

SISTEM REKOMENDASI RESTORAN HALAL BERBASIS COLLABORATIVE FILTERING DAN AUTOENCODER



Disusun Oleh:

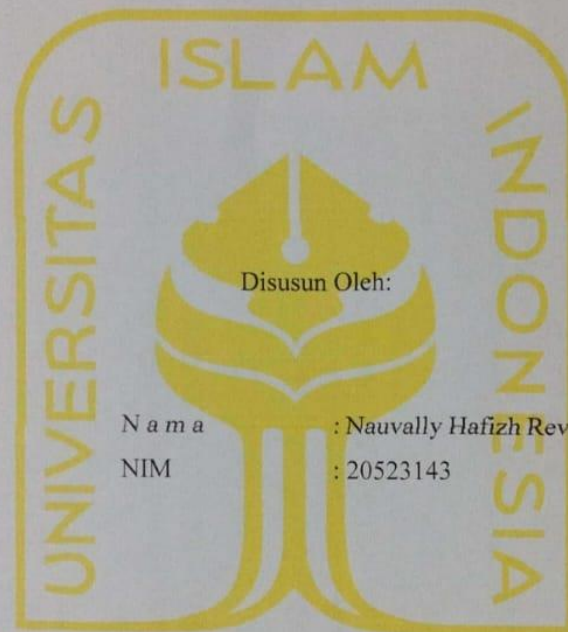
Nama : Nauvally Hafizh Revansa
NIM : 20523143

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2024**

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**SISTEM REKOMENDASI RESTORAN HALAL BERBASIS
COLLABORATIVE FILTERING DAN AUTOENCODER**

TUGAS AKHIR



N a m a : Nauvally Hafizh Revansa
NIM : 20523143

البعثة الإسلامية الأندلسية

Yogyakarta, 14 Oktober 2024

Pembimbing,

(Dr. Syarif Hidayat, S.Kom, M.I.T)

HALAMAN PENGESAHAN DOSEN PENGUJI

SISTEM REKOMENDASI RESTORAN HALAL BERBASIS COLLABORATIVE FILTERING DAN AUTOENCODER

TUGAS AKHIR

Telah dipertahankan di depan sidang pengujian sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 14 Oktober 2024

Tim Penguji

Dr. Syarif Hidayat, S.Kom., M.I.T.

Anggota 1

Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D

Anggota 2

Dr. Yudi Prayudi, S.Si., M.Kom.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana
Fakultas Teknologi Industri
Universitas Islam Indonesia

(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Nauvally Hafizh Revansa

NIM : 20523143

Tugas akhir dengan judul:

SISTEM REKOMENDASI RESTORAN HALAL BERBASIS COLLABORATIVE FILTERING DAN AUTOENCODER

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 14 Oktober 2024



(Nauvally Hafizh Revansa)

HALAMAN PERSEMBAHAN

Segala puji bagi Allah *Subhanahu wa ta'ala* atas segala nikmat yang telah Dia anugerahkan. Hanya kepada-Nya kita memanjatkan puji dan syukur, memohon pertolongan, serta meminta petunjuk. Dengan izin-Nya pula, kita masih diberi nikmat iman dan Islam hingga saat ini. Shalawat serta salam senantiasa tercurah kepada Nabi Muhammad *Shallallahu 'alaihi wa sallam* yang telah menuntun kita kepada cahaya Islam.

Penulis ingin menyampaikan rasa terima kasih yang mendalam kepada kedua orangtua penulis, yang telah memberikan kesempatan dan bimbingan luar biasa sehingga penulis bisa menuntut ilmu hingga tahap ini. Ucapan terima kasih juga penulis sampaikan kepada Bapak Dr. Syarif Hidayat, S.Kom, M.I.T., atas bimbingan yang diberikan dalam menyelesaikan laporan tugas akhir ini. Terima kasih juga kepada semua teman-teman dan orang-orang di sekitar penulis, terutama kakak penulis yang telah memberikan dukungan dan doa dalam proses penyelesaian tugas akhir ini.

Demikian yang dapat penulis sampaikan, penulis sadar bahwa naskah skripsi ini tidak sempurna, karena kesempurnaan hanya milik Allah *Subhanahu wa ta'ala* semata. Namun, penulis telah berjuang sekuat tenaga untuk menyelesaikan naskah skripsi ini. Semoga naskah skripsi ini dapat bermanfaat untuk banyak orang kedepannya.

HALAMAN MOTO

“Bacalah, dengan (menyebut) nama Tuhanmu yang menciptakan.” (QS. Al-Alaq: 1)

A little progress each day adds up to big results

“Sebaik-baik manusia adalah yang bermanfaat bagi manusia lain.” (HR. Ahmad & ath-Thabrani)

Every step forward, no matter how small, builds the foundation for success

“Belajar tanpa berpikir itu tidaklah berguna, tapi berpikir tanpa belajar itu sangatlah berbahaya!” (Ir. Soekarno)

KATA PENGANTAR

Assalamu 'alaikum Warahmatullahi Wabarakatuh

Alhamdulillahirabbil 'alamin, segala puji bagi Allah *Subhanahu wata'ala*, yang telah melimpahkan berbagai nikmat, baik nikmat iman, Islam, dan kesehatan, sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik. Shalawat dan salam semoga senantiasa tercurah kepada Nabi Muhammad Sallallahu 'alaihi wasallam, yang telah menyebarkan ajaran Islam, membawa umat dari zaman *jahiliyyah* menuju era penuh ilmu dan cahaya Islam.

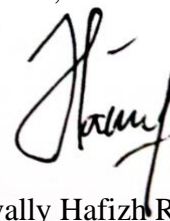
Dalam penyusunan skripsi ini, penulis menemui berbagai tantangan, namun berkat bimbingan, dukungan, dan doa dari banyak pihak, penulis berhasil mengatasinya. Oleh karena itu, penulis ingin mengucapkan rasa terima kasih dan apresiasi kepada:

1. Kedua orangtua yang selalu memberikan semangat, dukungan dan doanya selama ini.
2. DThomas Hatta Fudholi, S.T., M.Eng., Ph.D. selaku ketua Program Studi Informatika UII yang telah memberikan izin dalam penulisan skripsi ini.
3. Sheila Nurul Huda, S.Kom., M.Cs. selaku pembimbing akademik yang telah memberikan bimbingan dan arahan selama berkuliah di UII.
4. Dr. Syarif Hidayat, S.Kom., M.I.T. selaku pembimbing yang telah memberikan bimbingan dan arahan selama penelitian ini berlangsung.
5. Bapak dan Ibu dosen Jurusan Informatika yang telah memberikan ilmu yang sangat bermanfaat, semoga bapak dan ibu selalu diberikan keberkahan oleh Allah SWT.
6. Teman-teman dan orang di sekitar penulis yang selalu memberikan support dan semangatnya.

Semoga setiap pihak yang telah memberikan doa, dukungan, dan bimbingan selama proses ini senantiasa diberi kemudahan serta perlindungan oleh Allah *Subhanahu Wata'ala*. Penulis juga memohon maaf atas segala kekurangan yang masih ada dalam penelitian ini. Harapannya, semoga karya ini bisa memberikan manfaat serta menjadi sumber pengetahuan untuk memperluas wawasan kita semua.

Wassalamu 'alaikum Warahmatullahi Wabarakatuh

Yogyakarta, 14 Oktober 2024



(Nauvally Hafizh Revansa)

SARI

Sistem rekomendasi memainkan peran penting dalam membantu pengguna membuat keputusan, termasuk dalam memilih restoran. Namun, banyaknya pilihan restoran seringkali membuat masyarakat kebingungan, terutama bagi masyarakat Muslim yang membutuhkan kepastian terkait status halal dari restoran yang dikunjungi. Kekhawatiran ini muncul terutama ketika mereka berada di wilayah mayoritas non-Muslim, di mana restoran halal dan non-halal seringkali bercampur. Penelitian ini bertujuan untuk membangun sistem rekomendasi restoran halal dengan menggunakan metode *hybrid filtering* yang menggabungkan *collaborative filtering* dan *autoencoder*, serta penggunaan *cosine similarity* untuk menghitung kemiripan antar pengguna.

Penelitian ini dimulai dengan pengumpulan dataset yang terdiri dari 7.030 data restoran yang diperoleh melalui proses *scraping* dari situs *Google Maps*. Data ini kemudian diproses melalui tahapan pemfilteran, perhitungan *cosine similarity*, dan penerapan metode *collaborative filtering* serta *autoencoder*. Sistem yang dihasilkan diharapkan dapat memberikan rekomendasi yang lebih akurat dan relevan berdasarkan preferensi dan minat pengguna.

Evaluasi sistem yang dilakukan menggunakan metrik *Root Mean Square Error (RMSE)* menghasilkan nilai rendah. Namun, penggunaan teknik *overlap* dalam pembagian data menyebabkan nilai RMSE tampak lebih baik dari performa sebenarnya karena data pengujian menjadi terlalu mirip dengan data pelatihan. Meskipun belum teruji pada data yang beragam, sistem tetap dapat menghasilkan rekomendasi restoran kepada pengguna, sehingga dapat membantu pengguna, khususnya masyarakat Muslim, dalam membuat keputusan yang lebih tepat terkait pilihan restoran.

Kata kunci: sistem rekomendasi, restoran halal, *collaborative filtering*, *autoencoder*

GLOSARIUM

Cold Start	masalah ketika sistem rekomendasi kesulitan memberikan saran karena pengguna atau item baru tidak memiliki data historis.
Decoder	bagian model yang mengubah representasi tersembunyi (latent code) kembali menjadi data asli.
Encoder	komponen yang mengubah data asli menjadi representasi tersembunyi yang lebih ringkas.
Latent Code	representasi ringkas dari data yang dihasilkan oleh encoder dalam autoencoder.
Overlap	metode di mana sebagian data digunakan di set pelatihan dan set pengujian untuk memastikan evaluasi yang lebih menyeluruh.
Preprocessing	tahap persiapan data mentah sebelum digunakan dalam model, seperti normalisasi dan pembersihan data.
Scraping	pengambilan data secara otomatis dari situs web menggunakan skrip atau program.
Vektor	struktur data berbentuk deret angka yang merepresentasikan objek dalam ruang dimensi tertentu.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING	ii
HALAMAN PENGESAHAN DOSEN PENGUJI	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI	viii
GLOSARIUM	ix
DAFTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
1.6 Sistematika Penulisan	5
BAB II LANDASAN TEORI	7
2.1 Dasar Teori	7
2.1.1 Sistem Rekomendasi	7
2.1.2 Collaborative Filtering	8
2.1.3 Deep Learning	9
2.1.4 Autoencoder	10
2.1.5 Restoran/Tempat Makan	11
2.1.6 Web Scraping	11
2.1.7 Cosine Similarity	11
2.2 Kajian Literatur	12
BAB III METODOLOGI	16
3.1 Analisis Masalah	16
3.2 Pengumpulan Data	17
3.2.1 Teknik Pengumpulan Data	18
3.2.2 Pengambilan Fitur Restoran	19
3.3 Preprocessing	20
3.3.1 Combining Data	20
3.3.2 Data Cleaning	20
3.3.3 Filtering Data	21
3.3.4 Splitting Data	22
3.4 Pemodelan	22
3.4.1 Collaborative Filtering	23
3.4.2 Autoencoder	25
3.5 Rekomendasi	27
3.6 Evaluasi	27
BAB IV HASIL DAN PEMBAHASAN	29
4.1 Pengumpulan Dataset	29
4.1.1 Data Scraping	30

4.2	Preprocessing	37
4.2.1	Combining Data	37
4.2.2	Data Cleaning	38
4.2.3	Filtering	39
4.3	Splitting Data	41
4.4	Cosine Similarity	42
4.5	Autoencoder	43
4.6	Rekomendasi	47
4.7	Evaluasi	49
4.8	Hasil Penelitian	50
4.8.1	Hasil Uji Program	51
4.8.2	Pembahasan	52
4.8.3	Kelebihan dan Kekurangan	53
	BAB V KESIMPULAN DAN SARAN	54
5.1	Kesimpulan	54
5.2	Saran	54
	DAFTAR PUSTAKA	55
	LAMPIRAN	59

DAFTAR TABEL

Tabel 3.1 Matriks Item-user.....	23
Tabel 4.1 Perbandingan <i>rating</i> aktual dan <i>rating</i> prediksi	51

DAFTAR GAMBAR

Gambar 2.1 Struktur Autoencoder, sumber (Lopez Pinaya et al., 2019).....	10
Gambar 3.1 Langkah langkah penelitian	16
Gambar 4.1 Data restoran	29
Gambar 4.2 Kode <i>scraping</i> untuk akses situs <i>Google Maps</i>	31
Gambar 4.3 Kode <i>scraping</i> untuk mencari restoran	32
Gambar 4.4 Kode <i>scraping</i> pengambilan data restoran.....	33
Gambar 4.5 Kode <i>scraping</i> untuk menggulir halaman <i>review</i>	35
Gambar 4.6 Kode <i>scraping</i> untuk mengambil data pada halaman <i>review</i>	35
Gambar 4.7 Kode <i>scraping</i> untuk mengambil data pada halaman <i>about</i>	36
Gambar 4.8 Kode untuk proses menyimpan data	37
Gambar 4.9 Kode pada proses <i>combining data</i>	38
Gambar 4.10 Kode untuk proses <i>data cleaning</i>	39
Gambar 4.11 Kode <i>filtering</i> makanan halal	39
Gambar 4.12 Kode <i>filtering</i> minuman beralkohol	40
Gambar 4.13 Kode <i>filtering</i> berdasarkan kata kunci	41
Gambar 4.14 Kode untuk proses <i>splitting data</i>	42
Gambar 4.15 Kode untuk menghitung nilai <i>cosine similarity</i>	43
Gambar 4.16 Kode untuk proses <i>autoencoder</i>	45
Gambar 4.17 Grafik <i>loss</i> selama pelatihan model <i>autoencoder</i>	46
Gambar 4.18 Kode rekonstruksi matriks <i>user-item</i>	47
Gambar 4.19 Kode untuk mencari rekomendasi.....	48
Gambar 4.20 Daftar rekomendasi 10 restoran halal teratas	49
Gambar 4.21 Kode evaluasi menggunakan RMSE.....	50
Gambar 4.22 Distribusi <i>rating</i> aktual dan <i>rating</i> prediksi.....	52

BAB I

PENDAHULUAN

1.1 Latar Belakang

Halal merujuk pada segala sesuatu yang diizinkan oleh hukum syariah Islam untuk dilakukan, digunakan, atau diusahakan, asalkan tidak ada ikatan atau elemen yang melarang atau berpotensi membahayakan, dengan catatan bahwa proses memperolehnya dilakukan dengan mematuhi aturan-aturan yang berlaku, dan ini tidak melibatkan transaksi atau muamalah yang dilarang oleh Islam (Ali, 2016). Restoran halal merupakan tempat makan yang menyediakan makanan dan minuman yang diolah sesuai dengan aturan dan ketentuan dalam Islam. Makanan yang dianggap halal harus memenuhi beberapa kriteria, termasuk asal bahan, kandungan, serta kebersihan prosesnya.

Namun, ketika umat Muslim berada di lingkungan yang mayoritas penduduknya non-Muslim, mereka sering kali kesulitan menemukan restoran yang benar-benar hanya menyajikan makanan dan minuman halal. Kekhawatiran ini muncul karena ada kemungkinan restoran yang tidak mencantumkan status halal yang jelas. Selain itu, masyarakat Muslim semakin sadar akan pentingnya memastikan kehalalan makanan yang dikonsumsi dengan mencari informasi daftar rekomendasi restoran halal di internet. Akan tetapi, pencarian restoran halal melalui mesin pencari di internet sering kali tidak cukup memadai. Pencarian tersebut tidak selalu menghasilkan rekomendasi yang sesuai dengan minat atau preferensi pengguna, melainkan hanya berdasarkan kata kunci yang belum tentu sesuai dengan minat pengguna. Oleh karena itu, dibutuhkan sebuah sistem rekomendasi yang mampu memberikan rekomendasi yang lebih relevan dan sesuai dengan preferensi pengguna.

Seiring dengan tingginya kesadaran masyarakat Muslim terhadap produk makanan halal, terjadi peningkatan dalam perilaku pencarian informasi mandiri mengenai produk halal. Penelitian oleh (Yusoff & Adzharuddin, 2017) menunjukkan bahwa faktor kesadaran ini memiliki korelasi yang signifikan dengan perilaku pencarian informasi makanan halal di kalangan masyarakat Muslim. Situasi ini menggambarkan pentingnya penyediaan informasi yang lebih akurat dan tepat sasaran mengenai restoran halal, terutama di lingkungan yang mayoritasnya non-Muslim. Dengan adanya sistem rekomendasi yang difokuskan pada restoran halal, pengguna dapat menerima rekomendasi berdasarkan minat dan preferensi pribadi, sehingga memungkinkan mereka untuk membuat keputusan yang lebih baik dan sesuai dengan kebutuhan mereka.

Selain itu, tantangan lain muncul karena banyak restoran yang tidak mencantumkan label halal secara jelas, sehingga menimbulkan ketidakpastian mengenai kehalalan bahan-bahan yang digunakan. Situasi ini diperburuk oleh contoh kasus nyata di mana pelanggan Muslim tertipu oleh restoran yang menyajikan makanan yang ternyata mengandung bahan yang tidak halal (Budi, 2023; Mutiah, 2023). Sementara itu, penelitian oleh (Abdullah et al., 2019) mengungkapkan bahwa masih terdapat makanan yang diberi label halal meskipun statusnya belum jelas.

Disamping itu, Rasulullah Shalallahu ‘alaihi wasallam bersabda dalam salah sebuah hadis yang diriwayatkan oleh Jabir bin Abdillah, "Siapa yang beriman kepada Allah dan hari akhir, janganlah dia duduk di hidangan makanan yang di sana disediakan khamr." (HR Ahmad 15027, Turmudzi 2801). Hadis tersebut menegaskan bahwa seorang Muslim hendaknya menghindari restoran yang menjual minuman beralkohol. Namun, faktanya masih banyak restoran halal yang menyajikan alkohol, terutama tempat-tempat yang mayoritas masyarakatnya non-Muslim (Arbar, 2024). Hal itu akan membuat umat Muslim merasa kesulitan jika ingin berwisata ke tempat dengan mayoritas non-Muslim.

Dari penelitian tersebut, dapat diketahui bahwa masyarakat Muslim sebenarnya membutuhkan sebuah sistem yang dapat membantu mereka dalam mencari informasi mengenai restoran halal. Tak hanya itu, sistem juga harus mampu memberikan rekomendasi berdasarkan preferensi dari pengguna (Mahadi et al., 2018). Dalam kasus ini, sistem rekomendasi restoran halal yang akan dikembangkan dapat memainkan peran penting dalam membantu umat Muslim untuk menghindari tempat-tempat yang tidak terjamin kehalalannya, memberikan kenyamanan dan kepastian saat memilih tempat makan.

Terdapat pertimbangan utama dalam mengembangkan fitur utama dari sistem, yaitu memberikan rekomendasi secara tepat dan akurat, karena pilihan yang terlalu banyak dapat mempengaruhi kemampuan masyarakat dalam mengambil keputusan (Gerasimou & Papi, 2018). Selain itu, masyarakat cenderung menghindari atau menunda pemilihan ketika dihadapkan pada banyak alternatif yang beragam. Oleh karena itu, sistem rekomendasi hadir untuk membantu masyarakat dalam membuat pilihan yang lebih baik (Gunantohadi & Crysdiyan, 2022).

Pengembangan sistem rekomendasi untuk restoran yang dikembangkan oleh (Nurhayati & Widayani, 2021) menggunakan algoritma *collaborative filtering* sebagai algoritma modelnya. Belum ditemukan penelitian sejenis yang menggabungkan algoritma *collaborative filtering* dengan algoritma lain seperti *content-based filtering* maupun *deep learning*.

Menggabungkan algoritma *collaborative filtering* dengan metode *deep learning* seperti *autoencoder* dapat meningkatkan akurasi dalam proses rekomendasi.

Metode *Collaborative filtering* memungkinkan sistem rekomendasi untuk memberikan rekomendasi kepada pengguna berdasarkan informasi dan preferensi dari pengguna lain yang memiliki kesamaan dalam sejarah interaksi atau perilaku. Metode ini beroperasi dengan cara mencari kesamaan antara pengguna atau item, dan kemudian memberikan rekomendasi berdasarkan kesamaan tersebut. Beberapa kekurangan dari metode *collaborative filtering* adalah metode ini memerlukan data yang cukup besar untuk menghasilkan rekomendasi yang akurat dan relevan. Selain itu, metode ini juga rentan terhadap masalah *cold start*, yaitu ketika tidak ada atau sedikit data tentang pengguna atau item baru, sehingga sistem rekomendasi tidak dapat memberikan rekomendasi kepada pengguna dengan semestinya (Venil et al., 2020).

Metode pengembangan sistem rekomendasi yang lain adalah *content-based filtering* dan *deep learning*. Metode *content-based filtering* memungkinkan pengguna mendapat rekomendasi berdasarkan apa yang pengguna pilih sebelumnya. Metode ini tidak melibatkan pilihan pengguna yang lain, sehingga tidak memerlukan banyak data. Kekurangan dari metode ini adalah *over-specialization*, yaitu ketika rekomendasi menjadi terlalu spesifik dan monoton, sehingga mengurangi kepuasan dan variasi pengguna (B.Thorat et al., 2015). Sementara itu, *deep learning* merupakan bagian dari *machine learning* yang mengambil inspirasi dari struktur otak manusia. Struktur ini dikenal sebagai *Artificial Neural Networks* (ANN) atau yang sering disebut Jaringan Saraf Tiruan (Alfarizi et al., 2023). Penggunaan *deep learning* dalam sistem rekomendasi dapat memberikan dampak yang signifikan terhadap tingkat akurasi dan relevansi dari sistem tersebut (Singhal et al., 2017).

Penelitian tentang sistem rekomendasi restoran telah banyak dilakukan, namun sistem rekomendasi khusus untuk restoran halal masih jarang dijumpai. Penelitian sebelumnya tentang sistem rekomendasi restoran yang dilakukan oleh (Nurhayati & Widayani, 2021) hanya menggunakan metode *collaborative filtering*. Oleh karena itu, peneliti tertarik untuk mengembangkan sistem rekomendasi restoran halal dengan pendekatan *deep-learning based collaborative filtering*, yaitu dengan menggunakan metode *collaborative filtering* yang diperkuat dengan metode *deep learning*, sehingga rekomendasi yang diperoleh akan jauh lebih akurat dan relevan (Martins et al., 2020). Sistem yang akan dikembangkan adalah sistem rekomendasi yang menggunakan metode *collaborative filtering* dan *autoencoder* sebagai dasarnya dan didukung oleh *cosine similarity* sebagai metode untuk mengukur kemiripan antar

restoran, sehingga harapan peneliti sistem ini kedepannya akan dapat bermanfaat bagi masyarakat dan supaya sistem ini dapat menambah khazanah sistem rekomendasi restoran halal berbasis aplikasi kecerdasan buatan yang lebih mutakhir.

1.2 Rumusan Masalah

Meskipun terdapat banyak aplikasi yang menawarkan rekomendasi restoran, sistem rekomendasi restoran yang spesifik mengacu pada restoran halal masih sangatlah jarang. Sistem rekomendasi restoran halal yang dibuat selama ini masih menggunakan *collaborative filtering* saja dalam pengembangannya. Melalui pengembangan model kecerdasan buatan dengan menggabungkan *collaborative filtering* dengan *deep learning*, maka sangat memungkinkan sistem dapat memberikan rekomendasi restoran dengan lebih akurat dan relevan, sehingga bagaimanakah proses merancang, membangun, dan mengevaluasi model *deep collaborative filtering* pada sistem rekomendasi restoran halal?

1.3 Batasan Masalah

- a. Penelitian ini berfokus pada rekomendasi restoran, sehingga tidak akan mempertimbangkan aspek lain seperti kosmetik dan obat-obatan.
- b. Penelitian ini memanfaatkan dataset yang tersedia untuk pengembangan model. Keterbatasan data yang tersedia mungkin mempengaruhi kinerja sistem.
- c. Penelitian ini tidak akan mempertimbangkan aspek keberlanjutan dalam pemilihan restoran halal, seperti cara produksi makanan.
- d. Penelitian ini tidak akan memperhitungkan kendala hukum yang mungkin terkait dengan sertifikasi makanan halal atau peraturan yang berlaku di berbagai yurisdiksi.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk mengembangkan sebuah sistem rekomendasi restoran halal berbasis aplikasi kecerdasan buatan yang dapat membantu individu dalam mencari restoran halal yang sesuai dengan preferensi mereka. Sistem ini akan memanfaatkan teknik-teknik kecerdasan buatan untuk mengidentifikasi restoran halal, menganalisis preferensi pengguna, dan memberikan rekomendasi yang relevan.

1.5 Manfaat Penelitian

- a. Manfaat akademis

1. Penelitian ini dapat memberikan sumbangan baru dalam bidang kecerdasan buatan dan sistem rekomendasi, khususnya dalam konteks restoran halal. Ini akan memperkaya literatur akademis dalam bidang ini.
 2. Dapat memberikan peluang bagi mahasiswa dan peneliti muda untuk belajar tentang kecerdasan buatan dan aplikasinya dalam kehidupan sehari-hari, khususnya dalam konteks sistem rekomendasi.
- b. Manfaat praktis
1. Sistem rekomendasi restoran halal berbasis kecerdasan buatan dapat memberikan rekomendasi restoran yang lebih akurat dan sesuai dengan preferensi dan persyaratan halal pengguna. Ini dapat meningkatkan pengalaman kuliner mereka.
 2. Aplikasi ini dapat membantu pengguna dalam menemukan restoran, makanan, atau produk halal yang sesuai dengan lokasi dan keinginan mereka dan membuatnya lebih mudah untuk memenuhi kebutuhan makanan halal mereka.
 3. Aplikasi ini juga dapat membantu pelaku usaha halal untuk mempromosikan produk dan layanan mereka kepada pasar yang lebih luas, meningkatkan potensi pertumbuhan ekonomi dalam industri makanan halal.

1.6 Sistematika Penulisan

Dalam memberikan sebuah gambaran yang jelas mengenai pembahasan dalam pembuatan sistem rekomendasi restoran halal berbasis *collaborative filtering* dan *autoencoder*, berikut merupakan sistematika pembahasannya:

BAB I PENDAHULUAN

Bab ini menguraikan latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, metode penelitian, serta sistematika penulisan dari tugas akhir yang berfokus pada pengembangan sistem rekomendasi restoran halal menggunakan metode *collaborative filtering* dan *autoencoder*.

BAB II LANDASAN TEORI

Bab ini memberikan dasar teoritis yang mendukung proses pengembangan sistem. Pada bagian ini juga disajikan ringkasan penelitian-penelitian sebelumnya yang relevan dengan topik penelitian ini.

BAB III METODOLOGI PENELITIAN

Bab ini mencakup analisis kebutuhan yang diperlukan untuk mengembangkan sistem rekomendasi, serta menjelaskan metode pengumpulan data dan penerapan metode tersebut dalam sistem rekomendasi yang dikembangkan.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menjelaskan desain sistem yang telah dibangun serta hasil yang diperoleh setelah melewati proses penelitian. Selain itu, bagian ini juga berisi evaluasi kinerja dari sistem rekomendasi yang dihasilkan.

BAB V KESIMPULAN DAN SARAN

Bab terakhir ini menyajikan kesimpulan yang dapat diambil dari hasil penelitian, serta memberikan saran untuk pengembangan lebih lanjut yang dapat dilakukan dalam penelitian di masa mendatang.

BAB II

LANDASAN TEORI

2.1 Dasar Teori

2.1.1 Sistem Rekomendasi

Sistem rekomendasi adalah sebuah teknologi atau algoritma yang digunakan untuk merekomendasikan item, produk, atau konten kepada pengguna berdasarkan preferensi mereka. Sistem rekomendasi juga dapat membantu pengguna dalam membuat keputusan yang lebih baik dengan mengumpulkan informasi mengenai preferensi mereka dalam berbagai bidang, seperti produk dan layanan (García-Sánchez et al., 2020). Dengan adanya sistem rekomendasi, pengguna dapat memilih opsi terbaik dari berbagai pilihan yang ada sesuai dengan apa yang mereka butuhkan (Altan & Karasu, 2019). Banyak perusahaan terkenal di seluruh dunia, seperti Netflix, Amazon, dan YouTube, telah mengadopsi sistem rekomendasi dalam layanan mereka yang digunakan secara luas oleh pengguna internet. Tak hanya itu, sistem rekomendasi juga digunakan di banyak platform berita. Sistem ini berguna bagi pengguna dalam menemukan dan memilih berbagai item seperti film, buku, atau restoran dari beragam koleksi yang tersedia di situs web atau sumber informasi elektronik lainnya. Beberapa faktor dapat mempengaruhi keputusan sistem dalam memberikan rekomendasi kepada pengguna, diantaranya adalah perilaku pengguna, deskripsi item, dan juga preferensi serta kebiasaan dari kelompok pengguna yang serupa dalam menilai suatu item. Sistem rekomendasi telah terbukti sebagai alat efektif dalam mengelola informasi. Dengan kata lain, sistem rekomendasi berfokus pada karakteristik penyaringan informasi, yaitu menyaring produk atau item yang mungkin menarik bagi pengguna (Shao et al., 2021).

Sistem rekomendasi memperoleh informasi dari pengguna menggunakan berbagai macam metode. Namun, secara umum, terdapat tiga metode yang sering digunakan untuk membuat sistem rekomendasi. Metode tersebut diantaranya adalah *content-based filtering*, *collaborative filtering*, dan *hybrid method*. Metode *content-based filtering* bergantung pada informasi deskripsi item dan profil preferensi pengguna. Metode ini merekomendasikan item yang mirip dengan yang disukai oleh pengguna tertentu. Pada dasarnya, tindakan utama yang dilakukan oleh metode *content-based filtering* adalah mencocokkan informasi dasar pengguna seperti usia, jenis kelamin, lokasi, dan daftar item yang telah dinilai oleh pengguna di situs web yang tersimpan dalam akun mereka dengan item-item serupa yang memiliki spesifikasi yang sama, dengan tujuan merekomendasikan item baru yang sesuai dengan minat pengguna

tersebut. Sementara itu, metode *collaborative filtering* mengumpulkan dan menganalisis informasi perilaku pengguna dalam bentuk umpan balik, peringkat, preferensi, dan aktivitas mereka. Berdasarkan informasi ini, sistem kemudian memanfaatkan kemiripan di antara beberapa pengguna atau item untuk membuat rekomendasi yang sesuai. Adapun *hybrid method* adalah penggabungan antara metode *content-based filtering* dan metode *collaborative filtering* (Mohamed et al., 2019).

2.1.2 Collaborative Filtering

Collaborative filtering adalah salah satu pendekatan yang sangat populer dalam pengembangan sistem rekomendasi. Metode ini berfokus pada pengumpulan dan analisis informasi perilaku pengguna, seperti peringkat, ulasan, preferensi, atau aktivitas mereka dalam platform tertentu. Ide dasar di balik *collaborative filtering* adalah bahwa pengguna yang memiliki perilaku atau preferensi serupa dalam hal item (misalnya, produk atau konten) akan cenderung memiliki kesamaan dalam preferensi item lainnya. Oleh karena itu, sistem rekomendasi dapat memanfaatkan kemiripan ini untuk membuat rekomendasi yang personal dan relevan (Martins et al., 2020). Terdapat dua jenis utama *collaborative filtering*:

- a. *User-Based Collaborative Filtering*. Pada pendekatan ini, sistem mencari pengguna lain yang memiliki preferensi atau perilaku yang mirip dengan pengguna yang sedang diproses. Berdasarkan kesamaan ini, sistem kemudian memprediksi item yang belum dilihat atau dinilai oleh pengguna dan memberikan rekomendasi berdasarkan preferensi pengguna serupa. Misalnya, jika Pengguna A dan Pengguna B memiliki sejarah pembelian yang mirip, maka item yang disukai oleh Pengguna B tetapi belum dilihat oleh Pengguna A dapat direkomendasikan.
- b. *Item-Based Collaborative Filtering*. Dalam pendekatan ini, sistem fokus pada kesamaan antara item daripada pengguna. Sistem menganalisis seberapa sering item-item tertentu dikonsumsi bersama oleh pengguna yang berbeda. Ketika pengguna memilih atau menilai item tertentu, sistem dapat merekomendasikan item-item lain yang sering dipilih bersama dengan item tersebut. Misalnya, jika Pengguna A sering memilih film-film dengan genre yang sama, sistem dapat merekomendasikan film-film dengan genre yang serupa.

Meskipun *collaborative filtering* dapat menghubungkan item yang mirip antara pengguna dan mampu merekomendasikan item di luar preferensi pengguna, tidak dapat

dipungkiri bahwa metode ini akan kesulitan memberikan rekomendasi jika terdapat pengguna baru yang belum menentukan preferensi mereka (Mohamed et al., 2019).

2.1.3 Deep Learning

Deep learning adalah cabang dari pembelajaran mesin (*machine learning*) yang berfokus pada penggunaan jaringan saraf tiruan (*neural networks*) yang sangat kompleks untuk memahami dan memodelkan data (Chassagnon et al., 2020). *Deep learning* menggunakan arsitektur jaringan saraf yang terdiri dari banyak lapisan (disebut "*deep*" karena memiliki banyak lapisan) untuk mengotomatisasi ekstraksi fitur dari data mentah. Ini memungkinkan sistem *deep learning* untuk secara efektif memahami pola yang rumit dan abstrak dalam data, seperti gambar, suara, teks, dan banyak jenis data lainnya.

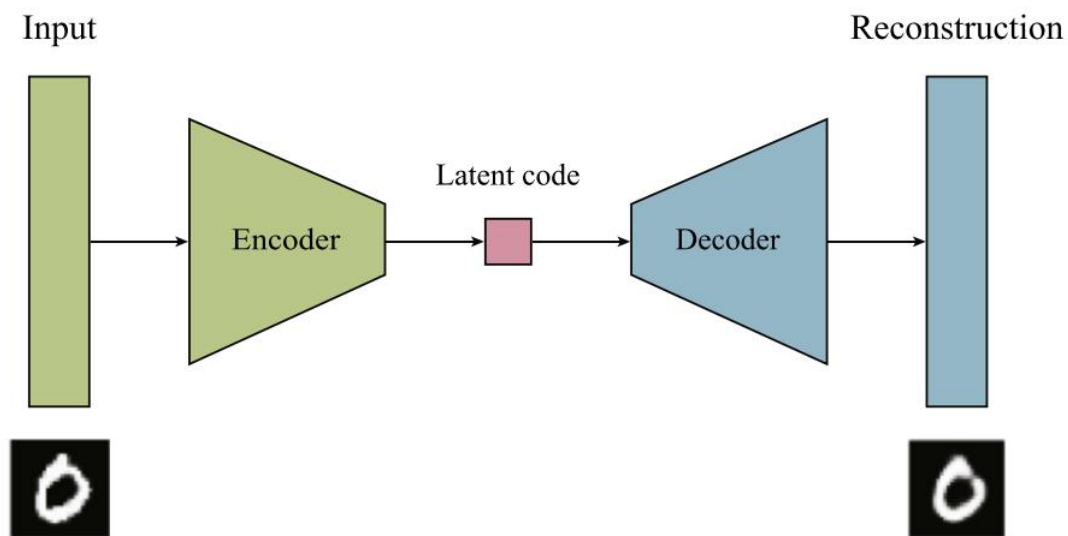
Konsep utama dalam *deep learning* adalah penggunaan jaringan saraf tiruan yang terdiri dari banyak neuron atau unit kecil yang bekerja secara paralel. Selama pelatihan, jaringan saraf ini mempelajari berbagai tingkat representasi dari data, dimulai dari fitur-fitur sederhana hingga fitur yang semakin kompleks. Ini memungkinkan *deep learning* untuk mengekstraksi fitur-fitur yang lebih tinggi, yang dapat digunakan untuk tugas-tugas seperti pengenalan gambar, pemrosesan bahasa alami, pengenalan suara, sistem rekomendasi, dan banyak lagi.

Deep learning telah mengalami perkembangan pesat dalam beberapa tahun terakhir dan telah menghasilkan kemajuan besar dalam berbagai bidang, termasuk pengenalan wajah, pengenalan suara, pemrosesan gambar, dan sebagainya. Ini juga digunakan secara luas dalam aplikasi teknologi canggih seperti kendaraan otonom, permainan video, dan aplikasi pencarian. *Deep learning* juga sudah merambah dalam bidang kedokteran. Sebagai contoh adalah sebuah sistem untuk mendiagnosis penyakit retinopati diabetik yang sangat sulit untuk didiagnosa oleh ahli manusia. *Deep learning* dapat menjadi alternatif solusi untuk memecahkan masalah tersebut (Dong et al., 2021).

Tidak diragukan lagi bahwa perkembangan *deep learning* yang begitu pesat juga mempengaruhi kemampuan manusia untuk mengembangkan model yang beragam. Seperti yang disebutkan dalam penelitian oleh (Dong et al., 2021), terdapat berbagai model *deep learning* yang dapat disesuaikan dengan kebutuhan kita. Diantaranya seperti *autoencoders*, *restricted boltzmann machines*, *deep belief networks*, *convolutional neural networks* (CNNs), dan masih banyak yang lain, yang semuanya memiliki karakteristik dan kemampuan yang berbeda. Sehingga, pemilihan model yang tepat bergantung pada karakteristik data dan tujuan analisis atau aplikasi yang ingin dikembangkan.

2.1.4 Autoencoder

Autoencoder atau *automatic encoder* adalah jenis arsitektur jaringan saraf buatan yang digunakan dalam pembelajaran mendalam (*deep learning*) untuk tugas-tugas seperti reduksi dimensi, *denoising*, rekonstruksi, atau ekstraksi fitur. *Autoencoder* telah digunakan dalam beragam aplikasi, termasuk segmentasi retakan pada gambar yang diambil dengan *smartphone*, pemodelan sistem atomistik untuk molekul dan material, prediksi struktur konektor otak, *person re-identification*, dan peningkatan hasil dari metode *collaborative filtering* dalam sistem rekomendasi. Arsitektur *autoencoder* dapat disesuaikan dengan berbagai fitur dan hiperparameter untuk mencapai berbagai tujuan, termasuk deteksi malware, pemrosesan bahasa alami, dan pengenalan gambar. *Autoencoder* terdiri dari dua bagian utama, yaitu *encoder* dan *decoder* (Lopez Pinaya et al., 2019). Struktur *autoencoder* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Struktur Autoencoder, sumber (Lopez Pinaya et al., 2019)

Berdasarkan gambar diatas, data input akan masuk ke dalam *encoder* untuk membuat representasi yang lebih sederhana dari input tersebut. Setelah melalui proses *encode*, data kemudian diproses di dalam *code layer/latent code* untuk mengambil fitur-fitur paling penting atau esensial dari data input, sebelum hasil representasi tersebut dikembalikan menjadi output yang sesuai dengan input awal oleh *decoder*.

2.1.5 Restoran/Tempat Makan

Tempat makan merujuk pada sebuah usaha kuliner yang bertujuan menyajikan berbagai hidangan kepada masyarakat, menyediakan tempat untuk dinikmati bersama, dan menetapkan tarif tertentu untuk makanan serta layanannya. Sebagian besar tempat makan umumnya menghadirkan makanan di lokasi fisik mereka, tetapi tidak jarang juga ada yang memberikan opsi layanan take-out atau pengiriman sebagai bentuk pelayanan tambahan kepada pelanggan mereka. Jenis tempat makan sangat beragam, mulai dari restoran formal hingga *cafe* yang santai, serta *food court* yang menawarkan berbagai pilihan kuliner. Inti dari tempat makan adalah menyediakan pengalaman makan yang memuaskan bagi pelanggan dengan berbagai jenis hidangan yang disajikan sesuai dengan preferensi dan gaya masing-masing tempat makan tersebut (Sahara & Baskoro, 2019).

2.1.6 Web Scraping

Web scraping atau *web crawling* mengacu pada langkah-langkah otomatis dalam mengekstraksi data dari situs web dengan menggunakan perangkat lunak khusus (Khder, 2021). Proses ini memungkinkan pengumpulan informasi secara efisien dari berbagai halaman web tanpa perlu intervensi manual. Teknik ini umumnya digunakan untuk mendapatkan data yang diperlukan dari berbagai sumber online secara cepat dan efektif. Selain itu, istilah lain seperti *web data extraction*, *web data scraping*, *web harvesting*, atau *screen scraping* juga merujuk pada proses ini. Web scraping, yang merupakan bagian dari data mining, bertujuan untuk mengambil data yang tidak terstruktur dari berbagai situs web dan mengkonversinya ke dalam format terstruktur, seperti spreadsheet, basis data, atau file CSV. Tujuan utama dari *web scraping* adalah untuk mengekstraksi informasi yang berasal dari sumber yang berbeda dan belum terstruktur, kemudian mengorganisasi data tersebut sehingga dapat dimengerti, seperti data penetapan harga barang, harga saham, laporan, informasi penetapan harga pasar, detail produk, dan informasi khusus lainnya.

2.1.7 Cosine Similarity

Cosine similarity adalah metode yang digunakan untuk mengukur sejauh mana dua vektor arah mendekati kesamaan atau keterkaitan satu sama lain. Dalam metode ini, dokumen atau teks dianggap sebagai vektor. Hal ini memungkinkan kita untuk mengukur sejauh mana kedua dokumen memiliki pola kata yang serupa, tanpa memperhatikan panjang dokumen atau frekuensi masing-masing kata. Metode *cosine similarity* ini umumnya diterapkan dalam bidang

analisis teks, pengelompokan dokumen, sistem rekomendasi, dan banyak lagi. *Cosine similarity* akan dihitung sebagai hasil perkalian titik dua vektor dibagi oleh magnitudo atau panjang dari masing-masing vektor (Mana & Sasipraba, 2021). *Cosine similarity* dapat dihitung dengan menggunakan persamaan (2.1).

$$sim(i, j) = \frac{i \cdot j}{\|i\| \cdot \|j\|} = \frac{\sum_u r_{ui} \cdot r_{uj}}{\sqrt{\sum_u r_{ui}^2} \cdot \sqrt{\sum_u r_{uj}^2}} \quad (2.1)$$

Dari persamaan diatas, i dan j masing-masing adalah dua vektor yang mewakili item atau dokumen. $i \cdot j$ adalah hasil dari perkalian titik (*dot product*) antara vektor i dan j , sedangkan $\|i\|$ dan $\|j\|$ masing-masing adalah magnitudo atau panjang dari vektor i dan j . Selain itu, \sum_u menyatakan penjumlahan atas semua *user* yang memberikan rating pada kedua item i dan j . Jika nilai *cosine similarity* antara dua vektor adalah 1, itu menunjukkan bahwa vektor-vektor tersebut memiliki arah yang identik dan dianggap sangat mirip. Sebaliknya, jika nilai *cosine similarity* mendekati 0, itu menunjukkan bahwa vektor-vektor tersebut memiliki arah yang berbeda dan dianggap kurang mirip (Mana & Sasipraba, 2021).

2.2 Kajian Literatur

Penelitian tentang sistem rekomendasi makanan halal sudah pernah dilakukan oleh beberapa peneliti. Penelitian pertama dilakukan oleh (Sihombing et al., 2020) pada tahun 2020. Pada penelitian tersebut, penulis menggunakan metode *social trust path* untuk membuat rekomendasi berdasarkan informasi kepercayaan antar *user* atau hubungan pertemanan. Metode *social trust path* akan memproses data input dengan menganalisis kemiripan antar pengguna berdasarkan dua aspek, yaitu kemiripan berdasarkan rating yang diberikan oleh pengguna dan kemiripan berdasarkan hubungan pertemanan. Data mengenai kepercayaan antar *user* dan rating akan digunakan sebagai faktor utama dalam memprediksi nilai pada matriks user-item.

Kemudian, pada tahun 2022, (Mahardika et al., 2022) melakukan penelitian tentang sistem rekomendasi pemilihan lokasi kuliner halal pada area kota Pekanbaru menggunakan metode *weighted product*. Penelitian tersebut menggunakan metode *weighted product* dengan menggabungkan beberapa kriteria yang diberi bobot. Kriteria tersebut diantaranya rata-rata

harga, waktu operasional, usia tahun berdiri, dan jarak. Kriteria-kriteria tersebut nantinya akan diranking untuk menghasilkan rekomendasi berdasarkan rating tertinggi. Fitur yang ada di aplikasi ini termasuk hasil rekomendasi, rute menuju lokasi, dan detail informasi lokasi. Sementara itu, hasil *usability testing* menunjukkan hasil yang cukup memuaskan dan keseluruhan atribut memiliki nilai di atas nilai 4.

Sementara itu, (Munir & Asqia, 2021) dalam penelitiannya menerapkan metode *Skyline Query* untuk memberikan rekomendasi pemilihan tempat makan berdasarkan preferensi para pengunjung. Data penelitian diambil dari 150 responden yang tersebar di tiga kota, yaitu kota Depok, kota Bogor, dan kota Tangerang. Penelitian ini berfokus pada empat aspek kriteria untuk menentukan preferensi tempat makan responden, yaitu kuliner, atmosfer, pelayanan, dan faktor kesehatan. Dari keempat aspek tersebut, aspek pelayanan menjadi pilihan utama responden dengan nilai 85,63%. Sedangkan aspek kuliner menjadi yang terendah dengan 67,82%. Penelitian ini juga menggunakan fitur GPS untuk menentukan lokasi tempat makan terdekat.

(Effendy et al., 2019) menggunakan metode Fuzzy Topsis dalam penelitiannya tentang sistem rekomendasi kuliner. Metode ini membantu pengguna dengan mengidentifikasi kriteria penting dalam memilih tempat makan. Kriteria-kriteria tersebut mencakup kenyamanan, variasi menu, harga, jarak, fasilitas parkir, fasilitas, pelayanan, area, kehalalan, dan jenis menu. Uji coba dan evaluasi sistem menunjukkan bahwa 75% output sistem sesuai dengan harapan, dan pengguna menilai fitur yang disediakan dalam sistem ini dengan skor 3.55 dari 4. Akan tetapi sistem masih perlu untuk dikembangkan lagi dengan penambahan fitur komunitas, sehingga semua pengguna dapat memberikan ulasan, komentar, dan peringkat destinasi kuliner.

Penelitian tentang rekomendasi tempat makan juga pernah dilakukan oleh (Fadhlullah et al., 2021) menggunakan *chatbot* berbasis aplikasi *mobile*. Penelitian ini menggunakan *Conversational Recommender Systems* (CRS) untuk membuat sistem rekomendasi berdasarkan preferensi pengguna. CRS adalah bagian dari *Knowledge-based Recommender Systems* yang dapat mengatasi *cold-start problem* pada metode *collaborative filtering*. Penelitian ini menggunakan data dari platform Traveloka.com dan PergiKuliner.com dengan jumlah 100 buah data. Dari data tersebut, didapatkan akurasi sebesar 40%. Kinerja rendah sistem disebabkan oleh keterbatasan basis pengetahuan yang tersedia. Oleh karenanya, penambahan

data mengenai tempat-tempat kuliner dalam basis pengetahuan dapat meningkatkan kualitas hasil rekomendasi.

Kemudian, (Mondi & Wijayanto, 2019) membuat penelitian rekomendasi tempat makan menggunakan metode *content-based filtering*. Penelitian ini menggunakan *cosine similarity* untuk mengukur kemiripan antara dua item. Terdapat 2599 data menu makanan dari 353 restoran yang digunakan dalam penelitian ini. Dari data tersebut dilakukan pengujian dan hasilnya menunjukkan bahwa rekomendasi yang dihasilkan memiliki rata-rata nilai presisi sebesar 0,8915 dan rata-rata nilai akurasi sebesar 0,5118.

Penelitian sejenis juga pernah dilakukan oleh (Sandi et al., 2022) dengan menggunakan metode *weighted product*. Penelitian ini berfokus pada pembuatan sistem rekomendasi makanan halal dengan studi kasus di provinsi Bali. Penelitian ini memiliki dua fitur utama, yaitu fitur *admin* dan *user*. Fitur admin pada penelitian ini dibuat menggunakan platform web yang berisi *dashboard* untuk memantau dan mengatur aktivitas *user*. Sementara itu, platform *mobile* digunakan untuk membuat fitur *user*. Pada platform tersebut, *user* dapat memasukkan nilai referensi yang dapat digunakan untuk membuat rekomendasi, setelah itu, *user* dapat melihat halaman daftar rekomendasi beserta detail dari tempat makan halal. Penelitian ini juga menggunakan *map* untuk mengetahui rute dari tempat makan yang dipilih. Dengan memperhatikan aspek-aspek seperti jarak, harga, dan jam operasi, dari 50 data tes, 45 diantaranya sudah sesuai. Dengan kata lain, penelitian ini menghasilkan tingkat akurasi hingga 90%.

Kemudian, pada 2021, (Asani et al., 2021) membuat penelitian tentang sistem rekomendasi restoran. Penelitian ini menggunakan metode *sentiment analysis* yang didasarkan pada pengolahan bahasa alami atau *natural language processing* untuk memproses teks komentar *user* dan mengekstrak nama-nama makanan yang diinginkan. Penelitian ini menggunakan metode *Wu-Palmer* untuk mengelompokkan nama-nama makanan. Sistem yang dibuat ini juga mempertimbangkan konteks, seperti preferensi *user*, lokasi, waktu, dan umpan balik dari semua *user* untuk merekomendasikan restoran yang sesuai dengan preferensi makanan *user*. Penelitian ini mengambil data dari situs web TripAdvisor pada tahun 2018 dan hasilnya menunjukkan bahwa sistem dapat memberikan tingkat presisi sebesar 92,8% dalam mode Top5.

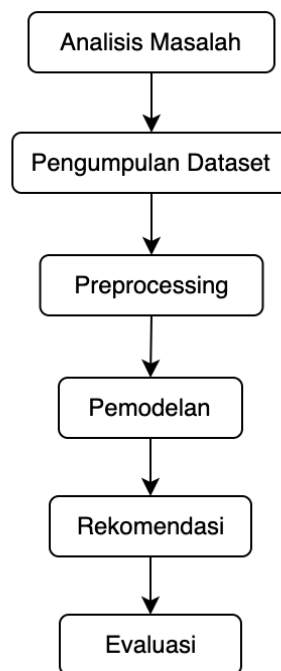
Penelitian tentang rekomendasi tempat makan juga pernah dilakukan oleh (Fakhri et al., 2019). Penelitian ini mengukur persamaan antara metode *user-based collaborative filtering* biasa dengan yang menggunakan *user attribute*. *User-based collaborative filtering* biasa

dihitung hanya dengan menggunakan *Pearson Similarity*. Sementara itu, *user-based collaborative filtering* yang menggunakan *user attribute* dihitung dengan memasukkan atribut *user*, seperti usia dan jenis kelamin. Pemilihan jumlah *user* dengan preferensi yang sama mempengaruhi nilai MAE. Semakin banyak jumlah *user* dengan preferensi yang sama, maka performa sistem akan semakin baik. Pada penelitian ini, *user-based collaborative filtering* biasa mendapat nilai MAE sebesar 1.492, sementara *user-based collaborative filtering* yang menggunakan *user attribute* memperoleh nilai MAE sebesar 2.166. Karena semakin kecil nilai MAE semakin baik, maka dapat disimpulkan dalam penelitian ini bahwa *User-based collaborative filtering* biasa lebih baik daripada *user-based collaborative filtering* yang menggunakan *user attribute*.

Dari beberapa penelitian di atas, dapat disimpulkan bahwa penelitian tentang sistem rekomendasi tempat makan telah banyak dilakukan sebelumnya. Terlepas dari itu, banyak metode yang digunakan dalam pengembangan sistem rekomendasi ini, seperti *weighted product*, *sentiment analysis*, *conversational recommender systems*, *content-based filtering*, dan *collaborative filtering*. Hasil dari beberapa tinjauan penelitian di atas telah memberikan panduan bagi peneliti dalam pengembangan sistem rekomendasi restoran halal. Langkah-langkah yang telah terbukti dari penelitian sebelumnya dapat menjadi acuan untuk menciptakan sistem rekomendasi yang akurat dan relevan.

BAB III METODOLOGI

Dalam rangka mengembangkan sistem rekomendasi restoran halal dalam penelitian ini, langkah-langkah dan proses tertentu harus dilalui. Proses tersebut melibatkan analisis kebutuhan, pengumpulan data, preprocessing data, pemodelan, rekomendasi, dan evaluasi. Rincian mengenai tahapan penelitian ini dapat ditemukan dalam Gambar 3.1.



Gambar 3.1 Langkah langkah penelitian

3.1 Analisis Masalah

Penelitian ini berfokus pada pengembangan sistem rekomendasi restoran halal menggunakan pendekatan *collaborative filtering*, yang dipadukan dengan *autoencoder* dan pengukuran kemiripan antar-item melalui metode *cosine similarity*. Dalam konteks sistem rekomendasi, *collaborative filtering* bertujuan untuk memanfaatkan pola perilaku kolektif pengguna untuk merekomendasikan restoran yang relevan. Salah satu tantangan utama dalam implementasi sistem ini adalah memastikan bahwa rekomendasi yang dihasilkan benar-benar sesuai dengan preferensi pengguna, terutama dalam hal menyajikan restoran yang menawarkan makanan halal, yang merupakan elemen penting dalam penelitian ini. Untuk mencapai tujuan tersebut, sistem harus mampu menemukan keterkaitan antara preferensi pengguna yang

berbeda, kemudian mengelompokkan restoran berdasarkan kemiripan yang dihitung dari ulasan, rating, dan fitur lainnya.

Sistem rekomendasi ini menggunakan *autoencoder*, yaitu teknik pembelajaran mendalam atau *deep learning* yang membantu mereduksi dimensi data sambil tetap mempertahankan informasi penting dari dataset. Dengan demikian, *autoencoder* dapat menangkap pola-pola kompleks dalam data, yang kemudian digunakan dalam proses *collaborative filtering* untuk membuat prediksi yang lebih akurat. *Cosine similarity* dipilih sebagai metode untuk mengukur kemiripan antar restoran, memungkinkan sistem untuk menghitung sejauh mana dua restoran dianggap mirip oleh pengguna. Tantangan lainnya terletak pada kemampuan sistem untuk menyaring data yang tidak relevan, seperti restoran yang tidak menyediakan makanan halal, dan mengatasi masalah *sparsity* dalam data rekomendasi, di mana banyak pengguna hanya memberikan sedikit informasi tentang preferensi mereka. Dengan menggabungkan pendekatan-pendekatan tersebut, penelitian ini bertujuan untuk menghasilkan rekomendasi yang lebih tepat, personal, dan sesuai dengan kebutuhan pengguna, sekaligus meminimalisasi potensi kesalahan dalam hasil rekomendasi yang disajikan.

3.2 Pengumpulan Data

Pada tahap ini, peneliti mengumpulkan dataset yang akan dijadikan dasar dalam penelitian ini. Dataset yang digunakan pada penelitian ini diperoleh melalui proses *scraping* dari situs *Google Maps*. Dataset berjumlah 7.030 data, yang diambil dari berbagai lokasi wisata, terutama di daerah yang masih terdapat campuran antara restoran yang menyajikan makanan halal dan non-halal, seperti beberapa wilayah di Bali, meliputi Denpasar, Gianyar, Ubud, Kuta, Canggu, dan Nusa Dua. Dataset ini mencakup informasi yang sangat penting untuk penelitian, seperti identitas pengguna, nama restoran, *rating*, ulasan (*review*), serta fitur penawaran yang tersedia di *Google Maps*.

Data dari situs *Google Maps* dipilih karena mengandung informasi *rating* atau interaksi pengguna yang nyata untuk tempat-tempat tertentu seperti restoran. Disamping itu, pengambilan data melalui proses *scraping* dapat memudahkan peneliti untuk mengambil data dari tempat yang spesifik. Alasan lain penggunaan data dari situs *Google Maps* adalah ketersediaan ulasan pengguna dan penawaran dari restoran, yang dapat membantu dalam proses penyaringan data untuk mendukung pengembangan sistem rekomendasi restoran halal. Meskipun data memiliki sparsitas tinggi, dengan menggunakan metode yang tepat, data

tersebut diharapkan dapat mendukung proses pengembangan sistem rekomendasi restoran halal secara lebih akurat dan komprehensif.

3.2.1 Teknik Pengumpulan Data

Proses pengumpulan data dalam penelitian ini menggunakan metode *web scraping*, yaitu sebuah teknik yang digunakan untuk mengekstraksi informasi semi-terstruktur dari halaman web yang berformat HTML atau XHTML. Teknik ini memungkinkan peneliti untuk memperoleh data langsung dari situs web secara otomatis dan efisien. Dalam konteks penelitian ini, *web scraping* dilakukan pada platform *Google Maps*, yang memuat berbagai informasi tentang restoran di daerah wisata, khususnya di Bali. Data yang diambil mencakup beberapa indikator penting yang dibutuhkan untuk membangun sistem rekomendasi restoran halal, seperti identitas pengguna, nama restoran, *rating*, ulasan, serta bagian penawaran yang tersedia di *Google Maps*.

Untuk melakukan proses *web scraping*, digunakan alat seperti *Jupyter Notebook* sebagai lingkungan pemrograman interaktif, dan *library Playwright* yang berfungsi untuk mengotomatisasi browser dalam menavigasi dan mengekstraksi data dari situs web. *Playwright* dipilih karena kemampuannya yang kuat dalam melakukan interaksi dengan elemen-elemen dinamis pada situs web, serta mendukung berbagai fitur yang memungkinkan proses *scraping* menjadi lebih stabil dan dapat diandalkan.

Proses pengambilan data dilakukan secara bertahap untuk memaksimalkan keakuratan hasil dan meminimalisir kesalahan. Tahap pertama adalah mengarahkan sistem otomatis ke halaman-halaman restoran di wilayah Bali yang ada di *Google Maps*. Wilayah-wilayah yang menjadi target pengumpulan data antara lain Denpasar, Gianyar, Ubud, Kuta, Canggu, dan Nusa Dua. Dalam tahap ini, sistem akan mengambil daftar restoran yang tersedia di setiap wilayah tersebut sebagai dasar pengumpulan data yang lebih mendetail.

Setelah data-data restoran berhasil dikumpulkan, semua informasi yang diperoleh disimpan dalam format CSV (*Comma-Separated Values*), yang merupakan format file yang umum digunakan untuk menyimpan data tabular agar mudah diproses dan dianalisis lebih lanjut. Format CSV dipilih karena kompatibilitasnya yang tinggi dengan berbagai alat analisis data serta kemudahan dalam manipulasi dan pengolahan menggunakan bahasa pemrograman seperti *python*. Data yang telah dikumpulkan akan melalui proses lanjutan untuk penyaringan dan pembersihan sebelum digunakan dalam model rekomendasi.

3.2.2 Pengambilan Fitur Restoran

Setelah tahap awal pengumpulan data selesai, langkah berikutnya adalah mengekstrak fitur-fitur penting dari setiap restoran yang telah terdaftar. Fitur-fitur tersebut mencakup nama restoran, rating, ulasan, identitas pengguna yang memberikan rating, serta bagian penawaran yang tersedia. Informasi ini sangat penting dalam pembuatan sistem rekomendasi restoran halal, karena memungkinkan sistem untuk mempersonalisasi rekomendasi berdasarkan preferensi pengguna dan data terkait restoran.

Proses pengambilan fitur restoran dilakukan dengan bantuan XPATH atau *XML Path Language*, yaitu sebuah bahasa query yang digunakan untuk menavigasi dan memilih bagian-bagian spesifik dari dokumen XML atau HTML pada halaman web. Dengan menggunakan XPATH, sistem dapat dengan tepat menargetkan elemen-elemen yang relevan di dalam struktur HTML dari situs *Google Maps*, seperti nama restoran, ulasan pengguna, dan *rating*. Pemilihan XPATH yang tepat menjadi kunci utama dalam proses ini, karena menentukan keakuratan dan kelengkapan data yang berhasil diekstraksi.

Seluruh proses ekstraksi ini dijalankan secara otomatis dengan menggunakan *library Playwright* yang sebelumnya telah didefinisikan dalam tahap pengumpulan data. *Playwright* mampu melakukan *scraping* pada halaman-halaman web yang dinamis dengan tepat dan cepat. Setiap kali sistem menemukan restoran baru, data-data terkait akan segera diekstraksi dan disimpan ke dalam dokumen CSV.

Setelah seluruh fitur restoran berhasil diekstraksi dan disimpan dalam format CSV, langkah selanjutnya adalah melakukan proses *filtering* dan pembersihan data. Proses ini dilakukan menggunakan bahasa pemrograman *python*, yang dilengkapi dengan berbagai pustaka untuk analisis data, seperti *pandas*. Penyaringan dilakukan untuk memastikan bahwa hanya data yang relevan dan berkualitas yang digunakan dalam pengembangan model rekomendasi. Data-data yang tidak sesuai atau mengandung informasi yang tidak lengkap akan dihapus atau disesuaikan, sehingga tidak mempengaruhi akurasi dan kualitas sistem rekomendasi yang akan dibangun.

Dengan melalui serangkaian proses pengumpulan dan pemrosesan data yang teliti, dataset akhir yang dihasilkan diharapkan dapat memberikan kontribusi yang signifikan dalam pengembangan sistem rekomendasi restoran halal, dengan fokus pada akurasi dan personalisasi rekomendasi yang tepat sesuai dengan preferensi pengguna.

3.3 Preprocessing

Preprocessing merupakan langkah awal dalam pengolahan data yang bertujuan untuk mengubah data mentah menjadi format yang lebih bersih dan siap diolah. Pada penelitian ini, *preprocessing* menjadi tahapan penting untuk memastikan bahwa data yang digunakan dapat menghasilkan model rekomendasi restoran halal yang akurat. Proses *preprocessing* terdiri dari empat bagian utama, yaitu penggabungan data atau *combining data*, pembersihan data atau *data cleaning*, penyaringan data atau *filtering data*, dan pembagian data atau *splitting data*.

3.3.1 Combining Data

Tahap pertama dalam *preprocessing* adalah penggabungan data atau *combining data*. Setelah data diperoleh melalui proses *scraping*, biasanya data tersebut masih terpisah dalam beberapa file berbeda berdasarkan wilayah atau kategori tertentu. Penggabungan data diperlukan untuk memudahkan analisis lebih lanjut serta mempercepat proses pengolahan data secara keseluruhan. Proses penggabungan ini dilakukan dengan menggunakan pustaka dari *python*, seperti *os*, yang memungkinkan peneliti untuk menggabungkan beberapa file CSV menjadi satu file data yang besar dan terstruktur.

Dalam penelitian ini, file-file data yang berisi informasi restoran dari berbagai wilayah wisata di Bali, seperti Denpasar, Gianyar, Ubud, Kuta, Canggu, dan Nusa Dua, digabungkan menjadi satu kesatuan. Penggabungan ini penting untuk memudahkan proses analisis berikutnya, serta memastikan bahwa semua data dari berbagai wilayah dapat dianalisis secara serentak untuk keperluan pengembangan model rekomendasi. Proses penggabungan dilakukan dengan teliti untuk memastikan tidak ada informasi yang hilang atau terduplikasi dalam file akhir.

3.3.2 Data Cleaning

Tahap berikutnya adalah *data cleaning*, yaitu proses pembersihan data untuk memastikan bahwa data yang digunakan bebas dari kesalahan atau ketidaksesuaian yang dapat mempengaruhi kualitas model. Data yang diperoleh dari proses *scraping* biasanya mengandung banyak masalah, seperti nilai yang hilang, duplikasi data, atau kesalahan pengetikan. Pembersihan data dilakukan secara bertahap untuk menangani setiap masalah tersebut dengan cara yang tepat.

Langkah pertama dalam pembersihan data adalah menangani nilai yang hilang. Nilai-nilai yang hilang dapat ditangani dengan dua pendekatan: mengisi nilai yang hilang dengan

estimasi, seperti nilai rata-rata, atau menghapus baris yang mengandung nilai yang hilang. Misalnya, jika terdapat ulasan yang kosong di kolom "*review_text*", maka kita dapat memilih untuk mengisi ulasan tersebut dengan estimasi berbasis ulasan lain yang serupa, atau menghapus baris data tersebut jika dirasa kurang relevan untuk analisis.

Selanjutnya, data duplikat juga harus dihapus untuk memastikan keunikan setiap entitas dalam dataset. Duplikasi dapat terjadi jika terdapat restoran yang sama dengan ulasan yang identik muncul lebih dari sekali dalam dataset. Dengan menghapus data duplikat, kita dapat menjaga integritas dataset serta meningkatkan efisiensi dalam proses pemodelan.

Selain itu, pembersihan data juga melibatkan penghapusan karakter yang tidak relevan, seperti digit dari teks ulasan. Terkadang pengguna mencampur teks ulasan dengan angka yang tidak memberikan informasi penting. Menghapus digit dari ulasan dapat meningkatkan kualitas data ulasan yang digunakan dalam proses rekomendasi.

3.3.3 Filtering Data

Setelah dilakukan proses *data cleaning*, langkah selanjutnya adalah *filtering data* atau penyaringan data. *Filtering* merupakan sebuah proses untuk menyaring data berdasarkan kriteria tertentu sehingga data dapat relevan dengan penelitian tentang sistem rekomendasi restoran halal. Langkah ini merupakan bagian penting dari proses *preprocessing* untuk data restoran dalam pengembangan sistem rekomendasi. Menyaring data dapat membantu meningkatkan relevansi dan akurasi sistem rekomendasi. Dalam konteks rekomendasi restoran halal, kata kunci seperti "Babi", "Pork", dan "Bacon" mungkin tidak membawa makna yang terkait dengan ulasan atau deskripsi restoran halal.

Sebagai contoh, pertimbangkan ulasan, "Tempat menyajikan daging babi yang sangat lezat." Setelah penyaringan data, restoran yang mengandung ulasan dengan kalimat seperti itu akan dihapus dari dataset. Dengan menghilangkan data dengan kata kunci tersebut, dapat lebih mempengaruhi akurasi model dan meningkatkan relevansi dalam proses rekomendasi.

Selain itu, dalam dataset terdapat fitur penawaran atau *offerings* yang diambil dari situs *Google Maps*. Fitur ini menunjukkan layanan yang disediakan oleh restoran, seperti jenis makanan atau fasilitas lainnya. Salah satu kata kunci penting dalam fitur ini penawaran berupa "Makanan halal", yang menandakan bahwa restoran tersebut menyajikan makanan halal. Selain itu, pada beberapa restoran terdapat pula penawaran berupa minuman beralkohol. Dengan menyaring data berdasarkan penawaran ini, proses rekomendasi menjadi lebih relevan dan sesuai dengan kebutuhan pengguna yang mencari restoran halal.

3.3.4 Splitting Data

Tahap terakhir dalam *preprocessing* adalah *splitting data* atau pembagian data menjadi beberapa bagian, yaitu set pelatihan dan set pengujian. Pembagian data ini penting untuk melatih model dengan data yang representatif serta menguji performa model menggunakan data yang tidak pernah "dilihat" oleh model sebelumnya. Dalam penelitian ini, digunakan teknik *overlap* untuk membagi data menjadi set pelatihan dan pengujian.

Teknik *overlap* atau teknik tumpang tindih memungkinkan sebagian data muncul di kedua set, baik set pelatihan maupun set pengujian. Metode ini memberikan manfaat dalam mengevaluasi model secara lebih mendalam, karena model diuji dengan data yang memiliki kemiripan tetapi tidak sepenuhnya sama dengan data yang digunakan selama pelatihan. Teknik ini juga membantu mengurangi risiko *overfitting*, di mana model terlalu menyesuaikan diri dengan data pelatihan sehingga performanya menurun saat dihadapkan dengan data baru.

Proporsi umum yang digunakan untuk membagi data adalah 80% untuk pelatihan dan 20% untuk pengujian. Set pelatihan digunakan untuk melatih model, sementara set pengujian digunakan untuk mengevaluasi kinerja model dalam memberikan rekomendasi. Dengan demikian, proses pembagian data ini membantu memastikan bahwa model yang dibangun dapat bekerja dengan baik dalam berbagai kondisi, baik pada saat pelatihan maupun saat diterapkan dalam dunia nyata.

Secara keseluruhan, tahap *preprocessing* dalam penelitian ini memastikan bahwa data yang digunakan dalam pengembangan sistem rekomendasi restoran halal bersih, relevan, dan siap digunakan untuk melatih dan menguji model rekomendasi. Proses ini berperan penting dalam menjaga kualitas data dan meningkatkan akurasi serta relevansi rekomendasi yang dihasilkan oleh sistem.

3.4 Pemodelan

Dalam penelitian ini, pendekatan yang digunakan untuk membangun sistem rekomendasi menggabungkan dua metode utama, yaitu *collaborative filtering* dan *autoencoder*. *Collaborative filtering* bertujuan untuk menemukan kesamaan antar item berdasarkan perilaku pengguna, sehingga dapat memberikan rekomendasi yang sesuai dengan preferensi pengguna berdasarkan item yang serupa. Sementara itu, metode *autoencoder* digunakan untuk merangkum dan mengekstrak fitur-fitur yang relevan dari data input secara otomatis. Integrasi kedua metode ini bertujuan untuk meningkatkan akurasi sistem rekomendasi dengan

memanfaatkan keunggulan masing-masing pendekatan. Meskipun menggunakan salah satu dari dua metode tersebut sudah dapat menghasilkan rekomendasi, penggabungan kedua metode mampu menghasilkan rekomendasi yang lebih baik dan relevan terhadap pengguna. Dengan demikian, diharapkan hasil akhir dari pemodelan ini dapat memberikan rekomendasi yang lebih tepat dan sesuai dengan preferensi dari para pengguna.

3.4.1 Collaborative Filtering

Collaborative filtering digunakan dalam pengembangan sistem rekomendasi karena mampu memberikan rekomendasi tanpa memerlukan data konten dari restoran yang direkomendasikan. Metode ini bekerja hanya dengan memanfaatkan data interaksi pengguna, seperti riwayat *rating* pengguna, sehingga cocok untuk data yang telah tersedia. Dengan menggunakan pola interaksi dari pengguna lain yang memiliki preferensi serupa, sistem dapat menghasilkan rekomendasi yang akurat dan relevan. Keunggulan lain adalah kemampuannya untuk meningkatkan akurasi seiring dengan bertambahnya data. Karena cara kerjanya adalah menangkap pola dari perilaku pengguna lain, metode ini bisa menghasilkan rekomendasi yang terasa lebih sesuai dengan selera pengguna, bahkan jika pengguna belum pernah melihat restoran tersebut sebelumnya.

Pembentukan Matriks Item-User

Langkah pertama dalam implementasi *collaborative filtering* adalah membentuk matriks item-user, yang merupakan representasi interaksi antara pengguna dan item yang dalam hal ini adalah restoran. Setiap baris dalam matriks mewakili satu item (restoran), dan setiap kolom mewakili satu pengguna. Nilai dalam setiap sel matriks adalah rating atau ulasan yang diberikan oleh pengguna terhadap restoran tersebut. Jika pengguna belum memberikan rating untuk sebuah restoran, sel tersebut akan diisi dengan nilai kosong atau nol.

Matriks ini berfungsi sebagai dasar untuk perhitungan kesamaan antar restoran. Misalnya Tabel 3.1 menunjukkan contoh matriks item-user, di mana beberapa pengguna telah memberikan rating untuk beberapa restoran.

Tabel 3.1 Matriks Item-user

	Andi	Dimas	Dani
RM. Minang	4	3	5
Bakso Malang	5	2	-
Ayam Pak Sunu	3	5	3
Soto lamongan	-	4	3

Pada tabel di atas, terlihat bahwa beberapa pengguna belum memberikan rating, misalnya Dani belum memberikan rating untuk Bakso Malang. Hal ini menimbulkan permasalahan umum dalam *collaborative filtering*, yaitu data yang hilang atau *sparsity*, yang dapat diatasi dengan beberapa teknik, seperti prediksi *rating* yang hilang menggunakan kesamaan antar item atau metode lain seperti *autoencoder*.

Perhitungan Kemiripan

Setelah matriks item-user terbentuk, langkah selanjutnya adalah menghitung kemiripan antar item (tempat makan). Penelitian ini menggunakan *cosine similarity* untuk melakukan perhitungan kemiripan. Perhitungan *cosine similarity* digunakan untuk mengukur tingkat kemiripan antara dua vektor atau dua item pada ruang vektor. Dengan metode ini, sistem dapat menilai seberapa mirip antara satu restoran dengan restoran yang lain. Perhitungan *cosine similarity* dapat dilakukan dengan menyelesaikan persamaan (2.1).

Dari matriks item-user pada Tabel 3.1, kita akan membandingkan tingkat kemiripan antara Bakso Malang dan Soto lamongan. Kita misalkan Bakso Malang adalah i_2 dan Soto lamongan sebagai i_4 . Kedua restoran masing-masing mendapat tiga *rating* dari empat pengguna. Dengan menggunakan persamaan (2.1), kita dapat melakukan perhitungan sebagai berikut:

$$sim(i_2, i_4) = \frac{(5.4) + (2.3) + (3.4)}{\sqrt{(5^2 + 2^2 + 3^2)} \cdot 4\sqrt{(4^2 + 3^2 + 4^2)}}$$

$$sim(i_2, i_4) = \frac{20 + 6 + 12}{\sqrt{38} \cdot \sqrt{41}}$$

$$sim(i_2, i_4) = \frac{38}{\sqrt{1558}}$$

$$sim(i_2, i_4) = \frac{38\sqrt{1558}}{1558}$$

$$sim(i_2, i_4) = \frac{\sqrt{1558}}{41} = 0.96$$

Nilai 0.96 menunjukkan bahwa Bakso Malang dan Soto Lamongan memiliki tingkat kemiripan yang sangat tinggi, mendekati nilai 1. Dengan demikian, sistem dapat merekomendasikan Soto Lamongan kepada pengguna yang menyukai Bakso Malang, karena keduanya dianggap serupa berdasarkan perilaku rating pengguna.

3.4.2 Autoencoder

Autoencoder merupakan jaringan saraf tiruan yang dirancang untuk melakukan tugas *dimensionality reduction* atau pengurangan dimensi dan rekonstruksi data. Dalam penelitian ini, *autoencoder* digunakan untuk mengekstraksi representasi yang lebih padat dan relevan dari preferensi pengguna terhadap restoran. Dengan menggunakan *autoencoder*, model dapat mengenali pola yang lebih kompleks dalam data dan menghasilkan rekomendasi yang lebih personal dan akurat. Metode ini juga dapat digabungkan dengan metode lain seperti *collaborative filtering*.

Dalam *collaborative filtering*, rekomendasi didasarkan pada pola interaksi pengguna dengan restoran seperti rating. Namun, data ini sering kali *spars* karena sebagian besar pengguna hanya berinteraksi dengan sebagian kecil restoran. *Autoencoder* membantu mengisi kekosongan data dengan memperkirakan nilai *rating* yang kosong berdasarkan pola yang ditemukan dalam data. Hasil dari pelatihan model *autoencoder* dapat digunakan untuk memperbarui matriks *rating* asli, sehingga semua data *rating* yang tadinya kosong dapat terisi. *Collaborative filtering* kemudian dapat menggunakan data ini untuk memprediksi *rating* dan menghasilkan rekomendasi.

Sebagai contoh, pada Tabel 3.1 terdapat beberapa data *rating* yang kosong atau tidak ada nilainya. *Autoencoder* dapat melatih dan menghasilkan perkiraan untuk nilai kosong dengan merekonstruksi inputnya, yaitu *rating* asli melalui lapisan-lapisan *encoder* dan *decoder*, sehingga menghasilkan matriks rekonstruksi yang lebih lengkap. Melalui matriks rekonstruksi ini, *collaborative filtering* kemudian dapat bekerja pada data yang lebih penuh untuk memberikan rekomendasi yang lebih baik.

Desain Model Autoencoder

Desain model *autoencoder* mencakup konfigurasi *encoder* dan *decoder*, dua bagian utama yang bekerja bersama untuk menghasilkan representasi yang optimal dari data. *Encoder* bertanggung jawab untuk mengkonversi input (fitur restoran) menjadi representasi yang lebih padat, sedangkan *decoder* merekonstruksi data dari representasi tersebut.

Arsitektur *autoencoder* ini dirancang dengan beberapa *layer* untuk mengekstraksi dan merekonstruksi fitur dari data restoran. Pada bagian *encoder*, terdapat empat *layer Dense* yang berfungsi untuk mengekstraksi fitur secara bertahap dengan jumlah neuron yang berkurang, mulai dari 256, 128, 64, hingga 50 neuron. Fungsi aktivasi ReLU digunakan di setiap *layer* untuk membantu menangkap pola non-linear yang kompleks. Selain itu, terdapat *layer Dropout* dengan rasio 0.2 yang membantu mencegah *overfitting*.

Pada bagian *decoder*, terdapat empat *layer Dense* yang secara bertahap memperbesar kembali dimensi representasi latennya. *Layer* pertama memiliki 64 neuron, diikuti oleh 128 neuron, 256 neuron, dan *layer* terakhir yang memiliki jumlah neuron sesuai dengan dimensi *input* awal yaitu *input_dim* dengan fungsi aktivasi Sigmoid. Fungsi aktivasi Sigmoid digunakan pada *layer output* untuk menghasilkan nilai prediksi yang berada dalam rentang 0 hingga 1, sesuai dengan skala rating pengguna. Kombinasi arsitektur *encoder* dan *decoder* ini memungkinkan *autoencoder* untuk efisien mengekstraksi fitur dan merekonstruksi data, menciptakan representasi yang kompak dari preferensi pengguna terhadap tempat makan.

Pelatihan Model Autoencoder

Proses pelatihan model *autoencoder* dilakukan dengan memanfaatkan dataset restoran yang telah terkumpul dan penggabungan dengan metode *collaborative filtering* yang telah dibuat sebelumnya. Selama proses iterasi, tujuan utama dari pelatihan adalah meminimalkan perbedaan antara input dan output, sehingga model dapat menghasilkan representasi yang semakin akurat dari preferensi pengguna terhadap restoran.

Dalam konteks ini, parameter pelatihan menjadi faktor krusial yang mendukung kesuksesan model. *Learning rate*, yang menentukan seberapa cepat model belajar dari data, pemilihan angka kecil, seperti 0.0001, dimaksudkan agar model belajar secara perlahan dan tidak terlalu cepat. Penyesuaian ini bertujuan membantu proses pelatihan agar tetap berjalan dengan stabil. Sebagai tambahan, pemilihan *batch size* yang optimal menjadi penting untuk memastikan bahwa model dapat memproses data secara efisien, sambil menghindari beban komputasional yang tidak perlu. Dengan konfigurasi parameter pelatihan yang cermat seperti ini, diharapkan model *autoencoder* mampu mengekstraksi fitur dan merekonstruksi data dengan akurat, menciptakan representasi yang lebih baik dari preferensi pengguna terhadap restoran.

3.5 Rekomendasi

Pada bagian ini, akan dipaparkan daftar rekomendasi restoran halal terbaik yang dihasilkan oleh sistem rekomendasi yang telah dibangun dalam penelitian ini. Rekomendasi tersebut disusun berdasarkan skor tertinggi yang dihitung dari kemiripan antar pengguna serta preferensi mereka terhadap restoran. Sistem ini memanfaatkan pendekatan *collaborative filtering* dan *autoencoder*, di mana setiap pengguna mendapatkan rekomendasi restoran yang personal dan relevan. Skor tertinggi diberikan kepada restoran yang memiliki kemiripan preferensi dengan pengguna, baik dari segi rating, ulasan, maupun fitur lainnya, seperti jenis makanan yang disajikan dan kenyamanan tempat.

Dengan mengutamakan skor tertinggi, sistem ini mampu memberikan rekomendasi yang relevan dan akurat, sehingga mempermudah pengguna dalam menemukan restoran yang sesuai dengan kebutuhan dan preferensi mereka. Rekomendasi ini tidak hanya mempertimbangkan kualitas makanan, tetapi juga kenyamanan tempat, layanan yang diberikan, dan yang terpenting, kesesuaian dengan prinsip-prinsip halal. Hal ini memastikan bahwa setiap rekomendasi yang dihasilkan relevan dengan gaya hidup pengguna yang memprioritaskan makanan halal dalam keseharian mereka.

Melalui pendekatan ini, sistem rekomendasi tidak hanya menawarkan opsi terbaik, tetapi juga dirancang untuk memberikan pengalaman kuliner yang optimal dan personal. Pengguna dapat menemukan restoran yang tidak hanya lezat, tetapi juga sesuai dengan standar kehalalan yang mereka pegang. Pada akhirnya, rekomendasi ini diharapkan dapat meningkatkan kenyamanan dan kepuasan pengguna dalam memilih restoran, sekaligus membantu mereka menemukan destinasi kuliner yang sejalan dengan nilai-nilai yang mereka anut.

3.6 Evaluasi

Evaluasi merupakan tahapan krusial dalam pengembangan sistem rekomendasi, karena memungkinkan peneliti untuk secara sistematis menilai performa sistem dalam memenuhi tujuan dan target yang telah ditetapkan. Evaluasi tidak hanya digunakan untuk mengukur kinerja keseluruhan, tetapi juga untuk mengidentifikasi kelemahan serta area yang memerlukan perbaikan. Pada penelitian ini, evaluasi dilakukan setelah seluruh tahapan pengolahan data, mulai dari *preprocessing* yang mencakup pembersihan data, penggabungan, penyaringan, hingga penghitungan kemiripan antar item menggunakan metode *cosine similarity*. Metode *cosine similarity* digunakan untuk menghitung kesesuaian antara preferensi pengguna dan restoran yang direkomendasikan. Dengan menghitung sudut kemiripan antara dua vektor

(misalnya preferensi pengguna dan fitur restoran), kita dapat menentukan seberapa relevan sebuah restoran bagi pengguna tertentu. Penggunaan *cosine similarity* memberikan dasar yang kuat untuk membangun rekomendasi yang berbasis pada pola perilaku pengguna yang serupa, sehingga hasil yang diberikan lebih tepat sasaran.

Tahap evaluasi pada penelitian ini menggunakan metrik utama *Root Mean Squared Error* (RMSE) untuk mengukur kinerja sistem. RMSE adalah salah satu metode evaluasi yang paling umum digunakan dalam sistem rekomendasi karena mampu memberikan gambaran yang jelas tentang seberapa besar selisih antara nilai yang diprediksi oleh sistem dan nilai aktual yang diharapkan oleh pengguna. Semakin kecil nilai RMSE, semakin baik akurasi prediksi yang dihasilkan oleh sistem. Selain RMSE, evaluasi ini juga melibatkan analisis mendalam terhadap kesalahan prediksi yang dihasilkan untuk menemukan pola atau masalah yang mungkin terjadi dalam model. Dengan begitu, jika nilai RMSE yang dihasilkan cukup besar, hal ini akan memberikan indikasi bahwa ada bagian dari sistem rekomendasi yang perlu disesuaikan atau dimodifikasi agar dapat menghasilkan prediksi yang lebih akurat. Evaluasi yang dilakukan tidak hanya berfungsi untuk menilai apakah sistem telah beroperasi sesuai harapan, tetapi juga memberikan wawasan tentang area yang membutuhkan peningkatan agar sistem rekomendasi dapat memenuhi kebutuhan pengguna secara lebih optimal.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pengumpulan Dataset

Dataset pada penelitian ini menggunakan data restoran yang diambil dari situs *Google Maps*. Proses pengambilan data melalui serangkaian proses yang disebut *scraping*. Dalam proses *scraping*, peneliti mengambil data dari beberapa tempat di Bali, seperti Denpasar, Gianyar, Ubud, Kuta, Canggu, dan Nusa dua. Pengambilan data di provinsi Bali bertujuan untuk memberikan rekomendasi restoran halal yang relevan bagi para pengguna. Pengumpulan data dilakukan dengan membuat sebuah *automation system* yang dapat dengan otomatis mengambil data. pengambilan data dilakukan dengan memperhatikan beberapa variabel. Variabel-variabel tersebut diantaranya adalah nama restoran, id restoran, nama pengguna, *rating* pengguna, ulasan pengguna, dan penawaran. Gambar 4.1 menunjukkan contoh dari dataset yang telah dikumpulkan.

	restaurant_id	restaurant_name	address	reviews_average	user_name	stars	review_text	latitude	longitude	offerings
0	0x2dd244859bbcbdb5:0x8b2bfcf77975f81	Rasa Selera Tanjung pinang	659H+C6F, Jl. Uluwatu II, Jimbaran, Kec. Kuta ...	4.4	rahayuteuk	4.0	Enak seperti masakan padang pada umumnya	-8.776405	115.098546	['Makanan halal']
1	0x2dd244859bbcbdb5:0x8b2bfcf77975f81	Rasa Selera Tanjung pinang	659H+C6F, Jl. Uluwatu II, Jimbaran, Kec. Kuta ...	4.4	toffuman 666	4.0	Sumpah ini enak. Banget seafoodnya , recomende...	-8.776405	115.098546	['Makanan halal']
2	0x2dd244859bbcbdb5:0x8b2bfcf77975f81	Rasa Selera Tanjung pinang	659H+C6F, Jl. Uluwatu II, Jimbaran, Kec. Kuta ...	4.4	Andy Yusuf Septanto	5.0	Enak kok	-8.776405	115.098546	['Makanan halal']
3	0x2dd244859bbcbdb5:0x8b2bfcf77975f81	Rasa Selera Tanjung pinang	659H+C6F, Jl. Uluwatu II, Jimbaran, Kec. Kuta ...	4.4	Kartika Sari	5.0	Enak dan nyaman	-8.776405	115.098546	['Makanan halal']
4	0x2dd244859bbcbdb5:0x8b2bfcf77975f81	Rasa Selera Tanjung pinang	659H+C6F, Jl. Uluwatu II, Jimbaran, Kec. Kuta ...	4.4	Berita Fajar Timur Bftnews (Bftnews)	3.0	Hmmmm yamick	-8.776405	115.098546	['Makanan halal']
7025	0x2dd21661b1e051ad:0x75b5e7c35aeaad57	Rumah Makan Padang Bukit Tinggi	F84H+GJ3, Jl. Kapten Dipta, Gianyar, Kec. Gian...	4.6	Emen Harmen	3.0	Lumayan enak	-8.656186	114.962623	['Kopi', 'Makanan halal']
7026	0x2dd21661b1e051ad:0x75b5e7c35aeaad57	Rumah Makan Padang Bukit Tinggi	F84H+GJ3, Jl. Kapten Dipta, Gianyar, Kec. Gian...	4.6	Dewa Zeruk	5.0	Murah meriah	-8.656186	114.962623	['Kopi', 'Makanan halal']
7027	0x2dd21661b1e051ad:0x75b5e7c35aeaad57	Rumah Makan Padang Bukit Tinggi	F84H+GJ3, Jl. Kapten Dipta, Gianyar, Kec. Gian...	4.6	Gede Eka	5.0	Mantap	-8.656186	114.962623	['Kopi', 'Makanan halal']
7028	0x2dd21661b1e051ad:0x75b5e7c35aeaad57	Rumah Makan Padang Bukit Tinggi	F84H+GJ3, Jl. Kapten Dipta, Gianyar, Kec. Gian...	4.6	divdossi	4.0	Menawarkan 'Nasi Padang' yang lezat. Harga mur...	-8.656186	114.962623	['Kopi', 'Makanan halal']
7029	0x2dd21661b1e051ad:0x75b5e7c35aeaad57	Rumah Makan Padang Bukit Tinggi	F84H+GJ3, Jl. Kapten Dipta, Gianyar, Kec. Gian...	4.6	Yves Boutin	3.0	Pilihan harga dan tempat yang sederhana dengan...	-8.656186	114.962623	['Kopi', 'Makanan halal']

Gambar 4.1 Data restoran

Pada Gambar 4.1 di atas, ditampilkan contoh dari dataset yang telah dikumpulkan. Dataset ini berjumlah 7.030 entri, yang berisi data dari berbagai restoran di wilayah Bali. Lima baris pertama dan lima baris terakhir dari dataset diperlihatkan sebagai representasi struktur data. Informasi penting yang dapat diambil dari dataset ini adalah variabel penawaran, yang memuat rincian layanan dan status halal dari restoran yang ada. Selain itu, ulasan pengguna juga memegang peranan penting karena memberikan perspektif langsung dari konsumen

terkait kualitas restoran, layanan, serta pengalaman mereka di tempat tersebut. Kedua elemen ini, yakni penawaran restoran dan ulasan pengguna, menjadi dasar untuk menentukan apakah restoran tersebut memenuhi kriteria sebagai restoran halal. Dengan demikian, data yang dikumpulkan tidak hanya mencakup informasi umum tentang restoran, tetapi juga aspek-aspek penting yang relevan bagi pengguna yang memprioritaskan pilihan makanan halal.

4.1.1 Data Scraping

Proses *scraping* pada situs *Google Maps* terdiri dari beberapa tahap, diantaranya yaitu mengakses situs *Google Maps* secara otomatis, mencari restoran berdasarkan *input* yang telah dimasukkan, mengambil data restoran secara umum, mengambil data restoran pada halaman *review* dan *about*, dan menyimpan data ke dalam format CSV dan excel. Data memuat berbagai macam variabel, termasuk nama restoran, nama pengguna, ulasan pengguna, *rating* pengguna, *latitude*, dan *longitude*.

Mengakses Situs Google Maps Secara Otomatis

Sebelum proses *scraping* dilakukan, pertama yang dilakukan adalah mengakses situs *Google Maps* secara otomatis. Penggunaan pustaka *playwright* memungkinkan sistem dapat terhubung dengan browser *chromium* yang akan memudahkan dalam proses *scraping* data. Namun sebelumnya, kita perlu untuk mendefinisikan beberapa variabel. Variabel-variabel tersebut antara lain yaitu *total*, *searchList*, *input_file_name*, dan *input_file_path*. Variabel *total* adalah variabel yang akan membantu kita untuk melakukan pencarian, sehingga kita beri nilai sebanyak mungkin untuk memaksimalkan jumlah pencarian. Kemudian, variabel *searchList* berguna untuk menyimpan daftar *input* yang telah dimasukkan ke dalam file input.txt. Daftar *input* tersebut disimpan di dalam variabel *input_file_name*. Dari situ, kita dapat menguraikan daftar *input* menggunakan variabel *input_file_path*. Setelah itu, kode akan menjalankan browser *chromium* dan mengakses situs *Google Maps* secara otomatis. Kode untuk melakukan akses situs *Google Maps* secara otomatis ditunjukkan pada Gambar 4.2.

```
def main():
    total = 1_000_000
    searchList = []
    input_file_name = 'input.txt'
    input_file_path = os.path.join(os.getcwd(), input_file_name)

    if os.path.exists(input_file_path):
        with open(input_file_path, 'r') as file:
            searchList = file.readlines()

    if len(searchList) == 0:
        print('Error occurred: You must add searches to input.txt')
        sys.exit()

    with sync_playwright() as p:
        browser = p.chromium.launch(headless=False)
        page = browser.new_page()

        page.goto("https://www.google.com/maps")
```

Gambar 4.2 Kode *scraping* untuk akses situs *Google Maps*

Mencari Restoran Berdasarkan *Input* yang Telah Dimasukkan

Setelah situs *Google Maps* berhasil diakses, langkah selanjutnya adalah melakukan pencarian restoran berdasarkan input yang telah dimasukkan pengguna, seperti nama lokasi atau jenis makanan. Proses ini melibatkan pemanfaatan fitur pencarian *Google Maps* untuk menampilkan hasil yang relevan. Untuk memastikan semua restoran yang relevan ditampilkan, halaman situs akan digulirkan secara otomatis ke bawah, memungkinkan pemuatan tambahan data restoran yang mungkin tersembunyi di awal. Proses *scrolling* otomatis ini berfungsi hingga jumlah data yang diinginkan tercapai, baik berdasarkan batas maksimal yang ditentukan atau setelah semua hasil telah dimuat. Setelah jumlah data restoran memenuhi kriteria yang ditetapkan, kode akan menghentikan proses pencarian secara otomatis. Kode untuk mencari data restoran berdasarkan *input* pengguna dapat dilihat pada Gambar 4.3.

```

for search_for_index, search_for in enumerate(searchList):
    print(f"-----\n{search_for_index} - {search_for}".strip())

    page.locator('//input[@id="searchboxinput"]').fill(search_for.strip())
    page.wait_for_timeout(3000)

    page.keyboard.press("Enter")
    page.wait_for_timeout(5000)

    page.hover('//a[contains(@href, "https://www.google.com/maps/place")]')

    previously_counted = 0
    while True:
        page.mouse.wheel(0, 10000)
        page.wait_for_timeout(3000)

        if (
            page.locator(
                '//a[contains(@href, "https://www.google.com/maps/place")]')
            ).count()
            >= total
        ):
            listings = page.locator(
                '//a[contains(@href, "https://www.google.com/maps/place")]')
            .all()[:total]
            listings = [listing.locator("xpath=..") for listing in listings]
            print(f"Total Scraped: {len(listings)}")
            break
        else:
            if (
                page.locator(
                    '//a[contains(@href, "https://www.google.com/maps/place")]')
                ).count()
                == previously_counted
            ):
                listings = page.locator(
                    '//a[contains(@href, "https://www.google.com/maps/place")]')
                .all()
                print(f"Arrived at all available\nTotal Scraped: {len(listings)}")
                break

```

Gambar 4.3 Kode *scraping* untuk mencari restoran

Mengambil Data Restoran Secara Umum

Setelah proses pencarian restoran selesai, langkah berikutnya adalah mengumpulkan data dari setiap restoran yang muncul dalam hasil pencarian. Pengumpulan data dilakukan secara sistematis dengan menggunakan *xpath* untuk mengekstrak informasi dari elemen-elemen spesifik di halaman situs. Dengan menggunakan *xpath*, kita dapat mengekstrak informasi

penting dari halaman situs *Google Maps*, seperti nama restoran, id restoran, alamat lengkap restoran, serta rata-rata *rating* yang diberikan oleh pengguna. *Xpath* digunakan untuk memastikan bahwa data yang diambil berasal dari elemen yang benar dan sesuai dengan kebutuhan. Kode untuk pengambilan data restoran dapat dilihat pada Gambar 4.4.

```

for listing in listings:
    try:
        listing.click()
        page.wait_for_timeout(3000)

        id = 'href'
        name_attribute = 'aria-label'
        address_xpath = '//button[@data-item-id="address"]//div[contains(@class, "fontBodyMedium")]'
        reviews_average_xpath = '//div[@jsaction="pane.reviewChart.moreReviews"]//div[@role="img"]'
        review_tab_xpath = '//button[contains(@class, "hh2c6")][2]/div[@class="LRkQ2"]/div[@class="Gpq6kf fontTitleSmall"]'
        username_xpath = '//button[contains(@class, "a16Kxe")]/div[contains(@class, "d4r55")]'
        stars_xpath = '//span[@role="img" and contains(@class, "kvMYJc")]'
        stars_xpath_fallback = '//span[@class="fzvQIb"]'
        text_xpath = '//div[@class="MyEneD"]//span[@class="wiI7pd"]'
        about_tab_xpath = '//button[contains(@class, "hh2c6")][3]/div[@class="LRkQ2"]/div[@class="Gpq6kf fontTitleSmall"]'
        offerings_xpath = '//h2[contains(text(), "Penawaran")]/following-sibling::ul[@class="ZQ6we"]/li/span'

        if listing.get_attribute(id):
            restaurant_id = listing.get_attribute(id).split("!1s")[-1].split("!8m2")[0]
        else:
            restaurant_id = ""
        if listing.get_attribute(name_attribute):
            restaurant_name = listing.get_attribute(name_attribute).split('.')[-1]
        else:
            restaurant_name = ""
        if page.locator(address_xpath).count() > 0:
            address = page.locator(address_xpath).first.inner_text()
        else:
            address = ""
        if page.locator(reviews_average_xpath).count() > 0:
            reviews_average = float(
                page.locator(reviews_average_xpath).first.get_attribute(name_attribute)
                .split()[0]
                .replace(',', '.')
                .strip())
        else:
            reviews_average = ""

```

Gambar 4.4 Kode *scraping* pengambilan data restoran

Mengambil Data Restoran Pada Halaman *Review* dan *About*

Beberapa informasi penting, seperti ulasan pengguna dan penawaran pada restoran, tidak dapat diakses hanya dengan membuka halaman awal dari restoran yang ditampilkan di *Google Maps*. Oleh karena itu, diperlukan kode tambahan yang dirancang untuk menggali data lebih dalam dari halaman-halaman yang lebih spesifik. Misalnya, untuk mengakses data ulasan, kode harus diarahkan ke halaman *review* masing-masing restoran. Proses ini memungkinkan kita untuk mendapatkan data yang lebih lengkap dan mendetail mengenai pengalaman pengguna di restoran tersebut. Penawaran khusus yang tidak ditampilkan di halaman utama juga dapat diambil dengan teknik serupa, memastikan semua informasi yang relevan diperoleh.

Setelah berhasil membuka halaman *review*, langkah selanjutnya adalah melakukan proses *scraping* untuk mengambil data ulasan. Kode akan mengotomatisasi proses pengguliran halaman untuk memastikan semua ulasan, terutama yang berada di bagian bawah, termuat sepenuhnya. Proses ini dilakukan secara hati-hati untuk menghindari terlewatnya data penting. Setelah halaman termuat, kode akan mengumpulkan berbagai informasi dari setiap ulasan, seperti nama pengguna yang memberikan ulasan, isi atau teks ulasan yang ditulis, serta *rating* yang diberikan. Penting untuk memastikan bahwa data yang diambil bersifat unik dan tidak ada duplikasi, terutama jika ulasan yang sama muncul kembali saat proses *scraping* berlangsung. Penghapusan duplikasi ulasan penting agar data yang diperoleh lebih bersih untuk proses selanjutnya. Kode untuk menggulir halaman *review* dapat dilihat pada Gambar 4.5, sedangkan kode untuk mengumpulkan data pada halaman *review* ditampilkan pada Gambar 4.6.

```
print(f"Clicking on Review tab for restaurant: {restaurant_id}")
if page.locator(review_tab_xpath).count() > 0:
    page.locator(review_tab_xpath).click()
    page.wait_for_timeout(8000)

    page.hover('//div[contains(@class, "jftiEf")]//div[contains(@class, "jJc9Ad")]')

    previously_counted_reviews = 0
    while True:
        page.mouse.wheel(0, 10000)
        page.wait_for_timeout(5000)

        if (
            page.locator(
                '//div[contains(@class, "jftiEf")]//div[contains(@class, "jJc9Ad")]'
            ).count()
            >= total
        ):
            reviewListings = page.locator(
                '//div[contains(@class, "jftiEf")]//div[contains(@class, "jJc9Ad")]'
            ).all()[total:]
            reviewListings = [reviewlisting.locator("xpath=..") for reviewlisting in reviewListings]
            print(f"Total Review Scraped: {len(reviewListings)}")
            break
        else:
            if (
                page.locator(
                    '//div[contains(@class, "jftiEf")]//div[contains(@class, "jJc9Ad")]'
                ).count()
                == previously_counted_reviews
            ):
                reviewListings = page.locator(
                    '//div[contains(@class, "jftiEf")]//div[contains(@class, "jJc9Ad")]'
                ).all()
                print(f"Arrived at all available\nTotal Review Scraped: {len(reviewListings)}")
                break
```

Gambar 4.5 Kode *scraping* untuk menggulir halaman *review*

```

for reviewlisting in reviewListings:
    try:
        user_name = reviewlisting.locator(username_xpath).first.inner_text() if reviewlisting.locator(username_xpath).count() > 0 else None
        # Mengambil data rating
        stars_element = reviewlisting.locator(stars_xpath).first
        if stars_element:
            stars = stars_element.evaluate("element => element.getAttribute('aria-label').split(' ')[0]")
        else:
            stars_element_fallback = reviewlisting.locator(stars_xpath_fallback).first
            if stars_element_fallback:
                stars = stars_element_fallback.inner_text().split('/')[0]
            else:
                stars = ""
        stars = float(stars) if stars else None
        review_text = reviewlisting.locator(text_xpath).first.inner_text() if reviewlisting.locator(text_xpath).count() > 0 else None
        review_id = f"{user_name}--{review_text}" # Memastikan data yang diambil bersifat unik

        if review_id not in seen_reviews:
            review = Review(
                restaurant_id=restaurant_id,
                restaurant_name=restaurant_name,
                address=address,
                reviews_average=reviews_average,
                user_name=user_name,
                stars=stars,
                review_text=review_text,
                latitude=None,
                longitude=None,
                offerings=[]
            )
            seen_reviews.add(review_id)
            reviewList.reviewList.append(review)
            print(f"Review for {review.user_name}: {review.review_text}")
        else:
            print(f"Duplicate review found: {review_text}")

```

Gambar 4.6 Kode *scraping* untuk mengambil data pada halaman *review*

Setelah proses pengumpulan data ulasan dari halaman *review* selesai, langkah berikutnya adalah mengambil informasi penting dari halaman *about* restoran. Halaman ini berisi berbagai informasi mendetail mengenai restoran, yang sangat relevan untuk penelitian ini, terutama terkait dengan variabel "*offerings*" atau penawaran. Penawaran ini mencakup layanan-layanan yang disediakan oleh restoran, seperti opsi makanan dan minuman, fasilitas tambahan, serta layanan khusus yang mungkin ditawarkan kepada pelanggan.

Salah satu variabel yang sangat penting dalam konteks penelitian ini adalah informasi mengenai status halal restoran. Halaman *about* sering kali mencantumkan apakah restoran tersebut menawarkan makanan halal, sebuah informasi penting untuk penelitian rekomendasi restoran halal ini. Selain status halal, halaman ini juga dapat memuat informasi tambahan seperti apakah restoran menyediakan reservasi, memiliki tempat makan di luar ruangan,

aksesibilitas bagi difabel, hingga ketersediaan *Wi-Fi*. Semua data ini akan diambil secara otomatis menggunakan teknik *scraping* yang sudah dirancang, memastikan bahwa variabel-variabel tersebut terkumpul dengan baik untuk keperluan analisis dalam penelitian. Dengan memperoleh data dari halaman *about*, kita dapat melengkapi kebutuhan model dan memberikan rekomendasi yang lebih relevan kepada pengguna. Kode untuk mendapatkan data pada halaman *about* ditunjukkan pada Gambar 4.7.

```
print(f"Clicking on About tab for restaurant: {restaurant_id}")
if page.locator(about_tab_xpath).count() > 0:
    page.locator(about_tab_xpath).click()
    page.wait_for_timeout(2000)

print(f"Checking for offerings list for restaurant: {restaurant_id}")
if page.locator(offerings_xpath).count() > 0:
    offerings = [offer.inner_text() for offer in page.locator(offerings_xpath).all()]
else:
    offerings = []
for review in reviewList.reviewList:
    if review.restaurant_id == restaurant_id:
        review.offerings = offerings
        review.latitude, review.longitude = extract_coordinates_from_url(page.url)
        print(f"Updated review for {review.user_name} with offerings and coordinates")
```

Gambar 4.7 Kode *scraping* untuk mengambil data pada halaman *about*

Menyimpan Data ke dalam Format CSV dan Excel

Setelah seluruh proses pengumpulan data selesai, tahap terakhir adalah menyimpan data ke dalam file dengan format CSV dan Excel untuk memudahkan penyimpanan dan analisis lebih lanjut. Sebelum data disimpan dalam file, langkah pertama yang dilakukan adalah mengonversi semua data yang telah dikumpulkan ke dalam format *dataframe* menggunakan pustaka *pandas*. Konversi ini penting karena *dataframe* menawarkan struktur data yang terorganisir dan mudah diakses, dengan baris dan kolom yang memudahkan manipulasi serta pembacaan data.

Dengan format *dataframe*, setiap elemen data seperti nama restoran, alamat, ulasan, *rating*, penawaran, dan status halal dapat diatur secara sistematis, mempermudah analisis lanjutan seperti filter, pembersihan, dan pengelompokan data. Setelah data tersusun rapi dalam *dataframe*, proses penyimpanan dilakukan dalam dua format, yaitu CSV dan Excel. Format CSV banyak digunakan karena ringan dan kompatibel dengan berbagai alat analisis data, sementara Excel memungkinkan penggunaan fitur-fitur tambahan seperti format sel dan rumus,

yang dapat bermanfaat saat mengolah data lebih lanjut. Penyimpanan dalam dua format ini memastikan fleksibilitas penggunaan data, baik untuk analisis manual maupun integrasi dengan alat-alat pemrosesan data lainnya. Kode untuk menyimpan data ke dalam format CSV dan excel dapat dilihat pada Gambar 4.8.

```
class ReviewList:
    reviewList: list[Review] = field(default_factory=list)
    save_at: str = 'output'

    def dataframe(self):
        return pd.json_normalize(
            (asdict(review) for review in self.reviewList), sep="_"
        )

    def save_to_excel(self, filename):
        if not os.path.exists(self.save_at):
            os.makedirs(self.save_at)
        self.dataframe().to_excel(f"{self.save_at}/{filename}.xlsx", index=False)

    def save_to_csv(self, filename):
        if not os.path.exists(self.save_at):
            os.makedirs(self.save_at)
        self.dataframe().to_csv(f"{self.save_at}/{filename}.csv", index=False)
```

Gambar 4.8 Kode untuk proses menyimpan data

4.2 Preprocessing

Pada tahap ini, dilakukan serangkaian proses *preprocessing* untuk memastikan bahwa data yang digunakan dalam penelitian ini relevan dengan topik sistem rekomendasi restoran halal. *Preprocessing* merupakan langkah penting yang melibatkan penggabungan data, pembersihan data, dan penyaringan data berdasarkan kriteria yang sesuai dengan tujuan penelitian. Tujuan dari tahap ini adalah mendapatkan data yang benar-benar berhubungan dengan restoran halal sehingga analisis yang dilakukan lebih akurat dan sesuai dengan fokus penelitian. Tahapan *preprocessing* secara lebih rinci dijelaskan sebagai berikut:

4.2.1 Combining Data

Langkah pertama dalam *preprocessing* adalah *combining* data atau penggabungan data dari beberapa file restoran yang telah dikumpulkan melalui proses *scraping*. Data ini berasal dari lima file terpisah yang mencakup restoran-restoran di berbagai wilayah di provinsi Bali.

Karena data restoran berasal dari beberapa wilayah, setiap file berisi informasi restoran di wilayah tertentu, dan digabungkan untuk mempermudah pengolahan lebih lanjut.

Proses penggabungan data ini dilakukan dengan memanfaatkan pustaka *os* untuk mengetahui direktori tempat file-file tersebut disimpan. Langkah pertama adalah mendefinisikan sebuah variabel dataframe untuk menyimpan hasil penggabungan data. Kemudian, semua file CSV yang berada di direktori *output* dibaca satu per satu dan digabungkan ke dalam variabel dataframe tersebut. Setiap file yang berhasil dibaca akan disatukan dalam satu dataset yang lebih besar. Proses ini bertujuan untuk memastikan bahwa semua informasi dari berbagai file dapat diproses secara bersamaan pada langkah-langkah selanjutnya. Setelah data digabungkan, hasilnya disimpan ke dalam dataframe yang siap untuk diproses lebih lanjut dalam tahap berikutnya. Kode untuk proses penggabungan data ditunjukkan pada Gambar 4.9.

```
[2]: directory_path = "/Users/devbsi/Documents/gmaps-scraping/gmaps_scraper/output"

[3]: dataframes = []

    for file_name in os.listdir(directory_path):
        if file_name.endswith('.csv'):
            file_path = os.path.join(directory_path, file_name)
            df = pd.read_csv(file_path)
            dataframes.append(df)
```

Gambar 4.9 Kode pada proses *combining data*

4.2.2 Data Cleaning

Setelah data digabungkan, langkah berikutnya adalah melakukan pembersihan data atau *data cleaning*. Pembersihan data merupakan proses penting untuk memastikan bahwa hanya data yang valid dan relevan yang digunakan dalam penelitian ini. Pada tahap ini, dilakukan beberapa prosedur untuk menghilangkan anomali data, data ganda, dan informasi yang tidak diperlukan.

Tahap pertama dalam *data cleaning* adalah menghapus baris yang memiliki nilai kosong pada kolom *offerings* dan *review_text*. Kolom-kolom ini memuat informasi penting, seperti jenis penawaran restoran dan ulasan dari pelanggan. Baris yang tidak memiliki nilai di kedua kolom ini dianggap tidak relevan untuk penelitian dan dihapus. Setelah itu, baris-baris yang

mengandung nilai ganda dihapus. Duplikasi data dapat mempengaruhi hasil analisis, sehingga eliminasi data ganda bertujuan untuk meningkatkan akurasi dan efisiensi data.

Selain itu, untuk mempermudah pengolahan data teks pada kolom *review_text*, dilakukan penghapusan karakter digit (angka). Digit-digit tersebut biasanya tidak memiliki makna yang signifikan dalam analisis ulasan restoran, sehingga dihapus untuk mengurangi kompleksitas data. Setelah *data cleaning*, dataset yang diperoleh menjadi lebih bersih, lebih terstruktur, dan siap untuk dianalisis lebih lanjut. Kode untuk proses ini dapat dilihat pada Gambar 4.10.

```
[5]: combined_df.dropna(subset=['offerings', 'review_text'], inplace=True)
      combined_df.drop_duplicates(inplace=True)
      combined_df['review_text'] = combined_df['review_text'].str.replace(r'\d+', '', regex=True)
```

Gambar 4.10 Kode untuk proses *data cleaning*

4.2.3 Filtering

Filtering adalah tahap penting dalam *preprocessing* yang bertujuan untuk menyaring data agar lebih relevan dengan penelitian tentang sistem rekomendasi restoran halal. Pada tahap ini, data restoran disaring berdasarkan beberapa kriteria khusus yang sesuai dengan konteks halal.

Proses *filtering* dimulai dengan menyaring restoran yang mencantumkan penawaran berupa "Makanan halal" di bagian *offerings*. Restoran yang tidak memiliki penawaran makanan halal tidak relevan dengan penelitian ini, sehingga data restoran tersebut dihapus dari dataset. Penyaringan ini membantu memastikan bahwa hanya restoran yang menawarkan makanan halal yang akan dianalisis lebih lanjut. Gambar 4.11 menunjukkan kode untuk melakukan proses *filtering* untuk penawaran "Makanan halal".

```
[6]: halalOfferings = 'Makanan halal'
      halalResto_df = combined_df[combined_df['offerings'].str.contains(halalOfferings)]
      halalResto_df.shape
```

Gambar 4.11 Kode *filtering* makanan halal

Selain itu, terdapat beberapa restoran yang mencantumkan penawaran yang tidak sesuai dengan prinsip halal pada bagian *offerings*, seperti penawaran alkohol atau minuman keras lainnya. Untuk memastikan kehalalan restoran secara keseluruhan, dilakukan *filtering*

tambahan dengan menyaring restoran yang menawarkan minuman beralkohol, seperti "Alkohol", "Bir", "Anggur", dan "Koktail". Restoran yang mencantumkan penawaran tersebut akan dihapus dari dataset, karena kehadiran minuman keras dapat menimbulkan keraguan terkait status halal restoran secara keseluruhan.

Filtering dilakukan secara cermat untuk memastikan bahwa hanya restoran yang benar-benar halal, baik dari sisi makanan maupun minuman yang ditawarkan, yang tersisa dalam dataset. Proses ini sangat membantu dalam menjaga akurasi dan kualitas data yang digunakan dalam penelitian, serta memastikan bahwa rekomendasi yang diberikan kepada pengguna sesuai dengan preferensi halal yang menjadi fokus utama penelitian ini. Kode untuk proses *filtering* minuman beralkohol ditunjukkan pada Gambar 4.12

```
[8]: nonHalalOfferings = ['Alkohol', 'Anggur', 'Bir', 'Koktail']
      stringPattern = "|".join(nonHalalOfferings)
      halalResto_df = halalResto_df[~halalResto_df['offerings'].str.contains(stringPattern, na=False)]
      halalResto_df.shape
```

Gambar 4.12 Kode *filtering* minuman beralkohol

Setelah menyaring restoran berdasarkan *offerings*, dilakukan penyaringan berdasarkan ulasan pengguna yang terdapat pada kolom *review_text*. Untuk lebih memastikan bahwa restoran yang tersisa benar-benar menyajikan makanan halal, dilakukan penghapusan ulasan yang mengandung kata-kata tertentu yang berhubungan dengan makanan non-halal. Ulasan yang memuat kata-kata seperti "Babi", "Pork", dan "Bacon" dihapus dari dataset. Penghapusan ulasan dengan kata kunci tersebut membantu meminimalkan kemungkinan bahwa restoran yang tersisa menyajikan makanan non-halal, meskipun menawarkan makanan halal di bagian *offerings*.

Dengan melakukan *filtering* pada tahap ini, dataset yang digunakan untuk penelitian diharapkan hanya mencakup restoran yang secara konsisten menawarkan makanan halal dan tidak terlibat dalam penyajian makanan non-halal berdasarkan ulasan pelanggan. Tahapan ini sangat penting untuk menjaga relevansi data dengan topik penelitian dan meningkatkan akurasi dalam pengembangan sistem rekomendasi restoran halal. Kode untuk proses *filtering* berdasarkan kata kunci dapat dilihat pada Gambar 4.13.

```
[7]: halalReviewText = ['Babi', 'Pork', 'Bacon']
testPattern = "|".join(halalReviewText)
halalReview_df = halalResto_df[~halalResto_df['review_text'].str.contains(testPattern, case=False, na=False)]
halalReview_df.shape
```

Gambar 4.13 Kode *filtering* berdasarkan kata kunci

4.3 Splitting Data

Splitting data merupakan proses penting dalam pengembangan model, di mana data dibagi menjadi dua bagian utama, yaitu set pelatihan dan set pengujian. Tujuan utama dari pemisahan ini adalah memastikan bahwa model yang dilatih dapat bekerja dengan baik pada data baru yang belum pernah dilihat sebelumnya, sekaligus menghindari terjadinya bias selama proses pelatihan. Jika seluruh data digunakan untuk melatih model, ada risiko model menjadi terlalu spesifik terhadap data pelatihan atau *overfitting*, sehingga kinerjanya menurun saat dihadapkan pada data baru. Pada penelitian ini, metode *splitting* dilakukan menggunakan pendekatan tumpang tindih atau *overlap*, yang didesain khusus untuk menangani keterbatasan data yang tersedia, yaitu dengan membagi sebagian data yang sama ke dalam kedua set, baik pelatihan maupun pengujian.

Metode *overlap* ini menggunakan variabel *overlap_ratio* dengan nilai 0.5, yang berarti 50% dari interaksi seorang pengguna akan di *overlap*, atau dimasukkan ke dalam kedua set, yaitu pelatihan dan pengujian. Data pengguna dikelompokkan berdasarkan *user_name*. Jika seorang pengguna memiliki lebih dari satu interaksi, maka 50% dari interaksinya diambil secara acak untuk dimasukkan ke dalam kedua set. Sisa data yang tidak di *overlap* akan dipisahkan lagi, di mana 80% masuk ke dalam set pelatihan dan 20% ke set pengujian. Jika hanya tersisa satu interaksi setelah pengambilan data untuk *overlap*, interaksi tersebut akan dimasukkan ke dalam kedua set, baik pelatihan maupun pengujian, untuk tetap menjaga representasi pengguna. Pendekatan ini memungkinkan lebih banyak data dapat digunakan untuk melatih model tanpa kehilangan kemampuan untuk melakukan pengujian yang representatif. Adapun data dari pengguna yang hanya memiliki satu interaksi dimasukkan ke dalam kedua set, baik pelatihan maupun pengujian, untuk memastikan bahwa setiap pengguna tetap terwakili dalam proses pelatihan dan evaluasi model. Gambar 4.14 menunjukkan proses *splitting data*.

```
[4]: train_list = []
test_list = []

overlap_ratio = 0.5

for user_id, group in dataframe.groupby('user_id'):
    if len(group) > 1:
        n_overlap = int(len(group) * overlap_ratio)
        # Mengambil data acak untuk di overlap
        group_overlap = group.sample(n=n_overlap, random_state=8)
        # Menyimpan sisa data yang tidak di overlap
        group_remaining = group.drop(group_overlap.index)

        if len(group_remaining) > 1:
            group_train, group_test = train_test_split(group_remaining, test_size=0.2, random_state=8)
        else:
            group_train = group_remaining
            group_test = group_remaining

        train_list.append(pd.concat([group_train, group_overlap]))
        test_list.append(pd.concat([group_test, group_overlap]))
    elif len(group) == 1:
        train_list.append(group)
        test_list.append(group)

train_data = pd.concat(train_list, ignore_index=True)
test_data = pd.concat(test_list, ignore_index=True)
```

Gambar 4.14 Kode untuk proses *splitting data*

4.4 Cosine Similarity

Setelah data diproses, langkah selanjutnya adalah menghitung nilai kemiripan antar restoran menggunakan metode *cosine similarity*. *Cosine similarity* adalah salah satu metode yang umum digunakan dalam sistem rekomendasi karena mampu mengukur kemiripan antara dua vektor, dalam hal ini restoran, berdasarkan pola *rating* yang diberikan oleh pengguna. Metode ini bekerja dengan menghitung sudut kosinus antara dua vektor, di mana semakin kecil sudut antara vektor, semakin tinggi tingkat kemiripannya. Dalam konteks penelitian ini, *cosine similarity* diterapkan untuk menilai kesamaan antara restoran berdasarkan ulasan dan *rating* yang diberikan oleh pengguna. Restoran yang memiliki pola *rating* yang mirip akan dianggap serupa, sehingga dapat digunakan untuk memberikan rekomendasi yang lebih relevan kepada pengguna.

Untuk menerapkan *cosine similarity*, data *rating* dari pengguna terhadap restoran tertentu terlebih dahulu diubah ke dalam bentuk *user-item matrix*. Matriks ini dibuat menggunakan pivot table, di mana setiap baris mewakili pengguna, dan setiap kolom mewakili restoran. Nilai dalam matriks ini merupakan *rating* yang diberikan oleh pengguna terhadap restoran. Dengan demikian, *user_item_matrix* berfungsi sebagai representasi data yang menunjukkan interaksi pengguna dengan restoran. Setelah mendapatkan matriks ini, langkah selanjutnya adalah

menghitung kemiripan antar restoran menggunakan fungsi *cosine similarity* yang disediakan oleh pustaka *sklearn*.

Dalam perhitungan kemiripan antar restoran, diperlukan transpose dari matriks user-item, di mana data restoran diubah menjadi baris dan data pengguna menjadi kolom dan didefinisikan sebagai *user_item_matrix.T*. Hal ini penting karena *cosine similarity* menghitung kemiripan antar baris dalam matriks. Dengan melakukan transpose, kita dapat membandingkan restoran satu sama lain berdasarkan rating yang diberikan oleh pengguna. Setelah perhitungan *cosine similarity* selesai, hasilnya disimpan dalam bentuk *dataframe* untuk memudahkan analisis lebih lanjut. *Dataframe* ini memberikan kemiripan antar restoran yang dapat diakses dan digunakan dalam proses rekomendasi, di mana restoran dengan nilai kemiripan yang tinggi akan lebih mungkin direkomendasikan kepada pengguna yang memberikan *rating* serupa pada restoran lain. Kode untuk menghitung nilai kemiripan menggunakan metode *cosine similarity* ditunjukkan pada Gambar 4.15.

```
[5]: user_item_matrix = train_data.pivot(index='user_id', columns='restaurant_id', values='stars').fillna(0)
[6]: item_similarity = cosine_similarity(user_item_matrix.T)
     item_similarity_df = pd.DataFrame(item_similarity, index=user_item_matrix.columns, columns=user_item_matrix.columns)
```

Gambar 4.15 Kode untuk menghitung nilai *cosine similarity*

4.5 Autoencoder

Langkah selanjutnya dalam pengembangan model sistem rekomendasi restoran halal ini adalah menerapkan *autoencoder*, sebuah teknik *deep learning* yang sering digunakan dalam rekonstruksi data dan pengurangan dimensi. Dalam penelitian ini, *autoencoder* bekerja secara bertahap dengan mengurangi ukuran data melalui beberapa lapisan *dense*, yang merupakan lapisan jaringan saraf yang terhubung penuh dengan lapisan berikutnya untuk mentransfer informasi. Setiap lapisan ini menggunakan fungsi aktivasi ReLU (*Rectified Linear Unit*) yang umum digunakan karena sifatnya yang efektif dalam menangani data non-linear. Proses ini bertujuan untuk mengekstraksi pola tersembunyi dalam data pengguna, di mana ukuran data akan semakin kecil setelah melewati beberapa lapisan, hingga akhirnya mencapai ukuran yang sangat ringkas, yaitu 50 neuron. Pada titik ini, data pengguna dikompresi sehingga hanya informasi-informasi penting yang tersimpan, yang kemudian dapat digunakan untuk memahami preferensi pengguna terhadap berbagai restoran.

Untuk mencegah *overfitting*, yang sering menjadi masalah dalam model *deep learning*, digunakan lapisan *dropout*. *Dropout* berfungsi dengan secara acak mematikan beberapa neuron selama proses pelatihan, sehingga model tidak terlalu bergantung pada neuron-neuron tertentu. Ini membuat model lebih *general* dan mampu bekerja lebih baik saat dihadapkan dengan data baru. Setelah data pengguna dikompresi di bagian *encoder*, selanjutnya data akan masuk ke bagian *decoder*. *Decoder* bertanggung jawab untuk mengembalikan data yang telah dipadatkan ke bentuk aslinya, yaitu menghasilkan versi rekonstruksi dari input awal yang diberikan. Model ini dilatih untuk meminimalkan perbedaan antara data asli dan data hasil rekonstruksi dengan menggunakan fungsi *loss* MSE (*Mean Squared Error*). Fungsi ini mengukur seberapa besar kesalahan yang terjadi antara nilai asli dan nilai yang diprediksi oleh model, dan tujuannya adalah untuk mengurangi kesalahan tersebut sebanyak mungkin. Model dilatih selama 20 *epoch* dengan *batch_size* sebesar 256.

Dengan cara ini, *autoencoder* tidak hanya mampu mengenali pola-pola penting dalam preferensi pengguna, tetapi juga dapat mengisi nilai-nilai yang belum ada atau hilang dalam matriks user-item. Misalnya, jika pengguna belum memberikan ulasan atau *rating* untuk beberapa restoran, *autoencoder* dapat memperkirakan nilai tersebut berdasarkan pola interaksi pengguna lainnya. Hal ini sangat berguna dalam meningkatkan kualitas rekomendasi, karena model bisa secara otomatis menyarankan restoran yang kemungkinan besar sesuai dengan preferensi pengguna, meskipun data interaksi mereka belum lengkap. Keunggulan utama dari pendekatan *autoencoder* ini adalah kemampuannya dalam menyederhanakan data kompleks sambil tetap menjaga informasi penting, yang pada akhirnya membuat sistem rekomendasi lebih akurat dan relevan bagi pengguna. Gambar 4.16 menunjukkan kode untuk proses *autoencoder*.

```
[7]: input_dim = user_item_matrix.shape[1]
      encoding_dim = 50

      input_layer = Input(shape=(input_dim,))

      encoder = Dense(256, activation='relu')(input_layer)
      encoder = Dense(128, activation='relu')(encoder)
      encoder = Dense(64, activation='relu')(encoder)
      encoder = Dropout(0.2)(encoder)
      encoder = Dense(encoding_dim, activation='relu')(encoder)

      decoder = Dense(64, activation='relu')(encoder)
      decoder = Dense(128, activation='relu')(decoder)
      decoder = Dense(256, activation='relu')(decoder)
      decoder = Dense(input_dim, activation='sigmoid')(decoder)

      autoencoder = Model(inputs=input_layer, outputs=decoder)
      autoencoder.compile(optimizer=Adam(learning_rate=0.0001), loss='mse')

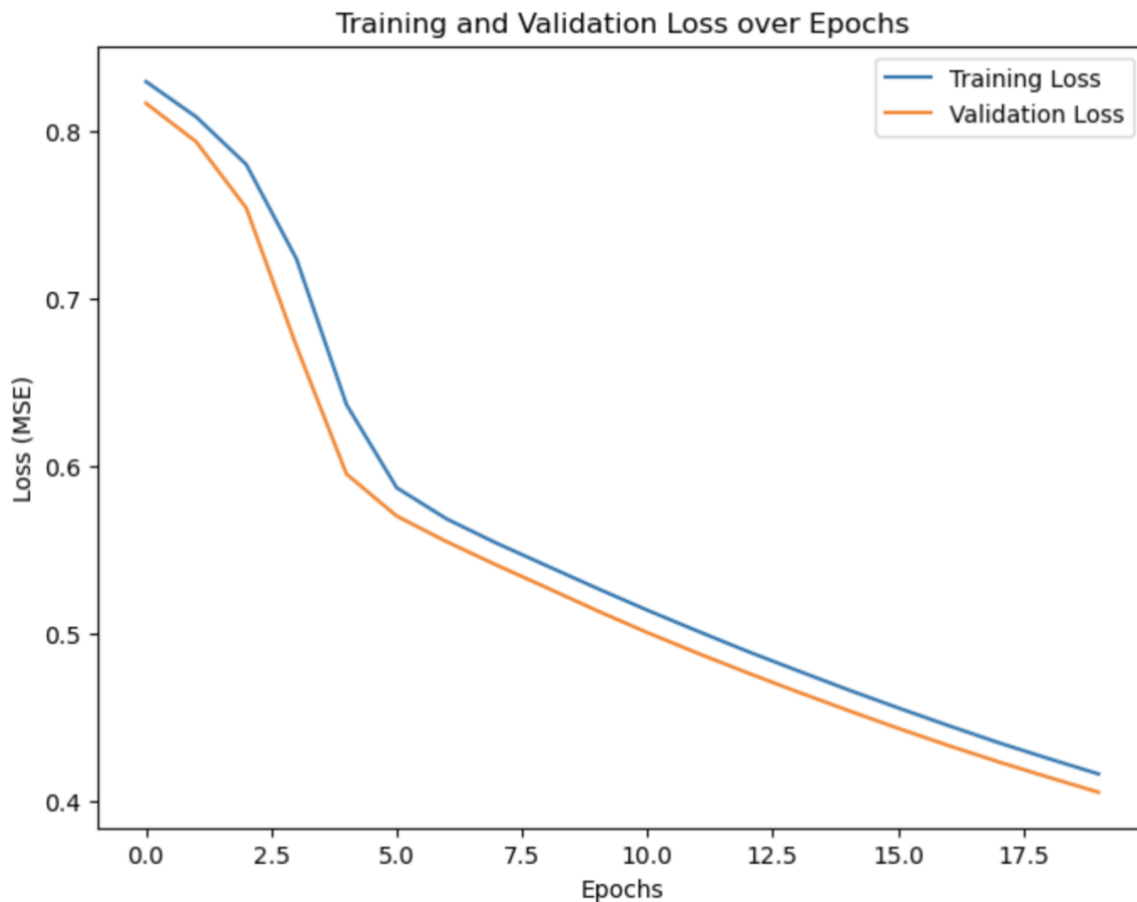
      history = autoencoder.fit(user_item_matrix.values, user_item_matrix.values,
                               epochs=20, shuffle=True, batch_size=256, validation_split=0.2, verbose=1)
```

Gambar 4.16 Kode untuk proses *autoencoder*

Dari model yang telah dilatih, kita dapat memvisualisasikan perubahan *loss (error)* seiring bertambahnya jumlah iterasi atau *epochs*. Sumbu X mewakili jumlah iterasi pelatihan (*epochs*) dan sumbu Y menunjukkan nilai *loss* yang dihitung menggunakan metrik MSE. Semakin kecil nilai *loss*, semakin baik model memprediksi output.

Dari grafik pada Gambar 4.17, garis berwarna biru mewakili *training loss*, yaitu *loss* pada data pelatihan. Garis ini terus menurun, yang menandakan model semakin baik dalam mengenali pola dari data pelatihan. Sementara itu, garis berwarna oranye menggambarkan *validation loss*, yang menunjukkan *error* model saat diuji pada data validasi (data yang belum dilihat oleh model). Penurunan stabil pada *validation loss*, menandakan bahwa model tidak hanya belajar dari data pelatihan, tetapi juga dapat melakukan prediksi yang cukup baik pada data baru (validasi).

Terlihat juga bahwa kedua kurva menurun dengan stabil, yang menunjukkan bahwa model beradaptasi dengan baik terhadap data tanpa mengalami *overfitting*. Tidak ada tanda *overfitting* yang jelas, karena *training loss* dan *validation loss* saling mengikuti dengan cukup baik. *Overfitting* biasanya terjadi jika *validation loss* mulai meningkat sementara *training loss* terus menurun, tetapi di sini, kedua kurva tetap berdekatan. Hal ini menandakan bahwa model memiliki performa yang baik, mampu mempelajari pola dari data pelatihan dan memprediksi dengan baik pada data validasi atau data yang belum pernah dilihat.



Gambar 4.17 Grafik *loss* selama pelatihan model *autoencoder*

Selanjutnya, setelah model *autoencoder* dilatih, kita dapat menggunakan model tersebut untuk menghasilkan rekonstruksi dari matriks user-item yang telah kita buat sebelumnya. Proses ini memungkinkan kita untuk membuat prediksi mengenai preferensi pengguna terhadap restoran yang belum mereka ulas atau beri rating. Model *autoencoder*, yang sudah dilatih untuk memahami pola tersembunyi dalam data pengguna, akan merekonstruksi matriks user-item dengan mengisi nilai-nilai yang hilang berdasarkan hubungan antar restoran dan pengguna yang telah dipelajari selama pelatihan.

Hasil rekonstruksi ini disimpan dalam variabel *reconstructed_matrix*, yang berbentuk *array NumPy*. Matriks ini berisi prediksi rating atau interaksi yang dihasilkan oleh model *autoencoder*. Variabel *index* digunakan untuk memastikan bahwa dataframe hasil rekonstruksi memiliki indeks dan kolom yang sama dengan *user_item_matrix*, sehingga setiap baris masih sesuai dengan pengguna dan setiap kolom sesuai dengan restoran dalam dataset. Dengan struktur yang konsisten ini, kita dapat dengan mudah menganalisis, membandingkan, dan menggunakan matriks rekonstruksi untuk memberikan rekomendasi yang lebih akurat kepada

pengguna berdasarkan prediksi model. Kode rekonstruksi matriks *user-item* dapat dilihat pada Gambar 4.18.

```
[8]: reconstructed_matrix = autoencoder.predict(user_item_matrix.values)
reconstructed_df = pd.DataFrame(reconstructed_matrix, index=user_item_matrix.index, columns=user_item_matrix.columns)
595/595 ————— 1s 1ms/step
```

Gambar 4.18 Kode rekonstruksi matriks *user-item*

4.6 Rekomendasi

Setelah *autoencoder* diterapkan dalam model, selanjutnya adalah proses rekomendasi. Disajikan 10 rekomendasi restoran halal kepada pengguna berdasarkan kemiripan dengan pengguna lain. *restaurant_dict* merupakan *dictionary* yang digunakan untuk mengambil nama restoran berdasarkan id nya. Fungsi *get_restaurants()* bertugas mengembalikan hasil rekomendasi dalam bentuk dataframe, yang terdiri dari tiga kolom, yaitu id restoran, nama restoran, dan skor kemiripan masing-masing restoran.

Fungsi *recommend_with_autoencoder()* digunakan ketika pengguna baru belum memiliki riwayat data. Fungsi ini membuat input baru berupa vektor nol yang mewakili pengguna baru, lalu memprediksi skor restoran menggunakan model *autoencoder*. Skor yang dihasilkan diurutkan dari yang tertinggi ke terendah. Restoran dengan skor tertinggi diambil sebanyak jumlah rekomendasi yang diminta. Selanjutnya, fungsi ini memanggil *get_restaurants()* dengan parameter berupa daftar rekomendasi dan skor yang telah dihitung.

Langkah selanjutnya adalah membuat fungsi untuk memberikan rekomendasi restoran halal. Jika pengguna belum memiliki riwayat data di *user_item_matrix*, fungsi akan memanggil *recommend_with_autoencoder()* untuk memberikan rekomendasi menggunakan model *autoencoder*. Namun, jika pengguna memiliki data, variabel *user_ratings* akan mengambil rating pengguna dari matriks *user_item_matrix*. Variabel *similar_scores* dan *autoencoder_scores* masing-masing menghitung nilai kemiripan sebelum dan setelah dilatih oleh model *autoencoder*. Selanjutnya kedua nilai tersebut dijumlahkan dan diurutkan berdasarkan nilai tertinggi. Sebelum fungsi mengambil restoran teratas sebanyak *num_recommendations* dan skor mereka untuk direkomendasikan. Kode untuk mencari rekomendasi ditunjukkan pada Gambar 4.19.

```

[11]: restaurant_names = train_data[['restaurant_id', 'restaurant_name']].drop_duplicates()
       restaurant_dict = restaurant_names.set_index('restaurant_id')['restaurant_name'].to_dict()

[12]: def get_restaurants(recommendations, scores):
       recommendations_df = pd.DataFrame(columns=['id restoran', 'nama restoran', 'skor'])
       count = 0

       for res_id, score in zip(recommendations, scores):
           recommendations_df.at[count, 'id restoran'] = res_id
           recommendations_df.at[count, 'nama restoran'] = restaurant_dict[res_id]
           recommendations_df.at[count, 'skor'] = score
           count += 1

       return recommendations_df

[13]: def recommend_with_autoencoder(num_recommendations):
       new_user_input = np.zeros((1, user_item_matrix.shape[1]))

       reconstructed_scores = autoencoder.predict(new_user_input)[0]

       restaurant_scores = pd.Series(reconstructed_scores, index=user_item_matrix.columns)
       restaurant_scores = restaurant_scores.sort_values(ascending=False)

       recommendations = restaurant_scores.head(num_recommendations).index
       scores = restaurant_scores.head(num_recommendations).values

       return get_restaurants(recommendations, scores)

[14]: def hybrid_recommendations(user_name, num_recommendations=10):
       if user_name not in user_item_matrix.index:
           return recommend_with_autoencoder(num_recommendations)

       user_ratings = user_item_matrix.loc[user_name]

       similar_scores = item_similarity_df.dot(user_ratings) / item_similarity_df.sum(axis=1)
       autoencoder_scores = reconstructed_df.loc[user_name]

       hybrid_scores = similar_scores + autoencoder_scores
       hybrid_scores = hybrid_scores.sort_values(ascending=False)

       # Menghapus rekomendasi pertama
       hybrid_scores = hybrid_scores.iloc[1:]

       recommendations = hybrid_scores.head(num_recommendations).index
       scores = hybrid_scores.head(num_recommendations).values

       return get_restaurants(recommendations, scores)

```

Gambar 4.19 Kode untuk mencari rekomendasi

Setelah sistem rekomendasi berhasil diimplementasikan, sistem dapat menampilkan daftar 10 restoran halal teratas yang direkomendasikan kepada setiap pengguna. Rekomendasi ini didasarkan pada kemiripan preferensi pengguna dengan pengguna lain yang memiliki pola ulasan atau *rating* yang serupa. Sistem menggunakan teknik *cosine similarity* untuk mengukur tingkat kemiripan antar pengguna, sehingga restoran yang lebih relevan akan diutamakan dalam daftar rekomendasi. Hasil rekomendasi ini kemudian ditampilkan dalam bentuk daftar yang diurutkan berdasarkan tingkat relevansi tertinggi, sebagaimana ditunjukkan pada Gambar 4.20. Dengan pendekatan ini, pengguna dapat menerima saran restoran yang paling sesuai

dengan preferensi mereka, meningkatkan pengalaman mereka dalam menemukan restoran halal yang diinginkan.

```
[18]: user_name = "Gede Eka"

[19]: print(f'Recommendation for user {user_name}: ')
      hybrid_recommendations(user_name)

Recommendation for user Gede Eka:
```

	id restoran	nama restoran	skor
0	0x2dd21645d2a8f22b:0x941d5332748be8c8	Rumah Makan Dharma Giri	0.223367
1	0x2dd2458835cc38a1:0xb3539b222a85e9	Minang Raya Masakan Padang	0.153567
2	0x2dd2411da6a41b05:0xc26ed909196114e5	Raa Cha Suki And BBQ - Level 21 Mall	0.14809
3	0x2dd2409546bf52fd:0x1f36859c78175f51	MM Juice	0.104793
4	0x2dd2166024343f7b:0x539ba5014fbda700	Rumah Makan Padang Ampera Gianyar	0.096301
5	0x2dd2476aa1775e77:0xcb4c01f571e2907c	Pizza Hut Delivery - PHD Kerobokan Bali	0.094294
6	0x2dd240f3bcfdb8b:0x4b4d9364f4f396ad	X.O Suki & Cuisine	0.019738
7	0x2dd2412e8a72f64b:0x9db60350d5fc8058	Grande Bistro Bali - Restoran & Cafe Terdekat ...	0.016124
8	0x2dd244c1eb9d14a1:0xd84fd2f5d30d34d2	D'SAMBAL & Uluwatu GWK, nasi tumpeng, nasi box...	0.013922
9	0x2dd2409537213425:0x13528bbc31ed3634	Ichiban Sushi Level 21 Denpasar	0.012151

Gambar 4.20 Daftar rekomendasi 10 restoran halal teratas

4.7 Evaluasi

Tahap terakhir dalam pembuatan sistem rekomendasi restoran halal adalah evaluasi model. Evaluasi ini bertujuan untuk mengukur kinerja sistem dalam memberikan rekomendasi yang akurat kepada pengguna. Salah satu metrik yang digunakan untuk mengukur akurasi model adalah RMSE (*Root Mean Squared Error*), yang merupakan indikator seberapa besar rata-rata kesalahan prediksi dibandingkan dengan *rating* sebenarnya yang diberikan oleh pengguna. RMSE menghitung akar dari rata-rata kuadrat perbedaan antara nilai prediksi dan nilai sebenarnya, memberikan gambaran tentang seberapa baik model menangkap pola *rating* pengguna.

Pada evaluasi yang dilakukan dalam penelitian ini, sistem rekomendasi menghasilkan nilai RMSE dibawah angka 1, yang menunjukkan performa yang cukup baik. Nilai RMSE yang rendah ini menggambarkan bahwa rata-rata kesalahan prediksi relatif kecil, sehingga sistem mampu memberikan rekomendasi yang cukup akurat kepada pengguna. Namun, sistem belum benar-benar teruji saat dihadapkan dengan data yang beragam. Penggunaan teknik *overlap* untuk membagi data menyebabkan data pengujian terlalu mirip dengan data pelatihan. Hal ini dapat membuat nilai RMSE tampak lebih baik dari performa sebenarnya, di mana data baru

tidak akan tumpang tindih dengan data sebelumnya. Meskipun demikian, seperti yang ditunjukkan pada Gambar 4.21, evaluasi ini menunjukkan nilai yang kecil pada data yang terbatas dan sistem tetap dapat memberikan saran restoran yang relevan dan sesuai dengan kebutuhan pengguna.

```
[15]: def get_user_predicted_ratings(user_name):
    if user_name not in user_item_matrix.index:
        return pd.Series(0, index=user_item_matrix.columns)

    user_ratings = user_item_matrix.loc[user_name]

    similar_scores = item_similarity_df.dot(user_ratings) / item_similarity_df.sum(axis=1)
    autoencoder_scores = reconstructed_df.loc[user_name]

    hybrid_scores = similar_scores + autoencoder_scores

    return hybrid_scores

[16]: def evaluate_rmse(test_data):
    predicted_ratings = []
    actual_ratings = []

    for index, row in test_data.iterrows():
        user_name = row['user_name']
        restaurant_id = row['restaurant_id']
        actual_rating = row['stars']

        predicted_rating = get_user_predicted_ratings(user_name).get(restaurant_id, 0)

        predicted_ratings.append(predicted_rating)
        actual_ratings.append(actual_rating)

    # agar rating prediksi berada pada rentang 1 - 5
    predicted_ratings = np.clip(predicted_ratings, 1, 5)
    rmse = sqrt(mean_squared_error(actual_ratings, predicted_ratings))

    return rmse, actual_ratings, predicted_ratings

rmse_value, actual_ratings, predicted_ratings = evaluate_rmse(test_data)
print(f"RMSE: {rmse_value}")
RMSE: 0.5828017706959349
```

Gambar 4.21 Kode evaluasi menggunakan RMSE

4.8 Hasil Penelitian

Penelitian ini bertujuan untuk mengimplementasikan sistem rekomendasi restoran halal berbasis *collaborative filtering* dan *autoencoder* menggunakan *cosine similarity* untuk menghitung kemiripan antar restoran. Sistem yang memanfaatkan data dari *Google Maps* ini berhasil menyediakan referensi dan rekomendasi baru kepada pengguna tentang restoran yang serupa dengan restoran yang mereka minati.

4.8.1 Hasil Uji Program

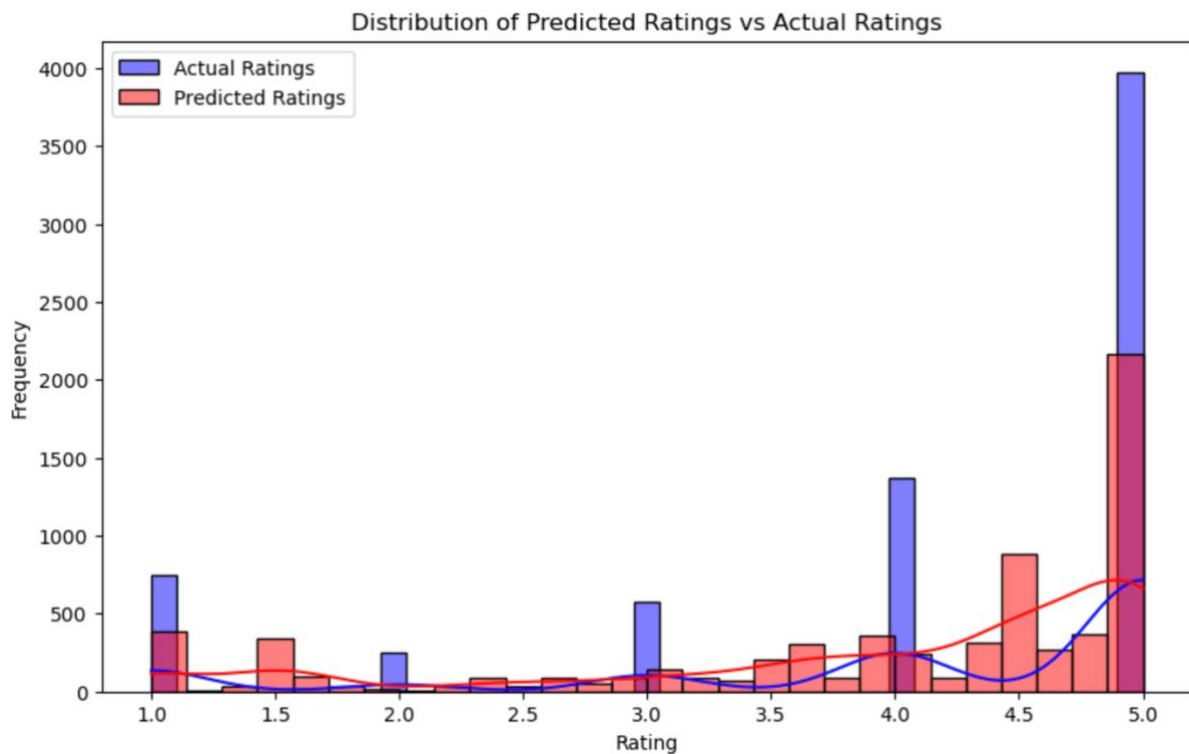
Pada penelitian ini, pengujian sistem dilakukan dengan menggunakan metrik *Root Mean Squared Error* (RMSE) untuk mengukur seberapa besar rata-rata kesalahan prediksi terhadap *rating* sebenarnya yang diberikan pengguna. Hasil pengujian menunjukkan bahwa sistem memiliki nilai RMSE sebesar 0.58280. Nilai RMSE yang rendah ini menunjukkan bahwa sistem mampu memberikan rekomendasi dengan tingkat akurasi yang cukup baik pada data yang terbatas.

Guna membuktikan nilai RMSE, dilakukan pengujian manual untuk membandingkan *rating* asli pengguna dengan *rating* prediksi. *Rating* aktual atau *rating* asli pengguna diambil dari set pengujian yang telah didefinisikan di awal, sedangkan *rating* prediksi didapat setelah proses pelatihan model. Pengujian ini dilakukan untuk mengetahui apakah model memberikan prediksi yang masuk akal. Sebagai contoh, diambil 10 indeks dari masing-masing data, baik *rating* aktual maupun *rating* prediksi. Tabel 4.1 menunjukkan perbandingan *rating* aktual dan *rating* prediksi.

Tabel 4.1 Perbandingan *rating* aktual dan *rating* prediksi

No	<i>rating</i> aktual	<i>rating</i> prediksi
1	4.0	3.89902987
2	5.0	4.74853651
3	2.0	2.14259541
4	5.0	4.9281588
5	5.0	5.0
6	5.0	5.0
7	5.0	5.0
8	5.0	4.015721
9	5.0	4.74853651
10	5.0	4.98706748

Sementara itu, distribusi dari *rating* aktual dan *rating* prediksi secara keseluruhan dapat dilihat pada Gambar 4.22.



Gambar 4.22 Distribusi *rating* aktual dan *rating* prediksi

4.8.2 Pembahasan

Nilai metrik RMSE sebesar 0.58280 menunjukkan bahwa sistem rekomendasi yang dibangun memiliki tingkat akurasi yang cukup tinggi dalam memprediksi *rating*. Nilai RMSE yang lebih rendah mengindikasikan bahwa rata-rata kesalahan prediksi sistem terhadap *rating* aktual relatif kecil, yang menunjukkan kemampuan model untuk merekomendasikan item yang sesuai dengan preferensi pengguna. RMSE dalam konteks ini memberikan informasi tentang seberapa jauh prediksi dari nilai *rating* sebenarnya, dengan penalti yang lebih besar diberikan untuk kesalahan prediksi yang lebih ekstrem.

Hal tersebut dibuktikan pada Tabel 4.1, terlihat bahwa nilai prediksi model sebagian besar mendekati *rating* aktual yang diberikan oleh pengguna. Seperti pada indeks ke-2 dan ke-4 yang masing-masing bernilai 4.748 dan 4.928 dari nilai sebenarnya, yaitu 5.0. Meskipun ada kasus dimana perbedaan nilai cukup besar, seperti pada indeks ke-8, yang mempunyai nilai prediksi 4.015 dari 5.0. Hal ini dapat diakibatkan oleh kurangnya interaksi pengguna terhadap restoran. Namun, perbedaan nilai yang signifikan hanya terjadi pada beberapa kasus dan membuktikan bahwa model tetap bekerja dengan baik dalam banyak kasus.

Dari distribusi *rating* aktual dan prediksi yang diperlihatkan pada Gambar 4.22, dapat dilihat bahwa prediksi dari sistem cukup konsisten dengan pola distribusi *rating* aktual. Meskipun ada beberapa perbedaan kecil, model tampak berhasil menjaga keseimbangan dalam prediksi untuk skala *rating* 1 sampai 5.

4.8.3 Kelebihan dan Kekurangan

Kelebihan

1. Sistem dapat memberikan rekomendasi restoran dengan jumlah yang cukup banyak.
2. Pengambilan data dilakukan secara otomatis dari *Google Maps*, sehingga efisien dalam mendapatkan informasi terbaru mengenai restoran.
3. Sistem menyediakan rekomendasi restoran populer bagi pengguna yang tidak memiliki riwayat data, membuat sistem tetap berguna meskipun pengguna baru.
4. Penggunaan *autoencoder* membantu model untuk mengisi data yang hilang dan memprediksi preferensi pengguna dengan lebih baik.

Kekurangan

1. Sistem bergantung pada proses *scraping* dari *Google Maps*, yang bisa mengalami kendala jika terjadi perubahan pada struktur situs atau kebijakan akses data.
2. Penyaringan restoran halal berdasarkan ulasan pengguna dengan kata-kata tertentu seperti "Babi", "Pork", "Bacon" bisa menyebabkan kesalahan, terutama jika pengguna memberikan ulasan ambigu atau tidak relevan dengan status halal restoran.
3. Penggunaan dataset terbatas pada satu wilayah (Bali) mungkin membuat rekomendasi kurang variatif atau kurang relevan bagi pengguna di luar area tersebut.
4. Penggunaan teknik *overlap* dapat menyebabkan data pengujian terlalu mirip dengan data pelatihan, sehingga menyebabkan nilai evaluasi tampak lebih baik daripada performa sebenarnya.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini berhasil menciptakan sebuah sistem rekomendasi restoran halal dengan menerapkan metode *collaborative filtering* yang didukung oleh *cosine similarity* dan *autoencoder*. Data diambil dari *Google Maps* melalui teknik *scraping*, mencakup informasi seperti nama restoran, ulasan pengguna, penawaran, dan rating pengguna. *Cosine similarity* digunakan untuk mengukur kesamaan antar restoran berdasarkan rating pengguna, sedangkan *autoencoder* membantu meningkatkan kualitas data dan akurasi rekomendasi. Sistem ini mampu memberikan rekomendasi yang relevan, termasuk kepada pengguna baru.

Hasil evaluasi model menggunakan RMSE menunjukkan performa yang baik dengan tingkat *error* yang rendah. Selain itu sistem telah berhasil memberikan rekomendasi restoran halal yang populer untuk pengguna baru tanpa riwayat data sebelumnya. Namun, penggunaan teknik *overlap* dalam pembagian data menyebabkan data pengujian dan pelatihan memiliki kemiripan, sehingga sistem ini belum teruji terhadap data yang beragam. Meskipun demikian, penelitian ini memberikan solusi yang bermanfaat untuk merekomendasikan restoran halal kepada pengguna berdasarkan preferensi mereka.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, ada beberapa saran yang dapat diterapkan untuk penelitian selanjutnya, yaitu:

- a. Jumlah dan kualitas data yang digunakan untuk sistem rekomendasi restoran halal dapat ditingkatkan dan diperluas.
- b. Pengumpulan data dapat diperluas ke lebih banyak daerah di luar Bali untuk membuat sistem lebih relevan bagi pengguna dari berbagai wilayah.
- c. Penambahan variabel seperti menu makanan, yang dapat membantu sistem merekomendasikan restoran halal lebih akurat.
- d. Pembaruan pada kode *scraping* secara berkala agar pengumpulan data tetap berjalan dengan baik, mengingat sistem *scraping* rentan terhadap perubahan struktur dan kebijakan *Google Maps*.

DAFTAR PUSTAKA

- Abdullah, M. R., Arno, A. K., Kamal, H., & Jamaluddin, F. (2019). *HALAL LABELISATION OF HARAM FOOD IN MAKALE TORAJA*.
- Alfarizi, M. R. S., Al-farish, M. Z., Taufiqurrahman, M., Ardiansah, G., & Elgar, M. (2023). Penggunaan Python Sebagai Bahasa Pemrograman untuk Machine Learning dan Deep Learning. *Karya Ilmiah Mahasiswa Bertauhid (KARIMAH TAUHID)*, 2(1), 1–6.
- Ali, M. (2016). Konsep Makanan Halal dalam Tinjauan Syariah dan Tanggung Jawab Produk Atas Produsen Industri Halal. *AHKAM: Jurnal Ilmu Syariah*, 16(2), 291–306. <https://doi.org/10.15408/ajis.v16i2.4459>
- Altan, A., & Karasu, S. (2019). the Effect of Kernel Values in Support Vector Machine To Forecasting Performance of Financial Time Series. *The Journal of Cognitive Systems*, 4(1), 17–21. <https://dergipark.org.tr/en/pub/jcs/570863>
- Arbar, T. (2024). *Berburu Makanan Halal di Singapura, Perhatikan Label Ini*. CNBC Indonesia. <https://www.cnbcindonesia.com/news/20240327200847-4-526120/berburu-makanan-halal-di-singapura-perhatikan-label-ini>
- Asani, E., Vahdat-Nejad, H., & Sadri, J. (2021). Restaurant recommender system based on sentiment analysis. *Machine Learning with Applications*, 6(July), 100114. <https://doi.org/10.1016/j.mlwa.2021.100114>
- B.Thorat, P., M. Goudar, R., & Barve, S. (2015). Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System. *International Journal of Computer Applications*, 110(4), 31–36. <https://doi.org/10.5120/19308-0760>
- Budi, M. (2023). *Viral Pembeli Muslim Marah Disajikan Pasta Babi, Pihak Restoran Minta Maaf*. News Detik.Com. <https://news.detik.com/berita/d-6771136/viral-pembeli-muslim-marah-disajikan-pasta-babi-pihak-restoran-minta-maaf>
- Chassagnon, G., Vakalopolou, M., Paragios, N., & Revel, M. P. (2020). Deep learning: definition and perspectives for thoracic imaging. *European Radiology*, 30(4), 2021–2030. <https://doi.org/10.1007/s00330-019-06564-3>
- Dong, S., Wang, P., & Abbas, K. (2021). A survey on deep learning and its applications. *Computer Science Review*, 40, 100379. <https://doi.org/10.1016/j.cosrev.2021.100379>
- Effendy, F., Nuqoba, B., & Taufik. (2019). Culinary recommendation application based on user preferences using fuzzy topsis. *IJUM Engineering Journal*, 20(2), 163–175. <https://doi.org/10.31436/iiumej.v20i2.1023>

- Fakhri, A. A., Baizal, Z. K. A., & Setiawan, E. B. (2019). Restaurant Recommender System Using User-Based Collaborative Filtering Approach: A Case Study at Bandung Raya Region. *Journal of Physics: Conference Series*, 1192(1). <https://doi.org/10.1088/1742-6596/1192/1/012023>
- García-Sánchez, F., Colomo-Palacios, R., & Valencia-García, R. (2020). A social-semantic recommender system for advertisements. *Information Processing and Management*, 57(2), 102153. <https://doi.org/10.1016/j.ipm.2019.102153>
- Gerasimou, G., & Papi, M. (2018). Duopolistic competition with choice-overloaded consumers. *European Economic Review*, 101, 330–353. <https://doi.org/10.1016/J.EUROECOREV.2017.10.002>
- Gunantohadi, T., & Crysdiyan, C. (2022). Review Penerapan Metode Klasifikasi Pada Sistem Rekomendasi Sosial Kemasyarakatan. *Jurnal Aplikasi Teknologi Informasi Dan Manajemen (JATIM)*, 3(2), 84–91. <https://doi.org/10.31102/jatim.v3i2.1578>
- Khder, M. A. (2021). Web scraping or web crawling: State of art, techniques, approaches and application. *International Journal of Advances in Soft Computing and Its Applications*, 13(3), 144–168. <https://doi.org/10.15849/ijasca.211128.11>
- Lopez Pinaya, W. H., Vieira, S., Garcia-Dias, R., & Mechelli, A. (2019). Autoencoders. In *Machine Learning: Methods and Applications to Brain Disorders*. Elsevier Inc. <https://doi.org/10.1016/B978-0-12-815739-8.00011-0>
- Mahadi, M. I., Zainuddin, N., Bazamah, N., Shah, A., Asyira Naziron, N., Fatimah, S., & Rum, M. (2018). E-Halal Restaurant Recommender System Using Collaborative Filtering Algorithm. *Journal of Advanced Research in Computing and Applications Journal Homepage*, 12(1), 22–34. www.akademiabaru.com/arca.html
- Mahardika, Ri., Yuliendi, R. R., & Yandri, Y. (2022). Sistem Rekomendasi Pemilihan Lokasi Kuliner Halal Area Kota Pekanbaru Menggunakan Metode Weighted Product. *JEKIN - Jurnal Teknik Informatika*, 2(3), 175–185. <https://doi.org/10.58794/jekin.v3i1.419>
- Mana, S. C., & Sasipraba, T. (2021). Research on cosine similarity and pearson correlation based recommendation models. *Journal of Physics: Conference Series*, 1770(1). <https://doi.org/10.1088/1742-6596/1770/1/012014>
- Martins, G. B., Papa, J. P., & Adeli, H. (2020). Deep learning techniques for recommender systems based on collaborative filtering. *Expert Systems*, 37(6), 1–21. <https://doi.org/10.1111/exsy.12647>

- Mohamed, M. H., Khafagy, M. H., & Ibrahim, M. H. (2019). Recommender Systems Challenges and Solutions Survey. *Proceedings of 2019 International Conference on Innovative Trends in Computer Engineering, ITCE 2019, February*, 149–155. <https://doi.org/10.1109/ITCE.2019.8646645>
- Mondi, R. H., & Wijayanto, A. (2019). Recommendation System With Content-Based Filtering Method for Culinary Tourism in Mangan Application. *ITSMART: Jurnal Ilmiah Teknologi Dan Informasi*, 8(2), 65–72.
- Munir, S., & Asqia, M. (2021). Implementasi Skyline Query pada Sistem Rekomendasi Pemilihan Tempat Kuliner di Kota Depok, Bogor, dan Tangerang. *Jurnal Teknologi Terpadu*, 7(2), 113–119. <https://doi.org/10.54914/jtt.v7i2.440>
- Mutiah, D. (2023). *Pengunjung Restoran Nyaris Tertipu Tanda No Pork No Lard, Dikira Tak Sajikan Makanan Mengandung Babi*. Liputan6.Com. <https://www.liputan6.com/lifestyle/read/5212028/pengunjung-restoran-nyaris-tertipu-tanda-no-pork-no-lard-dikira-tak-sajikan-makanan-mengandung-babi?page=3>
- Nurhayati, S. D., & Widayani, W. (2021). *Sistem Rekomendasi Wisata Kuliner di Yogyakarta dengan Metode Item-Based Collaborative Filtering*. 1(2), 55–63.
- Sahara, S., & Baskoro, A. P. (2019). Perancangan Logo Dan Corporate Identity Rumah Makan Biyunge “Ciri Khas Jawa.” *J-Ika*, 6(1), 24–32. <https://doi.org/10.31294/kom.v6i1.6184>
- Sandi, S., Made Mahendra, I. B., Muliantara, A., Gede Suhartana, I. K., Santiyasa, I. W., & Rahning Putri, L. A. A. (2022). Implementation of the Weighted Product Method for Recommendations Halal Dinning Places in Bali. *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, 11(3), 563. <https://doi.org/10.24843/jlk.2023.v11.i03.p12>
- Shao, B., Li, X., & Bian, G. (2021). A survey of research hotspots and frontier trends of recommendation systems from the perspective of knowledge graph. *Expert Systems with Applications*, 165, 113764. <https://doi.org/10.1016/j.eswa.2020.113764>
- Sihombing, O., Sihombing, S., Pasaribu, M. L., Kris, R., & Saragih, D. (2020). Website Rekomendasi Tempat Kuliner dengan Metode Social Trust Path. *Universitas Prima Indonesia Jl. Sekip, Petisah Tengah, Kec. Medan Petisah*, 3(1), 1–6.
- Singhal, A., Sinha, P., & Pant, R. (2017). Use of Deep Learning in Modern Recommendation System: A Summary of Recent Works. *International Journal of Computer Applications*, 180(7), 17–22. <https://doi.org/10.5120/ijca2017916055>
- Venil, P., Vinodhini, G., & Suban, R. (2020). A state of the art survey on cold start problem in a collaborative filtering system. *International Journal of Scientific and Technology*

Research, 9(4), 2606–2612.

Yusoff, S. Z., & Adzharuddin, N. A. (2017). Factor of Awareness in Searching and Sharing of Halal Food Product among Muslim Families in Malaysia. *SHS Web of Conferences*, 33, 00075. <https://doi.org/10.1051/shsconf/20173300075>

LAMPIRAN