

LAPORAN PROYEK UAS
PEMBELAJARAN MESIN (PRAKTIKUM)

"Sistem Rekomendasi Produk Skincare Berbasis Content-Based Filtering"



KELOMPOK 3

ANGGOTA:

[434221052] | [Amelia Nur Indah P]

[434221054] | [Andyka Salom]

[434221057] | Lintang Angraini N.LR]

PROGRAM STUDI D4 TEKNIK INFORMATIKA

**FAKULTAS VOKASI
UNIVERSITAS AIRLANGGA
SURABAYA**

2024

DAFTAR ISI

DAFTAR ISI.....	1
DAFTAR GAMBAR.....	3
BAB 1	
PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan.....	3
1.4 Manfaat.....	4
1.5 Batasan Masalah.....	4
BAB 2	
TINJAUAN PUSTAKA.....	6
2.1 Sistem Rekomendasi.....	6
2.2 Machine Learning.....	6
2.3 Content Based Filtering (CBF).....	7
2.4 Cosine Similarity.....	7
2.5 Skincare.....	8
BAB 3	
METODOLOGI.....	9
3.1 Desain Penelitian.....	9
3.2 Pengumpulan Dataset.....	10
3.3 Pre-Pemrosesan Data.....	11
3.4 Pra-Pemrosesan Data.....	12
3.5 Model Content-Based Filtering.....	12
3.6 Evaluasi Sistem.....	13
3.7 Implementasi Sistem.....	13
BAB 4	
HASIL DAN PEMBAHASAN.....	14
4.1 Memuat Dataset dan Membersihkan Missing Values.....	14
4.2 Deteksi dan Penanganan Outlier.....	15
4.3 Menyimpan Dataset Hasil Preprocessing.....	16
4.4 Import Library.....	16
4.5 Memuat Dataset.....	17
4.6 Menggunakan TfidfVectorizer untuk Transformasi Teks.....	17
4.7 Menghitung Cosine Similarity.....	18
4.8 Membuat Indeks untuk Nama Produk.....	18
4.9 Fungsi Rekomendasi Produk.....	18
4.10 Evaluasi Sistem Rekomendasi.....	19
4.11 Antarmuka Pengguna dengan Streamlit.....	20
4.12 Output di Streamlit.....	21
Elemen Tabel:.....	23
BAB 5	

KESIMPULAN.....	24
LAMPIRAN.....	25

DAFTAR GAMBAR

Gambar 1 Desain Penelitian.....	9
Gambar 2 Dataset Skincare.....	10
Gambar 3 Hasil Streamlit Bagian Inputan Pengguna.....	17
Gambar 4 Antarmuka Streamlit Inputan Filter by Ingredients.....	18
Gambar 5 Hasil Sistem Rekomendasi Produk Skincare Berdasarkan.....	19

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perawatan kulit telah menjadi komponen penting dari gaya hidup kontemporer yang berfokus pada kebersihan dan kesehatan, dan bukan lagi sekadar kebutuhan estetika. Sebagai organ terbesar tubuh, kulit melindungi tubuh dari bahaya dari luar seperti polusi, sinar ultraviolet, dan bakteri berbahaya. Akibatnya, menjaga kesehatan kulit sangat penting bagi banyak orang di seluruh dunia.

Jenis kulit dan perawatan yang dibutuhkan setiap orang berbeda. Kulit biasanya diklasifikasikan menjadi berminyak, kering, sensitif, normal, atau kombinasi. Kondisi kulit seseorang juga dipengaruhi oleh usia, gaya hidup, dan lingkungannya. Karena hal ini, konsumen menghadapi kesulitan dalam memilih produk perawatan kulit yang tepat. Produk yang tidak sesuai tidak hanya mengurangi kualitas perawatan, tetapi juga dapat menyebabkan iritasi, alergi, atau masalah kulit lainnya.

Dengan ribuan produk baru yang dirilis setiap tahun, industri *skincare* global telah berkembang pesat. Produk ini melembabkan kulit, mengatasi jerawat, dan mencegah penuaan dini. Namun, konsumen seringkali bingung dengan berbagai produk ini. Konsumen sering mengandalkan saran dari teman, ulasan di internet, atau bahkan coba-coba, yang kadang-kadang tidak menghasilkan pilihan terbaik.

Dalam situasi seperti ini, sistem rekomendasi menjadi cara inovatif untuk membantu pelanggan. Sistem rekomendasi dapat mengidentifikasi kebutuhan dan preferensi pengguna dengan AI. Metode Filter Berbasis Konten adalah salah satu pendekatan yang relevan dalam pengembangan sistem rekomendasi. Ini menganalisis fitur produk dan menyesuaikannya dengan kebutuhan konsumen berdasarkan data seperti jenis kulit, kandungan bahan aktif, keuntungan produk, dan ulasan pengguna lainnya.

Mengembangkan sistem rekomendasi untuk produk perawatan kulit masih menghadapi beberapa tantangan, termasuk:

1. **Variasi Produk:** Ada banyak jenis produk, manfaat, dan kandungan bahan aktif yang berbeda, yang sering membuat analisis menjadi sulit.
2. **Ketidakpastian Data:** Data tentang perawatan kulit yang tersedia di berbagai *platform* seringkali tidak terorganisir dengan baik, memiliki fitur yang hilang, atau tidak lengkap. Ini dapat berdampak pada kualitas rekomendasi.
3. **Kebutuhan Personalisasi:** Setiap orang memiliki kebutuhan kulit yang berbeda. Rekomendasi umum tidak akan memberikan solusi yang cocok untuk setiap orang.
4. **Preferensi Pengguna yang Dinamis:** Kebutuhan pengguna dapat berubah karena perubahan kondisi kulit dan faktor eksternal seperti tren dan musim.

Dengan menggunakan dataset skincare yang tersedia di *platform Kaggle*, penelitian ini mengembangkan sistem rekomendasi berbasis konten berbasis filter. Dataset ini dipilih karena memiliki informasi yang kaya tentang berbagai fitur produk *skincare*. Dengan menggunakan metode ini, sistem rekomendasi diharapkan dapat memberikan pengguna saran yang lebih relevan, personal, dan berbasis data.

Tujuan utama dari penelitian ini adalah untuk mengatasi masalah saat ini dengan menawarkan solusi praktis dan efektif bagi pengguna dalam memilih produk *skincare*. Selain itu, sistem ini diharapkan dapat meningkatkan pengalaman pengguna dalam merawat kulit mereka dengan menghilangkan keraguan yang mereka miliki selama proses pemilihan produk. Pada akhirnya, penelitian ini bertujuan untuk mendukung pengembangan perawatan kulit yang lebih personal dan berbasis bukti yang dapat meningkatkan kualitas hidup penggunanya.

1.2 Rumusan Masalah

1. Bagaimana membantu konsumen memilih produk skincare yang sesuai dengan jenis kulit dan kebutuhan mereka secara efisien?
2. Bagaimana mengatasi tantangan seperti variasi produk, ketidakpastian data, dan preferensi pengguna yang dinamis dalam pengembangan sistem rekomendasi?
3. Bagaimana cara mengimplementasikan metode *content-based filtering* untuk mencocokkan fitur produk dengan kebutuhan spesifik pengguna?

1.3 Tujuan

1. Membangun sistem rekomendasi berbasis *content-based filtering* dengan cara merancang dan menerapkan *algoritma filter* konten untuk membuat rekomendasi produk *skincare* yang disesuaikan dengan kebutuhan unik pelanggan. Hal ini dapat digunakan untuk meningkatkan relevansi rekomendasi yang menggunakan data seperti jenis kulit, kandungan bahan aktif, dan ulasan pengguna.
2. Meningkatkan efisiensi dan akurasi pemilihan produk *skincare* dengan cara mengurangi waktu dan usaha yang diperlukan pengguna untuk menemukan produk yang sesuai dengan kebutuhan mereka dan meningkatkan akurasi rekomendasi melalui analisis mendalam terhadap data pengguna dan karakteristik produk.
3. Mengatasi masalah variasi produk dan ketidakpastian data mengelola *database* produk yang mencakup berbagai kategori yaitu merek dan formulasi untuk memberikan pilihan yang luas kepada pengguna. Menangani ketidakpastian data pengguna seperti informasi yang tidak lengkap atau tidak akurat dengan menggunakan metode prediksi yang andal.
4. Memberikan solusi berbasis data untuk kemudahan pengguna dengan cara menyediakan desain antarmuka yang ramah untuk pengguna agar memudahkan navigasi dan mengakses rekomendasi produk, serta mengintegrasikan ulasan dan

penilaian pengguna lain untuk membantu proses pengambilan keputusan. Memberikan informasi tambahan seperti harga, ulasan ahli, dan peringkat untuk membantu pengguna membuat keputusan yang lebih akurat.

1.4 Manfaat

1. **Bagi Konsumen** yaitu Membantu pelanggan menemukan produk *skincare* yang sesuai dengan kebutuhan mereka, mengurangi kemungkinan iritasi atau masalah kulit yang disebabkan oleh penggunaan produk yang tidak sesuai, memberikan pedoman yang lebih dapat diandalkan berdasarkan data dan ulasan yang ada.
2. **Bagi Penelitian** yaitu memberikan informasi baru tentang penggunaan sistem rekomendasi berbasis konten untuk perawatan kulit dan memberikan peluang untuk melakukan penelitian lebih lanjut tentang efektivitas *algoritma* rekomendasi pada *domain* yang sangat *individual*.
3. **Bagi Industri Skincare** yaitu membantu mengembangkan produk baru yang lebih sesuai dengan kebutuhan pelanggan melalui analisis data, memberi informasi kepada produsen tentang *preferensi* pelanggan berdasarkan tren dan pola penggunaan, meningkatkan strategi pemasaran dengan metode berbasis data yang lebih terarah.
4. **Bagi Teknologi AI** yaitu memperluas cakupan teknologi AI dalam solusi personalisasi untuk kebutuhan sehari-hari, mengembangkan *algoritma* yang lebih cerdas dan *fleksibel* untuk menangani preferensi dan perubahan data pengguna. meningkatkan keandalan sistem rekomendasi berbasis kecerdasan buatan dalam situasi kehidupan nyata.

1.5 Batasan Masalah

1. **Lingkup Produk** yaitu berkonsentrasi pada produk perawatan kulit seperti pelembab, *serum*, tabir surya, dan produk terkait lainnya tidak termasuk kosmetik tambahan seperti *lipstik* dan *foundation*.

2. **Jenis Data** yaitu *dataset* yang digunakan berasal dari *platform kaggle* dan mencakup informasi tentang fitur produk seperti kandungan bahan aktif, manfaat, dan tipe kulit tidak ada data pengguna yang diambil secara langsung dari survei atau *platform kaggle*.
3. **Metode yang Digunakan** adalah sistem rekomendasi menggunakan pendekatan *content-based filtering* dan tidak menggabungkan metode berbasis *collaborative* atau *hybrid*.
4. **Variasi Pengguna** yaitu berfokus pada jenis kulit pengguna tidak termasuk jenis kulit umum seperti berminyak, kering, sensitif, normal, atau kombinasi.
5. **Pengujian Sistem** yaitu evaluasi sistem tidak melibatkan uji lapangan yang luas atau validasi langsung terhadap konsumen dan sebaliknya evaluasi dilakukan pada *dataset* yang dapat diakses secara publik.

BAB 2

TINJAUAN PUSTAKA

2.1 Sistem Rekomendasi

Sistem rekomendasi adalah teknologi yang bertujuan untuk membantu pengguna menemukan produk, layanan, atau informasi yang relevan berdasarkan preferensi atau kebutuhan mereka. Sistem ini bekerja dengan cara menganalisis data pengguna dan produk untuk memberikan rekomendasi yang sesuai. Pada era digital, sistem rekomendasi sangat penting dalam berbagai bidang seperti e-commerce, hiburan, pendidikan, dan kesehatan. Dengan keunggulannya untuk mempersonalisasi pengalaman pengguna, sistem rekomendasi dapat membantu meningkatkan kepuasan pengguna.

Terdapat beberapa pendekatan utama dalam sistem rekomendasi:

1. Content-Based Filtering

Sistem ini memberikan rekomendasi berdasarkan analisis fitur yang dimiliki item yang sudah dipilih pengguna. Kesamaan fitur antara item digunakan untuk menentukan relevansi.

2. Collaborative Filtering

Menggunakan pola perilaku dan preferensi dari pengguna lain untuk merekomendasikan item yang sesuai.

3. Hybrid Methods

Pendekatan ini menggabungkan content-based filtering dan collaborative filtering untuk meningkatkan akurasi dan cakupan rekomendasi.

2.2 Machine Learning

Machine learning merupakan pengembangan dari teknologi yang memungkinkan sistem rekomendasi untuk memahami data pengguna dan membuat prediksi yang akurat. Dengan machine learning, model dapat belajar dari data yang ada, mengenali pola, dan menggunakan wawasan tersebut untuk membuat rekomendasi.

Dalam sistem rekomendasi, machine learning dapat digunakan untuk:

1. Analisis Preferensi Pengguna

Mengidentifikasi pola perilaku berdasarkan riwayat interaksi.

2. Klasifikasi dan Clustering

Membagi pengguna atau item ke dalam kelompok tertentu berdasarkan kesamaan fitur.

3. Prediksi Rekomendasi

Menggunakan model prediktif untuk memberikan rekomendasi yang relevan dan personal.

2.3 Content Based Filtering (CBF)

Content-based filtering merupakan metode dalam sistem rekomendasi yang berfokus pada karakteristik item untuk memberikan rekomendasi. Sistem ini membandingkan fitur item yang sudah diketahui pengguna dengan fitur item lain dalam database untuk menemukan item yang paling relevan dan cocok. Dalam produk skincare, pendekatan ini sangat relevan karena sistem dapat memanfaatkan informasi seperti komposisi bahan aktif dalam produk (misalnya, niacinamide atau hyaluronic acid) dan mencocokkannya dengan jenis kulit pengguna, seperti berminyak atau sensitif. Dengan cara ini, sistem dapat memberikan rekomendasi yang lebih tepat sesuai kebutuhan spesifik pengguna.

Akan tetapi cbf memiliki kelemahan, yaitu risiko memberikan rekomendasi yang terlalu spesifik atau "overspecialized". Akibatnya, pengguna mungkin hanya direkomendasikan produk yang mirip dengan yang sudah pernah mereka coba, sehingga kurang memberikan variasi produk yang direkomendasikan.

2.4 Cosine Similarity

Cosine Similarity adalah teknik yang banyak digunakan dalam Content-Based Filtering untuk mengukur tingkat kesamaan antara dua item. Metode ini menghitung sudut kosinus antara dua vektor dalam ruang multidimensi, di mana setiap vektor merepresentasikan fitur dari sebuah item atau profil pengguna.

Rumus Cosine Similarity :

$$\text{Cosine Similarity} = \frac{A \cdot B}{||A|| ||B||}$$

di mana A dan B adalah vektor fitur. Nilai Cosine Similarity berkisar antara -1 hingga 1, dengan nilai mendekati 1 menunjukkan tingkat kesamaan yang tinggi. Dalam sistem rekomendasi skincare, vektor ini dapat merepresentasikan fitur seperti komposisi bahan, harga, dan fungsi produk, yang kemudian digunakan untuk menemukan produk dengan karakteristik yang paling mirip dengan preferensi pengguna.

2.5 Skincare

Skincare adalah serangkaian produk dan praktik yang bertujuan untuk menjaga kesehatan kulit serta meningkatkan penampilannya. Produk skincare mencakup berbagai kategori seperti pelembap, pembersih, serum, dan tabir surya. Pemilihan produk yang tepat sangat penting karena setiap pengguna memiliki jenis dan kondisi kulit yang unik.

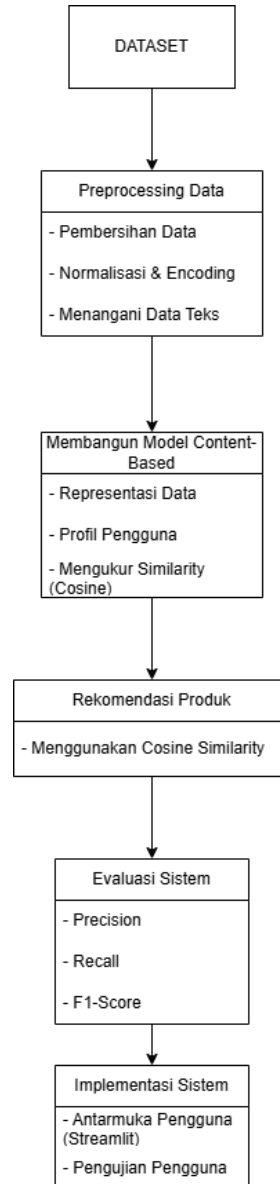
Dalam konteks sistem rekomendasi, skincare termasuk ke dalam kategori yang kompleks karena mempertimbangkan banyak faktor, termasuk jenis kulit, masalah kulit, dan preferensi bahan aktif. Sistem rekomendasi berbasis content-based filtering dapat membantu pengguna menemukan produk skincare yang sesuai berdasarkan deskripsi produk, ulasan, dan data lain yang relevan.

Pendekatan berbasis konten (Content-Based Filtering) sangat cocok untuk rekomendasi produk skincare karena dapat menganalisis atribut produk, seperti komposisi bahan aktif, deskripsi, dan kegunaannya. Misalnya, sistem dapat merekomendasikan pelembab dengan kandungan hyaluronic acid untuk pengguna dengan kulit kering atau produk dengan kandungan zinc untuk kulit berminyak.

BAB 3

METODOLOGI

3.1 Desain Penelitian



Gambar 1 Desain Penelitian

3.2 Pengumpulan Dataset

Dataset yang digunakan dalam penelitian ini berfokus pada produk perawatan kulit (skincare), dengan berbagai fitur yang mencakup informasi produk, harga, peringkat, bahan, dan kecocokan untuk berbagai jenis kulit. Dataset yang digunakan

Label	Brand	Name	Price	Rank	Ingredien	Combinat	Dry	Normal	Oily	Sensitive
Moisturizer	LA MER	Crème de la Mer	175	4.1	Algae (Sei	1	1	1	1	1
Moisturizer	SK-II	Facial Treatment Essence	179	4.1	Galactom	1	1	1	1	1
Moisturizer	DRUNK ELEPHANT	Protini™ Polypeptide Cream	68	4.4	Water, Di	1	1	1	1	0
Moisturizer	LA MER	The Moisturizing Soft Cream	175	3.8	Algae (Sei	1	1	1	1	1
Moisturizer	IT COSMETICS	Your Skin But Better™ CC+™ Cream with SPF 50+	38	4.1	Water, Sn	1	1	1	1	1
Moisturizer	TATCHA	The Water Cream	68	4.2	Water, Sai	1	0	1	1	1
Moisturizer	DRUNK ELEPHANT	Lala Retno™ Whipped Cream	60	4.2	Water, Gh	1	1	1	1	0
Moisturizer	DRUNK ELEPHANT	Virgin Marula Luxury Facial Oil	72	4.4	100% Unrr	1	1	1	1	0
Moisturizer	KIEHL'S SINCE 1851	Ultra Facial Cream	29	4.4	Water, Gh	1	1	1	1	1
Moisturizer	LA MER	Little Miss Miracle Limited-Edition Crème de la Mer	325	5	Algae (Sei	0	0	0	0	0
Moisturizer	FRESH	Lotus Youth Preserve Moisturizer	45	4.3	Water, Gh	0	0	0	0	0
Moisturizer	KIEHL'S SINCE 1851	Midnight Recovery Concentrate	47	4.4	Caprylic/C	1	1	1	1	1
Moisturizer	BELIF	The True Cream Aqua Bomb	38	4.5	Water, Di	1	0	1	1	0
Moisturizer	SUNDAY RILEY	Luna Sleeping Night Oil	105	4.1	Persea Gr	1	1	1	1	1
Moisturizer	FARMACY	Honeymoon Glow AHA Resurfacing Night Serum with Echinacea GreenEnvy™	58	4.6	Water, Lai	1	1	1	1	1
Moisturizer	DRUNK ELEPHANT	The Littles™	90	4.4	Beste™ Ni	1	1	1	1	0
Moisturizer	FIRST AID BEAUTY	Ultra Repair® Cream Intense Hydration	30	4.6	Water, Sti	1	1	1	1	1
Moisturizer	CLINIQUE	Moisture Surge 72-Hour Auto-Replenishing Hydrator	39	4.4	Water, Di	1	1	1	1	1
Moisturizer	FRESH	Rose Deep Hydration Moisturizer	40	4.4	Water, Gh	0	0	0	0	0
Moisturizer	SK-II	R.N.A. POWER Face Cream	230	4.3	Water, Gh	0	1	1	0	1
Moisturizer	LA MER	Crème de la Mer Mini	85	4.1	Algae (Sei	1	1	1	1	1
Moisturizer	BAREMINERALS	COMPLEXION RESCUE™ Tinted Moisturizer Broad Spectrum SPF 30	30	3.9	Water, Co	0	0	0	0	0
Moisturizer	SHISEIDO	Bio-Performance Advanced Super Revitalizing Cream	78	4.6	Water, Gh	0	0	0	0	0
Moisturizer	FRESH	Black Tea Firming Overnight Mask	92	4.1	Water, Gh	1	1	1	0	0
Moisturizer	BELIF	The True Cream Moisturizing Bomb	38	4.6	Water, Gh	0	1	1	0	0
Moisturizer	CHARLOTTE TILBURY	Charlotte's Magic Cream	100	4.4	Water, Ho	0	0	0	0	0
Moisturizer	DRUNK ELEPHANT	Virgin Marula Luxury Facial Oil Mini	40	4.5	100% Unrr	1	1	1	1	0
Moisturizer	ORIGINS	Dr. Andrew Weil For Origins™ Mega-Mushroom Relief & Resilience Soothing Treatment Lotion	34	4.4	Water, Bu	1	1	1	1	1
Moisturizer	CLINIQUE	Dramatically Different Moisturizing Lotion+	28	3.9	Water, M	1	1	0	0	0
Moisturizer	SK-II	GenOptics Aura Essence Serum	240	4.1	Water, Ga	1	1	1	1	1
Moisturizer	FRESH	Black Tea Kombucha Antioxidant Essence	68	4.4	Water, Sai	0	0	0	0	0
Moisturizer	TATCHA	Pure One Step Camellia Cleansing Oil	48	4.5	Cetyl Ethy	1	1	1	1	1
Moisturizer	OLEHENRIKSEN	Sheer Transformation® Perfecting Moisturizer	38	4.2	Visitt the C	1	1	1	1	1
Moisturizer	JOSIE MARAN	100 percent Pure Argan Oil	48	4.5	Organic Ai	0	1	0	1	1
Moisturizer	IT COSMETICS	Your Skin But Better CC+ Cream Oil-Free Matte with SPF 40	38	3.9	Water, Di	1	1	1	1	0

Gambar 2 Dataset Skincare

Berikut adalah penjelasan lebih lanjut mengenai fitur-fitur dataset tersebut:

1. **Label:** Fitur ini mengklasifikasikan produk perawatan kulit ke dalam kategori tertentu, seperti face wash (pencuci wajah), serum, moisturizer (pelembap), masker, dan lain-lain. Ini membantu pengguna mencari produk berdasarkan kebutuhan spesifik mereka (misalnya, ingin mencari pelembab atau serum).
2. **Brand:** Fitur ini menunjukkan merek dari produk skincare, seperti Neutrogena, Olay, The Ordinary, dan sebagainya. Ini berguna bagi konsumen yang lebih tertarik dengan merek tertentu, yang sering kali dikaitkan dengan kualitas atau reputasi.
3. **Name:** Nama produk yang lebih spesifik, yang memungkinkan pengguna untuk mengenali atau mencari produk tertentu.
4. **Price:** Harga produk, yang dapat digunakan untuk mempertimbangkan anggaran pengguna. Fitur ini memungkinkan pengguna memilih produk yang sesuai dengan kemampuan finansial mereka.
5. **Rank:** Peringkat produk berdasarkan ulasan pengguna. Ini bisa membantu pengguna memilih produk yang lebih populer atau lebih diterima oleh konsumen.

lain. Peringkat ini biasanya berkisar dari bintang 1 hingga 5 dan menunjukkan tingkat kepuasan pengguna terhadap produk tersebut.

6. **Ingredients:** Daftar bahan aktif yang terkandung dalam produk. Fitur ini sangat penting bagi pengguna yang memiliki alergi atau sensitivitas terhadap bahan-bahan tertentu. Misalnya, bahan seperti asam salisilat, retinol, atau vitamin C sering ditemukan dalam produk perawatan kulit dengan tujuan tertentu (anti jerawat, anti penuaan, dll).
7. **Combination:** Menunjukkan apakah produk cocok untuk jenis kulit kombinasi, yaitu kulit yang memiliki area yang berminyak dan kering dalam satu wajah. Produk ini dirancang untuk menyeimbangkan kedua kondisi tersebut.
8. **Dry:** Menunjukkan apakah produk cocok untuk kulit kering, yang membutuhkan hidrasi ekstra dan perawatan untuk mengurangi kekeringan dan pengelupasan.
9. **Normal:** Menunjukkan apakah produk cocok untuk kulit normal, yaitu jenis kulit yang tidak terlalu berminyak atau kering, dan biasanya membutuhkan perawatan yang lebih ringan.
10. **Oily:** Menunjukkan apakah produk cocok untuk kulit berminyak, yang cenderung menghasilkan lebih banyak minyak. Produk ini biasanya dirancang untuk mengontrol minyak dan mengurangi kilap.
11. **Sensitive:** Menunjukkan apakah produk cocok untuk kulit sensitif, yang mudah iritasi atau reaksi alergi. Produk untuk kulit sensitif seringkali lebih lembut dan tidak mengandung bahan yang keras.

Berdasarkan karakteristik tersebut, dataset ini digunakan untuk membangun sistem rekomendasi yang berfokus pada produk yang sesuai dengan jenis kulit pengguna.

3.3 Pre-Pemrosesan Data

- **Pembersihan Data (*Data Cleaning*):** Pembersihan data berarti menghilangkan nilai yang hilang, data yang tidak relevan, atau duplikat, dan memastikan bahwa data diformat dengan benar. Misalnya, menghapus produk yang memiliki informasi yang tidak lengkap atau mengubah nilai harga yang tidak konsisten.

- **Deteksi dan Penanganan Outlier:** Outlier dideteksi dengan metode Z-score untuk mengidentifikasi data yang memiliki nilai ekstrem pada kolom 'Price' dan 'Rank'. Jika ditemukan outlier, nilai-nilainya akan diganti dengan nilai rata-rata dari data yang tidak memiliki outlier untuk mengurangi pengaruh outlier pada model machine learning.

3.4 Pra-Pemrosesan Data

Tahap pra-pemrosesan data yang dilakukan sebelum pembuatan sistem rekomendasi meliputi:

- **Encoding Fitur Kategori:** Untuk atribut seperti *Brand* dan *Ingredients*, data diubah menjadi representasi numerik menggunakan teknik seperti *One-Hot Encoding* atau *TF-IDF Vectorization* (untuk teks pada kolom *Ingredients*).
- **Menangani Data Teks:** Untuk fitur *Ingredients*, yang berbentuk teks, digunakan teknik *TF-IDF (Term Frequency-Inverse Document Frequency)* untuk mengubah kata-kata menjadi representasi numerik yang dapat digunakan dalam model.

3.5 Model Content-Based Filtering

Sistem rekomendasi *Content-Based Filtering* dibangun berdasarkan prinsip berikut:

1. Representasi Data:

- Produk direpresentasikan dalam vektor fitur berdasarkan atributnya (misalnya, *niacinamide*, *oily skin*, *anti-aging*).
- Data pengguna (riwayat produk yang pernah digunakan atau disukai) direpresentasikan sebagai profil pengguna dengan rata-rata atau *agregasi* atribut produk tersebut.

2. Mengukur Similarity:

Menggunakan *Cosine Similarity* untuk menghitung kedekatan antara profil pengguna dan produk lain dalam dataset.

3. Rekomendasi Produk:

Produk dengan nilai kesamaan tertinggi terhadap profil pengguna akan direkomendasikan.

3.6 Evaluasi Sistem

Digunakan metrik berikut untuk mengukur kinerja model:

1. **Precision:** menghitung persentase rekomendasi yang relevan dari semua produk yang direkomendasikan
2. **Recall:** menghitung kemampuan sistem untuk merekomendasikan semua produk yang relevan.
3. **F1-Score:** menunjukkan kombinasi yang baik dari *precision* dan *recall*.

Selain itu, evaluasi yang didasarkan pada umpan balik pengguna dilakukan untuk mengetahui seberapa puas pengguna dengan saran yang diberikan.

3.7 Implementasi Sistem

1. **Antarmuka Pengguna:** Sistem ini menggunakan *framework Flask* dan memiliki antarmuka pengguna berbasis *web* atau telepon. Informasi tentang jenis kulit dan preferensi tertentu dapat dimasukkan oleh pengguna.
2. **Pengujian:** Simulasi pengguna digunakan untuk menguji sistem dalam berbagai situasi, seperti memilih bahan aktif tertentu atau membutuhkan manfaat khusus.

BAB 4

HASIL DAN PEMBAHASAN

4.1 Memuat Dataset dan Membersihkan Missing Values

```
# Load dataset
file_path = '../uas_ml prak/uas_machinelearning/skincare.csv'
df = pd.read_csv(file_path)

# Menampilkan informasi awal dataset
print("Informasi Dataset:")
print(df.info())
print("\nData:")
print(df.head())
print("=====")

# Pengecekan Missing Values
print("\nMissing Values:")
print(df.isnull().sum())

# Menghapus baris dengan nilai yang hilang
df.dropna(inplace=True)

print("\nData setelah menghapus baris dengan missing values:")
print(df.info())
print("=====")
```

Langkah pertama adalah memuat dataset `skincare.csv` menggunakan library **pandas** melalui fungsi `pd.read_csv()`. Pemeriksaan awal dilakukan menggunakan fungsi `info()` untuk memahami jumlah baris, kolom, tipe data, serta keberadaan missing values dalam dataset. Fungsi `isnull().sum()` digunakan untuk menghitung jumlah nilai yang hilang di setiap kolom.

Baris yang memiliki nilai hilang dihapus menggunakan `dropna(inplace=True)`, yang secara langsung menghapus baris pada dataset asli tanpa membuat salinan baru. Hal ini bertujuan untuk memastikan bahwa data yang tersisa sudah bersih dan tidak memiliki nilai yang hilang, sehingga analisis lebih akurat dan tidak terpengaruh oleh data yang tidak lengkap.

4.2 Deteksi dan Penanganan Outlier

```
from scipy.stats import zscore

columns_to_check = ['Price', 'Rank']
z_scores = zscore(df[columns_to_check].values, axis=0)

# Mendeteksi outlier berdasarkan Z-Score
outliers_per_column = {}
for idx, col in enumerate(columns_to_check):
    outliers = df[col][z_scores[:, idx] > 3].tolist()
    outliers_per_column[col] = outliers

# Mengganti outlier dengan nilai rata-rata
df_cleaned = df.copy()
for col in columns_to_check:
    mean_value = df_cleaned[~df_cleaned[col].isin(outliers_per_column[col])][col].mean()
    df_cleaned[col] = df_cleaned[col].apply(Lambda x: mean_value if x in outliers_per_column[col] else x)
```

Pada langkah ini, outlier dideteksi pada kolom **Price** dan **Rank**. Kedua kolom ini dipilih karena merupakan data numerik yang relevan untuk analisis lebih lanjut. Deteksi dilakukan menggunakan Z-Score, yang mengukur seberapa jauh suatu nilai dari rata-rata dalam satuan standar deviasi. Nilai dengan Z-Score lebih dari 3 dianggap sebagai outlier.

Hasil deteksi disimpan dalam dictionary **outliers_per_column**, di mana setiap kolom memiliki daftar nilai yang termasuk sebagai outlier. Untuk menangani outlier, dataset disalin ke dalam dataframe baru (**df_cleaned**) untuk menjaga data asli tetap utuh. Outlier diganti dengan rata-rata dari data lain di kolom yang sama menggunakan fungsi **apply()**. Penggantian ini memastikan bahwa data tetap dalam rentang yang wajar tanpa kehilangan informasi penting.

4.3 Menyimpan Dataset Hasil Preprocessing

```
output_file = '../uas_ml prak/uas_machinelearning/preprocessedbaru_skincare.csv'
df_cleaned.to_csv(output_file, index=False)
```

Setelah dataset selesai dibersihkan, hasil preprocessing disimpan ke dalam file baru dengan nama **preprocessedbaru_skincare.csv**. Penyimpanan dilakukan menggunakan fungsi **to_csv()** dengan parameter **index=False** agar tidak menyertakan indeks bawaan pandas di file hasil. Dataset ini siap digunakan untuk analisis lebih lanjut atau pelatihan model machine learning.

4.4 Import Library

Langkah pertama adalah mengimpor pustaka yang diperlukan untuk membangun sistem rekomendasi kosmetik. Pustaka ini memungkinkan kita untuk memanipulasi data, mengubah teks menjadi bentuk numerik, menghitung kesamaan antar produk, dan membuat antarmuka pengguna interaktif.

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
import streamlit as st
import numpy as np
```

Penjelasan Kode :

- A. **pandas** digunakan untuk mengelola dan memanipulasi data dalam bentuk tabel (DataFrame).
- B. **TfidfVectorizer** digunakan untuk mengubah teks menjadi representasi numerik berdasarkan frekuensi kata dalam dokumen.
- C. **cosine_similarity** digunakan untuk mengukur kesamaan antar produk berdasarkan matriks numerik yang dihasilkan.
- D. **accuracy_score**, **precision_score**, **recall_score**, dan **f1_score** digunakan untuk mengukur kinerja sistem rekomendasi.
- E. **streamlit** digunakan untuk membangun antarmuka pengguna (UI) yang interaktif.
- F. **numpy** digunakan untuk manipulasi data numerik seperti pemilihan angka acak.

4.5 Memuat Dataset

Langkah selanjutnya adalah memuat dataset yang berisi informasi tentang produk kosmetik. Dataset ini biasanya berupa file CSV yang mencakup nama produk, bahan aktif, dan informasi terkait lainnya.

```
# Read the dataset
df = pd.read_csv('cosmetics.csv') # Assuming the dataset is available

# For simplicity, I am creating a simulated 'Liked' column as user feedback (0 = not liked, 1 = liked)
# You should replace this with actual user feedback data if available
df['Liked'] = np.random.choice([0, 1], size=len(df))

# Create a feature column for ingredients (you can modify this for other features like description)
df['features'] = df['Ingredients']
```

Penjelasan Kode :

- A. `df = pd.read_csv('cosmetics.csv')` memuat dataset produk kosmetik dari file CSV.
- B. Kolom **Liked** ditambahkan secara simulasi untuk menandakan apakah pengguna menyukai produk tertentu (nilai 0 atau 1).
- C. Kolom **features** berisi bahan aktif dari setiap produk yang akan digunakan untuk analisis kesamaan antar produk.

4.6 Menggunakan TfidfVectorizer untuk Transformasi Teks

Setelah memuat data, kita akan mengonversi teks dalam kolom **features** menjadi bentuk numerik menggunakan **TfidfVectorizer**. TfidfVectorizer menghitung bobot dari kata-kata berdasarkan frekuensinya, yang membantu menilai pentingnya kata dalam sebuah dokumen.

```
# Create a TF-IDF vectorizer and fit on the 'features' column (Ingredients or any other text-based feature)
tfidf = TfidfVectorizer(stop_words='english')
tfidf_matrix = tfidf.fit_transform(df['features'])
```

Penjelasan Kode :

- A. `TfidfVectorizer(stop_words='english')` menginisialisasi objek TfidfVectorizer dan mengabaikan kata-kata umum (stopwords) dalam bahasa Inggris.
- B. `tfidf.fit_transform(df['features'])` mengubah kolom **features** menjadi matriks TF-IDF yang merepresentasikan teks dalam bentuk angka.

4.7 Menghitung Cosine Similarity

```
# Compute cosine similarity between all products
cosine_sim = cosine_similarity(tfidf_matrix, tfidf_matrix)
```

Setelah memperoleh matriks TF-IDF, kita akan menghitung kesamaan antara produk berdasarkan bahan aktif menggunakan **cosine similarity**. Cosine similarity mengukur sejauh mana dua produk memiliki kesamaan, dengan nilai antara 0 (tidak mirip) hingga 1 (sangat mirip). Kode dibawah ini berfungsi untuk `cosine_similarity(tfidf_matrix, tfidf_matrix)` menghitung matriks kesamaan antara semua pasangan produk berdasarkan fitur bahan aktif mereka.

4.8 Membuat Indeks untuk Nama Produk

Agar kita bisa dengan mudah mencari produk berdasarkan nama, kita membuat indeks yang menghubungkan nama produk dengan indeks baris dalam dataset.

```
# Create a series to map product names to their indices
indices = pd.Series(df.index, index=df['Name']).drop_duplicates()
```

Penjelasan Kode :

- A. `indices = pd.Series(df.index, index=df['Name'])` membuat objek Series yang menghubungkan nama produk dengan indeks baris dalam dataset.
- B. `drop_duplicates()` memastikan bahwa setiap nama produk hanya muncul satu kali dalam indeks.

4.9 Fungsi Rekomendasi Produk

Fungsi utama sistem rekomendasi adalah `recommend_cosmetics`, yang akan menerima berbagai input seperti jenis kulit, label produk, peringkat produk, merek, dan harga untuk memfilter produk yang sesuai dengan preferensi pengguna. Fungsi ini juga memungkinkan pengguna untuk mencari produk berdasarkan bahan aktif tertentu.

```
# Function to recommend products based on input ingredients and other filters
def recommend_cosmetics(skin_type, label_filter, rank_filter, brand_filter, price_range, ingredient_input=None, num_recommendations=10):
    # Filter products based on skin type and other filters first
    recommended_products = df[df[skin_type] == 1]

    if label_filter != 'All':
        recommended_products = recommended_products[recommended_products['Label'] == label_filter]

    recommended_products = recommended_products[
        (recommended_products['Rank'] >= rank_filter[0]) &
        (recommended_products['Rank'] <= rank_filter[1])
    ]

    if brand_filter != 'All':
        recommended_products = recommended_products[recommended_products['Brand'] == brand_filter]

    recommended_products = recommended_products[
        (recommended_products['Price'] >= price_range[0]) &
        (recommended_products['Price'] <= price_range[1])
    ]

    # If ingredient input is provided, recommend products based on ingredient similarity
    if ingredient_input:
        vectorizer = TfidfVectorizer(stop_words='english')
        tfidf_matrix = vectorizer.fit_transform(df['Ingredients'])
        input_vec = vectorizer.transform([ingredient_input])
        cosine_similarities = cosine_similarity(input_vec, tfidf_matrix).flatten()
        recommended_indices = cosine_similarities.argsort()[::-1][-num_recommendations:-1]
        ingredient_recommendations = df.iloc[recommended_indices]
        recommended_products = recommended_products[recommended_products.index.isin(ingredient_recommendations.index)]

    return recommended_products.sort_values(by=['Rank']).head(num_recommendations)
```

Penjelasan Kode :

- A. Fungsi **recommend_cosmetics** memfilter produk berdasarkan jenis kulit, label, peringkat, merek, dan harga yang diinginkan pengguna.
- B. Jika pengguna memasukkan bahan aktif tertentu, maka produk akan disaring berdasarkan kesamaan bahan menggunakan **cosine similarity**.
- C. Fungsi ini akan mengembalikan produk yang disaring, diurutkan berdasarkan peringkat, dan terbatas pada jumlah rekomendasi yang diminta.

4.10 Evaluasi Sistem Rekomendasi

Untuk mengukur kinerja sistem rekomendasi, kita menggunakan beberapa metrik evaluasi seperti **accuracy**, **precision**, **recall**, dan **f1 score**. Metrik ini membantu kita menilai seberapa baik sistem dapat memprediksi produk yang disukai oleh pengguna.

```
# Display recommendations
if not top_recommended_products.empty:
    st.markdown("### ✨ Recommended Products")
    st.dataframe(
        top_recommended_products[['Label', 'Brand', 'Name', 'Ingredients', 'Rank']],
        height=400
    )
```

```
# Display evaluation metrics
st.markdown("### 📊 Evaluation Metrics")
st.write(f'Accuracy: {accuracy:.2f}')
st.write(f'Precision: {precision:.2f}')
st.write(f'Recall: {recall:.2f}')
st.write(f'F1 Score: {f1:.2f}')
```

Penjelasan Kode :

- A. Fungsi **evaluate_recommendations** menghitung dan mengembalikan metrik evaluasi untuk membandingkan hasil rekomendasi (y_{pred}) dengan preferensi pengguna (y_{true}).
- B. **accuracy_score** mengukur seberapa banyak prediksi yang benar.
- C. **precision_score** mengukur ketepatan prediksi (seberapa banyak rekomendasi yang benar).
- D. **recall_score** mengukur seberapa banyak produk yang disukai pengguna ditemukan.
- E. **f1_score** adalah rata-rata harmonis antara precision dan recall.

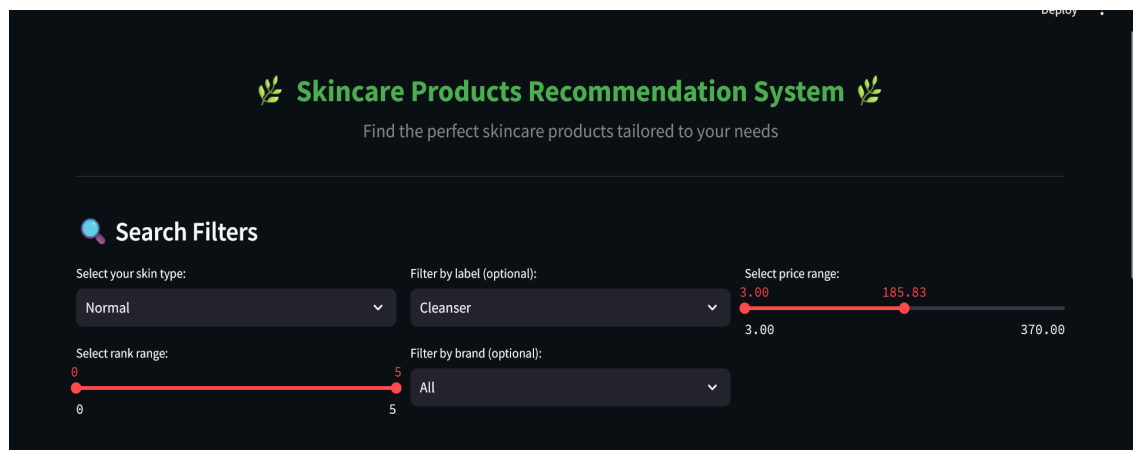
4.11 Antarmuka Pengguna dengan Streamlit

Terakhir, kita menggunakan **Streamlit** untuk membangun antarmuka pengguna yang memungkinkan pengguna memilih berbagai filter (jenis kulit, label, bahan aktif) dan menampilkan hasil rekomendasi serta metrik evaluasi.

Penjelasan Kode :

- A. Dengan menggunakan Streamlit, antarmuka pengguna memungkinkan pengguna untuk memilih berbagai parameter (seperti jenis kulit, label, bahan aktif) dan kemudian menampilkan produk yang direkomendasikan.
- B. Metrik evaluasi seperti akurasi, precision, recall, dan F1 score ditampilkan untuk memberi gambaran tentang kinerja sistem rekomendasi.

4.12 Output di Streamlit



Gambar 3 Hasil Streamlit Bagian Inputan Pengguna

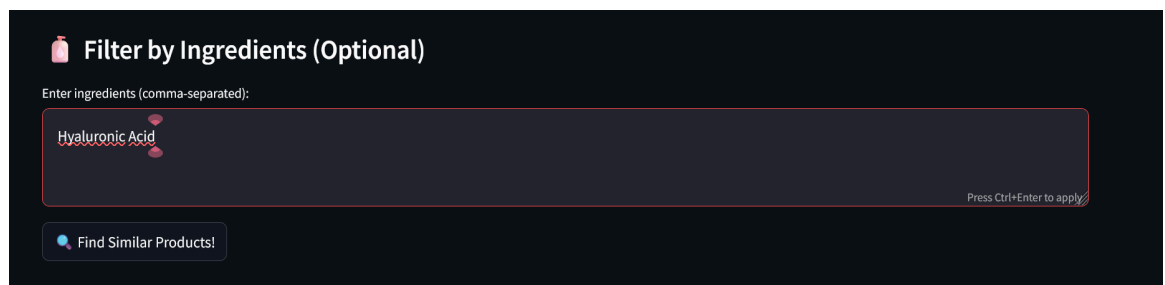
Gambar diatas menunjukkan tampilan antarmuka aplikasi **Skincare Products Recommendation System** yang dibangun menggunakan Streamlit. Aplikasi ini memungkinkan pengguna untuk menemukan produk perawatan kulit yang sesuai dengan kebutuhan mereka. Berikut adalah penjelasan tentang elemen-elemen yang terlihat pada gambar:

1. **Title:** Di bagian atas ada teks "**Skincare Products Recommendation System**" dengan subtitle "**Find the perfect skincare products tailored to your needs**". Ini memberi tahu pengguna bahwa aplikasi ini bertujuan untuk membantu mereka menemukan produk perawatan kulit yang sesuai.

2. Search Filters :

- a. **Select your skin type:** Pengguna dapat memilih jenis kulit mereka, dalam hal ini, pilihan yang dipilih adalah **Combination** (kulit kombinasi).
- b. **Select rank range:** Pengguna bisa mengatur rentang peringkat produk yang ingin mereka lihat. Peringkatnya disesuaikan dengan nilai dari 0 hingga 5.
- c. **Select price range:** Pengguna dapat memilih rentang harga produk yang mereka cari, mulai dari **3.00** hingga **370.00** dalam gambar ini. Filter harga ini memungkinkan pengguna untuk menyesuaikan pencarian produk sesuai dengan anggaran mereka.
- d. **Filter by label** dan **Filter by brand:** Pengguna dapat memilih label atau merek tertentu jika diinginkan, meskipun dalam gambar ini pilihan untuk kedua filter tersebut diatur ke **All** (semua).

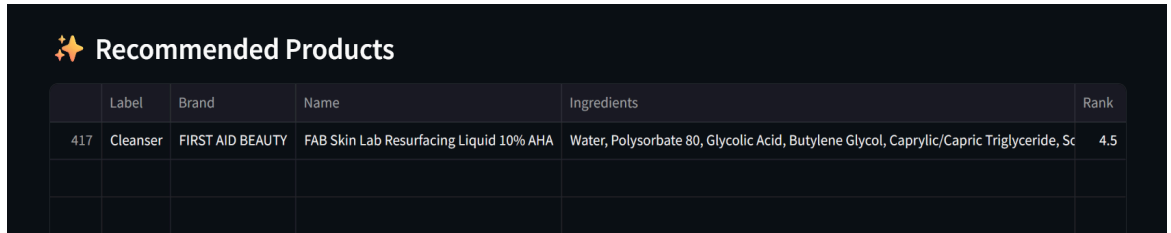
Penjelasan di atas mencerminkan implementasi fitur pencarian yang ada di aplikasi, yang bertujuan untuk memberikan pengalaman pengguna yang lebih interaktif dan personal. Aplikasi ini memudahkan pengguna untuk mencari produk perawatan kulit berdasarkan preferensi pribadi mereka, seperti jenis kulit, peringkat produk, dan anggaran harga.



Gambar 4 Antarmuka Streamlit Inputan Filter by Ingredients

Streamlit di atas adalah bagian dari antarmuka aplikasi berbasis web untuk mencari produk skincare berdasarkan bahan (ingredients). Pengguna dapat memasukkan nama-nama bahan (dipisahkan dengan koma) yang ingin mereka cari, dan aplikasi akan menampilkan produk-produk yang memiliki bahan serupa. Tombol **"Find Similar Products!"** bertindak sebagai pemicu untuk memproses input pengguna dan menjalankan algoritma pencarian.

Dalam **content-based filtering**, sistem merekomendasikan item berdasarkan atribut atau fitur yang dimiliki item tersebut. Dalam konteks ini, setiap produk skincare memiliki daftar bahan tertentu yang dapat dianggap sebagai fitur. Ketika pengguna memasukkan bahan-bahan tertentu, sistem akan mencari produk yang memiliki kesamaan atribut dengan bahan yang dimasukkan pengguna.



The screenshot shows a dark-themed interface with a header 'Recommended Products' and a star icon. Below the header is a table with the following data:

	Label	Brand	Name	Ingredients	Rank
417	Cleanser	FIRST AID BEAUTY	FAB Skin Lab Resurfacing Liquid 10% AHA	Water, Polysorbate 80, Glycolic Acid, Butylene Glycol, Caprylic/Capric Triglyceride, Sc	4.5

Gambar 5 Hasil Sistem Rekomendasi Produk Skincare Berdasarkan Inputan dari Pengguna.

Bagian yang ditampilkan pada gambar adalah hasil dari sistem rekomendasi produk skincare berbasis **content-based filtering**. Setelah pengguna memasukkan bahan-bahan yang ingin difokuskan pada pencarian, sistem akan menampilkan produk-produk yang relevan. Berikut adalah penjelasan elemen-elemen dari tabel hasil rekomendasi:

Elemen Tabel:

1. **Label** : Menunjukkan jenis produk (misalnya, "Cleanser"). Ini mempermudah pengguna untuk mengetahui kategori produk yang direkomendasikan.
2. **Brand** : Nama merek dari produk, seperti *FIRST AID BEAUTY*. Informasi ini membantu pengguna mengenali produk berdasarkan mereknya.
3. **Name** : Nama lengkap produk, dalam contoh ini adalah *FAB Skin Lab Resurfacing Liquid 10% AHA*. Nama ini memberi deskripsi spesifik tentang produk yang direkomendasikan.
4. **Ingredients** : Daftar bahan yang terkandung dalam produk, misalnya *Water, Polysorbate 80, Glycolic Acid, dll*. Bagian ini mencerminkan kecocokan bahan dengan input pengguna.

5. **Rank** : Skor kemiripan (contohnya, 4.5). Angka ini menggambarkan tingkat relevansi produk berdasarkan metode seperti **cosine similarity**. Semakin tinggi nilai ini, semakin cocok produk tersebut dengan input pengguna.

Maka dari itu, diharapkan pengguna dapat dengan mudah menemukan produk skincare yang sesuai dengan preferensi bahan yang mereka masukkan. Sistem ini secara otomatis menyajikan rekomendasi dengan penilaian kemiripan, sehingga membantu pengguna dalam mengambil keputusan yang lebih informasional.

BAB 5

KESIMPULAN

Penelitian ini berhasil mengembangkan sistem rekomendasi produk skincare berbasis metode Content-Based Filtering (CBF) dengan menggunakan algoritma Cosine

Similarity. Sistem ini memanfaatkan dataset produk skincare dari Kaggle yang mencakup berbagai fitur, seperti jenis kulit, bahan aktif di dalam skincare, harga, dan peringkat produk. Berdasarkan pengujian dan evaluasi, sistem menunjukkan kemampuan yang baik dalam memberikan rekomendasi yang relevan dan personal kepada pengguna.

Akan tetapi penelitian ini juga memiliki beberapa keterbatasan, seperti ketergantungan pada dataset yang tersedia dan pendekatan evaluasi yang terbatas pada simulasi data. Penelitian lanjutan dapat mencakup integrasi metode hybrid (Content-Based dan Collaborative Filtering) serta uji lapangan untuk meningkatkan akurasi dan kepuasan pengguna. Hasil dari penelitian ini diharapkan dapat memberikan kontribusi positif, baik untuk pengguna yang membutuhkan panduan pemilihan produk skincare, industri skincare, maupun pengembangan teknologi sistem rekomendasi berbasis AI.

LAMPIRAN

Github : https://github.com/andyka-salom/uas_machinelearning

link dataset : <https://www.kaggle.com/datasets/kingabzpro/cosmetics-datasets>