Penggunaan Latent Semantic Analysis (LSA) berbasis Singular Value Decomposition (SVD) dalam Rekomendasi Parfum

Muhammad Jordan Ferimeison - 13524047
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13524047@mahasiswa.itb.ac.id, ferimeison@gmail.com

***Abstract*—Parfum adalah salah satu cara seseorang mengekspresikan dirinya. Untuk mengenal kualitas suatu parfum, seseorang biasanya harus merasakan wanginya secara langsung atau mencari rekomendasi kesumber yang relevan. Rekomendasi parfum dapat diterapkan dengan Teknik Latent Semantic Analysis (LSA) pada bahan baku dan deskripsi dari suatu parfum. Teknik ini menyajikan hasil yang mampu melihat bahan baku dan deskripsi parfum secara menyeluruh.**

***Keywords*—Singular Value Decomposition, Recommendation System, .**

I. PENDAHULUAN

Parfum adalah salah satu barang yang digunakan oleh Masyarakat untuk meningkatkan kualitas diri. Mulai dari hal fungsional seperti wangi, hingga ke aspek social seperti status. Parfum dapat menjadi cara seseorang memperkenalkan diri.

Di era digital, pembelian parfum dapat dilakukan dimanapun dan kapanpun. Hal ini tentunya memudahkan mayoritas Masyarakat yang ingin mencari parfum. Namun, muncul masalah baru dimana Masyarakat tidak dapat mengetahui bau parfum tersebut secara langsung. Oleh karena itu, dibutuhkan suatu sumber yang mampu membantu Masyarakat dalam memilih parfum yang mereka inginkan.

Makalah ini dibuat dengan beberapa tujuan,

1) Apakah materi Aljabar Linier dan Geometri dapat diterapkan dalam rekomendasi parfum
2) Bagaimana cara menerapkan materi Aljabar Linier dan Geometri dalam rekomendasi parfum

Makalah ini merupakan makalah yang dibuat dalam rangka pemenuhan tugas makalah IF2123 Aljabat Linier dan Geometri, karena pertanyaan dari teman, dan dengan melakukan evaluasi pada Kaggle Notebook Aruneem Bhowmick [2].

II. DAFTAR PUSTAKA

Dalam pengerjaan makalah ini, penulis menggunakan konsep matriks, dekomposisi matriks dengan SVD, pencarian similaritas dengan kosinus, dan pembobotan dengan matriks tf-idf.

A. Matriks

Matriks adalah salah satu cara merepresintasikan data dalam baris dan kolom. Suatu matriks m x n adalah matriks dengan m baris dan n kolom [1].

$$A = [a_{ij}] = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

Gambar 2.1 Matriks
[https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-01-Review-Matriks-2025.pdf]

B. TF-IDF Matriks

Term Frequency-Inverse Document Frequency (TF-IDF) adalah salah satu representasi data dengan membangun matriks *term-document* yang berisi kata unik dan kemunculannya di tiap dokumen. Matriks *term-document* direpresentasikan seperti berikut dengan tiap baris menandakan kata unik dan tiap kolom menandakan dokumen unik.

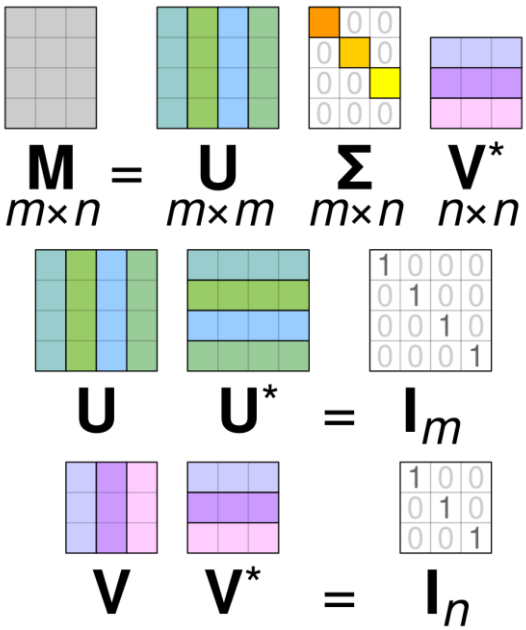
$$A = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1n} \\ f_{21} & f_{22} & \cdots & f_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ f_{m1} & f_{m2} & \cdots & f_{mn} \end{bmatrix}$$

Gambar 2.2 Matriks *term-document*
[https://drive.google.com/file/d/1TQNSAUgb0j3cthdf7uXwYvP_rH8VMPBU/view]

Matriks *term-document* kemudian diolah dengan mengalikan jumlah dokumen yang mengandung kata unik ke-i, dan menghitung bobotnya (matriks IDF), dengan hasil normalisasi setiap kolom matriks *term-document*, menghitung proporsi kemunculan kata di setiap dokumen(kolom).

C. Signular Value Decomposition

Singular Value Decomposition (SVD) adalah salah satu Teknik dekomposisi matriks menjadi matriks U (matriks singular kiri), matriks ∑ (nilai singular), dan matriks V (matrujs singular kanan) [1]. Dengan U dan V adalah matriks orthogonal, matriks persegi yang kolomnya saling orthogonal. Matriks U dan V didapat dari mengalikan matriks M dengan transpose matriks M. Nilai singular pada matriks ∑ adalah nilai eigen yang didpat dari matriks U atau V (nilai eigen unik keduanya sama).



Gambar 2.3 Singular Value Decomposition (SVD) [Cmglee - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=67853297]

D. Cosine Similarity

Kedekatan antara dua vector dapat dihitung dengan menggunakan cosine similarity. Pendekatan ini memanfaatkan hasil dari Singular Value Decomposition (SVD), dengan melakukan transformasi hasil SVD menjadi vector yang disebut sebagai *embeddings*. Nilai kedekatan (similarity) antara dua embeddings a dan b dapat dihitung menggunakan rumus,

$$\text{sim}(\mathbf{a},\mathbf{b}) = \frac{\mathbf{a}\cdot\mathbf{b}}{\|\mathbf{a}\|_2\|\mathbf{b}\|_2} = \frac{\sum_{i=1}^{k} a_i b_i}{\sqrt{\sum_{i=1}^{k} a_i^2}\sqrt{\sum_{i=1}^{k} b_i^2}},$$

Gambar 2.4 Cosine Similarity [https://drive.google.com/file/d/1TQNSAUgb0j3cthdf7uXwYvP_rH8VMPBU/view]

Dimana nilai kemiripannya berkisar antara [-1,1] dengan 1 menunjukkan kemiripan kuat (vektor searah) dan -1 menunjukkan kemiripan yang tidak kuat (arah vektor berbeda).

III. IMPLEMENTASI

A. Konsep

Sistem rekomendasi parfum akan dibuat dengan basis Latent Semantic Analysis (LSA). Data parfum akan berisi nama, bahan baku, dan deskripsi parfum. Bahan baku dan deskripsi nantinya akan diolah menjadi matriks Term Frequency-Inverse Document Frequency (TF-IDF0 dengan bahan baku berbobot lebih tinggi(akan dilakukan pengolahan data sehingga pembobotan bahan baku menjadi lebih signifikan). Matriks ini selanjutnya akan diolah menggunakan Singular Value Decomposition (SVD) untuk mencari fitur tersembunyi dari hubungan antar elemen unik pada matriks TF-IDF. Untuk meminimalisir overhead dan memaksimalkan Tingkat signifikan dari fitur tersembunyi, akan diambil k singular value teratas. Metode ini disebut dengan Truncated SVD. K singular value teratas kemudian diproyeksikan dengan basisnya pada V_k untuk membentuk *embeddings*. Tiap komponen embeddings merepresentasikan fitur tersembunyi tertentu. *Embeddings* yang didapat kemudian digunakan untuk membandingkan Tingkat kemiripan tiap parfum pada dataset dengan menggunakan *cosine similarity*.

Teknik Latent Semantic Analysis (LSA) dipilih karena dapat menapilkan hubungan tersembunyi antar kata yang muncul pada bahan baku dan deskripsi parfum.

B. Realisasi

Pada saat mencari dataset yang dapat digunakan di Kaggle, penulis menemukan dataset yang ada pada system rekomendasi parfum yang telah dibuat oleh Aruneem Bhowmick [2]. Sistem ini menggunakan pembobotan TF-IDF dalam implementasinya. Berikut Adalah kode sumber pada Kaggle Notebook yang penulis temukan,

```
  import pandas as pd

  from sklearn.feature_extraction.text
import TfidfVectorizer
  from sklearn.metrics.pairwise import
cosine_similarity

  import requests
  from IPython.display import display,
HTML

  perfumes =
pd.read_csv('/kaggle/input/perfume-
recommendation-
dataset/final_perfume_data.csv', encoding
= 'unicode_escape')
  perfumes = perfumes.rename(columns =
{'Notes' : 'Ingredients'})
  perfumes.fillna('', inplace = True)
```

```
  perfumes['Full Description'] =
perfumes['Description'] + ' ' +
perfumes['Ingredients']
  perfumes

  vectorizer = TfidfVectorizer(stop_words
= 'english')
  tfidf_matrix =
vectorizer.fit_transform(perfumes['Full
Description'])
  cosine_sim =
cosine_similarity(tfidf_matrix,
tfidf_matrix)

  def
recommend_perfumes(selected_perfumes, n =
10):

      selected_indices =
perfumes[perfumes['Name'].isin(selected_p
erfumes)].index
      sim_scores =
cosine_sim[selected_indices].mean(axis =
0)
      similar_indices =
sim_scores.argsort()[::-1]
      recommended_indices = [i for i in
similar_indices if i not in
selected_indices][:n]

      return
perfumes.iloc[recommended_indices][['Name
', 'Brand', 'Description', 'Ingredients',
'Image URL']]

  selected_perfumes = perfumes.sample(n =
5, random_state = None)['Name']
  print('Selected Perfumes: ' + ',
'.join(selected_perfumes) + '\n')

  recommendations =
recommend_perfumes(selected_perfumes)
  recommendations

  def display_perfume_info(perfumes):
      html_content = ""

      for _, row in perfumes.iterrows():
          perfume_html = f"""
          <div style="display: flex;
align-items: center; border: 1px solid
#ddd;
                      padding: 15px;
margin: 10px 0; border-radius: 10px;
                      background-color:
#fafafa;">
              <img src="{row['Image
URL']}" width="150" height="150"
                   style="border-radius:
10px; margin-right: 20px;">
              <div>
                  <h3 style="margin: 0;
color: #333;">{row['Name']}</h3>
                  <p style="margin: 5px
0; font-weight: bold; color:
#555;">{row['Brand']}</p>
                  <p style="margin: 5px
0; color:
#666;"><strong>Description:</strong>
{row['Description']}</p>
                  <p style="margin: 5px
0; color:
#666;"><strong>Ingredients:</strong>
{row['Ingredients']}</p>
              </div>
          </div>
          """
          html_content += perfume_html

    display(HTML(html_content))
display_perfume_info(recommendations)
```

Gambar 3.1 Kode Sumber Kaggle Notebook [https://www.kaggle.com/code/aruneembhowmick/eaumatch-a-perfume-recommendation-system/notebook]

Notebook tersebut juga berisi dataset dengan 5 kolom berjudul Name, Brand, Description, Notes, Image URL. Dataset ini cocok dengan pnerapan Latent Semantic Analysis (LSA) karena memiliki deskripsi produk dan bahan baku pada notes yang berbentuk teks.

Penulis kemudian menggunakan Kaggle Notebook tersebut sebagai bahan evaluasi untuk menerapkan system rekomendasi parfum menggunkan Latent Semantic Analysis (LSA).

1) Perubahan pada *import library*
   Penulis menambahkan fungsi TruncatedSVD dari library sklearn untuk dekomposisi matriks.

```
  import pandas as pd

  from
sklearn.feature_extraction.text
import TfidfVectorizer
  from sklearn.metrics.pairwise
import cosine_similarity
  from sklearn.decomposition import
TruncatedSVD

  import requests
from IPython.display import display,
HTML
```

Gambar 3.2 Kode Sumber 1

2) Menambah tingkat signifikan bahan baku

Penulis merasa bahan baku berperan lebih signifikan dibanding deskripsi produk. Bahan baku merupakan komponen pembangun dari parfum (komponen objektif) dan deskripsi merupakan persepsi terhadap produk (komponen subjektif). Secara subjektif, persepsi seseorang terhadap produk dapat berubah seperti seberapa "maskulin" suatu parfum. Namun, komponen pembangun dari parfum tidak akan berubah. Oleh karena itu, pembobotan bahan baku diubah.

```
  perfumes =
pd.read_csv('/kaggle/input/perfume-
recommendation-
dataset/final_perfume_data.csv',
encoding = 'unicode_escape')
  perfumes = perfumes.rename(columns
= {'Notes' : 'Ingredients'})
  perfumes.fillna('', inplace =
True)
  perfumes['Full Description'] =
perfumes['Description'] + ' ' +
perfumes['Ingredients']
  perfumes['Full Description'] =
perfumes['Description'] + ' ' +
perfumes['Ingredients'] * 2
```

Gambar 3.3 Kode Sumber 2

Dengan mengalikan kolom 'Ingredients' dengan suatu factor k, frekuensi kemunculannya akan bertambah. Hal ini akan memberikan pembobotan lebih ketika menghitung matriks TF-IDF. Misal, bahan baku 'Vanilla' awalnya memiliki bobotn 1/38. Karena dikalikan dengan k, bobotnya akan berubah menjadi (1+k)/(38+k). Bobotnya meningkat, membuat bahan baku menjadi lebih signifikan

3) Menambahkan *Truncated Singular Value Decomposition* (SVD)

```
  vectorizer =
TfidfVectorizer(stop_words =
'english')
  tfidf_matrix =
vectorizer.fit_transform(perfumes['F
ull Description'])
  svd = TruncatedSVD(n_components=5)
  newPerfumeMatrix =
svd.fit_transform(tfidf_matrix)

  cosine_sim =
cosine_similarity(tfidf_matrix,
tfidf_matrix)
cosine_sim =
cosine_similarity(newPerfumeMatrix)
```

Gambar 3.4 Kode Sumber 3

Truncated SVD ditambahkan untuk menghitung k 'vibes' teratas yang dimiliki oleh parfum. Penambahan ini diharapkan memberikan jawaban yang dapat melihat parfum secara holistic, bukan hanya berdasarkan pembobotan TF-IDF.

4) Fungsi bantuan

```
  def
recommend_perfumes(selected_perfumes
, n = 10):

      selected_indices =
perfumes[perfumes['Name'].isin(selec
ted_perfumes)].index
      sim_scores =
cosine_sim[selected_indices].mean(ax
is = 0)
      similar_indices =
sim_scores.argsort()[::-1]
      recommended_indices = [i for i
in similar_indices if i not in
selected_indices][:n]

      return
perfumes.iloc[recommended_indices][[
'Name', 'Brand', 'Description',
'Ingredients', 'Image URL']]

  selected_perfumes =
perfumes.sample(n = 5, random_state
= None)['Name']
  recommendations =
recommend_perfumes(selected_perfumes
)

  def
display_perfume_info(perfumes):
      html_content = ""

      for _, row in
perfumes.iterrows():
          perfume_html = f"""
          <div style="display: flex;
align-items: center; border: 1px
solid #ddd;
                      padding: 15px;
margin: 10px 0; border-radius: 10px;
                      background-
color: #fafafa;">
              <img src="{row['Image
URL']}" width="150" height="150"
                   style="border-
radius: 10px; margin-right: 20px;">
```

```
                <div>
                    <h3 style="margin:
0; color: #333;">{row['Name']}</h3>
                    <p style="margin:
5px 0; font-weight: bold; color:
#555;">{row['Brand']}</p>
                    <p style="margin:
5px 0; color:
#666;"><strong>Description:</strong>
{row['Description']}</p>
                    <p style="margin:
5px 0; color:
#666;"><strong>Ingredients:</strong>
{row['Ingredients']}</p>
                </div>
            </div>
            """
            html_content +=
perfume_html

        display(HTML(html_content))
```

Gambar 3.5 Kode Sumber 4

Penulis tidak terlalu mengubah fungsi bantuan secara keseluruhan. Penulis menggunakan fungsi bantuan seperti yang telah diimplementasikan oleh referensi [2]. Perubahan fungsi yang dilakukan untuk menampilkan hasil berdasarkan cosine_sim yang berbeda dengan menambahkan parameter pada recommend_perfumes.

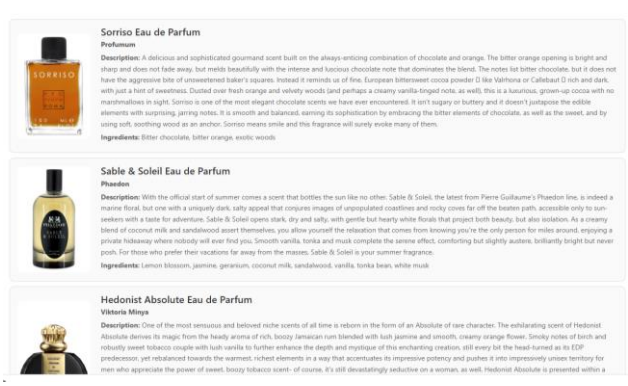
5) Menampilkan hasil rekomendasi

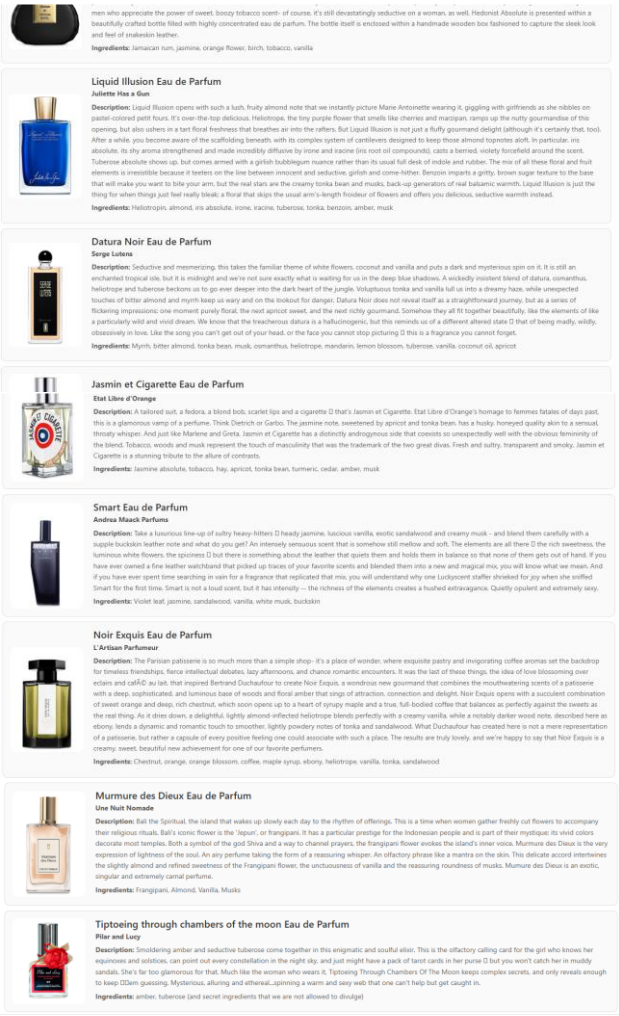
```
display_perfume_info(recommendations)
```

Gambar 3.6 Kode Sumber 5

IV. HASIL

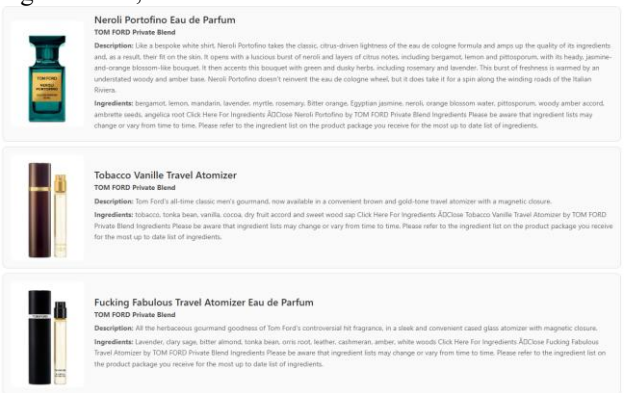
Sebuah tes dilakukan untuk menguji perbedaan penggunaan Teknik Latent Semantic Analysis (LSA) dan matriks Term Frequency-Inverse Document Frequency (TF-IDF) dengan menggunakan sampel parfum Vanille Absolu Eau de Parfum, Dent de Lait Eau de Parfum, Neroli Portofino Travel Atomizer, Nice Bergamote Eau de Parfum, My Own Private Teahupo'o Eau de Parfum. Pada tes menggunakan LSA, didapat hasil sebagai berikut

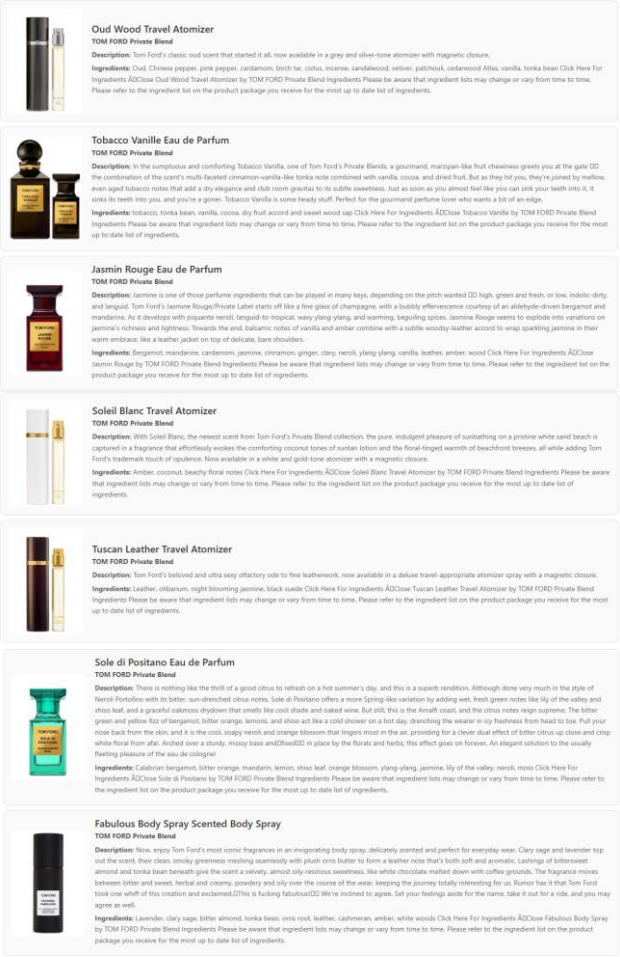




Gambar 3.6 Hasil Rekomendasi LSA

Sedangkan, menggunakan TF-IDF mendapatkan hasil sebagai berikut,





Gambar 3.7 Hasil Rekomendasi TF_IDF

V. KESIMPULAN DAN SARAN

Berdasarkan eksperimen yang dilakukan, penggunaan LSA memberikan hasil yang berbeda dengan TF-IDF yang telah diimplementasikan oleh referensi [2]. Penulis belum dapat menarik Kesimpulan apakah hasil yang ditunjukkan oleh LSA lebih baik daripada TF-IDF. Namun berdasarkan hasil observasi penulis, hasil yang ditunjukkan oleh TF-IDF lebih bersifat monoton dan berfokus pada satu parfum. Hasil pada TF-IDF di sampel ini lebih menonjol kepada TOM FORD dan Atomizer. Sedangkan, hasil dari LSA memberikan hasil yang lebih condong ke karakteristik dari parfumnya. Seperti karakter ‘sweet’ atau ‘sweetness’ yang lebih ditonjolkan dibandingkan dengan hasil pada TF-IDF.

Saran yang data diberikan adalah agar kedepannya tes yang dilakukan dapat lebih menyeluruh. Tes dapat dilakukan dengan melibatkan feedback Masyarakat secara langsung. Eksplorasi metode lain juga dapat digunakan untuk menghasilkan system rekomendasi yang lebih baik.

VI. LAMPIRAN

Google Colab berisi kode dari Kaggle Notebook penulis,
https://colab.research.google.com/drive/1NpEuEQS7Wec5xpOM5ehmP03zSoeRkl5W?usp=sharing

VII. ACKNOWLEDGMENT

Penulis berterima kasih kepada Tuhan Yang Maha Esa karena telah mengizinkan penulis membuat makalah ini. Penulis berterima kasih kepada Dr. Ir. Rinaldi Munir, M.T. sebagai dosen pengampu IF2123 K1. Penulis berterima kasih kepada Natanael Pangaribuan dan Yudhis Febrian yang karena telah bertanya apakah memungkinkan merekomendasikan parfum dengan teknologi. Penulis berterima kasih kepada pengguna Kaggle @aruneembhowmick karena telah membuat system rekomendasi parfum berbasis TF-IDF matriks.

REFERENCES

[1] R. Munir, *Aljabar Geometri*. Website. 24 Desember 2025. https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/algeo25-26.htm#SlideKuliah

[2] A. Bhowmick, *“EauMatch: A Perfume Recommendation System”*. Kaggle Notebook. 24 Desember 2024. https://www.kaggle.com/code/aruneembhowmick/eaumatch-a-perfume-recommendation-system/notebook

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 24 Desember 2025

Muhammad Jodan Ferimeion - 13524047