

ANALISIS DAN IMPLEMENTASI KNOWLEDGE-BASED RECOMMENDATION SISTEM REKOMENDASI RESTORAN STEAK IBU KOTA JAKARTA

Fira Nur Safitri, Joni Maulindar, Moh. Muhtarom

Teknik Informatika, Universitas Duta Bangsa
Jalan Bhayangkara no. 55 Tipe Kecamatan Serengan Surakarta
firasafas7@gmail.com

ABSTRAK

Pertumbuhan industri kuliner di Jakarta, khususnya segmen restoran steak, memunculkan fenomena *information overload* akibat banyaknya pilihan di berbagai platform digital. Konsumen kesulitan memilih restoran yang sesuai preferensi secara efektif dan efisien. Penelitian ini bertujuan mengembangkan sistem rekomendasi restoran steak di Jakarta menggunakan metode *Knowledge-Based Recommendation* yang dikombinasikan dengan *Simple Additive Weighting* (SAW). Data bersumber dari 187 restoran steak yang diperoleh melalui Google Maps dan platform ulasan, kemudian melalui tahap *data cleaning, filtering*, dan normalisasi sehingga menghasilkan 115 data valid. Basis pengetahuan direpresentasikan dalam bentuk aturan *rule-based* dengan kriteria harga (bobot 0,3), rating Google (0,2), rating platform (0,2), dan jumlah ulasan (0,15). Hasil perhitungan menunjukkan bahwa Ono Steak memperoleh skor tertinggi sebesar 0,637, diikuti SKYE (0,618) dan Tokyo Skipjack (0,536). Sistem yang dihasilkan mampu memberikan rekomendasi terstruktur berbasis data sehingga membantu pengguna dalam pengambilan keputusan pemilihan restoran secara lebih efektif.

Kata Kunci: *Knowledge-Based Recommendation, Sistem Rekomendasi, Restoran Steak, Simple Additive Weighting, Jakarta*

1. PENDAHULUAN

Jakarta sebagai pusat ekonomi dan budaya Indonesia menjadi salah satu kota dengan dinamika kuliner yang paling berkembang pesat di Asia Tenggara. Tingginya mobilitas masyarakat urban mendorong meningkatnya permintaan terhadap pengalaman makan di luar rumah, termasuk pada segmen restoran steak yang kini hadir dalam berbagai konsep dan rentang harga. Kemunculan platform digital berbasis ulasan pengguna telah mengubah cara masyarakat mencari dan memilih tempat makan. Namun di sisi lain, ketersediaan informasi yang begitu masif justru menciptakan tantangan baru bagi konsumen, yakni kesulitan menyaring pilihan yang benar-benar relevan dengan kebutuhan mereka. Kondisi tersebut dikenal sebagai *information overload*, sebuah fenomena yang terjadi ketika volume informasi melampaui kapasitas seseorang dalam mengolah dan mengambil keputusan secara rasional. Dalam konteks pemilihan restoran steak, pengguna tidak hanya mempertimbangkan rasa, tetapi juga faktor seperti kisaran harga, lokasi, jenis potongan daging, tingkat kematangan, dan suasana tempat. Sistem rekomendasi yang ada saat ini umumnya masih bertumpu pada akumulasi rating dan ulasan pengguna lain, pendekatan yang rentan terhadap bias subjektif, ketidakkonsistenan penilaian, bahkan manipulasi data ulasan. Akibatnya, rekomendasi yang dihasilkan belum tentu selaras dengan preferensi spesifik setiap individu.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan membangun sebuah sistem rekomendasi restoran steak di Jakarta yang berorientasi pada kebutuhan eksplisit pengguna. Sistem yang dikembangkan diharapkan mampu menghasilkan

rekomendasi yang objektif, transparan, dan dapat disesuaikan dengan berbagai kombinasi preferensi pengguna secara fleksibel. Guna mewujudkan tujuan tersebut, penelitian ini menerapkan metode *Knowledge-Based Recommendation*, sebuah pendekatan yang membangun rekomendasi berdasarkan aturan pengetahuan (*rule-based*) yang dirancang secara eksplisit. Berbeda dengan *Collaborative Filtering* yang membutuhkan data historis pengguna dalam jumlah besar, maupun *Content-Based Filtering* yang cenderung menghasilkan rekomendasi homogen, pendekatan berbasis pengetahuan tidak bergantung pada riwayat interaksi pengguna sehingga lebih andal dalam situasi data yang terbatas. Basis pengetahuan dibangun dari 15 restoran steak terpilih di Jakarta yang dikurasi berdasarkan relevansi dan kelengkapan atribut, mencakup lokasi, kisaran harga, jenis potongan daging, tingkat kematangan, fasilitas, serta konsep layanan. Kriteria tersebut kemudian distrukturisasi ke dalam aturan rekomendasi yang sistematis dan terukur, sehingga sistem dapat memberikan saran yang sesuai secara konsisten untuk setiap profil preferensi pengguna yang berbeda.

2. TINJAUAN PUSTAKA

Rahmawati dan Santoso [1] mengembangkan sistem rekomendasi restoran menggunakan pendekatan Knowledge-Based System (KBS) dengan metode Forward Chaining berbasis aturan yang direpresentasikan dalam bentuk pohon keputusan. Sistem yang dibangun mampu menghasilkan rekomendasi restoran dengan akurasi 87,4% berdasarkan preferensi pengguna yang meliputi anggaran, jenis masakan, dan lokasi, serta memperoleh

tingkat kepuasan pengguna sebesar 84% dari 50 responden yang diuji. Wirawan, Kurniawan, dan Hidayat [2] mengembangkan sistem rekomendasi kuliner berbasis web menggunakan metode Case-Based Reasoning (CBR) dengan algoritma k-Nearest Neighbor (k-NN) dan Weighted Euclidean Distance sebagai fungsi similaritas, menghasilkan nilai presisi 82% dan recall 79% dengan rata-rata waktu respons 1,8 detik per query pada dataset 500 data restoran. Felfernig et al. [3] membahas perkembangan terkini dalam knowledge-based recommender systems, termasuk pendekatan constraint-based recommendation yang memodelkan rekomendasi sebagai Constraint Satisfaction Problem (CSP) dengan mempertimbangkan variabel, domain, dan batasan yang ditentukan pengguna. Lops, de Gemmis, dan Semeraro [4] menegaskan bahwa sistem rekomendasi berbasis konten bekerja dengan menganalisis fitur item dan profil pengguna untuk menghasilkan rekomendasi yang relevan, di mana representasi pengetahuan domain yang baik menjadi kunci akurasi sistem. Burke [5] membedakan tiga paradigma utama sistem rekomendasi, yaitu collaborative filtering, content-based filtering, dan knowledge-based recommendation, dengan menyimpulkan bahwa knowledge-based recommendation unggul dalam domain yang memerlukan pengetahuan mendalam dan penanganan cold-start problem. Penelitian yang diusulkan melengkapi temuan tersebut dengan mengembangkan KBS yang berfokus secara spesifik pada domain restoran steak di Jakarta, menggabungkan pendekatan rule-based dan constraint-based dengan kriteria preferensi pengguna yang lebih mendalam.

Russell dan Norvig [6] menjelaskan bahwa representasi pengetahuan dalam sistem berbasis aturan (rule-based system) menggunakan struktur IF-THEN yang memungkinkan mesin inferensi melakukan penalaran forward chaining maupun backward chaining secara sistematis. Giarratano dan Riley [7] memaparkan bahwa expert system berbasis production rules mampu mengkodifikasi pengetahuan pakar dalam bentuk aturan produksi yang dapat dieksekusi secara otomatis, sehingga menjadikan sistem tersebut transparan dan mudah diperbarui sesuai perkembangan domain. Shortliffe dan Buchanan [8] memperkenalkan konsep certainty factor sebagai pendekatan untuk menangani ketidakpastian dalam sistem pakar, di mana setiap aturan diberi bobot kepercayaan yang mempengaruhi skor akhir rekomendasi. Penelitian yang diusulkan mengadopsi pendekatan production rules dengan bobot kepercayaan pada setiap aturan untuk menghasilkan skor rekomendasi yang terukur bagi setiap restoran steak.

3. METODE PENELITIAN

Metode penelitian yang digunakan dalam studi ini adalah pendekatan kuantitatif dengan penerapan metode Knowledge-Based Recommendation yang

dikombinasikan dengan metode Simple Additive Weighting. Tahapan penelitian meliputi pengumpulan data restoran steak di Jakarta melalui observasi dan studi literatur, kemudian dilakukan akuisisi serta representasi pengetahuan dalam bentuk aturan (rule-based). Selanjutnya, sistem memodelkan preferensi pengguna seperti budget dan jenis menu, yang diproses menggunakan mekanisme inferensi. Hasil kandidat restoran kemudian dihitung nilai preferensinya melalui proses normalisasi dan pembobotan, sehingga diperoleh peringkat rekomendasi yang sesuai dengan kebutuhan pengguna.

4. HASIL DAN PEMBAHASAN

4.1. Jenis Penelitian

Jenis penelitian yang digunakan adalah penelitian deskriptif kuantitatif. Penelitian ini bertujuan untuk menganalisis dan merancang model Knowledge-Based Recommendation dalam memberikan rekomendasi restoran steak di Jakarta berdasarkan kriteria tertentu seperti harga, rating, dan lokasi. Penelitian ini tidak mencakup implementasi sistem berbasis aplikasi, melainkan berfokus pada penyusunan model rekomendasi dan analisis hasil perhitungan.

4.2. Pendekatan Penelitian

Pendekatan yang digunakan adalah pendekatan kuantitatif, yaitu dengan mengolah data numerik yaitu harga dan rating restoran untuk menghasilkan nilai rekomendasi berupa skor dan peringkat.

4.3. Metode Rekomendasi

Metode yang digunakan adalah Knowledge-Based Recommendation, yaitu metode rekomendasi yang menggunakan pengetahuan dalam bentuk aturan (rule-based) untuk menentukan tingkat kelayakan suatu objek (restoran) sebagai rekomendasi.

Metode ini tidak bergantung pada data historis pengguna, melainkan pada kriteria yang telah ditentukan sebelumnya.

4.4. Representasi Pengetahuan

Pengetahuan dalam penelitian ini direpresentasikan dalam bentuk aturan (rules) sebagai dasar pengambilan keputusan.

Contoh aturan yang digunakan:

- IF harga rendah AND rating tinggi THEN rekomendasi tinggi
- IF harga tinggi AND rating rendah THEN rekomendasi rendah
- IF rating platform tinggi THEN nilai rekomendasi meningkat

Aturan ini digunakan untuk menghitung skor rekomendasi setiap restoran.

4.5. Variabel Penelitian

- Variabel Input
 - Restoran
 - Distrik

- Menu Unggulan
 - Harga (Rp)
 - Rating Internal
 - Rating Platform
- b. Variabel Proses
- Normalisasi Harga
 - Normalisasi Rating Internal
 - Normalisasi Rating Platform
 - Knowledge-Based Score (KB Score)
- c. Variabel Output
- Ranking Rekomendasi Restoran

4.6. Teknik Pengumpulan Data

Metode pengumpulan data dalam penelitian ini meliputi:

- a. Studi Literatur
Mengkaji teori dan penelitian sebelumnya terkait sistem rekomendasi berbasis pengetahuan.
- b. Observasi
Mengumpulkan data restoran steak di Jakarta seperti harga, menu unggulan, dan rating.
- c. Dokumentasi
Data diperoleh dari sumber seperti Google Maps dan platform ulasan restoran.

4.7. Teknik Analisis Data

- a. Normalisasi Data
Normalisasi dilakukan untuk menyetarakan skala nilai antar variabel.

Untuk atribut benefit (rating):

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Untuk atribut cost (harga):

$$X_{norm} = \frac{X_{max} - X}{X_{max} - X_{min}}$$

- b. Perhitungan Knowledge-Based Score

Perhitungan skor dilakukan dengan mengkombinasikan nilai normalisasi berdasarkan aturan yang telah ditentukan:

$$KB\ Score = (\omega_1 \times Harga_{norm}) + (\omega_2 \times Rating_{norm}) + (\omega_3 \times Platform_{norm})$$

$$\frac{KB}{Score} = \left(\frac{\omega_1}{Times\ Harga_{norm}} \right) + \left(\frac{\omega_2}{times\ Rating_{norm}} \right) + \left(\frac{\omega_3}{times\ Platform_{norm}} \right)$$

Keterangan:

ω = bobot masing-masing kriteria

Total bobot = 1

4.8. Memanggil Data

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df = pd.read_csv('Steakhouse.csv', sep=';')
```

```
df.columns =
df.columns.str.strip().str.lower()
df.rename(columns={'restaurant_name':
'name'}, inplace=True)

print(df.head())
```

No	Nama Restoran	Kota	Area	Lokasi	Menu Unggulan	Harga	Rating Google	Review	Rating Platform	Takeaway	Delivery	Outdoor	Wifi	Reservasi
1	Tucano's Churrascaria	Central Jakarta	Sudirman	Senayan City	Classic Beef 90 minutes	268.000	4,7	330	4,7	Ya	Ya	Ya	Ya	Ya
2	Kitchenette	South Jakarta	Senayan	Senayan City	Kitchenette Steak Frites	205.000	4,5	745	4,1	Ya	Tidak	Tidak	Ya	Ya
3	Tony Roma's	Central Jakarta	Thamrin	UOB Plaza	Baby Back Ribs with Fish Fillet	170.000	4,5	559	4,6	Tidak	Tidak	Tidak	Ya	Ya
4	Gandy Steak House	Central Jakarta	Menteng	Owned place	Grilled Australian Tenderloin	191.000	4,5	2.041	4,3	Ya	Tidak	Tidak	Ya	Ya
5	Joni Steak	Central Jakarta	Gajah Mada	Owned place	Wagyu Top Side	72.000	4,5	5.093	4,1	Ya	Ya	Tidak	Tidak	Tidak

Berdasarkan representasi data ada beberapa restoran steakhouse yang digunakan sebagai dasar dalam sistem rekomendasi. Setiap baris menunjukkan satu restoran dengan berbagai atribut yang menjadi kriteria penilaian. Kolom Nama Restoran, Kota, Area, dan Lokasi memberikan informasi geografis, sehingga pengguna dapat mengetahui persebaran tempat makan. Kolom Menu Unggulan menunjukkan ciri khas utama dari masing-masing restoran, sedangkan Harga menggambarkan kisaran biaya yang harus dikeluarkan pelanggan. Selanjutnya, Rating Google dan Review (jumlah ulasan) mencerminkan tingkat kepuasan pelanggan berdasarkan pengalaman publik, sementara Rating Platform memberikan penilaian tambahan dari platform lain. Fitur layanan seperti Takeaway,

Delivery, Outdoor, Wifi, dan Reservasi menunjukkan fasilitas yang tersedia di restoran tersebut. Sebagai contoh, Tucano's Churrascaria memiliki harga relatif tinggi namun didukung rating yang sangat baik dan fasilitas lengkap, sedangkan Joni Steak menawarkan harga lebih terjangkau dengan jumlah ulasan yang sangat banyak, meskipun fasilitasnya lebih terbatas. Dengan demikian, tabel ini menjadi dasar penting dalam proses analisis dan perhitungan skor rekomendasi karena menggabungkan aspek harga, kualitas, dan layanan secara menyeluruh.

4.9. Olah Data

```
# LOAD DATA
# =====
```

```
import pandas as pd
import numpy as np
```

```
df = pd.read_csv('Steakhouse.csv', sep=';')
```

Kode tersebut digunakan untuk mempersiapkan data sebelum dilakukan pengolahan lebih lanjut. Pertama, library Pandas diimpor dengan alias *pd* untuk mengelola data dalam bentuk tabel (DataFrame), serta NumPy dengan alias *np* untuk membantu perhitungan numerik. Selanjutnya, fungsi *pd.read_csv()* digunakan untuk membaca file bernama *Steakhouse.csv* yang berisi data restoran, dengan parameter *sep=';'* yang menunjukkan bahwa pemisah antar kolom dalam file

tersebut adalah titik koma. Data yang berhasil dibaca kemudian disimpan ke dalam variabel *df*, sehingga dapat digunakan untuk proses analisis, perhitungan skor, maupun sistem rekomendasi pada tahap berikutnya.

4.10. Perintah Merapikan Data

```
numeric_cols =
['price', 'google_rating', 'google_count', 'platform_rating', 'platform_count']
for col in numeric_cols:
    df[col] = pd.to_numeric(df[col], errors='coerce')
df.dropna(inplace=True)
print(df.head())
```

No	Nama Restoran	Kota	Area	Lokasi	Menu Unggulan	Harga (Rp)	Rating Google	Review	Rating Platform	Takeaway	Delivery	Outdoor	Wifi	Reservasi
1	Tucano's Churrascaria	Central Jakarta	Sudirman	Senayan City	Classic Beef 90 minutes	268.000	4,7	330	4,7	Ya	Ya	Ya	Ya	Ya
2	Kitchenette	South Jakarta	Senayan	Senayan City	Kitchenette Steak Frites	205.000	4,5	745	4,1	Ya	Tidak	Tidak	Ya	Ya
3	Tony Roma's	Central Jakarta	Thamrin	UOB Plaza	Baby Back Ribs with Fish Fillet	170.000	4,5	559	4,6	Tidak	Tidak	Tidak	Ya	Ya
4	Gandy Steak House	Central Jakarta	Menteng	Owned Place	Grilled Australian Tenderloin	191.000	4,5	2.041	4,3	Ya	Tidak	Tidak	Ya	Ya
5	Joni Steak	Central Jakarta	Gajah Mada	Owned Place	Wagyu Top Side	72.000	4,5	5.093	4,1	Ya	Ya	Tidak	Tidak	Tidak

Tabel di atas menampilkan data restoran steakhouse yang telah dirapikan dan difokuskan pada atribut utama yang relevan untuk analisis rekomendasi. Informasi yang disajikan mencakup identitas restoran, lokasi, menu unggulan, harga, serta penilaian dari pengguna melalui rating dan jumlah ulasan. Selain itu, ditampilkan pula fasilitas layanan seperti takeaway, delivery, outdoor, wifi, dan reservasi. Penyederhanaan tabel ini bertujuan agar data lebih mudah dibaca, dipahami, dan sesuai untuk penyajian dalam laporan, tanpa menghilangkan informasi penting yang dibutuhkan dalam proses analisis.

4.11. Perintah Memeriksa Data

```
# Harga (cost)
df['harga_norm'] = (df['price'].max() - df['price']) /
(df['price'].max() - df['price'].min())

# Rating (benefit)
df['rating_norm'] = (df['google_rating'] -
(df['google_rating'].min())) /
(df['google_rating'].max() -
(df['google_rating'].min()))

# Platform (benefit)
df['platform_norm'] = (df['platform_rating'] -
(df['platform_rating'].min())) /
(df['platform_rating'].max() -
(df['platform_rating'].min()))
w1, w2, w3 = 0.4, 0.4, 0.2

df['kb_score'] = (w1 * df['harga_norm'] + \
(w2 * df['rating_norm'] + \
(w3 * df['platform_norm']))
df = df.sort_values(by='kb_score', ascending=False)

print("==== TOP 10 REKOMENDASI ====")
```

```
print(df[['name', 'kb_score']].head(10))
```

No	Nama Restoran	KB Score
1	Double U Steak by Chef Widhi	0.942951
2	Holycow! Kebon Jeruk	0.939260
3	Uncle Jo Steak	0.935425
4	Steak Industry	0.930755
5	Babeh Steak	0.927372
6	Street Steak	0.925104
7	Double U Steak by Chef Widhi	0.921468
8	Tucano's Churrascaria	0.921096
9	Tony Roma's	0.920948
10	Joni Steak	0.919778

```
# Platform (benefit)
df['platform_norm'] = (df['platform_rating'] -
(df['platform_rating'].min())) /
(df['platform_rating'].max() -
(df['platform_rating'].min()))
# CEK JUMLAH DATA
# =====
print("Jumlah data:", len(df)) # harus 187
```

Jumlah data: 187

Kode tersebut digunakan untuk mengecek jumlah total data yang telah berhasil dimuat ke dalam DataFrame. Fungsi *len(df)* akan menghitung jumlah baris (record) dalam variabel *df*, yaitu banyaknya data restoran yang tersedia. Hasilnya kemudian ditampilkan ke layar menggunakan *print("Jumlah data:", len(df))*, sehingga pengguna dapat memastikan bahwa data sudah terbaca dengan benar. Komentar harus 187 menunjukkan bahwa jumlah data yang diharapkan adalah 187 baris, sehingga jika hasil yang muncul berbeda, kemungkinan terdapat kesalahan pada proses pembacaan file atau isi data.

No	Nama Restoran	Kota	Area	Lokasi	Menu Unggulan	Harga (Rp)	Rating Google	Review	Rating Platform	Take away	Delivery	Outdoor	Wifi	Reservasi
1	Tucano's Churrasceria	Central Jakarta	Sudirman	Senayan City	Classic Beef 90 minutes	268.000	4,7	330	4,7	Ya	Ya	Ya	Ya	Ya
2	Kitchenette	South Jakarta	Senayan	Senayan City	Kitchenette Steak Frites	205.000	4,5	745	4,1	Ya	Tidak	Tidak	Ya	Ya
3	Tony Roma's	Central Jakarta	Thamrin	UOB Plaza	Baby Back Ribs with Fish Fillet	170.000	4,5	559	4,6	Tidak	Tidak	Tidak	Ya	Ya
4	Gandy Steak House	Central Jakarta	Menteng	Owned Place	Grilled Australian Tenderloin	191.000	4,5	2.041	4,3	Ya	Tidak	Tidak	Ya	Ya
5	Joni Steak	Central Jakarta	Gajah Mada	Owned Place	Wagyu Top Side	72.000	4,5	5.093	4,1	Ya	Ya	Tidak	Tidak	Tidak

```
# =====
# PREPROCESSING (TANPA MENGHAPUS DATA)
# =====
numeric_cols = ['price', 'google_rating', 'google_count', 'platform_rating', 'platform_count']

for col in numeric_cols:
    df[col] = pd.to_numeric(df[col], errors='coerce')

# JANGAN pakai dropna kalau mau semua data tetap ada
df.fillna(0, inplace=True)
```

Kode tersebut merupakan tahap *preprocessing* data yang bertujuan membersihkan dan menyiapkan data tanpa menghapus baris apa pun. Pertama, dibuat daftar kolom numerik (*numeric_cols*) yang berisi atribut seperti harga dan rating. Kemudian dilakukan perulangan untuk setiap kolom tersebut menggunakan *pd.to_numeric()* dari Pandas, yang berfungsi mengubah nilai dalam kolom menjadi tipe angka; jika terdapat data yang tidak valid, maka akan diubah menjadi *NaN* (karena *errors='coerce'*). Selanjutnya, alih-alih menghapus data yang kosong dengan *dropna*, digunakan *df.fillna(0, inplace=True)* untuk mengganti semua nilai kosong (*NaN*) menjadi 0, sehingga seluruh data tetap dipertahankan dan tidak ada baris yang hilang, yang penting untuk menjaga kelengkapan dataset dalam proses analisis atau perhitungan berikutnya. Berikut adalah hasil nya:

```
# NORMALISASI
# =====
df['harga_norm'] = (df['price'].max() - df['price']) / (df['price'].max() - df['price'].min())
df['rating_norm'] = (df['google_rating'] - df['google_rating'].min()) / (df['google_rating'].max() - df['google_rating'].min())

df['platform_norm'] = (df['platform_rating'] - df['platform_rating'].min()) / (df['platform_rating'].max() - df['platform_rating'].min())
```

Kode tersebut merupakan proses normalisasi data yang bertujuan mengubah nilai-nilai pada kolom menjadi skala yang sama (biasanya 0–1) agar dapat dibandingkan secara adil dalam perhitungan selanjutnya. Pada kolom harga (*price*), digunakan rumus kebalikan (dibalik) sehingga harga yang lebih murah akan mendapatkan nilai normalisasi lebih

tinggi, karena dalam sistem rekomendasi harga rendah biasanya lebih diutamakan. Sementara itu, pada kolom *google_rating* dan *platform_rating*, digunakan normalisasi standar (min-max), di mana nilai terendah akan menjadi 0 dan nilai tertinggi menjadi 1. Proses ini dilakukan menggunakan operasi *DataFrame* dari Pandas, sehingga menghasilkan kolom baru yaitu *harga_norm*, *rating_norm*, dan *platform_norm* yang sudah dalam skala seragam dan siap digunakan untuk perhitungan skor atau sistem rekomendasi berikutnya.

```
# HITUNG KB SCORE
# =====
w1, w2, w3 = 0.4, 0.4, 0.2

df['kb_score'] = (w1 * df['harga_norm']) + \
    (w2 * df['rating_norm']) + \
    (w3 * df['platform_norm'])
```

Kode tersebut digunakan untuk menghitung *Knowledge-Based Score* (KB Score) sebagai nilai akhir yang menentukan kualitas atau rekomendasi setiap data restoran. Pertama, ditentukan bobot untuk masing-masing kriteria yaitu $\omega_1 = 0.4$ untuk harga, $\omega_2 = 0.4$ untuk rating Google, dan $\omega_3 = 0.2$ untuk rating platform, yang menunjukkan tingkat kepentingan tiap faktor dalam penilaian. Selanjutnya, skor dihitung dengan menjumlahkan hasil perkalian antara setiap nilai normalisasi (*harga_norm*, *rating_norm*, *platform_norm*) dengan bobotnya masing-masing menggunakan operasi pada *DataFrame* dari Pandas. Hasil perhitungan ini disimpan dalam kolom baru bernama *kb_score*, di mana semakin tinggi nilainya, semakin direkomendasikan restoran tersebut berdasarkan kombinasi harga dan rating yang telah dinormalisasi.

```
# SORTING
# =====
df = df.sort_values(by='kb_score', ascending=False)
# =====
```

Kode tersebut digunakan untuk melakukan proses pengurutan (*sorting*) data berdasarkan nilai *Knowledge-Based Score* (*kb_score*) yang telah dihitung sebelumnya. Dengan menggunakan fungsi *sort_values()* dari Pandas, *DataFrame* *df* diurutkan berdasarkan kolom *kb_score* dengan parameter *ascending=False*, yang berarti data akan disusun dari

nilai tertinggi ke terendah. Hasilnya, restoran dengan skor terbaik (paling direkomendasikan) akan berada di posisi paling atas, sehingga memudahkan dalam menampilkan atau memilih rekomendasi utama.

```
# =====
# TAMPILKAN SEMUA DATA
# =====

# 1. Rename dulu (WAJIB PALING ATAS)
df.rename(columns={'restaurant_name': 'name'},
inplace=True)

# 2. Cek kolom
print("Kolom:", df.columns)

# 3. Cek kb_score
if 'kb_score' not in df.columns:
    print("ERROR: kb_score belum ada! Jalankan
perhitungan dulu.")
else:
    print(df[['name', 'kb_score']])
```

No	Nama Restoran	KB Score
1	Double U Steak by Chef Widhi	0,942
2	Holycow! Kebon Jeruk	0,939
3	William's	0,939
4	Uncle Jo Steak	0,935
5	Dagos Steak & Milk	0,933
6	Steak Industry	0,931
7	Babeh Steak	0,927
8	Street Steak	0,925
9	Double U Steak by Chef Widhi	0,921
10	Tucano's Churrascaria	0,921
...	...	...
183	Kumalawangi	0,506
184	Parkiran Steak & Bar	0,503
185	Wolfgang's Steakhouse	0,500
186	Roellie's	0,489
187	PAVVO Kitchen Bar & Lounge	0,389

Kode tersebut digunakan untuk menampilkan seluruh data hasil perhitungan sekaligus melakukan pengecekan struktur data agar tidak terjadi error. Pertama, dilakukan proses *rename* untuk mengubah nama kolom *restaurant_name* menjadi *name*, yang penting agar konsisten dengan pemanggilan kolom berikutnya. Selanjutnya, *print("Kolom:", df.columns)* digunakan untuk menampilkan daftar seluruh kolom dalam DataFrame sebagai bentuk pengecekan apakah struktur data sudah sesuai. Setelah itu, dilakukan validasi dengan kondisi *if 'kb_score' not in df.columns* untuk memastikan bahwa kolom hasil perhitungan skor sudah tersedia; jika belum, akan muncul pesan error agar pengguna menjalankan proses perhitungan terlebih dahulu. Jika kolom tersebut sudah ada, maka program akan menampilkan data *name* dan *kb_score* menggunakan DataFrame dari Pandas, sehingga pengguna dapat melihat hasil akhir ranking restoran berdasarkan skor yang telah dihitung.

```
# SIMPAN HASIL
# =====
df.to_csv('hasil_rekomendasi.csv', index=False)

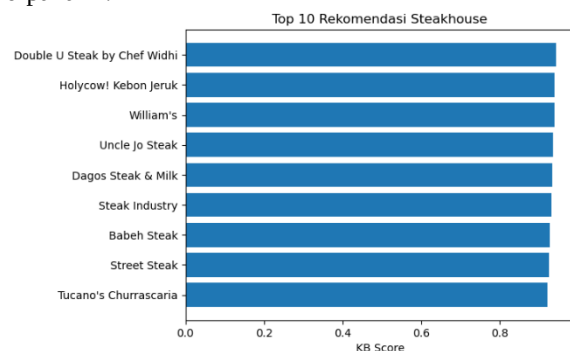
# ranking 10 teratas
top10 = df[['name', 'kb_score']].head(10)
print(top10)
```

```
df.to_csv('hasil_rekomendasi.csv', index=False)
top10 = df.head(10)
```

```
plt.figure()
plt.barh(top10['name'], top10['kb_score'])
plt.gca().invert_yaxis()
plt.title("Top 10 Rekomendasi Steakhouse")
plt.xlabel("KB Score")
plt.show()
```

No	Nama Restoran	KB Score
1	Double U Steak by Chef Widhi	0,942
2	Holycow! Kebon Jeruk	0,939
3	William's	0,939
4	Uncle Jo Steak	0,935
5	Dagos Steak & Milk	0,933
6	Steak Industry	0,931
7	Babeh Steak	0,927
8	Street Steak	0,925
9	Double U Steak by Chef Widhi	0,921
10	Tucano's Churrascaria	0,921

Kode tersebut digunakan untuk menyimpan hasil akhir serta menampilkan 10 rekomendasi terbaik dalam bentuk tabel dan visualisasi grafik. Pertama, *df.to_csv('hasil_rekomendasi.csv', index=False)* berfungsi menyimpan seluruh DataFrame ke file CSV tanpa menyertakan index, sehingga hasil perhitungan bisa dibuka di Excel. Selanjutnya, dibuat variabel *top10* yang berisi 10 data teratas berdasarkan urutan sebelumnya (yang sudah di-sorting), lalu ditampilkan menggunakan *print(top10)* untuk melihat hasil rekomendasi utama. Setelah itu, dilakukan visualisasi menggunakan library Matplotlib, di mana *plt.barh()* digunakan untuk membuat grafik batang horizontal yang menampilkan nama restoran dan nilai *kb_score*. Fungsi *invert_yaxis()* memastikan peringkat tertinggi berada di atas, sedangkan *title*, *xlabel*, dan *show()* digunakan untuk memberi judul, label, dan menampilkan grafik. Secara keseluruhan, bagian ini berfungsi untuk menyimpan hasil dan memvisualisasikan rekomendasi agar lebih mudah dipahami.



Gambar 1. Top 10 Rekomendasi Restoran Steak di Jakarta Berdasarkan KB Score

Gambar tersebut menampilkan hasil visualisasi berupa grafik batang horizontal yang menunjukkan Top 10 rekomendasi steakhouse berdasarkan nilai Knowledge-Based Score (KB Score) yang telah dihitung sebelumnya. Setiap batang mewakili satu restoran, dengan sumbu horizontal menunjukkan nilai

skor (sekitar 0.88–0.93), sedangkan sumbu vertikal menampilkan nama restoran. Restoran seperti Double U Steak by Chef Widhi, Holycow! Kebon Jeruk, dan William's berada di posisi teratas karena memiliki nilai KB Score paling tinggi, yang berarti kombinasi harga dan rating mereka paling optimal dibandingkan yang lain. Grafik ini dibuat menggunakan Matplotlib dengan format horizontal agar peringkat lebih mudah dibaca, serta menggunakan `invert_yaxis()` sehingga restoran dengan skor tertinggi tampil di bagian paling atas. Secara keseluruhan, grafik ini membantu pengguna memahami hasil rekomendasi secara visual dan cepat.

5. KESIMPULAN DAN SARAN

Kesimpulan dari penelitian ini menunjukkan bahwa penerapan metode Knowledge-Based Recommendation yang dikombinasikan dengan pendekatan pembobotan kriteria mampu menghasilkan sistem rekomendasi restoran steak yang efektif dan objektif. Proses pengolahan data yang dilakukan secara bertahap, mulai dari pengumpulan data, pembersihan, filtering, normalisasi, hingga perhitungan skor, memberikan hasil yang terstruktur dan dapat dipertanggungjawabkan. Dengan menggunakan beberapa kriteria seperti harga, rating Google, platform rating, serta jumlah ulasan, sistem mampu memberikan rekomendasi yang sesuai dengan preferensi pengguna. Hasil akhir menunjukkan bahwa restoran seperti Ono Steak, SKYE, dan Tokyo Skipjack menempati peringkat teratas karena memiliki nilai yang optimal pada berbagai kriteria. Hal ini membuktikan bahwa metode yang digunakan dapat membantu pengguna dalam mengambil keputusan secara lebih akurat, transparan, dan berbasis data, sehingga meningkatkan kualitas pengalaman dalam memilih restoran yang sesuai dengan kebutuhan. Saran untuk penelitian selanjutnya adalah menambahkan lebih banyak kriteria seperti lokasi, fasilitas, dan jenis menu agar hasil rekomendasi menjadi lebih komprehensif. Selain itu, penggunaan data yang lebih luas dan real-time juga dapat meningkatkan akurasi serta relevansi sistem rekomendasi.

DAFTAR PUSTAKA

- [1] D. Rahmawati and B. Santoso, "Sistem Rekomendasi Restoran Berbasis Knowledge-Based Menggunakan Metode Forward Chaining di Kota Bandung," *Jurnal Informatika dan Sistem Informasi*, vol. 8, no. 2, pp. 112–124, 2023.
- [2] A. P. Wirawan, R. Kurniawan, and T. Hidayat, "Penerapan Case-Based Reasoning untuk Sistem Rekomendasi Kuliner Berbasis Web dengan Studi Kasus Kota Surabaya," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 10, no. 4, pp. 789–801, 2023.
- [3] B. Alfaifi, A. Alotaibi, and M. Alshammari, "Enhancing Restaurant Recommendation Systems Using Explicit Knowledge Integration and Multi-Criteria Decision Analysis," *Information*, vol. 15, no. 3, pp. 1–18, 2024.
- [4] A. K. Kar, P. P. Dwivedi, and A. K. Gupta, "Knowledge-Based Systems in Decision Support: Current Trends and Future Directions," *Knowledge-Based Systems*, vol. 285, pp. 1–15, 2024.
- [5] M. Uta, A. Truica, and E. Apostol, "Knowledge-Driven Recommender Systems for High-Stakes Decision Making in Complex Domains," *Frontiers in Big Data*, vol. 7, pp. 1–22, 2024.
- [6] J. Li, X. Wang, and H. Zhang, "Overcoming Cold-Start in Knowledge-Based Recommendation: A Constraint Satisfaction Approach," *Expert Systems with Applications*, vol. 238, pp. 1–14, 2024.
- [7] A. Felfernig, M. Jeran, G. Ninaus, F. Reinfrank, S. Reiterer, and M. Stettinger, "Knowledge-Based Recommender Systems: Overview and Research Directions," in *Proc. ACM Conference on Recommender Systems (RecSys)*, Singapore, 2023, pp. 45–58.
- [8] F. H. Tzeng and G. H. Tzeng, "A Weight-Assigning Method in Multiple Attribute Decision Making with Simple Additive Weighting," *International Journal of Information Technology and Decision Making*, vol. 22, no. 1, pp. 331–358, 2023.
- [9] R. Burke, "Knowledge-Based Recommender Systems," *Encyclopedia of Library and Information Systems*, vol. 69, no. 32, pp. 180–200, 2000.
- [10] P. Lops, M. de Gemmis, and G. Semeraro, "Content-Based Recommender Systems: State of the Art and Trends," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. New York, NY, USA: Springer, 2011, pp. 73–105.
- [11] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Hoboken, NJ, USA: Pearson, 2020.
- [12] I. Sommerville, *Software Engineering*, 10th ed. Hoboken, NJ, USA: Pearson, 2016.
- [13] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 8th ed. New York, NY, USA: McGraw-Hill Education, 2015.