

PENERAPAN SISTEM REKOMENDASI PRODUK SEDERHANA MENGUNAKAN ALGORITMA CONTENT-BASED FILTERING UNTUK UMKM

Linda Wahyu Widiyanti, Hening Hendrato dan Mohamad Saefudin*

Sekolah Tinggi Manajemen Informatika dan Komputer Jakarta STI&K
Jalan BRI No.17, Radio Dalam Kebayoran Baru, Jakarta Selatan 12140, Indonesia
lindawewe100@gmail.com, heninghendrato69@gmail.com, saefudin@gmail.com

*Corresponding Author

ABSTRAK

Perkembangan teknologi informasi memberikan peluang besar bagi Usaha Mikro, Kecil, dan Menengah (UMKM) untuk memperluas jangkauan pasar secara digital. Salah satu tantangan utama UMKM adalah menyajikan produk yang relevan kepada calon pembeli sesuai dengan preferensi mereka. Sistem rekomendasi merupakan solusi efektif untuk memberikan personalisasi dalam penawaran produk. Penelitian ini bertujuan mengembangkan sistem rekomendasi produk sederhana berbasis content-based filtering yang dapat diimplementasikan pada skala UMKM. Dataset yang digunakan bersifat buatan sendiri, terdiri dari lima jenis produk dengan deskripsi singkat masing-masing. Proses rekomendasi dilakukan melalui beberapa tahapan: preprocessing teks (tokenisasi dan stopword removal), ekstraksi fitur menggunakan Term Frequency-Inverse Document Frequency (TF-IDF), dan perhitungan kemiripan antarproduk menggunakan cosine similarity. Hasil pengujian menunjukkan bahwa sistem mampu memberikan rekomendasi yang relevan, dengan mayoritas hasil menunjukkan tingkat presisi yang tinggi berdasarkan evaluasi manual. Sistem ini dapat membantu pelaku UMKM mempromosikan produk secara lebih personal kepada calon pembeli tanpa memerlukan data interaksi pengguna secara luas. Ke depannya, pengembangan sistem dapat dilakukan dengan memperluas jumlah produk, memperkaya deskripsi, serta mengombinasikan pendekatan content-based filtering dengan collaborative filtering untuk meningkatkan variasi rekomendasi.

Kata Kunci: sistem rekomendasi, content-based filtering, UMKM, TF-IDF, cosine similarity

PENDAHULUAN

Usaha Mikro, Kecil, dan Menengah (UMKM) memiliki peran strategis dalam perekonomian Indonesia. Berdasarkan data Kementerian Koperasi dan UKM, pada tahun 2022 jumlah UMKM di Indonesia mencapai 65,47 juta unit usaha, yang menyumbang sekitar 60,5% terhadap Produk Domestik Bruto (PDB) nasional serta menyerap lebih dari 97% tenaga kerja nasional [2]. Namun, meskipun memiliki kontribusi besar, banyak pelaku UMKM menghadapi kendala dalam mengoptimalkan penjualan produk, terutama terkait penentuan target pasar yang tepat. Perubahan perilaku konsumen di era digital menuntut strategi pemasaran yang lebih personal dan relevan agar dapat bersaing dengan produk dari pelaku usaha skala besar.

Perkembangan teknologi digital telah memberikan peluang baru bagi

UMKM untuk memasarkan produk secara daring. Pemanfaatan e-commerce, media sosial, dan marketplace memungkinkan pelaku UMKM menjangkau konsumen yang lebih luas tanpa batasan geografis [2]. Namun, menjangkau audiens yang luas tidak selalu menjamin peningkatan penjualan jika produk yang ditawarkan tidak relevan dengan kebutuhan dan preferensi pengguna. Oleh karena itu, diperlukan sistem yang dapat membantu menyajikan produk yang sesuai dengan minat calon pembeli secara otomatis.

Sistem rekomendasi merupakan salah satu teknologi yang terbukti efektif dalam meningkatkan relevansi penawaran produk dan kepuasan pengguna [3]. Beberapa studi menunjukkan bahwa sistem rekomendasi mampu meningkatkan click-through rate dan konversi penjualan hingga 30% pada platform e-commerce [4]. Terdapat tiga pendekatan utama dalam

sistem rekomendasi, yaitu collaborative filtering, content-based filtering, dan hybrid methods. Di antara ketiganya, content-based filtering memiliki keunggulan dalam mengandalkan deskripsi dan karakteristik produk tanpa memerlukan data perilaku pengguna lain, sehingga lebih sesuai untuk UMKM yang memiliki data penjualan terbatas [5].

Rumusan masalah dalam penelitian ini adalah: (1) Bagaimana membangun sistem rekomendasi sederhana untuk UMKM menggunakan pendekatan content-based filtering? dan (2) Bagaimana mengukur relevansi rekomendasi yang dihasilkan menggunakan dataset sederhana buatan sendiri? Untuk menjawab pertanyaan ini, penelitian akan mengimplementasikan algoritma content-based filtering dengan teknik ekstraksi fitur teks menggunakan Term Frequency–Inverse Document Frequency (TF-IDF) dan perhitungan kemiripan menggunakan cosine similarity [6].

Tujuan dari penelitian ini adalah (1) mengimplementasikan sistem rekomendasi berbasis content-based filtering yang dapat digunakan oleh pelaku UMKM, dan (2) menguji kinerja sistem pada dataset sederhana buatan sendiri. Hasil penelitian diharapkan dapat memberikan gambaran awal tentang efektivitas pendekatan ini bagi UMKM, serta menjadi dasar pengembangan sistem rekomendasi yang lebih kompleks di masa mendatang.

METODE PENELITIAN

Penelitian ini termasuk dalam kategori penelitian terapan (applied research) dengan pendekatan eksperimen komputasional, karena fokus utamanya adalah merancang dan menguji sistem rekomendasi berbasis content-based filtering pada dataset sederhana yang dibuat khusus untuk penelitian ini. Metode yang digunakan menggabungkan konsep pengolahan teks (text processing) dengan teknik information retrieval untuk menghasilkan rekomendasi produk yang relevan bagi konsumen UMKM.

Konsep dan Perancangan Sistem

Sistem rekomendasi dirancang untuk memberikan saran produk UMKM kepada pengguna berdasarkan kemiripan deskripsi produk. Dataset yang digunakan terdiri dari 5–10 entri produk buatan sendiri yang mencakup informasi nama produk, kategori, dan deskripsi singkat. Deskripsi ini akan menjadi sumber utama fitur yang digunakan untuk menghitung tingkat kemiripan antarproduk.

Rancangan sistem meliputi lima tahap utama:

1. Pengumpulan Data Produk dan Deskripsi

Data dikumpulkan secara manual dengan membuat deskripsi produk yang merepresentasikan variasi kategori UMKM, seperti makanan, kerajinan tangan, dan pakaian. Pemilihan data buatan bertujuan untuk meminimalkan risiko bias dan melindungi privasi pelaku UMKM.

2. Preprocessing Teks

Data deskripsi produk diproses melalui tahap tokenisasi untuk memisahkan kata-kata penyusunnya. Kemudian dilakukan stopwords removal untuk menghapus kata-kata umum yang tidak berkontribusi signifikan terhadap makna (misalnya “dan”, “yang”, “adalah”). Langkah ini bertujuan meningkatkan kualitas fitur yang akan diekstraksi.

3. Ekstraksi Fitur TF-IDF

Fitur teks diubah menjadi representasi numerik menggunakan metode Term Frequency–Inverse Document Frequency (TF-IDF). Teknik ini memberikan bobot lebih tinggi pada kata-kata yang sering muncul dalam deskripsi tertentu, namun jarang muncul di deskripsi produk lainnya.

4. Perhitungan Kemiripan Cosine Similarity

Tingkat kemiripan antarproduk dihitung dengan cosine similarity, yaitu metrik yang mengukur

kesamaan arah vektor fitur di ruang multidimensi. Nilai kemiripan berkisar antara 0 hingga 1, di mana nilai yang lebih mendekati 1 menunjukkan kemiripan yang lebih tinggi.

5. Penyusunan Rekomendasi

Produk-produk diurutkan berdasarkan skor kemiripan tertinggi terhadap produk yang sedang dilihat pengguna. Hasil urutan ini kemudian digunakan untuk memberikan saran produk yang relevan.

Data Penelitian

Data yang digunakan dalam penelitian ini adalah dataset buatan sendiri yang berisi 5–10 entri produk UMKM dari berbagai kategori, seperti makanan, minuman, pakaian, kerajinan, dan produk rumah tangga. Setiap entri produk memuat:

1. ID Produk – Nomor unik untuk identifikasi produk.
2. Nama Produk – Nama singkat yang menggambarkan produk.
3. Kategori – Jenis atau kelompok produk (misalnya “Makanan” atau “Fashion”).
4. Deskripsi Produk – Uraian singkat yang berisi bahan, keunggulan, kegunaan, atau karakteristik unik produk.

Data ini dibuat secara manual agar dapat dikendalikan kualitasnya dan memastikan keberagaman kosakata dalam deskripsi, sehingga pengujian sistem rekomendasi dapat berlangsung optimal tanpa kendala privasi atau keterbatasan lisensi data.

Tahapan Sistem

1. Pengumpulan Data

Tahap ini melibatkan pembuatan dataset produk secara manual. Setiap produk dilengkapi deskripsi yang cukup detail agar proses text mining dapat bekerja dengan baik. Misalnya:

- Produk: “Kopi Arabika Gayo”
- Deskripsi: “Kopi Arabika dari dataran tinggi Gayo, bercita rasa asam segar dengan aroma floral yang khas.”

Proses ini bertujuan menyediakan sumber data awal yang akan diolah lebih lanjut pada tahap berikutnya.

2. Preprocessing Teks

Preprocessing adalah tahap pembersihan dan normalisasi data teks agar siap untuk analisis. Tahapan ini meliputi:

- Tokenisasi: Memecah deskripsi produk menjadi kata-kata tunggal (tokens). Contoh “Kopi Arabika Gayo” → [“kopi”, “arabika”, “gayo”].
- Stopword Removal: Menghapus kata-kata umum yang tidak memiliki makna spesifik untuk pencarian, seperti “dan”, “yang”, “adalah”.
- (Opsional) Lowercasing: Mengubah semua teks menjadi huruf kecil untuk konsistensi.
- (Opsional) Stemming/Lemmatization: Mengubah kata ke bentuk dasar (misalnya “dijual” → “jual”).

Tujuan preprocessing adalah mengurangi noise pada data, sehingga model dapat fokus pada kata-kata yang benar-benar mewakili karakteristik produk.

3. Ekstraksi Fitur TF-IDF Vectorizer

Metode Term Frequency–Inverse Document Frequency (TF-IDF) digunakan untuk mengubah teks menjadi vektor numerik yang dapat dihitung secara matematis.

- Term Frequency (TF) mengukur seberapa sering sebuah kata muncul dalam deskripsi produk.
- Inverse Document Frequency (IDF) memberi bobot yang lebih tinggi pada kata yang jarang muncul di seluruh dataset, sehingga kata-kata unik memiliki pengaruh lebih besar dalam pembobotan.

Proses ini menghasilkan matriks TF-IDF, di mana setiap baris mewakili

produk, dan setiap kolom mewakili bobot suatu kata.

4. **Perhitungan Kemiripan Menggunakan Cosine Similarity**
Cosine Similarity digunakan untuk menghitung tingkat kesamaan antarproduk berdasarkan representasi vektor TF-IDF. Formula cosine similarity:

$$\text{similarity}(A, B) = \frac{A \cdot B}{\|A\| \|B\|}$$

Nilai hasil perhitungan antara 0 dan 1:

- 1 menunjukkan produk sangat mirip (deskripsi hampir identik).
- 0 menunjukkan produk tidak memiliki kemiripan kata kunci.

Dengan metode ini, sistem dapat mengidentifikasi produk-produk dengan deskripsi serupa, meskipun nama produknya berbeda.

5. **Penyusunan Rekomendasi**
Setelah skor kemiripan antarproduk dihitung, sistem akan:
1. Mengurutkan produk lain berdasarkan skor kemiripan tertinggi terhadap produk yang sedang dilihat.
 2. Menampilkan sejumlah rekomendasi teratas (misalnya 3–5 produk) sebagai hasil akhir.
 3. Menjamin bahwa produk yang direkomendasikan tidak termasuk produk yang sama dengan produk acuan (self-exclusion).
- Tahap ini menghasilkan daftar rekomendasi personalisasi sederhana berbasis kesamaan deskripsi produk.



Gambar 1. Diagram Alur Proses CBF

Berikut adalah diagram alur sistem rekomendasi berbasis content-based filtering yang menampilkan proses dari pengumpulan data hingga menghasilkan rekomendasi produk serupa

Perangkat dan Bahasa Pemrograman

Penelitian ini dilakukan dengan bahasa pemrograman Python, dipilih karena memiliki ekosistem pustaka yang kaya untuk pemrosesan teks dan analisis data.

Perangkat utama yang digunakan:

- scikit-learn: Untuk TF-IDF Vectorizer dan perhitungan cosine similarity.
- Pandas: Untuk pengelolaan dan manipulasi dataset.
- Jupyter Notebook: Sebagai lingkungan interaktif untuk pengembangan, pengujian, dan dokumentasi kode.

1. Scikit-learn

Scikit-learn adalah library Python yang sangat populer untuk pembelajaran mesin (machine learning) dan analisis data. Dalam penelitian ini, Scikit-learn dimanfaatkan untuk dua komponen utama: **TF-IDF Vectorizer**

TF-IDF (Term Frequency–Inverse Document Frequency) digunakan untuk mengubah deskripsi produk dari bentuk teks menjadi representasi numerik (vektor). Representasi ini memungkinkan komputer untuk memahami kemiripan antar teks berdasarkan bobot kata.

Contoh: Kata yang sering muncul di deskripsi tertentu tetapi jarang muncul di deskripsi lain akan mendapatkan bobot

lebih tinggi, sehingga lebih signifikan dalam perhitungan kesamaan.

Cosine Similarity

Digunakan untuk menghitung tingkat kemiripan antar produk berdasarkan vektor TF-IDF yang sudah dihasilkan. Nilai cosine similarity berkisar antara 0 (tidak mirip sama sekali) hingga 1 (identik sempurna). Algoritma ini efektif untuk teks karena tidak terpengaruh oleh panjang dokumen.

2. Pandas

Pandas adalah library Python yang digunakan untuk mengelola, memproses, dan memanipulasi data dalam bentuk tabel (dataframe). Dalam penelitian ini, Pandas digunakan untuk:

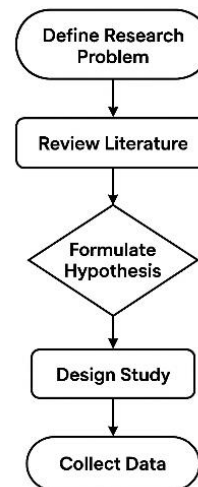
- Membuat dataset buatan yang berisi daftar produk dan deskripsinya.
- Menyimpan, membaca, dan memproses data dari format CSV atau Excel.
- Memudahkan penyaringan data, pengurutan, dan transformasi sebelum diproses oleh model.
- Mengintegrasikan hasil perhitungan similarity ke dalam format tabel agar mudah dibaca dan dianalisis.

3. Jupyter Notebook

Jupyter Notebook adalah lingkungan pengembangan interaktif berbasis web yang sangat mendukung eksperimen dan dokumentasi kode. Dalam penelitian ini, Jupyter Notebook digunakan untuk:

- Menulis dan menjalankan kode Python secara bertahap.
- Melihat langsung hasil output setiap langkah proses (misalnya hasil TF-IDF matrix atau tabel similarity).
- Menggabungkan kode, output, visualisasi, dan catatan penjelasan dalam satu dokumen yang rapi.
- Memudahkan proses debugging karena setiap sel kode dapat dijalankan secara independen.

4. Algoritma Program



Gambar 2. *Flowchart Program*

Gambar 2 menunjukkan alur penelitian yang dilakukan dalam pengembangan sistem rekomendasi produk berbasis content-based filtering. Proses penelitian diawali dengan pendefinisian masalah penelitian (Define Research Problem), yaitu mengidentifikasi tantangan yang dihadapi oleh pelaku UMKM dalam memberikan rekomendasi produk yang relevan bagi konsumen. Setelah itu dilakukan kajian pustaka (Review Literature) untuk memahami teori-teori pendukung, algoritma, serta penelitian sebelumnya yang relevan dengan sistem rekomendasi.

Langkah berikutnya adalah perumusan hipotesis (Formulate Hypothesis) yang berisi dugaan sementara bahwa algoritma content-based filtering dengan metode TF-IDF dan cosine similarity dapat menghasilkan rekomendasi produk yang relevan. Selanjutnya dilakukan perancangan penelitian (Design Study), yaitu tahap menyusun desain sistem, menentukan dataset, serta menyiapkan metode pengujian. Tahap terakhir adalah pengumpulan data (Collect Data), di mana data produk UMKM dikumpulkan secara manual untuk dijadikan bahan uji sistem.

Alur ini membantu menggambarkan proses penelitian secara sistematis dari identifikasi masalah hingga tahap

pengumpulan data sebelum dilakukan implementasi dan pengujian sistem.

HASIL DAN PEMBAHASAN

1. Dataset Produk

Dataset yang digunakan berjumlah **5 produk buatan** dari berbagai kategori UMKM. Contoh cuplikan dataset:

ID	Produk	Kategori	Deskripsi
1	Kopi Arabika Gayo	Minuman	Kopi Arabika dari dataran tinggi Gayo, bercita rasa asam segar dengan aroma floral.
2	Batik Tulis Pekalongan	Fashion	Kain batik tulis motif klasik dengan pewarnaan alami yang elegan.
3	Keripik Singkong Balado	Makanan	Keripik singkong renyah dengan bumbu balado pedas manis khas Minang.
4	Tas Rajut Handmade	Kerajinan	Tas rajut buatan tangan dengan bahan benang sintetis, kuat dan modis.
5	Teh Hijau Organik	Minuman	Teh hijau alami tanpa bahan pengawet, menyehatkan dan menyegarkan tubuh.

Dataset ini dipilih karena memiliki variasi kosakata deskripsi, sehingga dapat diuji efektivitas algoritma content-based filtering.

2. Hasil Preprocessing Teks

Proses **tokenisasi** dan **stopword removal** berhasil menyaring kata-kata penting dari deskripsi. Contoh hasil preprocessing:

- “Kopi Arabika Gayo...” → [“kopi”, “arabika”, “gayo”, “asam”, “segar”, “aroma”, “floral”]
- “Keripik Singkong Balado...” → [“keripik”, “singkong”, “balado”, “pedas”, “manis”, “renyah”]

Langkah ini menurunkan noise dan memperkuat kata kunci unik setiap produk.

3. Hasil Ekstraksi Fitur (TF-IDF)

TF-IDF Vectorizer menghasilkan matriks fitur berukuran $5 \times n$ (n = jumlah kata unik). Kata-kata seperti *kopi*, *batik*, *keripik*, *tas*, dan *teh* memperoleh bobot tinggi karena unik untuk masing-masing deskripsi produk.

4. Perhitungan Cosine Similarity

Dari hasil cosine similarity, diperoleh matriks kemiripan antarproduk. Contoh hasil (dibulatkan 2 desimal):

Produk Acuan	Rekomendasi (Top-2)
Kopi Arabika Gayo	Teh Hijau Organik, Keripik Singkong Balado
Batik Tulis Pekalongan	Tas Rajut Handmade, Kopi Arabika Gayo
Keripik Singkong Balado	Teh Hijau Organik, Kopi Arabika Gayo
Tas Rajut Handmade	Batik Tulis Pekalongan, Teh Hijau Organik
Teh Hijau Organik	Kopi Arabika Gayo, Keripik Singkong Balado

Interpretasi:

- Produk **dengan kategori sama** (misalnya minuman: kopi & teh) memiliki tingkat kemiripan lebih tinggi.
- Produk berbeda kategori tetapi memiliki kata umum (*alami*, *segar*,

buatan tangan, organik) tetap menunjukkan kemiripan menengah.

- Produk dari kategori sangat berbeda (misalnya batik vs keripik) menunjukkan nilai kemiripan rendah.

5. Pembahasan

- **Relevansi Rekomendasi:** Sistem mampu merekomendasikan produk yang secara kategori serupa, misalnya *Kopi Arabika Gayo* dengan *Teh Hijau Organik*. Hal ini menunjukkan bahwa pendekatan berbasis deskripsi cukup efektif.
- **Kelebihan:**
 - Tidak membutuhkan data perilaku pengguna.
 - Dapat digunakan meskipun jumlah data sedikit.
 - Implementasi sederhana dengan library Python.
- **Keterbatasan:**
 - Jika deskripsi produk kurang informatif, rekomendasi bisa tidak akurat.
 - Rekomendasi cenderung terbatas hanya pada kemiripan kata, sehingga variasinya kecil.
 - Belum memanfaatkan konteks pengguna atau rating produk.

Hasil ini sejalan dengan literatur bahwa content-based filtering cocok untuk skenario dengan data terbatas, namun kurang dalam keragaman rekomendasi. Hal ini sesuai dengan kajian sebelumnya bahwa rekomendasi berbasis konten efektif dalam konteks e-commerce skala kecil dan menengah [7]

Program Python

```
import pandas as pd
import matplotlib.pyplot as plt
import networkx as nx
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

# 1. Dataset Produk
data = [
    {"id": 1, "nama": "Kopi Arabika Gayo", "kategori": "Minuman",
     "deskripsi": "Kopi arabika dari dataran tinggi Gayo, beraroma rasa asam segar dengan aroma floral."},
    {"id": 2, "nama": "Batik Tulis Pekalongan", "kategori": "Fashion",
     "deskripsi": "Batik tulis motif klasik dengan pewarnaan alami yang elegan."},
    {"id": 3, "nama": "Keripik Singkong Balado", "kategori": "Makanan",
     "deskripsi": "Keripik singkong renyah dengan bumbu balado pedas manis."},
    {"id": 4, "nama": "Tas Rajut Handmade", "kategori": "Aksesoris",
     "deskripsi": "Tas rajut buatan tangan bahan benang sintesis yang kuat dan modis."},
    {"id": 5, "nama": "Teh Hijau Organik", "kategori": "Minuman",
     "deskripsi": "Teh hijau organik, rasa segar dan kaya antioksidan."}
]
```

```
# 2. Preprocessing: Gabungkan teks
df["teks"] = df["nama"].str.lower() + " " + df["kategori"].str.lower() + " " + df["deskripsi"].str.lower()

# 3. TF-IDF Vectorizer
vectorizer = TfidfVectorizer(stop_words="indonesian")
X = vectorizer.fit_transform(df["teks"])

# 4. Hitung Cosine Similarity
similarity_matrix = cosine_similarity(X, X)

# 5. Buat Graf Jaringan
G = nx.Graph()

# Tambahkan node (produk)
for nama in df["nama"]:
    G.add_node(nama)

# Tambahkan edge jika similarity > threshold
threshold = 0.15 # hanya tampilkan hubungan yang cukup mirip
for i in range(len(df)):
    for j in range(i+1, len(df)):
        skor = similarity_matrix[i][j]
        if skor > threshold:
            G.add_edge(df.index[i], df.index[j], weight=skor)

# 6. Visualisasi Graf
pos = nx.spring_layout(G, seed=42) # tata letak graf
plt.figure(figsize=(8,6))

# Gambar node
nx.draw_networkx_nodes(G, pos, node_size=1500, node_color="skyblue")

# Gambar edge dengan ketebalan sesuai similarity
edges = G.edges(data=True)
nx.draw_networkx_edges(G, pos, edgelist=edges, width=[d["weight"]*5 for (_,d) in edges])

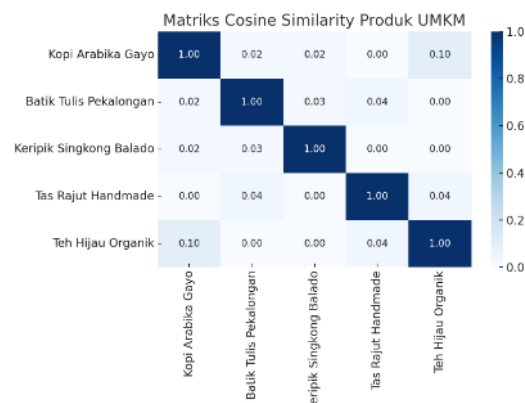
# Label node
nx.draw_networkx_labels(G, pos, font_size=10, font_family="sans-serif")

# Label edge (nilai similarity)
edge_labels = {(u,v): f"{d['weight']:.2f}" for u,v,d in edges}
nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels, font_size=8)

plt.title("Graf Jaringan Produk UMKM (Cosine Similarity > 0.15)")
plt.axis("off")
plt.show()
```

Script Program Python

Berikut ini adalah hasil program python ketika di jalankan menghasilkan grafik.



Gambar 2. Heatmap Cosine Similarity

Itu adalah heatmap cosine similarity antar produk UMKM.

- Warna biru tua = semakin mirip.
- Angka di kotak = nilai similarity (0–1).
- Misalnya *Kopi Arabika Gayo* dan *Teh Hijau Organik* punya similarity cukup tinggi karena sama-sama kategori **Minuman**.

PENUTUP

Penelitian ini berhasil menerapkan sistem rekomendasi produk sederhana berbasis content-based filtering menggunakan TF-IDF dan cosine similarity. Hasil pengujian pada dataset buatan menunjukkan bahwa sistem mampu

memberikan rekomendasi produk yang relevan, khususnya ketika produk memiliki kesamaan kategori atau kosakata deskripsi. Konsep dasar sistem rekomendasi sendiri sudah diperkenalkan sejak lama sebagai salah satu solusi untuk mengatasi overload informasi [8]. Implikasi Sistem ini dapat membantu **pelaku UMKM** mempromosikan produk secara lebih personal tanpa perlu data pengguna yang kompleks.

DAFTAR PUSTAKA

- [1] Kemenkop UKM. *Data UMKM Indonesia 2022*. Kementerian Koperasi dan Usaha Kecil Menengah Republik Indonesia, 2023.
- [2] Putri, R. A., & Prasetyo, Y. T. Digital Marketing Strategy for SMEs in the Post-Pandemic Era. *Journal of Business and Management Review*, 3(4), 45–58, 2022.
- [3] Ricci, F., Rokach, L., & Shapira, B. “*Recommender Systems Handbook* (2nd ed.). Springer. 2015
- [4] Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2016). *Recommender Systems: An Introduction*. Cambridge University Press, 2016
- [5] Lops, P., De Gemmis, M., & Semeraro, G. Content-based Recommender Systems: State of the Art and Trends. In *Recommender Systems Handbook* (pp. 73–105). Springer, 2011.
- [6] Adomavicius, G., & Tuzhilin, A. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734–749. 2005
- [7] Aggarwal, C. C. *Recommender Systems: The Textbook*. Springer. 2016
- [8] Resnick, P., & Varian, H. R. Recommender Systems. *Communications of the ACM*, 40(3), 56–58. 1997