

SISTEM REKOMENDASI CONTENT-BASED FILTERING SKINCARE PRIA DI E-COMMERCE SHOPEE

CONTENT-BASED FILTERING MEN'S SKINCARE RECOMMENDATION SYSTEM IN SHOPEE E-COMMERCE

Ananto Dwi Saputro¹, Fatkhul Amin²

^{1,2}Teknik Informatika Fakultas Teknologi Informasi dan Industri, Unisbank Semarang
anantosaputro24@gmail.com¹, fatkhulamin@edu.unisbank.ac.id²

ABSTRACT

This research was conducted to create a system that makes it easier for users to buy men's skincare face wash in shopee e-commerce. The problem that is often experienced by users is that they want to replace the products they usually use with new products, but they are confused to find suitable products. Based on these problems, this recommendation system was created using the Content-based Filtering method with the TF-IDF algorithm and cosine similarity. This Content-based Filtering method does not require reviews or opinions of others in determining the results of recommendations. The TF-IDF algorithm is used to weight words and cosine similarity to find the level of similarity of each word that has been weighted. The data collection method used is primary data by searching for male skincare face wash products directly in shopee e-commerce and secondary data by reading related research journals. The trial of this recommendation system was carried out directly by researchers using face wash products used previously by researchers. The way the system works is to input skincare face wash products to be compared, in this case the researcher uses 15 products from a variety of different products. The results obtained are ranked from those with the highest level of similarity to those with the lowest level of similarity.

Keywords: *Recommendation System, Content-Based Filtering, Men's Skincare*

ABSTRAK

Penelitian ini dilakukan untuk membuat suatu sistem yang memudahkan pengguna dalam membeli skincare face wash pria di e-commerce shopee. Permasalahan yang seringkali dialami oleh pengguna yaitu mereka ingin mengganti produk yang biasa mereka pakai dengan produk baru, tetapi mereka bingung untuk mencari produk yang sesuai. Berdasarkan permasalahan tersebut maka dibuatlah sistem rekomendasi ini dibuat menggunakan metode Content-based Filtering dengan algoritma TF-IDF dan cosine similarity. Metode Content-based Filtering ini tidak membutuhkan ulasan ataupun pendapat orang lain dalam menentukan hasil rekomendasi. Algoritma TF-IDF digunakan untuk melakukan pembobotan kata dan cosine similarity untuk mencari tingkat kemiripan setiap kata yang sudah dibobot. Metode pengambilan data yang digunakan yaitu data primer dengan mencari produk skincare face wash pria secara langsung di e-commerce shopee dan data sekunder dengan membaca jurnal-jurnal penelitian terkait. Uji coba sistem rekomendasi ini dilakukan secara langsung oleh peneliti dengan menggunakan produk face wash yang dipakai sebelumnya oleh peneliti. Cara kerja sistem yaitu melakukan input produk skincare face wash yang akan dibandingkan, dalam hal ini peneliti menggunakan 15 produk dari berbagai macam produk yang berbeda. Hasil yang didapat dirangking dari yang memiliki tingkat similaritas paling tinggi hingga yang memiliki tingkat similaritas yang paling rendah.

Kata Kunci Sistem Rekomendasi, Pemfilteran Berbasis Konten, Perawatan Kulit Pria.

PENDAHULUAN

Kemajuan *Content-based Filtering* (CBF) mengikuti perkembangan teknologi dan internet. Penyaringan berbasis konten menjadi metode yang populer diaplikasikan dalam metode perekomendasi di berbagai macam platform seperti media sosial, platform streaming, e-commerce dan lain-lain. Salah satu penerapan Metode *Content-based Filtering* yang saat ini diterapkan

yaitu dalam bidang industri di Indonesia meliputi bidang *e-commerce*, media dan hiburan, periklanan digital dan lain-lain. *Content-based Filtering* dalam *e-commerce* dapat digunakan untuk memberikan rekomendasi produk kepada pengguna berdasarkan preferensi dan perilaku mereka.

Shopee merupakan satu dari banyak toko online yang digemari di negara Indonesia kita ini. Shopee menyediakan

kebutuhan elektronik, perawatan kulit / *skincare*, perlengkapan tempat tinggal, *fashion* dan lain-lain. Salah satu kebutuhan yang paling dicari adalah perawatan kulit / *skincare*. *Skincare* Pria yang disediakan shopee terdapat berbagai macam brand terkenal maupun brand yang baru mulai merintis. Banyaknya pilihan *skincare* membuat kaum pria bingung untuk menentukan produk mana yang paling sesuai untuk dibeli. Pemilihan *skincare* pria harus sesuai dengan jenis kulit, supaya tidak mengakibatkan ketidakcocokan *skincare* yang mengakibatkan munculnya permasalahan kulit lainnya seperti timbulnya jerawat, noda kehitaman, warna kulit tampak kusam dan masih banyak lagi.

Sistem rekomendasi diperlukan dalam mengatasi masalah ini. Berdasarkan permasalahan yang ada akan dibuat sistem rekomendasi dengan algoritma Penyaringan berbasis konten. Penyaringan berbasis konten tidak menyertakan pemakai lainnya saat menetapkan perekomendasi, melainkan pemakai tersebutlah yang menjadi acuannya. Mengacu atas yang pengguna cari, algoritma hanya akan menentukan artikel dengan konten serupa untuk disarankan kepada pengguna. Metode penyaringan berbasis konten mengambil data yang terkandung dalam sebuah item dan membandingkannya dengan informasi dalam artikel yang sebelumnya pernah dibeli, ditinjau atau disenangi oleh pemakai. Maka dari itu penulis melakukan penelitian untuk merekomendasikan *skincare* pria yang paling sesuai di *e-commerce* Shopee

METODE

Banyaknya produk *skincare* pria yang ada di *e-commerce* shopee membuat kaum pria bingung untuk memilih *skincare face wash* yang paling cocok untuk mereka. Terdapat suatu kondisi dimana pengguna sudah cocok dengan *skincare face wash* yang biasa dipakai, namun untuk harganya lumayan mahal.

Pemilihan *skincare facewash* dengan manfaat yang sama dan juga harga yang relatif lebih terjangkau sangat diperlukan. Penelitian ini dilakukan untuk membuat suatu sistem rekomendasi berupa program untuk merekomendasikan *skincare* pria di shopee menggunakan metode *content-based filtering*.

Pengumpulan Data

Adapun sumber data yang digunakan dalam penelitian ini dibagi menjadi dua yaitu :

- a. Data primer merupakan data yang berkaitan dengan indikator penelitian yang diperoleh langsung dari lokasi penelitian melalui observasi langsung di lapangan. Data primer penelitian ini diambil secara langsung dengan mencari produk *skincare face wash* pria di *e-commerce* shopee dengan pemilihan untuk semua jenis kulit, harga yang relatif sama serta merek *skincare face wash* pria yang familiar.
- b. Data sekunder merupakan data yang mendukung informasi dari berbagai sumber. Data sekunder untuk penelitian ini dikumpulkan melalui tinjauan pustaka yaitu mencari studi literatur untuk informasi dasar tentang isu-isu yang berkaitan dengan penelitian ini. Data sekunder penelitian ini meliputi buku-buku, artikel-artikel, dataset *skincare* pria, jurnal-jurnal ilmiah mengenai sistem informasi, rekayasa perangkat lunak, pemrograman web dan tools python.

Proses Implementasi Metode Content-based Filtering

Content-based filtering pada dasarnya melakukan 3 proses yaitu, pra-pemrosesan data, proses representasi data menggunakan TF-IDF dan perhitungan similaritas dengan *cosine similarity*. Proses metode *content-based filtering* adalah sebagai berikut :

- a. Pra-pemrosesan data
 1. Melakukan *case folding*

- Case folding* dilakukan untuk mengubah semua karakter huruf menjadi huruf kecil (lowercase).
- Melakukan *remove punctuation*
Remove punctuation dilakukan untuk menghapus tanda baca seperti titik, koma, tanda tanya, tanda seru dan kutipan.
 - Melakukan *remove multispace*
remove multispace dilakukan untuk menghapus atau mengganti beberapa spasi (spasi berulang) dalam teks dengan satu spasi.
- b. Representasi data menggunakan TF-IDF
- Persamaan Tf (*term frequency*) dapat dilihat pada persamaan 1.

$$TF = \frac{t}{d}$$

Dimana t adalah jumlah kemunculan kata tertentu dalam dokumen d. Sementara d merupakan total keseluruhan kata pada dokumen. Persamaan idf (*inverse document frequency*) dapat dilihat pada persamaan 2.

$$idf = \log \left(\frac{N}{df(t)} \right)$$

Dimana N adalah total dokumen yang ada, serta df(t) : jumlah dokumen yang memiliki kata t. Persamaan TD-IDF dapat dilihat pada persamaan 3.

$$TFidf = TF \cdot idf$$

- c. Perhitungan Similaritas dengan Cosine Similarity
- Hitung kemiripan vektor [manfaat] pada setiap produk *skincare face wash* pria yang ada pada dataset. Kemiripan produk berdasarkan vektor [manfaat] dihitung dengan menggunakan algoritma *cosine similarity* menggunakan rumus sebagai berikut :

$$\cos(\theta) = \frac{x \cdot y}{\|x\| \|y\|}$$

$$= \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}}$$

A = Vektor A, yang akan dibandingkan kemiripannya

B = Vektor B, yang akan dibandingkan kemiripannya

A • B = dot product antara vektor A dan vektor B

|A| = panjang vektor A

|B| = panjang vektor B

|A||B| = cross product antara |A| dan |B|

HASIL DAN PEMBAHASAN

Penerapan metode content-based filtering dalam pembuatan sistem rekomendasi skincare face wash pria di e-commerce shopee menggunakan algoritma TF-IDF dan cosine similarity. Pembuatan sistem rekomendasi ini sudah dirancang dengan baik sehingga untuk hasil yang didapat akan akurat. Berikut adalah implementasi sistemnya:

4.1. Import Library

Library yang digunakan untuk membuat program python ini dijelaskan pada gambar 4.1.

```
[1] import pandas as pd
import numpy as np
import re
import nltk
pd.set_option('display.max_columns', None)
```

Gambar 4.1

Gambar 4.1 menjelaskan *library* yang digunakan yaitu pandas, numpy, re dan nltk.

4.2. Import modul dtype dan dataset

Modul yang diimpor adalah modul *dtype* dari file *numpy.core.multiarray* ditunjukkan pada gambar 4.2

```
[2] from numpy.core.multiarray import dtype
skin = pd.read_excel("data_skincare_pria.xlsx")
skin.drop(['no'], axis=1, inplace=True)
skin.head()
```

Gambar 4.2

Gambar 4.2 menjelaskan modul *numpy.core.multiarray* menyediakan fungsi *dtype* di *numpy* digunakan untuk memeriksa tipe data (*dtype*) untuk array *numpy*.

4.3. Text Preprocessing

Tahap ini dilakukan text processing untuk membuat kalimat yang terdapat pada

atribut manfaat supaya bisa lebih terfokus untuk dilakukan pembobotan yang digunakan dalam proses similaritas nantinya. Text preprocessing yang dilakukan antara lain case folding, stopword, normalisasi, dan juga stemming.

4.3.1. Case Folding

Tahap case folding ditunjukkan pada gambar 4.3

```
[5] # Convert lowercase and remove numbers, punctuations, spaces, etc.
skin['clean_manfaat'] = skin['manfaat'].str.lower()
skin['clean_manfaat'] = skin['clean_manfaat'].apply(lambda x: re.sub('[^a-z]', '', x))
skin['clean_manfaat'] = skin['clean_manfaat'].apply(lambda x: re.sub('[^a-z]', '', x))
skin['clean_manfaat']

0      membuat wajah lebih halus
1      membantu merawat skin barrier
2      memudarkan bintik hitam
3      membantu mencerahkan kulit
4      melembabkan dan melembutkan kulit
5      membuat wajah lebih halus
6      membuat kulit berisi dan kencang
7      menenangkan kulit kemerahan dan iritasi
8      melembabkan dan menjaga elastisitas kulit wajah
9      meredakan kemerahan pada kulit wajah
10     membantu mengurangi ruam yang tidak merata da...
11     menghilangkan kotoran dan sebum berlebih
12     mencerahkan dan menghaluskan wajah
13     mencerahkan dan melembutkan kulit wajah
14     mengatasi polusi yang menempel pada wajah
Name: clean_manfaat, dtype: object
```

Gambar 4.3

Gambar 4.3 menjelaskan tentang tahap case folding yang dilakukan yaitu lowercase digunakan untuk pemrosesan teks yang melibatkan perbandingan teks tanpa memperhatikan huruf besar atau kecil, sehingga mengubah teks menjadi huruf kecil (*lowercase*)

4.3.2. Stopword

Proses stopwords ditunjukkan pada gambar 4.4

```
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords

# ----- get stopwords from NLTK stopwords -----
# get stopwords indonesia
list_stopwords = stopwords.words('indonesian')

# ----- manually add stopwords -----
# append additional stopwords
list_stopwords.extend(['ye', 'dg', 'rt', 'dgn', 'ny', 'd', 'klo',
                        'kelo', 'amp', 'bian', 'bikin', 'bilang',
                        'gak', 'ga', 'knn', 'ngg', 'mb', 'aduh',
                        'si', 'tau', 'tdk', 'tuh', 'utk', 'ya',
                        'jd', 'jgn', 'dh', 'aja', 'n', 't',
                        'ngg', 'hehe', 'pen', 'w', 'man', 'loh', 'rk',
                        'smp', 'yah'])

# ----- add stopwords from txt file -----
# read txt stopwords using pandas
txt_stopword = pd.read_csv('stopwords.txt', names= ['stopwords'], header = None)
# convert stopwords string to list & append additional stopwords
list_stopwords.extend(txt_stopword['stopwords'][0].split(' '))

# ----- convert list to dictionary -----
list_stopwords = set(list_stopwords)
```

Gambar 4.4

Gambar 4.4 menjelaskan tentang tahap stopwords yang digunakan dalam pemrosesan teks. Perintah `nltk.download('stopwords')` digunakan untuk mengunduh dataset 'stopwords' dari NLTK

4.3.3. Normalisasi dan Stemming

Proses normalisasi dan stemming ditunjukkan pada gambar 4.5.

```
import Sastrawi package
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import suifter

# create stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

# stemmed
def stemmed_upper(term):
    return stemmer.stem(term)

term_dict = {}

for document in skin['normalized_manfaat']:
    for term in document:
        if term not in term_dict:
            term_dict[term] = 1
        print(len(term_dict))
        print("-----")

for term in term_dict:
    term_dict[term] = stemmed_upper(term)
    print(term, " ", term_dict[term])
    print("-----")

# apply stemmed term to dataframe
def get_stemmed_term(document):
    return [term_dict[term] for term in document]

skin['stemmed_manfaat'] = skin['normalized_manfaat'].suifter.apply(get_stemmed_term)
skin['stemmed_manfaat']
```

Gambar 4.5

Gambar 4.5 menjelaskan modul sastrawi diperlukan untuk melakukan proses *stemming* ini untuk mengembalikan kata-kata yang sudah dinormalisasi dalam atribut manfaat menjadi bentuk bakunya.

4.3.4. Hasil Preprocessing

Pada tahap ini, setelah kalimat-kalimat pada kolom manfaat melalui proses case folding, stopwords, normalisasi, dan stemming, maka didapatkan hasil pra-pemrosesan seperti pada Tabel 4.1

Tabel 4.1 Hasil Preprocessing

Doc	Kode	Nama Produk	Manfaat	Setelah Preprocessing
Q	-	Untuk mencerahkan kulit wajah	Membantu mencerahkan kulit	bantu cerah kulit
D1	F0001	The Originote Cicamide Facial Cleanser	Membuat wajah lebih halus	wajah halus
D2	F0002	Kahf Skin Energizing and Brightening Face Wash	Membantu merawat skin barrier	bantu rawat skin barrier
D3	F0003	Nivea Man Personal Care Men Extra Bright	Memudarkan bintik hitam	pudar bintik hitam
D4	F0004	Scarlett Whitening Facial Wash	Membantu mencerahkan kulit	bantu cerah kulit
D5	F0005	Hada Labo Shirojyun Face Wash	Melembabkan dan melembutkan kulit	lembab lembut kulit
D6	F0006	Facial Wash MS Glow For Men	Membuat wajah lebih halus	wajah halus
D7	F0007	Bioaqua facial wash 24k gold	Membuat kulit berisi dan kencang	kulit isi kencang
D8	F0008	Ms Glow Facial Wash	Menenangkan kulit kemerahan dan iritasi	Tenang kulit merah iritasi
D9	F0009	Elvicto Deep Clean Face Wash	Melembabkan dan menjaga elastisitas kulit wajah	lembab jaga elastis kulit wajah
D10	F010	Himalaya Men Power Glow Licorice Face Wash	Meredakan kemerahan pada kuit wajah	reda merah kulit wajah
D11	F011	Vaseline Men	Membantu	bantu kurang

		Face Wash Healthy Bright	mengurangi warna yang tidak merata dan bintik hitam	warna rata bintik hitam
D12	F012	Something Low pH Gentle Jelly Cleanser	Menghilangkan kotoran dan sebum berlebih	hilang kotor sebum lebih
D13	F013	Whitelab Brightening Facial Wash	Mencerahkan dan menghaluskan wajah	cerah halus wajah
D14	F014	Facial Wash Brightening Benings	Mencerahkan dan melembabkan kulit wajah	cerah lembab kulit wajah
D15	F015	Ponds Men Pollution Out Face Wash	Mengatasi polusi yang menempel pada wajah	atas polusi tempel wajah

4.4. Pembobotan TF-IDF

Pembobotan dilakukan untuk manfaat produk yang berisikan *keyword*, dengan memberikan nilai 1 untuk setiap dokumen yang berisikan istilah tersebut yang bisa dilihat pada table 4.2

Tabel 4.2 Bobot Term

term	tf		
	bantu	cerah	kulit
Q	1	1	1
D1			
D2	1		
D3			
D4	1	1	1
D5			1
D6			1
D7			1
D8			1
D9			1
D10			1
D11	1		
D12			
D13		1	
D14		1	1
D15			

Nilai DF merupakan jumlah term dalam setiap dokumen. Hitung IDF menggunakan rumus $IDF = \log(n/DF)$. Hasil perhitungan IDF ditunjukkan pada Tabel 4.3

Tabel 4.3 Hasil perhitungan idf

term	DF	D/DF	IDF
bantu	4	3,75	0,5740
cerah	4	3,75	0,5740
kulit	9	1,67	0,2218

Tabel 4.4 Hasil Perhitungan Wdt

term	Wdt=TF.IDF		
	bantu	cerah	Kulit
Q	0,5740	0,5740	0,2218
D1	0,0000	0,0000	0,0000
D2	0,5740	0,0000	0,0000
D3	0,0000	0,0000	0,0000
D4	0,5740	0,5740	0,2218
D5	0,0000	0,0000	0,2218
D6	0,0000	0,0000	0,2218
D7	0,0000	0,0000	0,2218
D8	0,0000	0,0000	0,2218
D9	0,0000	0,0000	0,2218
D10	0,0000	0,0000	0,2218
D11	0,5740	0,0000	0,0000
D12	0,0000	0,0000	0,0000
D13	0,0000	0,5740	0,0000
D14	0,0000	0,5740	0,2218
D15	0,0000	0,0000	0,0000

4.5. Perhitungan cosine similarity

Hitung kemiripan antara kueri q dan masing-masing dokumen yang ada memakai rumus kemiripan kosinus. Hitung hasil kalkulasi perkalian skalar antara Q dengan 10 dokumen lainnya. Hasilnya lalu ditotalkan dan hasilnya ditunjukkan pada Tabel 4.5

Tabel 4.5 Perkalian Skalar

	WD*Wdi			
				Total
D1	0,0000	0,0000	0,0000	0,0000
D2	0,3295	0,0000	0,0000	0,3295
D3	0,0000	0,0000	0,0000	0,0000
D4	0,3295	0,3295	0,0492	0,7081
D5	0,0000	0,0000	0,0492	0,0492
D6	0,0000	0,0000	0,0492	0,0492
D7	0,0000	0,0000	0,0492	0,0492
D8	0,0000	0,0000	0,0492	0,0492
D9	0,0000	0,0000	0,0492	0,0492
D10	0,0000	0,0000	0,0492	0,0492
D11	0,3295	0,0000	0,0000	0,3295
D12	0,0000	0,0000	0,0000	0,0000
D13	0,0000	0,3295	0,0000	0,3295
D14	0,0000	0,3295	0,0492	0,3787
D15	0,0000	0,0000	0,0000	0,0000

Hitung panjang vektor masing-masing dokumen, termasuk Q, dengan mengakarkan jumlah wdt pada kolom total. Hasil kalkulasi panjang vektor ditunjukkan pada Tabel 4.6.

Tabel 4.6 Perhitungan Panjang Vektor

WD*Wdi				Total	Akar
Q	0,3295	0,3295	0,0492	0,7081	0,8415
D1	0,0000	0,0000	0,0000	0,0000	0,0000
D2	0,3295	0,0000	0,0000	0,3295	0,5740
D3	0,0000	0,0000	0,0000	0,0000	0,0000
D4	0,3295	0,3295	0,0492	0,7081	0,8415
D5	0,0000	0,0000	0,0492	0,0492	0,2218
D6	0,0000	0,0000	0,0492	0,0492	0,2218
D7	0,0000	0,0000	0,0492	0,0492	0,2218
D8	0,0000	0,0000	0,0492	0,0492	0,2218
D9	0,0000	0,0000	0,0492	0,0492	0,2218
D10	0,0000	0,0000	0,0492	0,0492	0,2218
D11	0,3295	0,0000	0,0000	0,3295	0,5740
D12	0,0000	0,0000	0,0000	0,0000	0,0000
D13	0,0000	0,3295	0,0000	0,3295	0,5740
D14	0,0000	0,3295	0,0492	0,3787	0,6154
D15	0,0000	0,0000	0,0000	0,0000	0,0000

Terapkan rumus cosine similarity

$$\begin{aligned} \cos(\theta) &= \frac{x \cdot y}{\|x\| \|y\|} \\ &= \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}} \end{aligned}$$

Dengan melakukan kalkulasi kemiripan antara dokumen D1-D15. Contoh kalkulasi hanya diberikan untuk dokumen D2, D4, D5, D14, karena dokumen D1, D3, D6, D7, D8, D9, D10, D11, D12, D15 mempunyai nilai kalkulasi yang sama dengan dokumen-dokumen pada contoh di bawah ini.

$$\text{Similarity (Q,D2)} = 0,3295 / (0,8415 * 0,5740)$$

$$= 0,3295 / 0,4830$$
$$= 0,292168$$

$$\text{Similarity (Q,D4)} = 0,7081 / (0,8415 * 0,8415)$$

$$= 0,7081 / 0,7081$$
$$= 1$$

$$\text{Similarity (Q,D5)} = 0,0492 / (0,8415 * 0,2218)$$

$$= 0,0492 / 0,1866$$
$$= 0,263679$$

Similarity (Q,D13) = 0,3295 / (0,8415 * 0,5740)

$$= 0,0492 / 0,1866$$
$$= 0,393025$$

$$\begin{aligned} \text{Similarity (Q,D14)} &= 0,3787 / (0,8415 \\ &\quad * 0,6154) \\ &= 0,3787 / 0,5178 \\ &= 0,561108 \end{aligned}$$

4.6. Pengubahan teks menjadi representasi numerik

Tahap perubahan teks ini ditunjukkan pada gambar 4.6

```
[19] from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
tfidf = TfidfVectorizer()
features = tfidf.fit_transform(skin2['manfaat_join'])
```

Gambar 4.6

Gambar 4.6 menjelaskan `sklearn.feature_extraction.text` untuk mengambil data teks dari kolom manfaat dan mengubahnya ke dalam bentuk angka. `TfidfVectorizer` untuk menghitung bobot nilai TF IDF pada setiap data di kolom manfaat.

4.7. Pembuatan matriks *cosine similarity*.

Tahap pembuatan matriks *cosine similarity* ini menggunakan pustaka skit-learn yaitu *metrics.pairwise* dan *cosine similarity*

[illegible]

Gambar 4.7

Gambar 4.7 menjelaskan `sklearn.metrics.pairwise` digunakan untuk menghitung distansi atau tingkat kemiripan tiap pasang data dalam kolom manfaat. Modul `cosine_similarity` digunakan untuk mengukur kemiripan kosinus antara pasangan vektor di kolom manfaat pada dataset 'data skincare pria'.

4.8. Mencari rekomendasi produk *face wash* pria dengan kemiripan produk Scarlett Whitening Facial Wash

Pengguna ingin diberikan rekomendasi *skin care facewash* pria yang dilihat berdasarkan manfaatnya dengan kemiripan produk sebelumnya yang dipakai oleh pengguna yaitu Scarlett Whitening Facial Wash yang ditunjukkan pada gambar 4.9.

```
[24] index = pd.Series(skin2['nama_produk'])
      index.head()

0      The Originote Cicamide Facial Cleanser
1      Kahf Skin Energizing and Brightening Face Wash
2      Nivea Man Personal Care Men Extra Bright
3      Scarlett Whitening Facial Wash
4      Hada Labo Shirojyun Face Wash
      Name: nama_produk, dtype: object

[25] def recommend_skincare(title):
      skincare = []
      idx = index[index == title].index[0]
      # print(idx)
      score = pd.Series(cosine_sim[idx]).sort_values(ascending=False)
      top10 = list(score.iloc[1:11].index)
      # print(top10)

      for i in top10:
          skincare.append(skin2['nama_produk'][i])
      return skincare
```

Gambar 4.8

4.9. Nilai similaritas yang dihasilkan dari perekomendasi skincare face wash dengan kemiripan produk Scarlett Whitening Facial Wash.

Nilai similaritas yang dihasilkan ini berdasarkan perangkingan top 10 produk yang paling mirip nilai manfaatnya dari produk Scarlet Whitening Facial Wash yang ditunjukkan pada gambar 5.0.

```
[28] pd.Series(cosine_sim[3]).sort_values(ascending=False)

3      1.000000
13     0.561108
12     0.393025
1      0.253957
10     0.213203
4      0.202086
6      0.184124
7      0.160223
8      0.156518
0      0.000000
2      0.000000
5      0.000000
9      0.000000
11     0.000000
14     0.000000
      dtype: float64
```

Gambar 5.0

SIMPULAN

Berdasarkan hasil penelitian ini, dapat disimpulkan sebagai berikut:

1. Penelitian ini bisa dipakai untuk mencari rekomendasi *skincare face wash* pria yang mirip dengan produk yang biasanya dipakai pengguna sebelumnya menggunakan metode

content-based filtering dengan algoritma TF IDF dan *cosine similarity*.

2. Data *skincare face wash* yang digunakan berjumlah 15 produk dari berbagai macam merek yang berbeda. Data diambil dari official store di *e-commerce* shopee.
3. Nilai tertinggi dari perekomendasi skincare face wash pria yang mirip dalam manfaat dengan produk 'Scarlett Whitening Facial Wash' yaitu produk 'Facial Wash Brightening Benings' dengan nilai similaritas 0,5611

DAFTAR PUSTAKA

- Alkaff, M., Khatimi, H., & Eriadi, A. (2020). Sistem Rekomendasi Buku pada Perpustakaan Daerah Provinsi Kalimantan Selatan Menggunakan Metode Content-Based Filtering. *MATRIK: Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, 20(1), 193–202.
- Astungkoro, F. P., Kartika Dewi, R., & Brata, K. C. (2019). *Sistem Rekomendasi Lapangan Futsal berbasis Android dengan Metode TOPSIS* (Vol. 3, Issue 9).
- Fajriansyah, M., Adikara, P. P., & Widodo, A. W. (2021). Sistem Rekomendasi Film Menggunakan Content Based Filtering. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 5(6), 2188-2199.
- Huda, A. A., Fajarudin, R., & Hadinegoro, A. (2022). Sistem Rekomendasi Content-based Filtering Menggunakan TF-IDF Vector Similarity Untuk Rekomendasi Artikel Berita. *Building of Informatics, Technology and Science (BITS)*, 4(3).
- Jaja, V. L., Susanto, B., & Sasongko, L. R. (2020). Penerapan Metode Item-Based Collaborative Filtering Untuk Sistem Rekomendasi Data MovieLens. *d'CARTESIAN: Jurnal Matematika dan Aplikasi*, 78-83.

- Larasati, F. B. A., & Februariyanti, H. (2021). Sistem Rekomendasi Product Emina Cosmetics Dengan Menggunakan Metode Content-Based Filtering. *Jurnal Manajemen Informatika Dan Sistem Informasi*, 4(1), 45-54.
- Masdalena, A., Dalimunthe, R. A., & Saputra, E. (2022). Implementasi Metode Analytical Hierarchy Process Pada Sistem Pendukung Keputusan Rekomendasi Ikan Budidaya Berbasis Web. *Building of Informatics, Technology and Science (BITS)*, 4(2).
- Muliawan, A., Badriyah, T., & Syarif, I. (2022). Membangun Sistem Rekomendasi Hotel dengan Content Based Filtering Menggunakan K-Nearest Neighbor dan Haversine Formula. *Technomedia Journal*, 7(2 October), 231-247.
- Putri, N. I., Herdiana, Y., & Munawar, Z. (2021). Sistem Rekomendasi Hibrid Pemilihan Mobil Berdasarkan Profil Pengguna dan Profil Barang. *TEMATIK*, 8(1), 56-68.
- Ratna Sari, K., Suharso, W., & Azhar, Y. (2020). Pembuatan Sistem Rekomendasi Film dengan Menggunakan Metode Item Based Collaborative Filtering pada Apache Mahout. *REPOSITOR*, 2(6), 767–774.
- Sipayung, E. M., Fiarni, C. F., & Sutopo, S. (2021). Sistem Rekomendasi Tempat Kost di Sekitar Kampus ITHB Menggunakan Metode Analytical Hierarchy Process (AHP). *Jurnal Nasional Teknologi Dan Sistem Informasi*, 7(2), 52–60.
- Syah, R. D. (2020). Performa Algoritma User K-Nearest Neighbors pada Sistem Rekomendasi di Tokopedia. *Jurnal Informatika Universitas Pamulang*, 5(3), 302-306.
- Yoshua, I., Bunyamin, H., & Si, S. (2021). *Pengimplementasian Sistem*

Rekomendasi Musik Dengan Metode Collaborative Filtering (Vol. 3)