguna (g_x) untuk setiap kelompok menu yang dibentuk lalu membuat profil pengguna yang merepresentasikan kelompok pengguna (g_x). Profil pengguna tersebut dirumuskan dengan persamaan (5).

$$\vec{t}_{g_x} = (p_{g_x,r}:1, p_{g_x,p}:1, p_{g_x,k}:1, p_{g_x,pl}:1, p_{g_x,f}:1) \quad (5)$$

\vec{t}_{g_x} merupakan vektor profil pengguna pada kelompok pengguna g_x , $p_{g_x,r}$ adalah properti *rate* rasa dari vektor profil pengguna yang diperoleh dari total jumlah *rating* yang telah diberikan kepada menu dalam kelompok menu g_x , $p_{g_x,p}$ adalah properti *rate* porsi dari vektor profil pengguna yang diperoleh dari total jumlah *rating* yang telah diberikan kepada menu di kelompok menu g_x , $p_{g_x,k}$ adalah properti *rate* kebersihan dari vektor profil pengguna yang diperoleh dari total jumlah *rating* yang telah diberikan kepada menu di kelompok menu g_x , $p_{g_x,pl}$ adalah properti *rate* pelayanan dari vektor profil pengguna yang diperoleh dari total jumlah *rating* yang telah diberikan kepada menu di kelompok menu g_x , $p_{g_x,f}$ merupakan properti frekuensi dari profil pengguna yang didapatkan dari jumlah menu yang sudah dinilai di kelompok menu g_x , 1 adalah nilai maksimum dari setiap properti dari vektor profil pengguna dengan range nilai dari nol sampai dengan satu.

7. Membuat vektor untuk merepresentasikan pengguna. Vektor ini dirumuskan dalam persamaan (6).

$$\vec{p}_{i,g_x} = (p_{g_x,r}:s_{g_x,r}^i, p_{g_x,p}:s_{g_x,p}^i, p_{g_x,k}:s_{g_x,k}^i, p_{g_x,pl}:s_{g_x,pl}^i, p_{g_x,f}:s_{g_x,f}^i) \quad (6)$$

\vec{p}_{i,g_x} adalah vektor *user property*, $s_{g_x,r}^i$ adalah derajat *user* U_i memiliki properti $p_{g_x,r}$, $s_{g_x,p}^i$ adalah derajat *user* U_i memiliki properti $p_{g_x,p}$, $s_{g_x,k}^i$ adalah derajat *user* U_i memiliki properti $p_{g_x,k}$, $s_{g_x,pl}^i$ adalah derajat *user* U_i memiliki properti $p_{g_x,pl}$, $s_{g_x,f}^i$ adalah derajat *user* U_i memiliki properti $p_{g_x,f}$.

8. Menghitung derajat dari $s_{g_x,r}^i$ dengan menggunakan persamaan (7).

$$s_{g_x,r}^i = \frac{\sum_{y=1}^n w_{x,y} \cdot R_{i,y}}{n \cdot R^{max}} \quad (7)$$

$R_{i,y}$ adalah nilai *rating* yang diberikan oleh pengguna U_i pada menu O_y , $w_{x,y}$ adalah derajat menu O_y pada kelompok menu g_x , n adalah jumlah menu yang sudah dinilai oleh U_i dalam g_x , R^{max} adalah nilai maksimal *rating*, dan persamaan (7) digunakan juga dalam menghitung derajat dari $s_{g_x,p}^i$.

9. Menghitung derajat dari $s_{g_x,k}^i$ dengan menggunakan persamaan (8).

$$s_{g_x,k}^i = \frac{\sum_{y=1}^n R_{i,y}}{n \cdot R^{max}} \quad (8)$$

$R_{i,y}$ adalah nilai *rating* yang diberikan oleh U_i pada menu O_y , n adalah jumlah menu yang

sudah dirate oleh U_i dalam g_x , R^{max} adalah nilai maksimal rating, dan persamaan (8) digunakan juga dalam menghitung derajat dari $s_{g_x,pl}^i$.

10. Menghitung derajat dari $s_{g_x,f}^i$ dengan menggunakan persamaan (9).

$$s_{g_x,f}^i = \frac{N_{x,i}}{N_i} \quad (9)$$

$N_{x,i}$ adalah jumlah menu yang sudah dinilai oleh pengguna U_i dalam kelompok menu g_x , N_i adalah jumlah menu yang ada dalam seluruh kelompok menu.

11. Menghitung nilai tipikalitas setiap pengguna terhadap profil pengguna. Nilai tipikalitas didapatkan dengan menggunakan persamaan (10).

$$T_{g_x}(U_i) = \frac{s_{g_x,r}^i + s_{g_x,p}^i + s_{g_x,k}^i + s_{g_x,pl}^i + s_{g_x,f}^i}{5} \quad (10)$$

$T_{g_x}(U_i)$ adalah nilai tipikalitas dari pengguna U_i dalam kelompok menu g_x , semakin tinggi nilai derajat menunjukkan semakin tingginya kemiripan antara vektor pengguna dengan profil kelompok pengguna.

12. Mencari tetangga (*neighbor*) dari pengguna aktif dengan menggunakan persamaan (11).

$$\vec{N}_i = \{U_j \mid Sim(U_i, U_j) \geq \gamma\} \quad (11)$$

\vec{N}_i adalah vektor yang menyimpan pengguna yang merupakan pengguna aktif, γ merupakan *threshold* yang digunakan untuk memilih pengguna U_j yang dapat menjadi tetangga dari pengguna U_i dan untuk aplikasi ini γ bernilai 0.5.

13. Memprediksi *rate* pengguna aktif terhadap setiap menu yang ada didalam kelompok menu. Prediksi *rate* dilakukan dengan menggunakan persamaan (12).

$$R(U_i, O_j) = \frac{\sum_{U_x \in \vec{N}_i} R(U_x, O_j) \cdot Sim(U_x, U_i)}{\sum_{U_x \in \vec{N}_i} Sim(U_x, U_i)} \quad (12)$$

$R(U_i, O_j)$ adalah nilai rating dari pengguna U_i pada menu O_j , U_x merupakan pengguna dalam daftar tetangga dari U_i , $R(U_x, O_j)$ merupakan *rating* dari pengguna U_x pada menu O_j , $Sim(U_x, U_i)$ merupakan kemiripan antara U_x dengan U_i , dan persamaan (12) digunakan untuk menghitung nilai total *rating* yang diberikan oleh seluruh tetangga dari U_i pada O_j .

14. Menghitung beban total dari setiap menu, beban ini dihitung dengan menggunakan persamaan (13)

$$b_{total} = (k_r \cdot R(U_i, O_j)_r) + (k_p \cdot R(U_i, O_j)_p) + (k_k \cdot R(U_i, O_j)_k) + (k_{pl} \cdot R(U_i, O_j)_{pl}) \quad (13)$$

b_{total} merupakan nilai beban total menu, k_r merupakan nilai konstanta dari rating rasa, k_p merupakan nilai konstanta dari rating porsi, k_k merupakan nilai konstanta dari rating kebersihan, k_{pl} merupakan nilai konstanta dari rating pelayanan.

15. Menjumlah total beban dari setiap menu pada tempat makan. Total beban dari tempat makan dihitung dengan persamaan (14)

$$B_x = \frac{\sum_1^{n_x} b_{total,x}}{n_x} \quad (14)$$

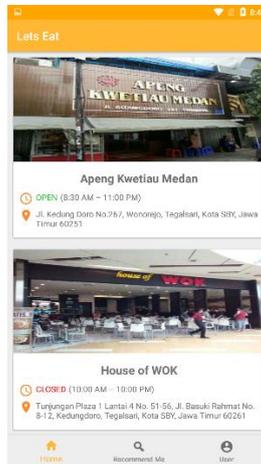
B_x adalah total beban tempat makan (B_x), $\sum_1^n b_{total,x}$ adalah jumlah total beban dari setiap menu yang sesuai dengan kategori yang dicari yang dimiliki tempat makan (B_x), n adalah jumlah menu yang sesuai dengan kategori yang dicari oleh pengguna aktif yang dimiliki tempat makan (B_x).

16. Mengurutkan tempat makan berdasarkan total beban dari terbesar ke terkecil.

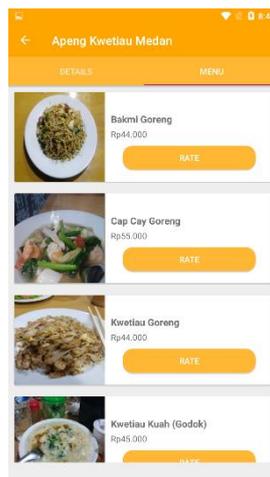
Desain pengumpulan dilakukan untuk memperjelas cara pengumpulan dan data apa saja yang perlu dikumpulkan. Dalam desain pengumpulan data yang perlu dikumpulkan berupa *rating* terhadap 10 tempat makan yang masing-masing memiliki 10 menu. Data tersebut dikumpulkan dengan menggunakan survei. Data yang telah dikumpulkan akan digunakan sebagai data latih sistem rekomendasi.

Hasil implemementasi aplikasi dapat dilihat pada Gambar 4 sampai dengan Gambar 5.

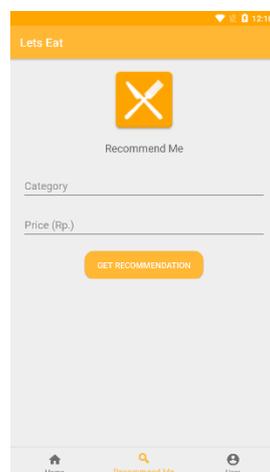
Tempat makan yang tersimpan dalam aplikasi ditampilkan dalam bentuk daftar seperti pada Gambar 4. Ketika pengguna memencet tombol navigasi “menu”, pengguna akan diperlihatkan daftar menu yang ada di tempat makan yang dipilih pengguna seperti pada Gambar 8. Dan ketika pengguna memilih “Recommend Me”, tampilan dari halaman cari rekomendasi akan ditampilkan seperti pada Gambar 10.



Gambar 4 Halaman Daftar Tempat Makan



Gambar 5 Halaman Daftar Menu



Gambar 6 Halaman Cari Rekomendasi

Setelah selesai mengimplementasikan seluruh desain, dilakukan pengujian. Pengujian dilakukan untuk mengetahui tingkat akurasi dari hasil sistem rekomendasi. Tingkat akurasi dari hasil sistem rekomendasi didapatkan dengan menghitung MAE. Semakin besar nilai MAE maka hasil prediksi *rating* dari sistem rekomendasi semakin tidak akurat. Dan sebaliknya semakin kecil nilai MAE maka hasil prediksi *rating* dari sistem rekomendasi semakin akurat. Dalam pengujian ini nilai MAE paling kecil adalah nol dan nilai terbesar MAE adalah lima. Karena memerlukan nilai *rating* pengguna untuk dapat dibandingkan dengan prediksi *rating* maka dilakukan survei.

Pada survei yang digunakan dalam pengujian terdapat dua buah skenario pertanyaan. Skenario pertama dibuat untuk menyaring responden. Penyaringan ini diperlukan untuk memastikan bahwa responden mengetahui mengenai tempat makan yang akan ditanyakan pada skenario kedua. Pada skenario kedua responden akan diminta untuk memberikan penilaian mengenai seberapa mungkin responden tersebut akan mendatangi setiap tempat makan pada skenario tertentu.

Dalam survei yang dilakukan terdapat empat buah pertanyaan yang ditanyakan untuk mengetahui demografi dari responden, delapan pertanyaan yang ditanyakan untuk menyaring responden dan lima buah pertanyaan untuk mengetahui *rating* pengguna.

Untuk mengetahui *rating* pengguna untuk setiap tempat makan maka diberikanlah lima jenis skenario, dimana satu skenario untuk sebuah pertanyaan. Kelima skenario ini antara lain

Skenario 1, “Jika anda ingin makan Nasi Goreng Seafood dengan kisaran harga 30 ribu rupiah dan saat ini anda berada di depan hotel Majapahit di jalan Tunjungan”.

Skenario 2, “Jika anda ingin makan Udang Goreng Mayonaise dengan kisaran harga 45 ribu rupiah dan saat ini anda berada di depan Monumen Kapal Selam di jalan Pemuda”.

Skenario 3, “Jika anda ingin makan Ayam Goreng dengan kisaran harga 50 ribu rupiah dan saat ini anda berada di depan mall Ciputra World di jalan Mayjen Sungkono”.

Skenario 4, “Jika anda ingin makan Sapi Lada Hitam dengan kisaran harga 65 ribu rupiah dan saat ini anda berada di depan Pasar Atom di jalan Bunguran”.

Skenario 5, “Jika anda ingin makan Cap Cay dengan kisaran harga 40 ribu rupiah dan saat ini anda berada di depan Galaxy Mall di jalan Dharmahasada Indah Timur”.

Apabila responden pada skenario pertanyaan pertama berhasil menjawab dengan benar minimum empat pertanyaan maka jawaban responden tersebut dapat digunakan untuk skenario kedua. Dari Hasil Survei didapatkan 10 responden tidak lolos skenario pertama sehingga dari total 30 responden hanya digunakan jawaban dari 20 responden. Tabel II merupakan hasil survei dari skenario pertama. Tabel III merupakan hasil survei dari skenario kedua. Tabel IV merupakan hasil survei dari skenario ketiga. Tabel V merupakan hasil survei dari skenario keempat. Tabel VI merupakan hasil survei dari skenario kelima. Dan Tabel VII merupakan tabel yang berisi rata-rata MAE dari setiap tempat makan pada lima skenario.

Tabel 2 Hasil Survei Skenario Ke-1

Tempat Makan	Rata-rata Nilai	Prediksi Rating	MAE
Qua-li Noodle & Rice	3.65	3.52	0.13
Liu Fu	3.95	3.55	0.4
Moi Garden Hakka Chinese Restaurant	3.4	0	3.4
Ta Wan	4.35	3.6	0.75
Apeng Kwetiau Medan	1	0	1
LAOPAN	1	0	1
Jade Imperial	2.85	3.57	0.72
Tokyo Resto	1	0	1

House of Wok	1	0	1
Porong Wei	1	0	1

Tabel 3 Hasil Survei Skenario Ke-2

Tempat Makan	Rata-rata Nilai	Prediksi Rating	MAE
Qua-li Noodle & Rice	3.65	3.61	0.04
Liu Fu	2.7	3.54	0.84
Moi Garden Hakka Chinese Restaurant	2.9	3.57	0.67
Ta Wan	1	0	1
Apeng Kwetiau Medan	1	0	1
LAOPAN	1	0	1
Jade Imperial	2.75	0	2.75
Tokyo Resto	2.9	3.55	0.65
House of Wok	3.75	0	3.75
Porong Wei	2.9	3.44	0.54

Tabel 4 Hasil Survei Skenario Ke-3

Tempat Makan	Rata-rata Nilai	Prediksi Rating	MAE
Qua-li Noodle & Rice	1	0	1
Tempat Makan	Rata-rata Nilai	Prediksi Rating	MAE
Liu Fu	1	0	1
Moi Garden Hakka Chinese Restaurant	2.6	0	2.6
Ta Wan	1	0	1
Apeng Kwetiau Medan	1	0	1
LAOPAN	1	0	1
Jade Imperial	3.45	3.62	0.17
Tokyo Resto	1	0	1
House of Wok	4.1	3.49	0.61
Porong Wei	1.05	0	1.05

Tabel 5 Hasil Survei Skenario Ke-4

Tempat Makan	Rata-rata Nilai	Prediksi Rating	MAE
Qua-li Noodle & Rice	1	0	1
Liu Fu	3.4	3.5	0.1
Moi Garden Hakka Chinese Restaurant	3.6	3.62	0.02
Ta Wan	4.2	0	4.2
Apeng Kwetiau Medan	1	0	1
LAOPAN	1	0	1

Jade Imperial	1	0	1
Tokyo Resto	3.35	0	3.35
House of Wok	4.05	0	4.05
Porong Wei	2.9	3.4	0.5

Tabel 6 Hasil Survei Skenario Ke-5

Tempat Makan	Rata-rata Nilai	Prediksi Rating	MAE
Qua-li Noodle & Rice	3.05	0	3.05
Liu Fu	4.05	0	4.05
Moi Garden Hakka Chinese Restaurant	3.25	3.52	0.27
Ta Wan	1	0	1
Apeng Kwetiau Medan	3.15	3.52	0.37
LAOPAN	4.8	0	4.8
Jade Imperial	1	0	1
Tokyo Resto	3.7	0	3.7
House of Wok	3.55	3.55	0
Porong Wei	2.7	3.47	0.77

Dari Tabel VII diketahui bahwa rata-rata MAE dari hasil rekomendasi tempat makan pada aplikasi ini bernilai 1.366. Rata-rata MAE yang bernilai lebih dari satu ini disebabkan karena *neighbor* atau pengguna-pengguna lain yang memiliki pola kesukaan yang mirip dengan pengguna aktif tidak memberikan *rating* kepada menu dalam skenario. Tidak diberikannya *rating* tersebut mempengaruhi hasil prediksi *rating* dari sistem rekomendasi yang menghasilkan nilai nol untuk prediksi *rating*. Hal tersebut diperkirakan karena kurangnya data latihan yang tersedia.

Tabel 7 Hasil Rata-Rata MAE Setiap Tempat Makan

Tempat Makan	Rata-rata MAE per Skenario
Qua-li Noodle & Rice	1.044
Liu Fu	1.278
Moi Garden Hakka Chinese Restaurant	1.392
Ta Wan	1.59
Apeng Kwetiau Medan	0.874
LAOPAN	1.76
Jade Imperial	1.128
Tokyo Resto	1.94
House of Wok	1.882
Porong Wei	0.772
Rata-Rata Total MAE	1.366

4. KESIMPULAN

Berdasarkan desain, implementasi dan pengujian dapat disimpulkan bahwa tugas akhir ini dapat merancang dan membangun aplikasi rekomendasi tempat makan dengan menggunakan algoritma *typicality based collaborative filtering* dengan nilai rata-rata MAE sebesar 1.366 yang dikarenakan kurangnya data *training* atau data latih.

5. SARAN

Saran untuk aplikasi ini adalah

1. penambahan data *training* untuk memperbaiki tingkat akurasi dari prediksi *rating* sistem rekomendasi.
2. Optimasi sistem rekomendasi agar dapat memberikan rekomendasi lebih cepat.
3. Mengubah faktor-faktor yang digunakan dalam sistem rekomendasi.

DAFTAR PUSTAKA

- [1] D. Pratama and S. Hansun, "Aplikasi Rekomendasi Tempat Makan Menggunakan Algoritma Slope One pada Platform Android," *IJCCS*, pp. 11-20, 2017.
- [2] V. Wirawan, S. Hansun and H. T. Nugroho, "Implementasi Algoritma Squeezer dan Term," *ULTIMA Computing*, pp. 25-32, 2014.
- [3] K. A. Wijaya, "PENERAPAN METODE COLLABORATIVE FILTERING PADA SISTEM REKOMENDASI TEMPAT MAKAN DI DAERAH KABUPATEN BULELENG," *UAJY*, 2017.
- [4] C. Yi, L. Ho-fung, L. Qing, M. Huaqing, T. Jie and L. Juanzi, "Typicality-Based Collaborative Filtering," *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, pp. 1-13, 2014.
- [5] P. B. Thorat, R. M. Goudar and S. Barve, "Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System," *International Journal of Computer Applications*, pp. 31-36, 2015.
- [6] N. P. Kumar and f. zhenzhen, "Hybrid User-Item Based Collaborative Filtering," *19th International Conference on Knowledge Based and Intelligent Information and Engineering*, vol. 60, pp. 1453-1461, 2015.
- [7] W. Wang and Y. Lu, "Analysis of the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) in Assessing Rounding Model," *ICMEMSCE*, pp. 1-10, 2018.
- [8] "Places SDK for Android," 13 November 2018. [Online]. Available: <https://developers.google.com/places/android-sdk/intro>.
- [9] "Distance Matrix API," 26 November 2018. [Online]. Available: <https://developers.google.com/maps/documentation/distance-matrix/intro>.