Sistem Rekomendasi Menggunakan Content-Based Filltering pada Restoran di Kota Palangka Raya

Thomas Richard Wini Ngele[1)], Efrans Christian[2)], Jadiaman Parhusip[3)]

[1)2)3)]Teknik Informatika, Fakultas Teknik, Universitas Palangka Raya Kampus Tanjung Nyaho, Jalan Yos Sudarso, Palangka Raya, Kalimantan Tengah, Indonesia
[1)]thomasrichardwiningele@mhs.eng.upr.ac.id
[2)]efrans@it.upr.ac.id
[3)]parhusip.jadiaman@it.upr.ac.id

Abstrak

Pertumbuhan industri kuliner yang pesat di Kota Palangka Raya menciptakan tantangan bagi masyarakat dalam menentukan pilihan restoran yang sesuai dengan preferensi pribadi, sebuah fenomena yang dikenal sebagai paradoks pilihan. Penelitian ini bertujuan untuk membangun sistem rekomendasi restoran berbasis website yang mampu memberikan saran personal dan relevan bagi pengguna di Palangka Raya.

Metodologi yang digunakan adalah *Content-Based Filtering* (CBF) dengan pendekatan *Dual Similarity Measurement*. Metode ini menggabungkan *Jaccard Similarity* untuk pencocokan token kategorikal inti dan *Cosine Similarity* yang diimplementasikan dengan teknik *TF-IDF Character N-gram*. Integrasi kedua algoritma ini melalui Agregasi Skor Tertimbang bertujuan untuk meningkatkan akurasi rekomendasi serta memberikan ketahanan terhadap kesalahan pengetikan (*typo*) pada input pengguna. Atribut restoran yang digunakan meliputi nama restoran, *menu tags*, *place types*, harga, rating, lokasi, fasilitas, deskripsi, dan menu populer.

Hasil penelitian menunjukkan bahwa sistem berhasil memberikan rekomendasi yang selaras dengan keinginan pengguna. Berdasarkan hasil pengujian, sistem ini mencapai performa yang sangat baik dengan nilai Mean Average Precision (MAP) sebesar 0.90 (90%). Selain itu, evaluasi menggunakan metrik lainnya menunjukkan tingkat akurasi yang tinggi, yaitu Precision@k sebesar 0.85 (85%), Recall@k sebesar 0.82 (82%), dan F1-score sebesar 0.83 (83%). Hal ini membuktikan bahwa penerapan *Dual Similarity Measurement* efektif dalam menjaga relevansi rekomendasi meskipun terdapat variasi pada input pengguna.

Kata Kunci: Sistem Rekomendasi, *Content-Based Filtering*, *Jaccard Similarity*, *Cosine Similarity*, *Character N-gram*, Restoran, Palangka Raya.

Abstract

The rapid growth of the culinary industry in Palangka Raya City offers diverse choices for the community, yet simultaneously creates a "paradox of choice" phenomenon that complicates consumers' ability to determine restaurants matching their personal preferences. This research aims to design and develop a web-based restaurant recommendation system capable of providing personalized and relevant suggestions for the community in Palangka Raya.

The methodology implemented in this system is Content-Based Filtering (CBF) with a Dual Similarity Measurement approach. This method integrates Jaccard Similarity for processing core categorical tokens and Cosine Similarity combined with TF-IDF Character N-gram techniques. The use of weighted score aggregation from both algorithms aims to improve recommendation accuracy and provide robustness against typographical errors (typos) in user input. Content features used include restaurant name, menu tags, place types, price, rating, location, facilities, description, and popular menus. The system development followed the Waterfall model.

The results indicate that the system successfully provides restaurant recommendations consistent with user preferences. Based on system testing results, the performance achieved is significant, with a Mean Average Precision (MAP) value of 0.90 (90%). Further evaluation shows

DOI:https://doi.org/10.47111/jointecoms.v6i3 *Received: 01-09-2026*
Accepted: 10-09-2026

high accuracy levels with a Precision@k of 0.85 (85%), Recall@k of 0.82 (82%), and an F1-score of 0.83 (83%). These results confirm that the application of Dual Similarity Measurement is effective in maintaining the relevance of recommendation lists despite variations in user input.

Keywords: Recommendation System, Content-Based Filtering, Jaccard Similarity, Cosine Similarity, Character N-gram, Restaurant, Palangka Raya.

1. PENDAHULUAN

Kota Palangka Raya, sebagai pusat pemerintahan dan ekonomi di Kalimantan Tengah, menunjukkan tren pertumbuhan industri kuliner yang signifikan. Pertumbuhan jumlah tempat makan yang pesat, mulai dari kafe modern hingga warung tradisional, mencerminkan peningkatan permintaan masyarakat akan keragaman kuliner. Namun, banyaknya opsi ini justru menimbulkan fenomena *paradox of choice* atau paradoks pilihan, di mana pengguna mengalami kesulitan kognitif dalam menentukan tempat makan yang paling sesuai dengan preferensi pribadi mereka. [1] kondisi ini membuat masyarakat membutuhkan alat digital yang mampu menyaring informasi secara efektif untuk menawarkan rekomendasi yang akurat.

Faktor penentu utama dalam pengambilan keputusan konsumen meliputi atribut seperti jenis makanan, rentang harga, profil rasa, dan ketersediaan fasilitas. Sistem rekomendasi berbasis web diyakini sebagai solusi tepat karena memungkinkan aksesibilitas yang luas dan memfasilitasi pengguna untuk menemukan restoran tanpa perlu melakukan pencarian manual yang memakan waktu. Dalam konteks ini, pendekatan *Content-Based Filtering* (CBF) dipilih karena kesesuaiannya dengan domain restoran yang memiliki atribut data kaya dan stabil seperti *menu tags*, harga, dan *place types*. [2] menjelaskan bahwa CBF bekerja dengan mencocokkan karakteristik item dengan profil preferensi pengguna untuk menyarankan item lain dengan kemiripan konten. Keunggulan krusial CBF adalah ketahanannya terhadap masalah *cold start*, di mana sistem tetap dapat memberikan rekomendasi pada platform baru tanpa memerlukan data interaksi dari pengguna lain.

Untuk mengukur tingkat kemiripan, penelitian ini mengadopsi *Dual Similarity Measurement*. Metode utama yang digunakan adalah *Jaccard Similarity*, sebuah teknik perhitungan kesamaan berbasis himpunan yang efektif untuk data kategorikal. Namun, karena algoritma berbasis token murni sensitif terhadap kesalahan pengetikan (*typo*), sistem ini mengintegrasikan *Cosine Similarity* dengan teknik *TF-IDF Character N-gram*. Penggunaan *character n-gram* sangat efektif untuk mendeteksi kesamaan pada level karakter sehingga sistem tetap dapat memberikan hasil yang relevan meskipun terdapat variasi ejaan pada *input* pengguna. Pendekatan CBF ini juga didukung oleh temuan [3] yang menunjukkan bahwa penggunaan atribut seperti rasa, harga, dan rating mampu memberikan akurasi yang signifikan dalam meningkatkan relevansi rekomendasi restoran.

2. TINJAUAN PUSTAKA

2.1 Teori-Teori Pendukung

Berdasarkan penelitian yang dilakukan [4], Performa sistem dievaluasi menggunakan *precision, recall, F1-score,* dan gain kumulatif yang dinormalisasi. Hasil pada top@10 untuk *precision* adalah 0,619, 0,650, untuk masing-masing berbasis emosi, piksel, dan metadata. Mengenai *recall*, pada top@10, *recall* adalah 0,5420, 0,5045, dan 0,4929 untuk masing-masing berbasis emosi, *piksel*, dan metadata. Pada gain kumulatif yang dinormalisasi, emosi, piksel, dan anotasi memiliki top@10 sebesar 0,302, 0,299, dan 0,285. Hasil menunjukkan bahwa rekomendasi gambar berdasarkan emosi mencapai hasil yang lebih baik daripada menggunakan fitur-fitur mutakhir, seperti kesamaan piksel dan metadata.

Penerapan Metode *Content Based Filtering* Dalam Implementasi Sistem Rekomendasi Tanaman PanganBerdasarkan penelitian yang dilakukan oleh [5]

menggunakan algoritma *Vector Space* Model dengan menerapkan metode *Content Based Filtering*, sistem rekomendasi ini berhasil memberikan informasi kepada pedagang mengenai petani yang memiliki profil serupa dengan mereka, serta memberikan informasi kepada petani mengenai pedagang dengan profil serupa. Dalam pengujian presisi, sistem ini menunjukkan hasil rata-rata presisi sebesar 78.40% dari 10 profil pedagang yang diuji dengan 15 rekomendasi kelompok tani teratas. Namun, untuk meningkatkan kualitas sistem, disarankan untuk menggabungkan pengujian menggunakan recall agar dapat memastikan bahwa seluruh kelompok tani yang relevan bagi pedagang dapat direkomendasikan oleh sistem.

Berdasarkan penelitian yang dilakukan oleh [6] sistem rekomendasi menggunakan metode *Collaborative filltering* dengan *Jaccard Similarity* dikembangkan untuk meningkatkan akurasi rekomendasi dalam waktu komputasi yang lebih rendah. Sistem ini dirancang untuk membantu pengguna menemukan item yang sesuai dengan preferensi mereka berdasarkan kesamaan antar item yang dihitung menggunakan indeks Jaccard.

Metode yang digunakan mempertimbangkan seluruh vektor rating pengguna untuk mengklasifikasikan lingkungan yang relevan dan menghasilkan rekomendasi yang lebih akurat. Selain itu, penelitian ini menyoroti kelemahan dari pendekatan kesamaan tradisional yang hanya mempertimbangkan item yang diberi peringkat bersama, sehingga model *Jaccard Similarity* yang diusulkan dapat memberikan hasil yang lebih akurat dengan mempertimbangkan semua data pengguna. Hasil eksperimen menunjukkan bahwa pendekatan ini memberikan performa lebih baik dibandingkan dengan metode kesamaan lainnya seperti *Cosine Similarity dan Pearson Correlation***,** dengan tingkat akurasi yang lebih tinggi dan efisiensi komputasi yang lebih baik

Berdasarkan penelitian yang dilakukan [7] Penelitian ini menerapkan metode *Content-Based Filtering* dalam pembuatan sebuah sistem rekomendasi film pada platform streaming, metode ini digunakan untuk mencari kesamaan-kesamaan film dari deskripsi filmnya dengan memberikan pembobotan kata pada deskripsi film. Pembobotan dilakukan dengan menghitung bobot dari atribut deskripsi menggunakan metode *TF-IDF* dan *Cosine Similarity*, dengan membagi film berdasarkan tipenya, yaitu data dengan tipe movie dan data dengan tipe show, sebuah sistem rekomendasi film pada plaform streaming berhasil dibuat. Dengan membagi film sesuai tipenya, pengguna akan mendapatkan rekomendasi film yang lebih akurat, karena bukan hanya membuat rekomendasi berdasarkan kemiripan film saja, namun juga rekomendasi sesuai dengan tipe film. Skor evaluasi sistem rekomendasi menggunakan *mean average precision* (MAP), film pada *platform streaming* menggunakan metode *content-based filtering* ini adalah 0.7973 untuk sistem rekomendasi film tipe *show* dan 0.7022 untuk sistem rekomendasi film tipe *movie.1*

2.2 Teori Pendukung

2.2.1. Restoran

Restoran merupakan usaha penyediaan jasa makanan dan minuman dilengkapi dengan peralatan dan perlengkapan untuk proses pembuatan, penyimpanan, dan penyajian di suatu tempat tetap yang tidak berpindah –pindah dengan tujuan memperoleh keuntungan [8]

2.2.2. Sistem Rekomendasi

Sistem rekomendasi merupakan sebuah (*web*) alat personalisasi yang menyediakan pengguna sebuah informasi daftar item-item yang sesuai dengan keinginan masing-masing pengguna. Sistem rekomendasi menyimpulkan preferensi pengguna dengan menganalisis ketersediaan data pengguna, informasi tentang pengguna dan lingkungannya . Tujuan dari sistem rekomendasi adalah untuk memanfaatkan informasi dan katalog produk dan memahami preferensi pengguna berdasarkan pilihan mereka, dan

merekomendasikan mereka produk yang tidak mungkin untuk dipilih melalui ruang produk secara manual [9] .

2.2.3. *Content-Based Filltering*

Content-based filtering adalah teknik dalam sistem rekomendasi yang melihat perilaku pengguna untuk mengidentifikasi pola perilaku dan menyarankan produk yang sesuai dengan pola tersebut. Strategi ini menyelidiki kencenderungan perilaku klien untuk membuat model, yang kemudian dikontraskan dengan item yang disertakan dengan yang disarankan[2].

Teori *Content-Based Filtering* (CBF) berfokus pada analisis konten dari item yang disediakan untuk rekomendasi. Sistem ini mempelajari preferensi pengguna dengan menganalisis atribut atau fitur dari item yang disukai atau dinilai positif oleh pengguna tersebut. Berdasarkan analisis ini, sistem kemudian mencocokkan atribut ini dengan atribut dari item lain untuk memberikan rekomendasi. Langkah pertama dalam CBF adalah ekstraksi fitur, di mana setiap item dianalisis untuk mengekstraksi fitur-fitur relevan. Misalnya, dalam kasus film, fitur bisa berupa genre, aktor, sutradara, atau sinopsis. Selanjutnya, representasi pengguna dibuat berdasarkan item yang telah dinilai positif oleh pengguna tersebut, mencerminkan preferensi pengguna terhadap fitur-fitur tertentu.

Pengukuran kesamaan antara profil pengguna dan item lain kemudian dilakukan menggunakan metode seperti *Jaccard Similarity* atau korelasi Pearson. *Jaccard Similarity* adalah teknik yang mengukur kesamaan antara dua himpunan dengan membandingkan jumlah elemen yang sama di kedua himpunan terhadap jumlah total elemen unik dalam kedua himpunan tersebut. Nilai *Jaccard Similarity* berkisar antara 0 hingga 1, di mana nilai yang mendekati 1 menunjukkan bahwa kedua himpunan memiliki banyak kesamaan, sedangkan nilai yang mendekati 0 menunjukkan sedikit atau tidak ada kesamaan. Dalam konteks *Content-Based Filtering* (CBF), profil pengguna dan item baru direpresentasikan sebagai himpunan fitur, dan *Jaccard Similarity* digunakan untuk menentukan tingkat kemiripan antara preferensi pengguna dengan item yang direkomendasikan.

Selain *cosine similarity*, *Jaccard Similarity* merupakan teknik penting dalam *Content-Based Filtering* (CBF) yang digunakan untuk mengukur kesamaan antara dua item berdasarkan jumlah elemen yang sama dalam himpunan fitur mereka. *Jaccard Similarity* dihitung dengan membandingkan jumlah elemen yang sama dalam dua himpunan terhadap jumlah total elemen unik dari kedua himpunan tersebut. Nilai *Jaccard Similarity* berkisar antara 0 hingga 1, di mana nilai yang lebih tinggi menunjukkan tingkat kesamaan yang lebih besar. Dalam konteks CBF, teknik ini digunakan untuk menentukan tingkat kemiripan antara item yang telah disukai oleh pengguna dengan item lain yang tersedia dalam sistem rekomendasi.

2.2.4. *Jaccard Simmilarity*

Jaccard Similarity atau Jaccard Coefficient merupakan algoritma yang fungsinya untuk membandingkan dua sample yaitu dokumen yang satu dengan yang lainnya berdasarkan kata yang dimilikinya. Jaccard similarity biasanya digunakan untuk membandingkan dokumen dan menghitung nilai kemiripan (similarity) dari dua buah objek atau dokumen. Jaccard similarity dapat dirumuskan sebagai berikut:

$$Similarity\ (X,Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

Dimana:
X = Dokumen 1
Y = Dokumen 2

Rumus 1 merupakan dari rumus Jaccard Similarity atau Jaccard Coefficient yang digunakan untuk mencari persamaan dan perbedaan pada dua sample. sebagai contoh diketahui X "Magister Teknik Informatika Yogyakarta", dan Y "Magister Teknik Sipil Yogyakarta". Maka akan menghasilkan nilainya:

$$Similarity\ (X,Y) = \frac{|\text{Magister Teknik Yogyakarta}|}{|\text{Magister Teknik Informatika Sipil Yogyakarta}|}$$

$$Similarity\ (X,Y) = \frac{3}{5} = 0{,}6\ x100\% = 60\%$$

Dari dua contoh sample di atas setelah dihitung kesamaannya menggunakan Jaccard similarity, bahwa kedua sample tersebut memiliki kesamaan atau kemiripan sebesar 60% [10].

2.2.5. *Cosine Simmilarity*

Cosine similarity merupakan metode untuk menghitung kesamaan antara dua buah objek yang dinyatakan dalam dua buah *vector* [11]. Rumus yang umum digunakan dalam *cosine similarity* adalah:

$$cosine\ similarity(A,B) = \frac{A\ .\ B}{\|A\|\ x\ \|B\|}$$

Penjelasan :

A	:	Vektor A, yang akan dibandingkan kemiripannya
B	:	Vektor B, yang akan dibandingkan kemiripannya
$A\ .\ B$	:	Hasil kali *dot* antara dua vektor
$\|A\|\ x\ \|B\|$	:	Magnitudo (norma) dari vektor A dan B

Dalam sistem rekomendasi restoran berbasis *content-based filtering*, *Cosine Similarity* digunakan untuk mengukur tingkat kemiripan antara teks pada data restoran (seperti nama menu, menu tags, atau deskripsi) dengan kata kunci atau query yang dimasukkan pengguna.
Metode ini diimplementasikan menggunakan TF-IDF character n-gram, yang mengubah teks menjadi representasi numerik berdasarkan frekuensi kemunculan karakter. Pendekatan ini memberikan toleransi terhadap kesalahan pengetikan (*typo*) atau variasi kecil dalam penulisan kata.

Dengan perhitungan *Cosine Similarity*, sistem dapat mengidentifikasi restoran yang memiliki kemiripan tertinggi terhadap preferensi atau input pengguna, sehingga hasil rekomendasi tetap relevan meskipun terjadi[11].

2.2.6. TF-IDF *(Term Frequency-Inverse Document Frequency)*

TF-IDF atau Term Frequency Inverse Document Frequency adalah suatu proses untuk melakukan transformasi data dari data teks menjadi data numerik untuk dilakukan pembobotan pada tiap kata pada tiap dokumen yang ada. TF merupakan frekuensi munculnya kata pada setiap dokumen yang diberikan menunjukkan seberapa bagusnya dan pentingnya kata di dalam tiap dokumen tersebut [12].

2.2.7. N-gram

N-gram adalah potongan sejumlah n karakter dari sebuah string yang diaplikasikan untuk pembangkitan kata atau karakter. Metode n-gram ini digunakan untuk mengambil potongan-potongan karakter huruf sejumlah n dari sebuah kata yang secara kontinuitas dibaca dari teks sumber hingga akhir dari dokumen. N-gram dibedakan berdasarkan jumlah potongan karakter sebesar n. Penambahan garis bawah (blank) pada

awal dan akhir kata digunakan untuk membantu menentukan kondisi awal dan akhir kata. Sebagai contoh : kata ”ROSE” dapat diuraikan ke dalam beberapa n-gram berikut (”_” merepresentasikan blank) :

uni-grams : R,O,S,E
bi-grams : _R,RO,OS,SE,E_
tri-grams : _RO,ROS,OSE,SE_
quad-grams : _ROS, ROSE, OSE_
quint-grams : _ROSE, ROSE_

Pada pembangkitan kata metode N-gram digunakan untuk mengambil potongan kata sejumlah n dari kalimat yang secara kontinuitas dibaca dari teks sumber hingga akhir [13].

2.2.8. *Whitespace Tokenization*

Whitespace tokenization adalah metode dasar dalam pemrosesan bahasa alami (NLP) yang memisahkan teks menjadi token berdasarkan spasi. Metode ini membagi teks menjadi kata-kata individu dengan memisahkan setiap kata saat ditemukan spasi, tanpa mempertimbangkan tanda baca atau struktur linguistik lainnya[14].

Whitespace tokenization sangat cocok digunakan bersama dengan *Jaccard similarity* dalam pengukuran kesamaan teks. *Jaccard similarity* mengukur kesamaan antara dua set dengan membandingkan ukuran irisan dan gabungan dari set tersebut. Dengan menggunakan *whitespace tokenization*, teks dapat dengan mudah diubah menjadi set kata-kata, memungkinkan perhitungan *Jaccard similarity* yang efisien[15].

2.2.9. *Mean Average Presicion (MAP)*

Mean Average Precision (MAP) digunakan untuk mengevaluasi akurasi sistem rekomendasi dengan menghitung rata-rata presisi dari semua pengguna. Metrik ini memberikan gambaran yang jelas tentang seberapa baik sistem dalam meranking item yang relevan bagi pengguna. MAP mengukur performa sistem rekomendasi dengan mempertimbangkan relevansi item yang direkomendasikan dan sangat berguna dalam konteks di mana pengguna memiliki preferensi yang berbeda, sehingga sistem perlu menyesuaikan rekomendasi berdasarkan interaksi pengguna sebelumnya. Untuk menghitung MAP, pertama-tama presisi dihitung untuk setiap pengguna berdasarkan item yang relevan yang mereka terima, kemudian rata-rata dari semua nilai presisi ini diambil untuk mendapatkan nilai MAP keseluruhan[16].

Evaluasi dengan MAP terdiri dari tiga tahap, yaitu menghitung precision, mean precision, dan nilai MAP.

Precision adalah proporsi jumlah dokumen yang ditemukan dan dianggap relevan untuk kebutuhan pencari informasi, atau rasio jumlah dokumen yang ditemukan kembali dengan total jumlah dokumen yang dianggap relevan. Precision juga bisa diartikan kepersisan atau kecocokan antara permintaan informasi dengan jawaban terhadap permintaan tersebut[17].

$$Precision : (R_{jk)} = \frac{Jumlah\ dokumen\ relevan}{k}$$

2.2.10. *Precision@k*

Precision@k merupakan sebuah pengujian yang menentukan tingkat proporsi jumlah dokumen relevan yang telah ditemukan oleh sebuah sistem. Pencarian sebanyak nilai k akan dianggap sebagai parameter dalam menentukan tingkat relevansi sistem untuk kebutuhan pengujian pencarian informasi [18] . Berikut adalah bentuk rumus perhitungan nilai Precision@k:

$$Prescision@k = \frac{Number\ of\ relevant\ items\ in\ top{-}k}{X}$$

2.2.11. *Recall@k*

Recall@k merupakan metrik yang mengukur berapa banyak dokumen yang berada di antara k hasil teratas. Metrik evaluasi ini populer dalam deep retrieval dan recommendation systems
[19]. Berikut adalah bentuk rumus perhitungan nilai Recall@k:

$$Recall@k = \frac{|\{relevant\ items\} \cap \{Top - k\ recommended\ items\}|}{|\{relevant\ items\}|}$$

2.2.12. *F1-score@k*

F1-score@k adalah rata-rata harmonik dari *Precision@k* dan *Recall@k*. Metrik evaluasi ini berguna untuk menyeimbangkan *trade off* antara ketepatan dan cakupan, terutama jika keduanya sama pentingnya. Nilai *F1-score@k* yang tinggi menunjukkan bahwa sistem tidak hanya tepat, tetapi juga mampu menjangkau banyak item relevan[20]. Berikut adalah bentuk rumus perhitungan nilai *F1-score@k:*

$$F1 - score@k = 2x\ \frac{Precision@k\ \times Recall@k}{Precision@k + Recall@k}$$

3. METODE PENELITIAN

3.1. Pendekatan Penelitian

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak (model Waterfall) untuk tahap analisis kebutuhan, desain, implementasi, pengujian, dan pemeliharaan sistem rekomendasi restoran berbasis web. Tujuan utamanya adalah merancang, mengimplementasikan, dan mengevaluasi performa rekomendasi berbasis *content-based filltering.*

3.2. Lokasi dan sampel data

Data restoran dikumpulkan secara *scraping* untuk cakupan Kota Palangka Raya—termasuk nama, menu dan harga, fasilitas, rating, serta tags menu — dengan total sampel sebanyak 75 restoran. Sumber data utama: Google Maps dan Gofood sebagai titik awal pengumpulan atribut restoran.

3.3. Perangkat dan Bahan

Pengembangan dilakukan pada lingkungan SQLYOG dengan backend Flask/Python dan frontend web; pengolahan data dan model menggunakan pandas, scikit-learn, Diagram dan flowchart dibuat untuk mendukung desain **.**

3.4. Pengumpulan dan Preprocessing Data

Data dikumpulkan secara manual dari Google Maps dan GoFood, mencakup atribut nama restoran, *menu tags, place types*, harga, rating, lokasi, fasilitas, deskripsi, dan menu populer. Tahap *preprocessing* meliputi:

1. Normalisasi teks : Mengubah teks menjadi huruf kecil dan standarisasi istilah (misal: "Jalan" menjadi "jl").

Teks Asli	**Teks Hasil**
Kueri: " masakan khas dayak,seafood,steak"	Kueri Normalisasi: "masakan khas dayak seafood steak ikan seafood"
Dokumen: Restoran menyajikan ikan patin bakar khas Dayak. Selain itu juga ada menu steak dan chicken."	Dokumen Normalisasi (combined): "restoran menyajikan ikan patin bakar khas dayak selain itu juga ada menu steak,chicken dan ikan"

Tabel 3.1 Contoh Hasil Normalisasi teks

2. *Whitespace Tokenizer* : Memecah teks menjadi token berdasarkan spasi.

Teks Input	**Proses Kunci**	**Output Token (List Kata)**
Kueri Normalisasi: "masakan khas	Tokenisasi berdasarkan spasi.	['masakan', 'khas', 'dayak', 'seafood', 'steak', 'ikan', 'seafood']

dayak seafood steak ikan seafood"		
Dokumen Normalisasi: "restoran menyajikan ikan patin bakar khas dayak selain itu juga ada menu steak dan chicken ikan"	Tokenisasi berdasarkan spasi.	['restoran', 'menyajikan', 'ikan', 'patin', 'bakar', 'khas', 'dayak', 'selain', 'itu', 'juga', 'ada', 'menu', 'steak', 'dan', 'chicken', 'ikan']

Tabel 3.2 Contoh hasil *whitespace tokenizer*

3. Penghapusan *Stopwords* : Menghilangkan kata umum yang tidak memiliki makna penting.

3.5. Vektorisasi fitur *Jaccard*

Representasi fitur dari restoran diproses menggunakan metode *Tf-Idfvektorizer* menggunakan *Word N-gram (1-3)*, di mana setiap kata kunci diubah menjadi vektor biner (1 jika ada, 0 jika tidak ada) untuk perhitungan *Jaccard*.

Token (Word N-gram)	**ayam**	**ikan**	**patin**	**bakar**	**khas dayak**	**seafood**	**steak**	**Jumlah Token**
Kueri Pengguna	0	1	0	0	1	1	1	5
Restoran A	1	1	1	1	1	0	1	6
Restoran B	0	0	0	0	0	1	0	2

Tabel 3.3 Hasil *Tf-Idfvektorizer*

3.6. Perhitungan *cosine (character n-gram)*

Cosine Similarity diimplementasikan menggunakan TF-IDF *character n-gram* untuk mengukur kemiripan bentuk karakter dan memberikan toleransi *typo*. Perhitungan ini penting saat *query* pengguna memiliki kesalahan pengetikan.

Menu Tags (Data Restoran)	Kategori Preferensi (Query User)
Masakan khas dayak	Masakan khas dayk

Tabel 3.4 contoh kasus *typo*

Jaccard similarity : Meskipun terdapat kesalahan pengetikan (typo) pada preferensi pengguna, *Jaccard Similarity* tetap dapat menghitung kemiripan berdasarkan kata-kata yang ditulis dengan benar. Sebagai contoh, pada frasa 'Masakan khas dayk', algoritma akan tetap mengenali kata 'Masakan' dan 'khas', meskipun nilai kemiripan akhirnya akan lebih kecil karena adanya kata yang tidak cocok

Cosine similarity : Vektor TF-IDF *n-gram* akan menangkap kemiripan urutan karakter seperti *n-gram* "Day", "aya", "yak", dan "day". Karena ada banyak *n-gram* yang cocok, *Cosine Similarity* dapat menghasilkan nilai kemiripan yang tinggi.

3.7. Agregasi skor kemiripan (*final score agregation)*

Perhitungan Kemiripan Sistem menggunakan Agregasi Skor Tertimbang untuk menggabungkan dua sinyal kemiripan dengan bobot seimbang

$$\text{Skor Final} = (\omega_{Jaccard} \times \text{Skor Jaccard}) + (\omega_{Cosine} \times \text{Skor Cosine})$$

Rumus akhir sebagai berikut : ($\omega Jaccard$ = 0.5 dan ωCosine = 0.5)

Penetapan bobot yang sama antara *Jaccard Similarity* dan *Cosine Similarity* dilakukan untuk menyeimbangkan kontribusi kedua metode dalam proses pengukuran kemiripan.

4. PEMBAHASAN

4.1 Pengembangan Sistem Rekomendasi

Pada bab ini, dilakukan pembahasan mengenai hasil implementasi sistem rekomendasi restoran di Kota Palangka Raya yang dibangun menggunakan metode *Content-Based Filtering* (CBF). Fokus utama pengembangan adalah pada penerapan *Dual Similarity Measurement* yang menggabungkan *Jaccard Similarity* dan *Cosine Similarity* berbasis *Character N-gram* untuk meningkatkan akurasi dan ketahanan sistem terhadap variasi input pengguna.

4.1.1. Tahap preprocessing data

Tahap ini bertujuan untuk mengubah data mentah restoran menjadi format yang siap diproses oleh algoritma. Proses ini sangat krusial karena kualitas rekomendasi sangat bergantung pada kebersihan data tekstual yang digunakan.

Normalisasi Teks (*Text Normalization*): Setiap atribut tekstual seperti nama restoran, deskripsi, dan menu diubah menjadi huruf kecil (*lowercase*) untuk memastikan konsistensi. Selain itu, dilakukan standarisasi istilah, seperti mengubah kata "Jalan" atau "Jln" menjadi "jl", serta teknik injeksi kata kunci (misalnya, menambahkan kata "ikan" jika terdapat kata "patin" atau "lele") untuk memperkuat sinyal pencocokan.

	Teks Asli (Input)	Teks Hasil (Normalisasi dan Gabungan)
Kueri Pengguna	"ayam bakar di jl tjilik riwut"	Kueri Normalisasi:"ayam bakar di jl tjilik riwut"
Dokumen Restoran	Restoran menyajikan menu ayam bakar spesial. Lokasi di Jalan Tjilik Riwut.	Dokumen Normalisasi (Combined):"restoran menyajikan menu ayam bakar spesial lokasi di jl tjilik riwut"

Tabel 4.1 *Output* normalisasi teks

Tokenisasi (*Whitespace Tokenizer*): Teks yang telah dinormalisasi kemudian dipecah menjadi unit-unit kata (token) berdasarkan spasi. Penggunaan *whitespace tokenizer* dipilih karena kesederhanaannya yang efektif untuk bahasa Indonesia yang menggunakan spasi sebagai pemisah kata.

Input user	**(Whitespace)**	**Unigram**	**Bigram**	**Trigram**
Ayam bakar di jl.tjilik riwut	‘ayam’, ‘bakar’ , ‘di’, ‘jl.tjilik’ , ‘riwut’	‘ayam’ , ‘bakar’ , ‘jl.tjilik’ , ‘riwut’	‘ayam bakar’ , ‘bakar di’ , ‘di jl.tjilik’ , ‘jl.tjilik riwut’	‘ayam bakar di’ , ‘jl.tjilik riwut’

Tabel 4.2 *Output whitespacetokenizer*

Penghapusan *Stopwords*: Kata-kata umum yang sering muncul namun tidak memiliki nilai informasi penting dalam konteks kuliner (seperti "dan", "di", "ke", "yang")

dihilangkan dari daftar token. Hal ini dilakukan agar algoritma hanya fokus pada kata kunci yang merepresentasikan karakteristik restoran, seperti "pedas", "bakar", atau "seafood".

4.1.2. Implementasi *Dual Similarity Measurement*

Sistem ini memproses dua representasi vektor secara paralel untuk mendapatkan hasil yang optimal:

Vektorisasi *Jaccard* (Biner): Data restoran diubah menjadi vektor biner (nilai 1 jika kata hadir, 0 jika tidak) menggunakan *Word N-gram* (rentang 1 hingga 3 kata). Representasi ini digunakan untuk menghitung kecocokan konseptual atau kata kunci inti antara preferensi pengguna dan fitur restoran.\

Fitur (Word N-gram 1-3)	TF-IDF Weight (Desimal)	Biner (Bin=1)	Penjelasan
ayam	0.21	1	Bobot desimal (rendah karena umum), tetapi Biner=1 (Hadir).
bakar	0.25	1	Bobot desimal (lebih tinggi dari ayam), tetapi Biner=1 (Hadir).
di	(Dihilangkan)	0	Merupakan Stopword; token ini tidak dimasukkan ke dalam Matriks Vektor.
ayam bakar	0.35	1	Bobot desimal, Biner=1.
jl tjilik	0.48	1	Bobot desimal tertinggi karena spesifik dan jarang (IDF tinggi). Biner=1 (Hadir).
tjilik riwut	0.40	1	Bobot desimal, Biner=1 (Hadir).
ayam bakar tjilik	0.55	1	Trigram dengan bobot tinggi, Biner=1 (Hadir).

Tabel 4.3 *Output* dari *Tfidfvectorizer (jaccard)*

Vektorisasi Cosine (TF-IDF Character N-gram): Data direpresentasikan sebagai vektor TF-IDF berbasis urutan karakter (3 hingga 5 karakter). Keunggulan teknik ini adalah kemampuannya menangkap kemiripan bentuk kata meskipun terjadi kesalahan pengetikan (*typo*). Sebagai contoh, input "ikan bkar" tetap dapat dicocokkan dengan data "ikan bakar" karena memiliki irisan karakter n-gram yang sangat besar.

Fitur (Char N-gram 3-5)	Bobot TF-IDF (Contoh Nilai Desimal)	Penjelasan
Ayam (4-gram)	0.315	Bobot Tinggi: Urutan karakter ayam muncul 1 kali di kueri.
bakar (5-gram)	0.301	Bobot Sedang: Urutan karakter lengkap bakar penting.
baka (4-gram)	0.288	Bobot Sedang: Potongan baka juga berkontribusi pada kecocokan.
tjili (5-gram)	0.420	Bobot Tertinggi: Karakter ini sangat spesifik (IDF Tinggi).
ilik (4-gram)	0.385	Bobot Tinggi: Karakter dari kata tjilik.

riwut (5-gram)	0.350	Bobot Tinggi: Karakter lengkap riwut spesifik lokasi.
riwu (4-gram)	0.330	Bobot Tinggi: N-gram dari riwut.

Tabel 4.4. *output* hasil dari *TF-IDF character n-gram (cosine)*

4.1.3. Perhitungan Agregasi Skor Akhir

Setelah mendapatkan nilai dari kedua algoritma, sistem melakukan penggabungan menggunakan Agregasi Skor Tertimbang. Dalam pengujian ini, bobot yang diberikan adalah seimbang (ωJaccard = 0.5 dan ωCosine = 0.5) untuk memastikan stabilitas hasil.

doc_id	Jaccard (J)	Cosine (C)	Perhitungan Agregasi (0.5 +0.5C)	FinalScore (Laporan)
3	0,6667	0,8783	(0.5 x 0.6667) + (0.5 x 0.8738) = 0.7495	0,7495
0	0,1000	0,4416	(0.5 x 0.1000) + (0.5 x 0.4416) =0.2366	0,2366
2	0,2000	0,1712	(0.5 x 0.2000) + (0.5 x 0.1712) = 0.1885	0,1885
1	0,0667	0,1648	(0.5 x 0.0667) + (0.5 x 0.1648) = 0.1059	0,1095

Tabel 4.5 *Output* skor kemiripan
Jaccard, *Cosine* dan Penggabungan kedua perhitungan.

Sebagai tambahan, diterapkan mekanisme *Field Boosting* pada variabel inti seperti 'Menu Populer' dan 'Menu Tags'. Variabel 'Menu Populer' diberikan bobot tertinggi (0.30) karena merupakan identitas utama yang paling sering dicari oleh pengguna. Sebaliknya, variabel 'Lokasi' diberikan bobot terendah (0.10) karena sifatnya yang hanya sebagai faktor pendukung.

4.1.4. Hasil Peringkat Rekomendasi

dokumen dengan *FinalScore* tertinggi akan menempati peringkat teratas. Contoh pengujian menunjukkan bahwa untuk kueri "ayam bakar pedas di jl.tjilik riwut", sistem berhasil memberikan skor tertinggi sebesar 0.7495 pada restoran yang paling relevan. Restoran yang memiliki skor di bawah ambang batas minimum (*min_score threshold*) akan secara otomatis dihapus dari daftar untuk menjamin kualitas hasil yang diterima pengguna.

4.2. Evaluasi Sistem

Evaluasi dilakukan untuk mengukur akurasi prediksi dan kualitas urutan (*ranking*) rekomendasi menggunakan 169 kueri uji.

TopK	MAP	Precision@k	Recall@k	F1-score@k
1	0.9112	0.9172	0.1214	0.1868

3	0.8874	0.8935	0.3030	0.3835
5	0.8869	0.8734	0.4446	0.5010
10	**0.9044**	**0.7269**	**0.7077**	**0.6225**
Default Nilai Evaluasi Terbaik @K =10				

Tabel 4.6 Evaluasi *Mean average precision, precision,recall* dan *f1-score*

Mean Average Precision (MAP): Sistem mencapai nilai MAP@10 yang sangat tinggi yaitu 0.9044. Hal ini membuktikan bahwa sistem sangat akurat dalam meranking item relevan pada posisi teratas daftar rekomendasi.

Precision vs *Recall*: Terjadi *trade-off* yang wajar seiring penambahan panjang daftar (@K). Pada @K=1, presisi sangat tinggi (0.9172) namun *recall* rendah, sementara pada @K=10, *recall* meningkat tajam menjadi 0.7077 karena lebih banyak item relevan yang berhasil ditemukan oleh sistem.

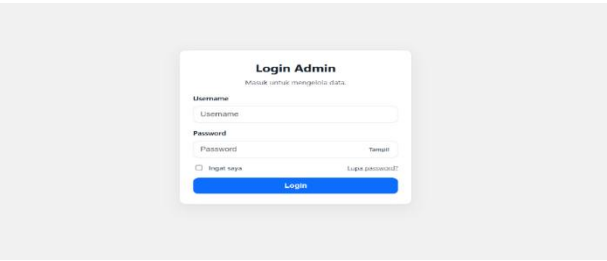
F1-Score: Kinerja sistem mencapai keseimbangan optimal pada @K=10 dengan nilai F1-score sebesar 0.6225. Titik ini merupakan rekomendasi praktis terbaik bagi pengguna untuk mendapatkan hasil yang akurat sekaligus komprehensif.

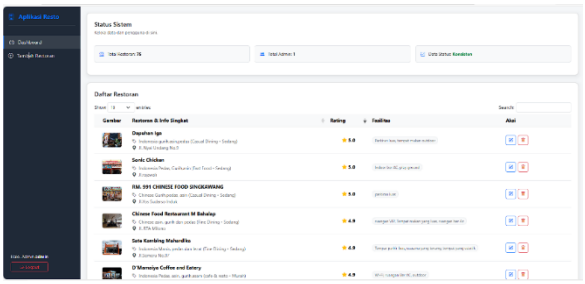
4.3. Implementasi antarmuka (*User Interface)*

Sistem diimplementasikan dalam bentuk aplikasi berbasis website. Halaman utama (*homepage*) menyediakan fitur pencarian gabungan di mana pengguna dapat memasukkan kueri kompleks (menu, rasa, dan lokasi). Hasil rekomendasi ditampilkan dalam format kartu yang memuat persentase kecocokan secara transparan. Selain itu, halaman detail dilengkapi dengan peta interaktif Google Maps untuk memudahkan navigasi langsung ke lokasi restoran.

4.3.1. Admin

Halaman admin berisikan halaman login, dashboard statistik (total user, total restoran), halaman *Manage Restoran.*

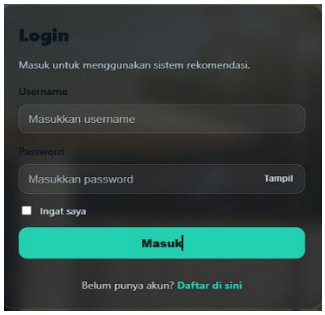


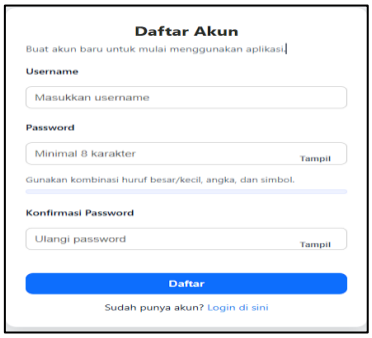


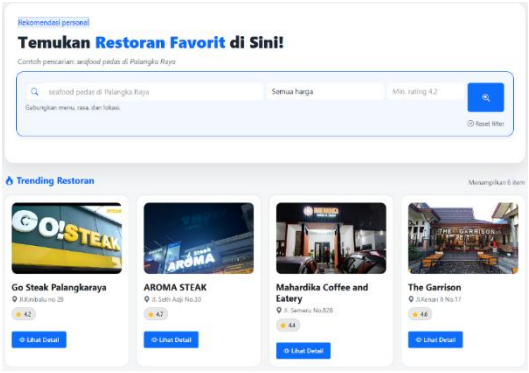
4.3.2. Pengguna

Halaman pengguna mencakup beranda (*homepage*), halaman awal yang menampilkan restoran dengan rating tertinggi, halaman login, halaman register dan halaman mencari restoran sesuai preferensi pengguna.









5. KESIMPULAN

Penelitian ini berhasil mengimplementasikan sistem rekomendasi restoran berbasis CBF yang akurat bagi masyarakat Palangka Raya. Integrasi *Dual Similarity Measurement* terbukti tangguh terhadap variasi ejaan dan kesalahan pengetikan. Evaluasi menunjukkan bahwa sistem mencapai performa optimal pada @K=10 dengan tingkat akurasi yang tinggi.

Saran untuk pengembangan selanjutnya adalah menguji metode *Hybrid Filtering* dengan mengintegrasikan *Collaborative Filtering* untuk mempertimbangkan ulasan dan interaksi antar pengguna, serta mengembangkan platform ke aplikasi *mobile* guna meningkatkan aksesibilitas.

6. DAFTAR PUSTAKA

[1] T. Ridwansyah, B. Subartini, and S. Sylviani, "Penerapan Metode Content-Based Filtering pada Sistem Rekomendasi," *Math. Sci. Appl. J.*, vol. 4, no. 2, pp. 70–77, 2024.

[2] A. Zakharia, A. D. Ulhaq, A. B. Suryono, N. C. Nugroho, D. F. Hafith, and D. N. Andrade Gusmao, "Sistem Rekomendasi Film Indonesia Menggunakan Metode Content-Based Filtering," *Log. J. Ilmu Komput. dan Pendidik.*, vol. 2, no. 4, pp. 671–678, 2024, [Online]. Available: https://journal.mediapublikasi.id/index.php/logic

[3] I. Ilhamsyah, S. Rahmayuda, D. M. Midyanti, and S. Martha, "Pemodelan Sistem Rekomendasi Restoran berdasarkan Preferensi Pengguna dengan Pendekatan Content-Based Filtering," *J. Edukasi dan Penelit. Inform.*, vol. 10, no. 1, p. 154, 2024, doi: 10.26418/jp.v10i1.74008.

[4] M. Leonor and L. Pereira, "Emotion-based Image Recommendation System," 2025.

[5] P. Nastiti, "Penerapan Metode Content Based Filtering Dalam Implementasi Sistem Rekomendasi Tanaman Pangan," *Teknika*, vol. 8, no. 1, pp. 1–10, 2019, doi: 10.34148/teknika.v8i1.139.

[6] S. Bag, S. K. Kumar, and M. K. Tiwari, "Ilmu Informasi Pembuatan rekomendasi yang efisien menggunakan kemiripan Jaccard yang relevan," vol. 483, pp. 53–64, 2019.

[7] J. M. Azri Saputra, L. M. Huizen, and D. B. Arianto, "Sistem Rekomendasi Film pada Platform Streaming Menggunakan Metode Content-Based Filtering," *J. Transform.*, vol. 22, no. 1, pp. 10–21, 2024, doi: 10.26623/transformatika.v22i1.7041.

[8] D. R. KUSNADI and N. HASTI, "Analisis Dan Perancangan Aplikasi Reservasi Dan Order Menu Berbasis Web Pada Restoran Bebek Van Java," *Maj. Ilm. UNIKOM*, vol. 14, no. 1, pp. 129–140, 2016, doi: 10.34010/miu.v14i1.167.

[9] N. I. P. Zen Munawar, Rustiyana1, Yudi Herdiana2, "Jurnal Teknologi Informasi dan Komunikasi Vol. 10 No. 1, Juni 2021 25," vol. 10, no. 1, pp. 25–34, 2021.

[10] S. Sunardi, A. Yudhana, and I. A. Mukaromah, "Implementasi Deteksi Plagiarisme Menggunakan Metode N-Gram Dan Jaccard Similarity Terhadap Algoritma Winnowing,"

Transmisi, vol. 20, no. 3, p. 105, 2018, doi: 10.14710/transmisi.20.3.105-110.

[11] M. Habibi, "Implementation of Cosine Similarity in an automatic classifier for comments," *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 3, no. 2, p. 110, 2019, doi: 10.14421/jiska.2018.32-05.

[12] H. Aditya, Ardiansyah, Sidik, and W. Gata, "Analisis Sentimen Penggunaan Twitter Terhadap Penggunaan Cairan Desinfektan Menggunakan Metode Term Frequency-Inverse Document Frequency Dan Support Vector Machine," *J. Ilmu-ilmu Inform. dan Manaj.*, vol. 14, no. 2, pp. 167–174, 2020, [Online]. Available: https://media.neliti.com/media/publications/414201-none-8fa7a6de.pdf

[13] M. Fachrurrozi and A. A. Manik, "Perbaikan Ejaan Kata pada Dokumen Bahasa Indonesia dengan Metode Cosine Similarity," *Jur. Tek. Inform. Univ. Sriwij. Kampus Unsri Indralaya Ogan Ilir*, no. October 2015, 2016.

[14] H. Manning, C. D., Raghavan, P., & Schütze, "Introduction to Modern Information Retrieval (2nd edition)," *Libr. Rev.*, vol. 53, no. 9, pp. 462–463, 2004, doi: 10.1108/00242530410565256.

[15] A. Riyanto, A., & Azis, "Implementation of the Jaccard Similarity Algorithm on Answer Type Description," *IJIIS Int. J. Informatics Inf. Syst.*, vol. 5, no. 2, pp. 76–83, 2022, doi: 10.47738/ijiis.v5i2.130.

[16] W. G. S. Parwita*1 and Edi Winarko2, "Hybrid Recommendation System," vol. 9, no. 2, 2015.

[17] T. T. Prabowo, "Efektivitas Sistem Temu Kembali Informasi Perpustakaan Digital Institut Seni Indonesia (ISI) Yogyakarta dalam Tinjauan Recall dan Precision," *Media Pustak.*, vol. 28, no. 1, pp. 37–48, 2021, doi: 10.37014/medpus.v28i1.1087.

[18] M. F. Ilmi, P. P. Adikara, and Indriati, "Pencarian Dokumen Skripsi menggunakan BM25 dan Faceted Search berdasarkan Kata Kunci Abstrak (Studi Kasus: Universitas Muhammadiyah Sidoarjo)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 9, pp. 4175–4180, 2022, [Online]. Available: http://j-ptiik.ub.ac.id

[19] I. G. N. A. Jayarana, I. G. W. Darma, I. W. A. Juliantara, and I. M. A. W. Putra, "Jurnal SUTASOMA | 61 STUDY LITERATUR INFORMATION RETRIEVAL MODEL: TEKNIK DAN APLIKASI," *J. Sutasoma*, vol. 3, no. 1, pp. 61–69, 2025.

[20] A. Rasida Hanum, E. Sakti Pramukantoro, and D. Primanita Kartikasari, "Studi Perbandingan Kinerja TF-IDF dan IndoBERT untuk Rekomendasi Resep Berdasarkan Ketersediaan Bahan Makanan Berbasis Website," vol. 9, no. 10, pp. 1–8, 2025.