

Pengembangan Sistem Rekomendasi Lokasi Bengkel Berdasarkan Penilaian Pengguna Menggunakan Integrasi Google Maps API

Risal^{1*}, Ricard Aprian Pamean¹, Rismayani², Angdy Erna²

¹Sistem Informasi, Universitas Dipa Makassar, Makassar, Indonesia.

²Universitas Dipa Makassar, Makassar, Indonesia.

e-mail: ^{*}risallrusli@gmail.com, ²aprianpameanricard@gmail.com, ³rismayani@undipa.ac.id, ⁴angdy@undipa.ac.id

Abstrak

Permasalahan yang sering dialami pengendara adalah kesulitan menemukan bengkel ketika kendaraan mengalami kerusakan mendadak serta kebingungan dalam memilih bengkel yang sesuai dengan kebutuhan karena banyaknya pilihan pada Google Maps. Penelitian ini bertujuan mengembangkan sistem rekomendasi lokasi bengkel berbasis web untuk memberikan rekomendasi yang sesuai dengan preferensi pengguna. Metode yang digunakan adalah Collaborative Filtering dengan algoritma Cosine Similarity untuk menghitung tingkat kesamaan antar pengguna, sedangkan proses prediksi rating dilakukan menggunakan metode Weighted Sum berdasarkan bobot kesamaan tersebut. Dataset yang digunakan dalam penelitian ini diperoleh melalui Google Maps API dengan total 204 bengkel. Pengujian menggunakan MAE dilakukan pada tiga skenario dengan jumlah pengguna berbeda untuk mengevaluasi akurasi sistem serta mengukur selisih antara rating sebenarnya dan prediksi sistem, menghasilkan nilai 0,759 untuk 5 pengguna, 0,715 untuk 10 pengguna, dan 0,710 untuk 20 pengguna. Hasil menunjukkan bahwa metode yang diterapkan pada sistem ini memberikan rekomendasi dengan tingkat akurasi yang cukup baik.

Kata kunci: Sistem Rekomendasi, Collaborative Filtering, Cosine Similarity, Weighted Sum, MAE.

I. PENDAHULUAN

Pemilihan bengkel terdekat khususnya di Kota Makassar merupakan permasalahan yang penting, mengingat kendaraan bermotor menjadi sarana transportasi utama yang paling banyak digunakan masyarakat untuk menunjang aktivitas sehari-hari maupun perjalanan jarak jauh. Berdasarkan data Badan Pusat Statistik Sulawesi Selatan, jumlah kendaraan bermotor di Kota Makassar pada tahun 2025 mencapai 2.091.501 unit. Kondisi ini menunjukkan aktivitas transportasi yang sangat tinggi, namun juga berpotensi menimbulkan berbagai permasalahan teknis pada kendaraan.

Seiring meningkatnya jumlah pengguna kendaraan, kebutuhan layanan perawatan dan perbaikan, khususnya bengkel yang cepat dan mudah diakses, juga semakin meningkat [1]. Kendala yang sering dialami pengendara adalah kesulitan menemukan bengkel terdekat, terutama pada kondisi darurat seperti kerusakan mesin atau ban bocor dan kempes di tengah perjalanan [2].

Pencarian lokasi bengkel di Kota Makassar masih sering menjadi kendala bagi pengguna kendaraan. Banyaknya bengkel tidak sebanding dengan ketersediaan informasi lokasinya [2]. Pemanfaatan Google Maps API dapat membantu pengguna menemukan lokasi bengkel secara langsung. Namun, banyaknya bengkel yang terdaftar di Google Maps masih sering membingungkan pengguna dalam menentukan pilihan.

Informasi yang kurang lengkap membuat proses perbandingan memakan waktu, yang pada akhirnya menimbulkan ketidaknyamanan, terutama dalam situasi darurat yang memerlukan perbaikan cepat [1].

Sistem rekomendasi dapat membantu para pengguna dalam memberikan saran rekomendasi item dari banyaknya pilihan yang tersedia [3]. Sistem rekomendasi berfungsi menyediakan dan merekomendasikan suatu item untuk membantu pengguna mengambil keputusan [4]. Penerapannya dilakukan dengan memprediksi item yang akan direkomendasikan kepada pengguna berdasarkan karakteristik item atau penilaian (rating), baik yang berasal dari pengguna itu sendiri maupun dari pengguna lain [5].

Dalam pengembangan sistem rekomendasi, terdapat dua metode yang umum digunakan, yaitu Content Based Filtering dan Collaborative Filtering [3]. Content Based Filtering memberikan rekomendasi berdasarkan kesamaan item sesuai karakteristik pengguna, sedangkan Collaborative Filtering memanfaatkan riwayat penilaian sebagai dasar rekomendasi [6]. Namun, pengembangan sistem rekomendasi sering terkendala oleh kurangnya informasi pada data, seperti sparsity dan cold start problem [3], [7].

Metode Collaborative Filtering terbagi menjadi dua pendekatan, yaitu User-Based dan Item-Based [8]. Pendekatan Item-Based dinilai lebih unggul karena mampu menghasilkan prediksi yang lebih baik dan rekomendasi yang lebih spesifik

dengan mencocokkan item yang telah dinilai oleh berbagai pengguna [9].

Beberapa penelitian telah menerapkan Item-Based Collaborative Filtering, antara lain pada sistem rekomendasi rumah makan menggunakan Adjusted Cosine Similarity yang menghasilkan nilai MAE terendah sebesar 0,623 [10], serta pada sistem rekomendasi destinasi wisata di Yogyakarta menggunakan Cosine Similarity dengan 1.069 destinasi yang memperoleh nilai MAE sebesar 2,537 [11].

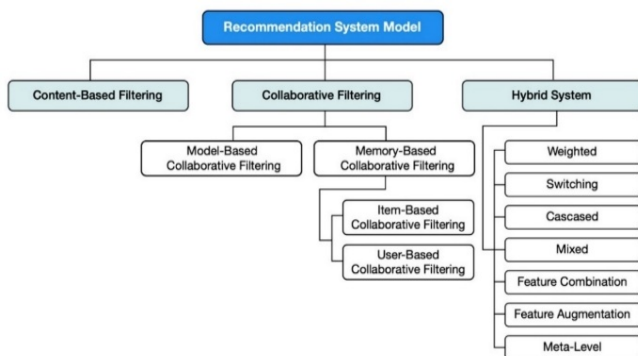
Berdasarkan latar belakang tersebut, penelitian ini mengimplementasikan metode Collaborative Filtering menggunakan algoritma Cosine Similarity dalam sistem rekomendasi lokasi bengkel berbasis web. Sistem ini diharapkan mampu memberikan rekomendasi sesuai kebutuhan pengguna sekaligus mempermudah pencarian bengkel terdekat melalui integrasi Google Maps API.

II. METODOLOGI PENELITIAN

Dalam penelitian ini, data diperoleh melalui *Google Maps API* dengan mengumpulkan informasi lokasi bengkel secara otomatis dan terstruktur. Pengambilan data dilakukan dengan menggunakan parameter wilayah Kota Makassar, sehingga bengkel yang terdaftar di dalam cakupan wilayah tersebut dapat terdeteksi dan dimasukkan ke dalam sistem. Data yang diperoleh berjumlah 204 bengkel. Atribut utama yang digunakan antara lain *title* (nama bengkel), *latitude*, *longitude*, *address*, *phone*, *operating_hours*, *thumbnail* (gambar), serta *type* untuk mengidentifikasi kategori bengkel. Data tersebut selanjutnya digunakan sebagai *dataset* utama dalam pengembangan sistem rekomendasi.

A. Sistem Rekomendasi

Sistem rekomendasi merupakan sistem atau aplikasi yang dibuat untuk dapat menyediakan dan memberikan rekomendasi dari suatu *item* untuk membuat suatu keputusan yang diinginkan oleh pengguna sistem [12].



Gambar 1. Model Sistem Rekomendasi

Sistem rekomendasi dapat dikatakan sebagai sebuah perangkat lunak yang memberikan prediksi berupa pilihan *item* kepada pengguna sesuai dengan karakteristik dan preferensi pengguna dengan tujuan sebagai sistem pendukung keputusan untuk pengguna dapat menyukai suatu *item* [5]. Dalam

pengembangan sistem rekomendasi, terdapat tiga pendekatan utama yang umum digunakan, yaitu *Content-Based Filtering*, *Collaborative Filtering*, dan *Hybrid Filtering* [5], [13].

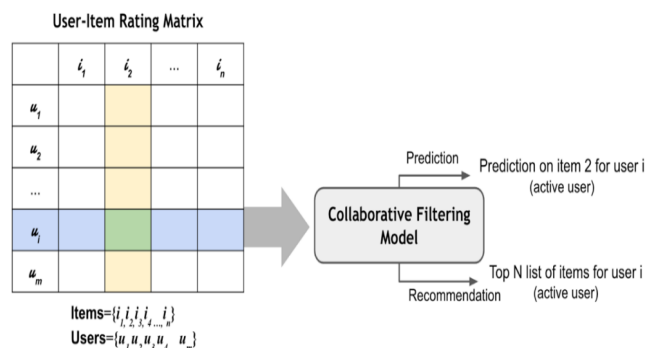
B. Collaborative Filtering

Collaborative Filtering merupakan salah satu teknik dalam sistem rekomendasi yang membangun basis data preferensi pengguna berdasarkan data penilaian (*rating*) pengguna untuk memprediksi *item* yang sesuai dengan selera pengguna, kemudian digunakan sebagai dasar rekomendasi [13]. Metode *Collaborative Filtering* memberikan rekomendasi dengan memanfaatkan pendapat dari pengguna lain yang memiliki pola atau kebiasaan serupa dengan pengguna [14]. Pendapat tersebut dapat diperoleh baik secara *eksplisit* maupun *implisit* [7]. Namun demikian, *Collaborative Filtering* memiliki beberapa kelemahan, seperti *cold start problem* dan *data sparsity* [3], [7].

Collaborative Filtering dapat dibedakan ke dalam dua pendekatan utama [8], [13], yaitu:

1. *User-Based Collaborative Filtering*, sistem menghasilkan rekomendasi dengan mengidentifikasi pengguna lain yang memiliki pola preferensi serupa. Proses ini dilakukan dengan menghitung tingkat kemiripan antar pengguna untuk menentukan *user* terdekat (*user neighbors*), kemudian nilai *rating* dari *user* tersebut digunakan sebagai dasar rekomendasi bagi *user* target.
2. *Item-Based Collaborative Filtering*, sistem menghasilkan rekomendasi dengan mengukur kemiripan antar *item* berdasarkan *rating* pengguna. Berbeda dengan pendekatan *user-based* yang menekankan kemiripan antar pengguna, metode ini fokus pada hubungan antar *item* yang telah diberi *rating*, kemudian memilih *item* yang paling mirip untuk diprediksi dan direkomendasikan kepada pengguna.

Secara umum, proses *Collaborative Filtering* terdiri dari dua tahap utama, yaitu *prediction* untuk memperkirakan *rating item* yang berpotensi disukai *user* aktif, dan rekomendasi untuk menghasilkan daftar *Top-N Recommendation item* paling diminati [15], [13].



Gambar 2. Proses Prediksi dan Rekomendasi *Collaborative Filtering*

Proses *Collaborative Filtering* dalam menghasilkan rekomendasi mencakup perhitungan tingkat *similarity* antar

item serta prediksi *rating* item yang belum dinilai oleh *user*, dengan pengukuran kemiripan menggunakan algoritma seperti *Cosine Similarity*, *Pearson Correlation Coefficient*, dan *Adjusted Cosine Similarity* [14].

C. Cosine Similarity

Cosine Similarity merupakan salah satu metode yang dimanfaatkan untuk menilai tingkat kemiripan antara dua *user* atau dua *item*, dengan memperhatikan pola pemberian *rating* [14]. Metode ini bekerja dengan menganalisis kemiripan antar *item* yang telah diberi *rating* oleh *user*, sehingga sistem mampu menyarankan *item* lain yang serupa [16], [17].

$$\text{sim}(k, l) = \frac{\sum_{u=1}^m R_{u,k} \cdot R_{u,l}}{\sqrt{\sum_{u=1}^m R_{u,k}^2} \cdot \sqrt{\sum_{u=1}^m R_{u,l}^2}} \quad (1)$$

Keterangan:

$\text{sim}(k, l)$: nilai *similarity* antara *item-k* dan *item-l*.

m : jumlah *user* yang memberikan *rating* pada *item-k* dan *item-l*.

$R_{u,k}$: *rating* yang diberikan oleh *user u* pada *item k*.

$R_{u,l}$: *rating* yang diberikan oleh *user u* pada *item l*.

D. Weighted Sum

Weighted Sum merupakan salah satu metode yang digunakan dalam *collaborative filtering* untuk menghitung nilai prediksi *rating* terhadap *item* yang akan direkomendasikan [16]. Metode ini melakukan perhitungan prediksi dengan membandingkan *rating* yang sebelumnya diberikan oleh *user* terhadap *item* tertentu dengan nilai kemiripan antara *item* tersebut dan *item* lainnya [18].

$$P(u, j) = \frac{\sum_{i \in j} r_{u,i} * \text{sim}(i, j)}{\sum_{i \in j} \text{sim}(i, j)} \quad (2)$$

Keterangan:

$P(u, j)$: prediksi untuk *user u* pada *item j*.

$\sum_{i \in j}$: himpunan *item* yang mirip dengan *item j*.

$r_{u,i}$: *rating user* u pada *item i*.

$\text{sim}(i, j)$: nilai *similarity* antara *item i* dan *item j*.

E. Normalisasi Multi Kriteria Rekomendasi

Sistem rekomendasi multi-kriteria memerlukan proses normalisasi untuk mengatasi perbedaan rentang penilaian antar kriteria agar tidak terjadi dominasi kriteria [6]. Pada penelitian ini metode *Min-Max* digunakan untuk mentransformasi data ke rentang 0–1 guna menyeragamkan skala antar kriteria [18].

$$x' = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (3)$$

Di mana x_i merupakan nilai asli, x' adalah hasil normalisasi, $\min(x)$ nilai terkecil, dan $\max(x)$ nilai terbesar. Normalisasi diterapkan pada jarak bengkel menggunakan *Min-*

Max terbalik dan *rating* hasil prediksi *Collaborative Filtering* menggunakan *Min-Max* standar. Nilai hasil normalisasi selanjutnya digunakan dalam perhitungan rekomendasi *Top-N Recommendation* berdasarkan skor gabungan jarak dan *rating*.

F. Mean Absolute Error (MAE)

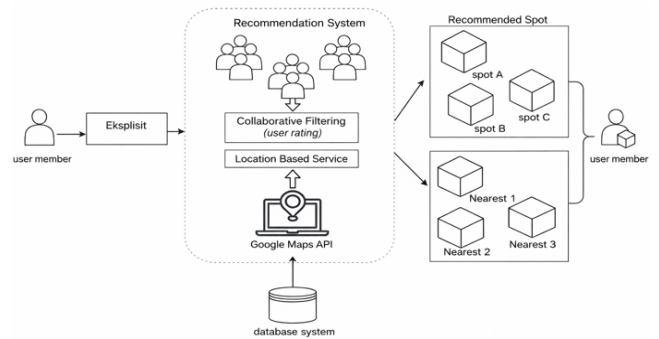
Mean Absolute Error (MAE) merupakan salah satu metode yang sering digunakan dalam pengujian sistem rekomendasi [4]. *MAE* digunakan untuk menghitung selisih antara *rating* prediksi dan *rating* sebenarnya dari *user* [19]. Nilai *MAE* yang lebih kecil menunjukkan bahwa prediksi sistem rekomendasi semakin akurat dan kualitas sistem semakin baik [8].

$$MAE = \frac{\sum_{u=1}^N |P_{u,i} - R_{u,i}|}{N} \quad (4)$$

Dimana n merepresentasikan jumlah keseluruhan *item* dalam data rekomendasi milik pengguna aktif yang memiliki nilai prediksi. Simbol $P_{u,i}$ Menunjukkan nilai *rating* hasil perhitungan prediksi yang diberikan oleh pengguna u terhadap *item i*, sedangkan $R_{u,i}$ mengacu pada nilai *rating* sebenarnya yang telah diberikan oleh pengguna u untuk *item* tersebut.

III. HASIL DAN PEMBAHASAN

A. Perancangan Solusi



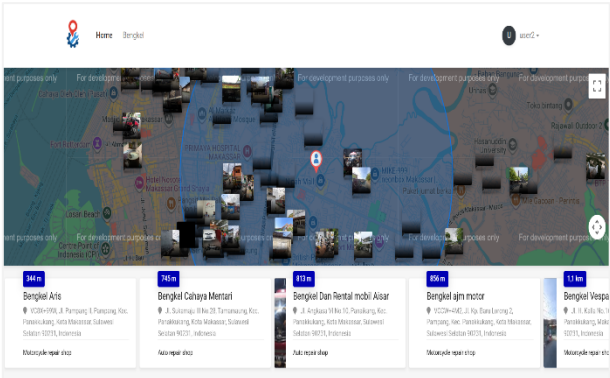
Gambar 3. Perancangan Solusi

Gambar 3 perancangan solusi, menggambarkan alur kerja dari sistem rekomendasi lokasi bengkel ini dimulai dari memasukkan penilaian secara *eksplisit* terhadap salah satu bengkel, seperti pemberian *rating* langsung oleh *user*. Kemudian data tersebut menjadi masukan ke dalam sistem rekomendasi yang terbagi menjadi dua modul, pertama yaitu *Collaborative Filtering* yang menyaring bengkel-bengkel berdasarkan kemiripan *item rating* (*rating* yang diberikan *user* terhadap bengkel), kedua yaitu *Location Based Service* yang dibantu dengan *Google Maps API* untuk menyaring informasi berdasarkan jarak terdekat antara posisi *user* dengan bengkel yang direkomendasikan. Output sistem berupa daftar bengkel hasil penyaringan *Collaborative Filtering* dan *Location Based Service* yang merekomendasikan bengkel sesuai dengan preferensi penilaian dan jarak terdekat dari *user*.

B. Implementasi Sistem

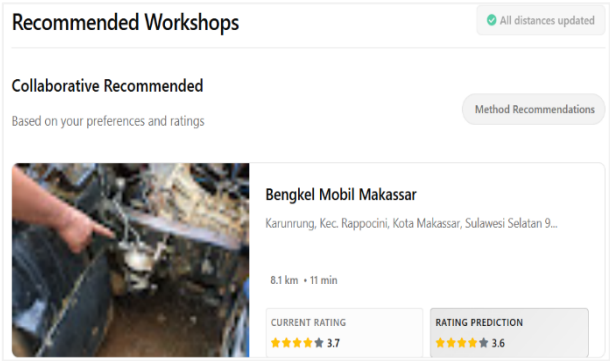
Pada tahapan ini merupakan penerapan sistem yang telah dirancang dan dikembangkan dengan menggunakan bahasa pemrograman *PHP* dan *Framework Laravel*. Berikut hasil implementasi sistem rekomendasi yang telah dikembangkan:

- 1) Tampilan *Home* Sistem Rekomendasi Lokasi Bengkel



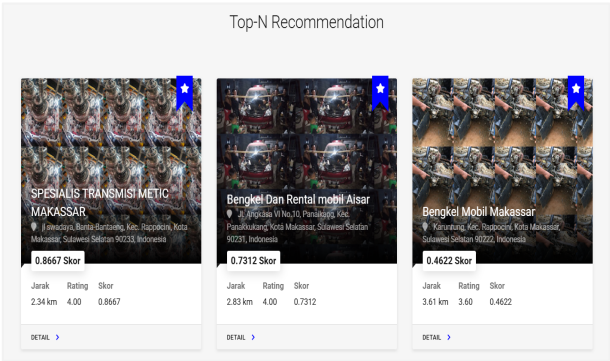
Gambar 4. Tampilan *Home* Sistem Rekomendasi Lokasi Bengkel

- 2) Tampilan Prediksi *Rating Collaborative Filtering*



Gambar 5. Tampilan rekomendasi berdasarkan penilaian pengguna

- 3) Tampilan Hasil *Top-N Recommendation*

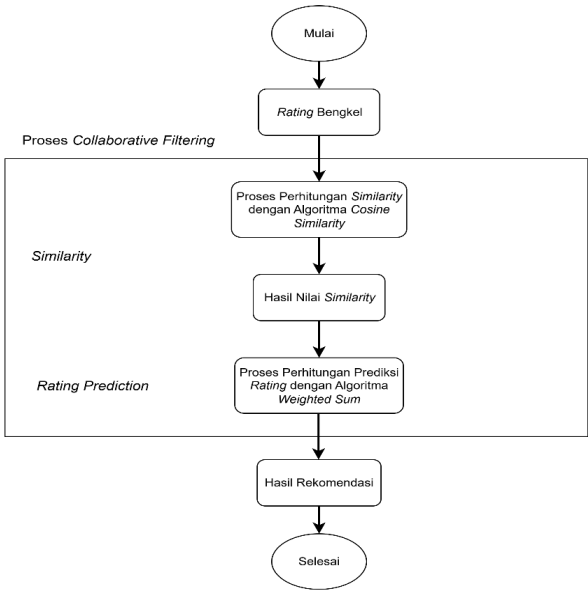


Gambar 6. Tampilan Hasil *Top-N Recommendation*

C. Perhitungan Metode *Collaborative Filtering*

Pada penelitian ini, metode *Collaborative Filtering* dengan pendekatan *item-based collaborative filtering* digunakan untuk merekomendasikan bengkel berdasarkan *rating* pengguna. Data yang digunakan berupa *rating* bengkel yang telah

diberikan oleh pengguna. Tahapan proses tersebut ditunjukkan pada Gambar 7.



Gambar 7. Alur Proses Perhitungan Metode *Collaborative Filtering*

Proses *similarity* merupakan tahapan untuk menghitung tingkat kemiripan antar bengkel. Dalam penelitian ini digunakan pendekatan *item-based collaborative filtering* dengan mengukur kemiripan antara *item* yang sama-sama diberi *rating* oleh pengguna yang sama, bertujuan untuk mengetahui tingkat kemiripan antar bengkel.

Sebelum melakukan perhitungan *similarity*, langkah awal yang dilakukan adalah menentukan nilai *rating* yang diberikan oleh *user*. Nilai *rating* ini digunakan sebagai acuan dalam proses perhitungan *Cosine Similarity*.

Tabel 1. Data *Rating Bengkel*

User / Bengkel	Bengkel 1	Bengkel 2	Bengkel 3	Bengkel 4	Bengkel 5
user1	5	4	3	4	0
user2	4	0	2	4	5
user3	5	5	3	0	5
user4	0	4	2	3	4
user5	5	0	3	4	5

Keterangan:

Bengkel 1: Bengkel dan Rental mobil Aisar

Bengkel 2: Bengkel Mobil Makassar

Bengkel 3: Bengkel Mahaputra

Bengkel 4: Spesialis Transmisi Metic Makassar

Bengkel 5: Bengkel Grand Nurfadillah (spesialis mobil)

Langkah selanjutnya adalah menghitung nilai *similarity* dari data *rating* pada tabel 1 menggunakan algoritma *Cosine Similarity*. Sebagai contoh, dilakukan perhitungan dua skenario tingkat kemiripan, yaitu antara bengkel 1 dan bengkel 2, serta antara bengkel 1 dan bengkel 3.

- 1) Perhitungan *Similarity* bengkel 1 dan 2:

$$\begin{aligned} \text{sim}(B1, B2) &= \frac{(5 \times 4) + (4 \times 0) + (5 \times 5) + (0 \times 4) + (5 \times 0)}{\sqrt{5^2 + 4^2 + 5^2 + 0^2 + 5^2} \times \sqrt{4^2 + 0^2 + 5^2 + 4^2 + 0^2}} \\ &= \frac{20 + 0 + 25 + 0 + 0}{\sqrt{91} \times \sqrt{57}} = \frac{45}{9.539 \times 7.550} = \frac{45}{72.021} = 0.625 \end{aligned}$$

2) Perhitungan *Similarity* bengkel 1 dan 3:

$$\begin{aligned} \text{sim}(B1, B3) &= \frac{(5 \times 3) + (4 \times 2) + (5 \times 3) + (0 \times 2) + (5 \times 3)}{\sqrt{5^2 + 4^2 + 5^2 + 0^2 + 5^2} \times \sqrt{3^2 + 2^2 + 3^2 + 2^2 + 3^2}} \\ &= \frac{15 + 8 + 15 + 0 + 15}{\sqrt{91} \times \sqrt{35}} = \frac{53}{9.539 \times 5.916} = \frac{53}{56.436} = 0.939 \end{aligned}$$

Hasil lengkap perhitungan *similarity* antar bengkel ditunjukkan pada tabel 2 berikut:

Tabel 2. Hasil Perhitungan *Similarity* antar Bengkel

Bengkel	Bengkel 1	Bengkel 2	Bengkel 3	Bengkel 4	Bengkel 5
Bengkel 1	1.000	0.625	0.939	0.778	0.769
Bengkel 2	0.625	1.000	0.784	0.491	0.569
Bengkel 3	0.939	0.784	1.000	0.851	0.851
Bengkel 4	0.778	0.491	0.851	1.000	0.722
Bengkel 5	0.769	0.569	0.851	0.722	1.000

Dari tabel 2 di atas, terlihat bahwa nilai *similarity* tertinggi terdapat pada bengkel 1 dan bengkel 3 sebesar 0.939, diikuti oleh bengkel 3 dengan bengkel 4 dan bengkel 5 sebesar 0.851. Hal ini menunjukkan adanya kesamaan pola penilaian dari pengguna pada bengkel-bengkel tersebut. Sebaliknya, *similarity* terendah diperoleh pada bengkel 2 dan bengkel 4 sebesar 0.491, yang menandakan perbedaan preferensi pengguna terhadap kedua bengkel tersebut. Nilai *similarity* 1.000 artinya bengkel dibandingkan dengan dirinya sendiri, sehingga kemiripannya sempurna.

Hasil lengkap perhitungan *similarity* antar bengkel ditampilkan pada Tabel 3, yang menunjukkan tingkat kemiripan antar bengkel berdasarkan nilai *similarity*.

Tabel 3. Hasil *Similarity* antar Bengkel

<i>Similarity</i> antar Bengkel	Nilai <i>Similarity</i>
<i>sim</i> (Bengkel dan Rental mobil Aisar, Bengkel Mobil Makassar)	0.625
<i>sim</i> (Bengkel dan Rental mobil Aisar, Bengkel Mahaputra)	0.939
<i>sim</i> (Bengkel dan Rental mobil Aisar, Spesialis Transmisi Metic Makassar)	0.778
<i>sim</i> (Bengkel dan Rental mobil Aisar, Bengkel Grand Nurfadillah (spesialis mobil))	0.769
<i>sim</i> (Bengkel Mobil Makassar, Bengkel Mahaputra)	0.784
<i>sim</i> (Bengkel Mobil Makassar, Spesialis Transmisi Metic Makassar)	0.491
<i>sim</i> (Bengkel Mobil Makassar, Bengkel Grand Nurfadillah (spesialis mobil))	0.569
<i>sim</i> (Bengkel Mahaputra, Spesialis Transmisi Metic Makassar)	0.851
<i>sim</i> (Bengkel Mahaputra, Bengkel Grand Nurfadillah (spesialis mobil))	0.851

<i>Similarity</i> antar Bengkel	Nilai <i>Similarity</i>
<i>sim</i> (Spesialis Transmisi Metic Makassar, Bengkel Grand Nurfadillah (spesialis mobil))	0.722

Setelah diperoleh nilai kemiripan antar bengkel, langkah selanjutnya adalah menghitung nilai prediksi terhadap bengkel yang akan direkomendasikan. Proses ini dilakukan dengan menggunakan perhitungan berdasarkan algoritma *Weighted Sum*.

Pada skenario ini, dilakukan perhitungan nilai prediksi pada bengkel yang belum diberi *rating* oleh pengguna. Nilai prediksi tersebut digunakan sebagai dasar dalam menentukan rekomendasi. Sebagai contoh, pada tabel 1 *rating* bengkel terdapat dua *user*, yaitu *user 2* dan *user 5*, yang belum memberikan *rating* pada bengkel 2. Proses perhitungan prediksi dilakukan menggunakan algoritma *Weighted Sum* sebagai berikut:

1) Prediksi *user 2*, bengkel 2:

$$\begin{aligned} P(\text{user2}, B2) &= \frac{(4 \times 0.625) + (2 \times 0.784) + (4 \times 0.491) + (5 \times 0.569)}{0.625 + 0.784 + 0.491 + 0.569} \\ &= \frac{8.877}{2.469} = 3.60 \end{aligned}$$

2) Prediksi *user 5*, bengkel 2:

$$\begin{aligned} P(\text{user5}, B2) &= \frac{(5 \times 0.625) + (3 \times 0.784) + (4 \times 0.491) + (5 \times 0.569)}{0.625 + 0.784 + 0.491 + 0.569} \\ &= \frac{10.286}{2.469} = 4.17 \end{aligned}$$

Berdasarkan hasil perhitungan, prediksi *rating* untuk *user 2* pada Bengkel 2 adalah 3.60, sedangkan untuk *user 5* adalah 4.17. Hal ini menunjukkan bahwa kemungkinan besar *user 5* akan memberikan *rating* lebih tinggi terhadap bengkel 2 dibandingkan dengan *user 2*. Dengan demikian, nilai *rating* yang sebelumnya kosong kini telah terisi berdasarkan hasil prediksi tersebut. Berdasarkan hasil perhitungan *weighted sum*, maka dapat dihasilkan prediksi *rating* sebagai berikut:

Tabel 4. Hasil Prediksi *Rating*

User / Bengkel	Bengkel 1	Bengkel 2	Bengkel 3	Bengkel 4	Bengkel 5
<i>user1</i>	5	4	3	4	3.97
<i>user2</i>	4	3.60	2	4	5
<i>user3</i>	5	5	3	4.4	5
<i>user4</i>	3.15	4	2	3	4
<i>user5</i>	5	4.17	3	4	5

D. Perhitungan Skor Akhir Rekomendasi

Tahap selanjutnya adalah menghitung skor akhir setiap bengkel dengan melakukan normalisasi nilai jarak dan *rating* menggunakan metode *Min-Max*. Dalam penelitian ini, jarak diberi bobot 60% dan *rating* 40%, karena jarak dianggap lebih berpengaruh dalam pemilihan bengkel. Sebagai contoh,

dilakukan perhitungan skor rekomendasi untuk *user* 1 pada bengkel 1 dengan parameter jarak minimum 2,34 km, maksimum 4,51 km, serta *rating* minimum 3 dan maksimum 5.

Perhitungan bobot jarak dilakukan menggunakan normalisasi *Min-Max* terbalik, sebagai berikut:

$$\text{Bobot Jarak} = 1 - \frac{2,83 - 2,34}{4,51 - 2,34} = 1 - \frac{0,49}{2,17} = 1 - 0,2258 = 0,774$$

Perhitungan bobot *rating* dilakukan menggunakan normalisasi *Min-Max* standar, sebagai berikut:

$$\text{Bobot Rating} = \frac{5 - 3}{5 - 3} = \frac{2}{2} = 1$$

Setelah bobot jarak dan bobot *rating* diperoleh, skor akhir rekomendasi dihitung menggunakan persamaan berikut:

$$\begin{aligned} \text{finalScore} &= (0,6 \times 0,774) + (0,4 \times 1) \\ &= 0,464 + 0,4 = 0,864 \end{aligned}$$

Nilai skor akhir digunakan untuk menentukan peringkat rekomendasi, di mana bengkel dengan skor tertinggi dipilih sebagai yang paling sesuai bagi pengguna. Pada contoh ini, bengkel 1 menjadi rekomendasi terbaik karena memiliki jarak yang lebih dekat dan *rating* prediksi tertinggi. Berdasarkan skor yang diperoleh, sistem menghasilkan *Top-N Recommendation*, yaitu daftar bengkel paling sesuai bagi pengguna.

Tabel 5. Hasil Nilai Akhir Rekomendasi

Id.	Jarak (km)	Bobot Jarak	Bobot Rating	Skor Rekomendasi	Ranking
b1	2.83	0.774	1	0.864	1
b2	2.34	1	0.5	0.8	2
b3	3.61	0.414	0.5	0.448	3
b4	4.31	0.092	0.485	0.249	4
b5	4.51	0	0	0	5

Berdasarkan hasil perhitungan skor akhir rekomendasi pada Tabel 5, sistem rekomendasi mempertimbangkan dua kriteria utama, yaitu jarak dan *rating* hasil prediksi. Pada penelitian ini, jarak memiliki bobot sebesar 60%, sedangkan *rating* memiliki bobot sebesar 40%, sehingga faktor jarak menjadi pertimbangan utama dalam menentukan rekomendasi bengkel.

Pada kondisi bengkel dengan jarak dekat dan *rating* tinggi, bengkel memperoleh skor rekomendasi tertinggi dan menjadi prioritas utama sistem. Hal ini menunjukkan bahwa kombinasi jarak yang dekat dan kualitas layanan yang baik sangat berpengaruh dalam menghasilkan rekomendasi terbaik bagi pengguna.

Pada kondisi bengkel dengan jarak dekat namun memiliki *rating* sedang, bengkel tetap direkomendasikan oleh sistem, tetapi berada pada peringkat menengah. Faktor jarak yang dekat masih memberikan nilai yang cukup besar, namun *rating* yang tidak terlalu tinggi menyebabkan skor rekomendasi lebih rendah dibandingkan bengkel dengan *rating* tinggi.

Selanjutnya, pada kondisi bengkel dengan jarak dekat tetapi memiliki *rating* rendah, skor rekomendasi bengkel mengalami penurunan. Meskipun jarak bengkel dekat dengan pengguna, kualitas layanan yang rendah berdasarkan penilaian pengguna lain menyebabkan bengkel tersebut tidak menjadi prioritas utama dalam rekomendasi.

Sementara itu, bengkel dengan jarak yang relatif jauh namun memiliki *rating* tinggi masih dapat muncul dalam daftar rekomendasi. Namun, posisinya berada pada peringkat yang lebih rendah karena faktor jarak yang memiliki bobot lebih besar dibandingkan *rating*. Hal ini menunjukkan bahwa sistem lebih mengutamakan kemudahan akses bagi pengguna tanpa mengabaikan kualitas layanan bengkel.

Berdasarkan hasil tersebut, dapat disimpulkan bahwa sistem rekomendasi mampu menyeimbangkan faktor jarak dan *rating* dalam menentukan prioritas bengkel yang direkomendasikan kepada pengguna.

E. Pengujian Sistem

Pengujian sistem dilakukan untuk dapat mengetahui seberapa bagus sistem rekomendasi diberikan sistem kepada *user*. Pada penelitian ini, metode pengujian yang digunakan yaitu dengan *Mean Absolute Error (MAE)* sebagai metrik evaluasi akurasi rekomendasi dan *Blackbox Testing* sebagai menguji apakah sistem telah berfungsi dengan baik.

1) Pengujian MAE

Pengujian ini dilakukan dengan menggunakan *Mean Absolute Error (MAE)* untuk menilai akurasi rekomendasi. Perhitungan *MAE* dilakukan dengan menghitung selisih antara *rating* prediksi yang dihasilkan oleh sistem dengan *rating* asli terhadap *user*.

Sebagai contoh perhitungan, misal seorang *user* yang telah memberi *rating* pada satu bengkel, kemudian mendapatkan hasil rekomendasinya berupa 3 bengkel. Bengkel pertama memperoleh 4,25 dengan *rating* asli 5. Bengkel kedua memperoleh 3,80 dengan *rating* asli 4. Bengkel ketiga memperoleh 2,90 dengan *rating* asli 3. Maka perhitungan *MAE* ditulis sebagai berikut:

$$\begin{aligned} \text{MAE} &= \frac{|4,25 - 5| + |3,80 - 4| + |2,90 - 3|}{3} \\ &= \frac{0,75 + 0,20 + 0,10}{3} \\ &= \frac{1,05}{3} \\ &= 0,35 \end{aligned}$$

Data uji diperoleh dari penilaian *user* berupa pemberian *rating* terhadap bengkel yang direkomendasikan sistem. Pada tahap pengujian ini digunakan tiga skenario, yaitu skenario pertama dengan melibatkan 5 *user*, skenario kedua dengan melibatkan 10 *user*, dan skenario ketiga dengan melibatkan seluruh *user* sebanyak 20 *user*.

Melalui ketiga skenario tersebut, dapat dianalisis apakah jumlah *user* yang memberikan *rating* berpengaruh terhadap akurasi sistem. Semakin kecil nilai *MAE* yang diperoleh, maka semakin baik pula kualitas rekomendasi yang dihasilkan. Berikut adalah hasil pengujian *MAE* berdasarkan 3 skenario pengujian:

a. Skenario pengujian MAE dengan 5 user

Tabel 6. Hasil uji MAE 5 user

User	Jumlah Rating Data	Jumlah Nilai Selisih	MAE
user1	24	17.06	0.711

User	Jumlah Rating Data	Jumlah Nilai Selisih	MAE
user2	19	14.49	0.763
user3	17	13.52	0.795
user4	14	10.87	0.777
user5	12	9.02	0.752
Rata-rata			0.759

b. Skenario pengujian MAE dengan 10 user

Tabel 7. Hasil uji MAE 10 user

User	Jumlah Rating Data	Jumlah Nilai Selisih	MAE
user1	24	17.06	0.711
user2	19	14.49	0.763
user3	17	13.52	0.795
user4	14	10.87	0.777
user5	12	9.02	0.752
user6	14	10.56	0.754
user7	15	10.79	0.720
user8	11	8.29	0.754
user9	8	4.75	0.593
user10	17	9.17	0.539
Rata-rata			0.715

c. Skenario pengujian dengan 20 user (all user)

Tabel 8. Hasil uji MAE 20 user

User	Jumlah Rating Data	Jumlah Nilai Selisih	MAE
user1	24	17.06	0.711
user2	19	14.49	0.763
user3	17	13.52	0.795
user4	14	10.87	0.777
user5	12	9.02	0.752
user6	14	10.56	0.754
user7	15	10.79	0.720
user8	11	8.29	0.754
user9	8	4.75	0.593
user10	17	9.17	0.539
user11	8	4.75	0.593
user12	11	7.62	0.693
user13	7	5.87	0.839
user14	9	5.26	0.584
user15	15	11.43	0.762
user16	8	5.93	0.742
user17	7	4.69	0.670
user18	10	7.09	0.709
user19	8	6.14	0.768
user20	17	11.75	0.691
Rata-rata			0.710

Berdasarkan hasil pengujian MAE dengan 3 skenario, skenario pertama dengan jumlah user 5 menghasilkan nilai MAE sebesar 0,759, skenario kedua dengan jumlah user 10

menghasilkan nilai MAE sebesar 0,715, dan pada skenario ketiga dengan jumlah user 20 menghasilkan nilai MAE sebesar 0,710. Dari ketiga skenario tersebut dapat dilihat bahwa semakin banyak jumlah user semakin kecil nilai MAE. Hal ini dikarenakan semakin banyak user semakin kecil sparsity yang dihasilkan oleh sistem. Sparsity merupakan kekosongan data.

Ketika sistem terdiri dari sedikit user, rating yang diberikan user terhadap item tidak sebanyak rating yang diberikan oleh user pada skenario dengan jumlah user yang lebih banyak. Semakin banyak user yang me-rating, maka akan semakin banyak item yang direkomendasikan sistem kepada user, sehingga selisih rating antara prediksi dan rating aktual menjadi lebih kecil.

2) Blackbox Testing

Tahap pengujian selanjutnya dilakukan dengan menggunakan blackbox testing. Pengujian ini bertujuan untuk memastikan bahwa sistem telah berjalan dengan baik dan dapat digunakan oleh pengguna sesuai dengan fungsi yang dirancang. Proses pengujian dilakukan dengan menjalankan seluruh fungsi yang terdapat dalam program secara menyeluruh untuk mengetahui kemampuan sistem dalam merespons input dari pengguna serta memastikan output yang dihasilkan sesuai dengan kebutuhan pengguna. Berikut adalah hasil pengujian blackbox:

Tabel 9. Hasil Blackbox Testing

Aspek Uji	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Uji
Menu Login	User mengisi form login dengan email dan password yang valid, lalu menekan button login	Sistem menampilkan halaman utama sesuai peran user (admin atau user biasa)	Berhasil
	User mengisi form login dengan email atau password yang salah	Sistem menampilkan pesan error bahwa email atau password tidak sesuai	Berhasil
Data Bengkel	Admin memilih menu bengkel	Sistem menampilkan daftar data bengkel yang tersimpan	Berhasil
Data User	Admin membuka menu user	Sistem menampilkan daftar data user dan form tambah data user baru	Berhasil
Create, Read, Update, Delete User	Admin mengisi form tambah user (nama, email, password) lalu menekan button create	Sistem menyimpan data dan menampilkan user baru pada daftar menu data user	Berhasil
	Admin melihat daftar user	Sistem menampilkan semua data user yang tersimpan	Berhasil
	Admin mengubah data user, lalu menekan button update	Sistem memperbarui data user sesuai perubahan	Berhasil
	Admin memilih salah satu user lalu menekan button delete	Sistem menghapus user dari daftar Menu data user	Berhasil

Aspek Uji	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Uji
Details Bengkel	User memilih bengkel	Sistem akan menampilkan detail bengkel tersebut	Berhasil
	User melakukan scroll pada halaman detail	Sistem menampilkan rekomendasi bengkel serupa berdasarkan penilaian dari user lain	Berhasil
Rating Bengkel	User memilih bengkel	Sistem akan menampilkan detail bengkel	Berhasil
	User memberi rating dan ulasan	Sistem menampilkan rekomendasi berdasarkan rating atau rekomendasi dari pengguna lain	Berhasil
Menu Logout	Kedua aktor memilih menu logout	Sistem akan keluar dari aplikasi dan menampilkan menu awal	Berhasil

IV.KESIMPULAN

Berdasarkan hasil penelitian penerapan Collaborative Filtering pada pengembangan sistem rekomendasi lokasi bengkel, dapat disimpulkan bahwa sistem rekomendasi yang dikembangkan mampu memberikan rekomendasi berdasarkan kemiripan bengkel serta penilaian bengkel oleh pengguna lain. Selain itu, hasil pengujian menunjukkan bahwa semakin banyak jumlah pengguna, nilai Mean Absolute Error (MAE) yang dihasilkan semakin kecil, yang mengindikasikan bahwa metode Collaborative Filtering dengan algoritma Cosine Similarity mampu menghasilkan tingkat akurasi yang cukup baik dalam memberikan rekomendasi bengkel sesuai dengan preferensi pengguna, serta meningkatkan kualitas rekomendasi seiring dengan bertambahnya data penilaian yang tersedia.

V. SARAN

Penelitian ini masih memiliki keterbatasan sehingga diperlukan pengembangan lebih lanjut dalam implementasi sistem rekomendasi. Penelitian selanjutnya diharapkan dapat memanfaatkan algoritma lain dalam proses rekomendasi item guna meningkatkan kualitas rekomendasi. Selain itu, penambahan jumlah data pengujian disarankan agar akurasi yang diperoleh menjadi lebih optimal. Pengembangan sistem rekomendasi selanjutnya dapat dilakukan dalam bentuk aplikasi mobile, karena faktor lokasi dan koneksi perangkat memengaruhi akurasi perhitungan jarak.

REFERENSI

[1] D. F. Alwan, A. R. Adriyanto, and D. Yudiarti, "Perancangan Prototype Aplikasi Mobile untuk Bengkel Motor di Wilayah Kota Bandung," 2025.

[2] S. Aisa, "Aplikasi Pencarian Bengkel Aktif dengan Google Maps API Berbasis Web," *DoubleClick: Journal*

of Computer and Information Technology, vol. 4, Feb. 2021.

[3] D. Patel, F. Patel, and U. Chauhan, "Recommendation Systems: Types, Applications, and Challenges," *International Journal of Computing and Digital Systems*, vol. 13, no. 1, pp. 851–868, 2023, doi: 10.12785/ijcds/130168.

[4] Agustian, S. P. A. Alkadri, and Istikoma, "Penerapan Metode Collaborative Filtering Untuk Rekomendasi Tempat Kos," *JIP (Jurnal Informatika Polinema)*, 2024.

[5] Y. Afoudi, M. Lazaar, and M. Al Achhab, "Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network," *Simul Model Pract Theory*, vol. 113, Dec. 2021, doi: 10.1016/j.simpat.2021.102375.

[6] S. Eliyas and P. Ranjana, "Recommendation Systems: Content-Based Filtering vs Collaborative Filtering," in *International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 1360–1365. doi: 10.1109/ICACITE53722.2022.9823730.

[7] S. Ismail, A. Abdul-Barik, and S. Abdul-Mumin, "A Review of Recommender System Strategies for Enhanced Service Recommendation," *International Journal of Software & Hardware Research in Engineering*, vol. 10, no. 1, Jan. 2022, doi: 10.26821/ijshre.10.1.2022.100112.

[8] D. B. Rajesh and A. Kumar, "Collaborative filtering models an experimental and detailed comparative study," *Sci Rep*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-15096-4.

[9] F. Yuniardini and T. Widiyaningtyas, "Analisis Perbandingan Pearson Correlation dan Cosine Similarity pada Rekomendasi Musik berbasis Collaborative Filtering," *Edumatic: Jurnal Pendidikan Informatika*, vol. 8, no. 2, pp. 555–564, Dec. 2024, doi: 10.29408/edumatic.v8i2.27781.

[10] A. Yusmar, L. K. Wardhani, and H. B. Suseno, "Restaurant Recommender System Using Item Based Collaborative Filtering and Adjusted Cosine Algorithm Similarity," *Jurnal Teknik Informatika*, vol. 14, no. 1, pp. 93–100, Sep. 2021, doi: 10.15408/jti.v14i1.21102.

[11] M. M. Dewi, R. Andriani, and M. Nuraminudin, "Performance Analysis of the Item-Based Collaborative Filtering Model in Yogyakarta Tourism Recommendations," 2025. [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JAIC>

[12] H. Februariyanti, A. D. Laksono, J. S. Wibowo, and M. S. Utomo, "Implementasi Metode Collaborative Filtering Untuk Sistem Rekomendasi Penjualan Pada Toko Mebel," *Jurnal Khatulistiwa Informatika*, 2021, [Online]. Available: www.unisbank.ac.id

[13] H. Ko, S. Lee, Y. Park, and A. Choi, "A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields," Jan. 01, 2022, *MDPI*. doi: 10.3390/electronics11010141.

- [14] P. Kumar, M. K. Gupta, C. R. S. Rao, M. Bhavsingh, and M. Srilakshmi, "A Comparative Analysis of Collaborative Filtering Similarity Measurements for Recommendation Systems," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, pp. 184–192, Feb. 2023, doi: 10.17762/ijritcc.v11i3s.6180.
- [15] A. Simangunsong, R. M. Simanjorang, F. Amalia, and P. Khairunnisa, "Implementasi Sistem Rekomendasi dengan Collaborative Filtering dalam Pemilihan Produk Skincare," *Jurnal Sains Manajemen Informatika dan Komputer*, pp. 56–63, 2025, [Online]. Available: <https://ojs.trigunadharma.ac.id/index.php/jis/index>
- [16] F. Kurniawan, K. Ningsih, and A. Komarudin, "Sistem Rekomendasi Channel YouTube Resep Masakan Menggunakan Collaborative Filtering," 2024.
- [17] A. K. Rai, P. Khanna, P. Singh, A. K. Rai, P. Khanna, and P. Singh, "Movie Recommender System: Collaborative Filtering Methods," *Journal of Management and Service Science*, vol. 04, no. 066, pp. 1–11, 2024, doi: 10.54060/a2zjourna.
- [18] N. Ifada, M. K. Sophan, and N. F. D. Putri, "A MinMax Item-based Method for Multi-Criteria Recommendation Systems," in *Procedia Computer Science*, Elsevier B.V., 2023, pp. 1020–1029. doi: 10.1016/j.procs.2023.10.611.
- [19] M. Alghamdi, H. Elazhary, and A. Mojahed, "Evaluation of Collaborative Filtering for Recommender Systems," 2021. [Online]. Available: www.ijacsa.thesai.org