

# Analisis Ulasan pada Restoran Bebek Ngarasan dengan Penerapan Latent Semantic Analysis

Muhammad Aufa Farabi - 13523023  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
[iniabi838@gmail.com](mailto:iniabi838@gmail.com), [13523023@std.stei.itb.ac.id](mailto:13523023@std.stei.itb.ac.id)

**Abstrak**— Dengan kemajuan teknologi, kemudahan dalam menyampaikan informasi atau pesan pada dunia maya memudahkan banyak konsumen membuat ulasan pada suatu tempat makanan, tidak terkecuali pada Restoran Bebek Ngarasan sebagai salah satu restoran bebek terenak di Bandung. Ulasan-ulasan yang diberikan pada Restoran Bebek tentunya sangat berpengaruh pada pengembangan restoran atau perbaikan kualitas produk. Penerapan dari *Singular Value Decomposition* dapat membantu untuk melakukan analisis tersebut, yakni dalam bentuk implementasi *Latent Semantic Analysis*. Dengan *Latent Semantic Analysis*, dapat diketahui topik ulasan yang umum diulas oleh konsumen sehingga dapat membantu pemilik Restoran Bebek Ngarasan dalam mengembangkan restorannya.

**Keywords**—Ulasan, Restoran Bebek Ngarasan, Singular Value Decomposition, Latent Semantic Analysis

## I. PENDAHULUAN

Seiring dengan kemajuan teknologi informasi khususnya untuk dunia kuliner, penyampaian suatu informasi dan komunikasi semakin mudah dan cepat sehingga semakin banyak konsumen yang dapat membagikan hasil pengalaman atau ulasan mereka terhadap suatu restoran atau produk makanan. Dalam industri kuliner, ulasan pelanggan memiliki peran yang sangat penting dalam mengetahui preferensi pelanggan dan untuk memperbaiki kualitas pelayanan yang diberikan oleh restoran.



**Gambar 1.** Restoran Bebek Ngarasan

Sumber : <https://maps.app.goo.gl/3fqTR2uE2bEongi6A>

Sebagai salah satu restoran bebek terpopuler di Bandung yang terletak di Jalan Tubagus Ismail Raya dengan hidangan bebek ovennya yang lezat, Bebek Ngarasan tentunya mendapatkan banyak ulasan terutama pada Google Maps mengenai produk dan layanan yang diberikan restoran. Dengan banyaknya ulasan bervariasi yang diberikan, tentunya diperlukan analisis yang baik agar pemilik restoran dapat mengambil informasi inti dari ulasan-ulasan tersebut untuk perbaikan kualitas restoran dan memahami pasar konsumen. Salah satu teknik efektif yang dapat digunakan untuk menemukan kata-kata kunci dari ulasan-ulasan tersebut adalah dengan menggunakan metode *Latent Semantic Analysis*. Teknik ini merupakan penerapan dari salah satu metode dekomposisi matriks yakni *Singular Value Decomposition (SVD)* untuk mereduksi dimensi informasi dan mengungkap pola tersembunyi dari data-data teks dalam hal ini teks setiap ulasan,

Penerapan *Latent Semantic Analysis* dari *SVD* ini diharapkan membantu bagi usaha Restoran Bebek Ngarasan dalam mengetahui topik ulasan apa yang umum diulas oleh konsumen untuk memperbaiki pelayanan dan mengetahui persepsi dari konsumen.

## II. DASAR TEORI

### A. Matriks

Matriks didefinisikan sebagai susunan angka atau symbol yang diatur sedemikian dalam baris dan kolom dalam bentuk persegi panjang. Matriks direpresentasikan dengan  $[M]_{m \times n}$  dengan  $m$  adalah jumlah baris pada matriks dan  $n$  adalah jumlah kolom dari matriks. Bilangan-bilangan atau symbol yang terdapat pada matriks disebut dengan elemen matriks. Apabila jumlah kolom  $m$  sama dengan jumlah baris  $n$  maka matriks tersebut dinamakan matriks persegi (*square matrix*).

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

**Gambar 2.** Bentuk dari Matriks

Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-01-Review-Matriks-2023.pdf>

Suatu matriks dapat dilakukan operasi-operasi sebagai berikut:

1. Penjumlahan dan Pengurangan Matriks,

Matriks C merupakan hasil penjumlahan atau pengurangan dari matriks A dan matriks B dengan cara  $C_{ij} = A_{ij} + B_{ij}$ , dengan  $i = 1, 2, \dots$ , jumlah baris (m) dan  $j = 1, 2, \dots$ , jumlah kolom (n).

2. Perkalian Matriks dengan Matriks

Matriks C merupakan hasil perkalian dari matriks A dan matriks B dengan cara  $C_{ij} = A_{i1}B_{1j} + A_{i2}B_{2j} + \dots + A_{in}B_{nj}$ . Operasi ini akan berlaku jika jumlah kolom pada matriks A sama dengan jumlah baris matriks B.

3. Perkalian Matriks dengan skalar

Matriks A dapat dikalikan dengan nilai skalar dengan cara  $cA = [ca_{ij}]$ .

4. Transpose Matriks

Transpose matriks adalah matriks dengan posisi elemen pada baris dan kolomnya dibalik seperti berikut:

$$b_{ji} = a_{ij}$$

dengan  $i = 1, 2, \dots, m$  dan  $j = 1, 2, \dots, n$ .

Matriks Identitas (I) adalah suatu matriks persegi yang elemen pada posisi diagonal utamanya bernilai 1 dan bernilai 0 pada posisi lain.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Gambar 3. Bentuk Matriks Identitas 2x2

Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-01-Review-Matriks-2023.pdf>

Hasil perkalian suatu matriks identitas dengan sembarang matriks akan menghasilkan matriks sembarang itu sendiri,  $AI = IA = A$ .

Determinan matriks, disimbolkan dengan  $\det(A)$ , adalah suatu nilai skalar yang didapat dari matriks persegi (*square matrix*) sebagai representasi ciri dari suatu matriks. Terdapat beberapa cara untuk mendapatkan nilai determinan matriks, yakni dengan sarrus rule (untuk orde kecil), ekspansi kofaktor, dan reduksi baris dengan Operasi Baris Elementer (OBE). Matriks disebut sebagai matriks singular (*singular matrix*) apabila determinannya bernilai nol.

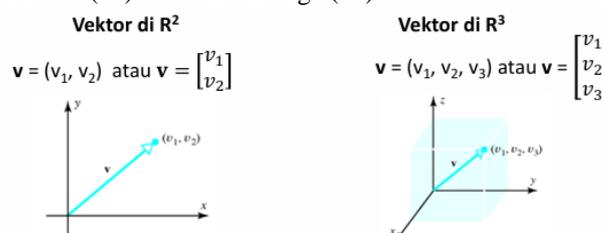
Suatu matriks juga dapat memiliki balikan matriks atau *inverse*, yakni matriks yang jika dikalikan dengan matriks itu sendiri akan menghasilkan matriks Identitas. Balikan matriks A disimbolkan dengan  $A^{-1}$ .

$$A A^{-1} = A^{-1}A = I$$

Ada dua cara utama untuk mendapatkan matriks balikan, yakni dengan metode kofaktor matriks dan menggunakan metode Eliminasi Gauss-Jordan. Tidak semua matriks memiliki balikan, matriks singular yang memiliki determinan nol tidak memiliki balikan (*not invertible*).

B. Vektor

Vektor didefinisikan sebagai suatu kuantitas fisik yang memiliki besar dan arah. Vektor dinotasikan dengan huruf kecil yang dicetak tebal (**v**) atau huruf yang memakai tanda panah di atasnya ( $\vec{v}$ ). Ruang vektor atau juga disebut ruang eucledian adalah ruang tempat vektor didefinisikan. Vektor dapat digambarkan pada ruang Eucledian apabila memiliki dimensi dua ( $R^2$ ) atau dimensi tiga ( $R^3$ ).



Gambar 4. Ruang Vektor dua dimensi dan tiga dimensi

Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-01-Review-Matriks-2023.pdf>

Vektor ditulis sebagai  $\mathbf{v} = (v_1, v_2)$ ,  $\mathbf{v} = (v_1, v_2, v_3)$ , atau  $\mathbf{v} = (v_1, v_2, \dots, v_n)$  tergantung dari jumlah dimensinya dan semuanya berawal dari titik asal O (komponen bilangannya nol semua).

Vektor memiliki sifat-sifat operasi aritmatika dasar sebagai berikut:

1. Penjumlahan dan Pengurangan dua vektor

Jika  $\mathbf{v} = (v_1, v_2, \dots, v_n)$  dan  $\mathbf{w} = (w_1, w_2, \dots, w_n)$  maka  $\mathbf{v} \pm \mathbf{w} = (v_1 \pm w_1, v_2 \pm w_2, \dots, v_n \pm w_n)$

2. Perkalian vektor dengan scalar

Perkalian vektor dengan nilai scalar akan menghasilkan vektor yang panjangnya  $|k|$  kali panjang suatu vektor

Pada konsep vektor, Panjang sebuah vektor **v** dinamakan dengan norma (*norm*) **v** atau disimbolkan sebagai  $\|\mathbf{v}\|$ . Panjang vektor merupakan representasi skala dari suatu vektor. Rumus untuk menghitung Norma vektor adalah sebagai berikut:

$$\|\vec{v}\| = \sqrt{v_1^2 + v_2^2 + v_3^2 + \dots + v_n^2}$$

Vektor yang panjangnya bernilai satu disebut dengan vektor satuan (*unit vector*) yang dilambangkan dengan  $\hat{v}$ . Rumus untuk menghitung vektor satuan dari vektor atau disebut normalisasi vektor adalah seperti berikut:

$$\hat{v} = \frac{\vec{v}}{|\vec{v}|}$$

dengan vektor  $\mathbf{v}$  tidak nol.

Terdapat dua jenis perkalian pada vektor, yakni perkalian titik (*dot product*) dan perkalian silang (*cross product*). Perkalian titik antara dua vektor akan menghasilkan suatu nilai skalar, sedangkan perkalian silang akan menghasilkan vektor baru dengan besar dan arahnya. Pada makalah ini, perkalian silang tidak akan dibahas lebih lanjut karena tidak berkorelasi dengan perhitungan yang digunakan dalam proses *Latent Semantic Analysis*.

Perkalian titik pada misalnya vektor  $\mathbf{u}$  dan vektor  $\mathbf{v}$  dapat dilakukan dengan dua cara, yakni dengan mengalikan panjang  $\mathbf{u}$  dan panjang  $\mathbf{v}$  serta cosinus dari sudut antara  $\mathbf{u}$  dan  $\mathbf{v}$

$$\mathbf{u} \cdot \mathbf{v} = |\mathbf{u}||\mathbf{v}|\cos(\theta)$$

atau dengan mengalikan setiap komponen dari dua vektor yang berkoresponden

$$\mathbf{u} \cdot \mathbf{v} = u_x v_x + u_y v_y$$

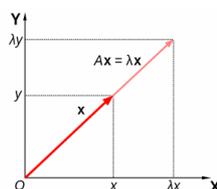
### C. Nilai Eigen dan Vektor Eigen

Eigen berasal dari bahasa Jerman yang berarti asli atau karakteristik. Nilai Eigen ( $\lambda$ ) menyatakan suatu nilai karakteristik dari matriks berukuran  $n \times n$ . Vektor Eigen sendiri merupakan matriks kolom yang jika dikalikan dengan nilai eigen akan menghasilkan kelipatan dari vektor eigen. Hubungan antara vektor eigen dan nilai eigen dapat dijelaskan dalam bentuk ini

$$A\mathbf{x} = \lambda\mathbf{x}$$

dengan  $\lambda$  sebagai nilai eigen dari  $A$  dan  $\mathbf{x}$  adalah vektor eigen yang berkoresponden dengan  $\lambda$ .

Operasi  $A\mathbf{x} = \lambda\mathbf{x}$  juga dapat disimpulkan sebagai operasi untuk memanjangkan atau menyusutkan  $\mathbf{x}$  tergantung dari faktor  $\lambda$  dengan arah dari  $\lambda$ .



Gambar 5. Grafik Vektor Eigen

Sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-21-Singular-value-decomposition-Bagian1-2023.pdf>

Nilai-nilai eigen dari matriks  $A$  dapat dicari dengan penurunan rumus berikut:

$$\begin{aligned} A\mathbf{x} &= \lambda\mathbf{x} \\ A\mathbf{x} &= \lambda I\mathbf{x} \\ I A\mathbf{x} &= \lambda I\mathbf{x} \\ \det(\lambda I - A) &= 0 \end{aligned}$$

dengan  $I$  adalah matriks identitas dan  $\lambda$  adalah Solusi nilai-nilai eigen yang ingin dicari.

Vektor eigen dapat ditemukan setelah mendapatkan nilai eigen ( $\lambda$ ) dengan rumus berikut:

$$(\lambda I - A)\mathbf{x} = 0$$

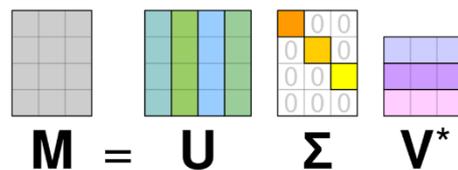
Dekomposisi matriks artinya memfaktorkan suatu matriks menjadi hasil perkalian dari sejumlah matriks lain. Hal ini dapat disimbolkan seperti berikut:

$$A = P_1 \times P_2 \times \dots \times P_k$$

$P_1, P_2,$  sampai  $P_k$  adalah matriks-matriks dekomposisi dari matriks  $A$ .

Terdapat beberapa metode untuk mendekomposisi matriks, tetapi pada makalah ini yang dipakai adalah metode dekomposisi nilai singular (*Singular Value Decomposition*).

### D. Singular Value Decomposition



Gambar 6. Dekomposisi SVD

Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-21-Singular-value-decomposition-Bagian1-2023.pdf>

*Singular Value Decomposition (SVD)* adalah salah satu metode dekomposisi matriks yang mendekomposisi matriks berukuran  $m \times n$  menjadi tiga matriks, yakni matriks  $U, \Sigma,$  dan  $V$  sedemikian sehingga berlaku

$$A = U\Sigma V^T$$

dengan  $U$  sebagai matriks orthogonal  $m \times m,$   $V$  adalah matriks ortogonal  $n \times n,$  dan  $\Sigma$  adalah matriks  $m \times n$  yang elemen pada diagonal utamanya merupakan nilai singular  $A,$  sedangkan sisanya bernilai 0.

Matriks orthogonal yang menjadi  $U$  dan  $V$  pada *SVD* adalah suatu matriks persegi yang kolom-kolomnya adalah vektor yang saling orthogonal satu sama lain atau bisa dikatakan hasil perkalian antar vektornya adalah 0.

Nilai singular ( $\sigma$ ) adalah akar dari nilai-nilai eigen pada suatu matriks. Nilai-nilai singular memiliki perbandingan yang sebanding dengan nilai-nilai eigen sehingga jika  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$  maka juga berlaku  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$

Terdapat dua cara untuk melakukan dekomposisi *SVD*

yakni seperti berikut:

- **Cara 1**

1. Buat matriks  $A^T A$  lalu dicari nilai dan vektor eigennya. Vektor eigen yang didapat akan digunakan untuk membentuk matriks  $V$  (sebagai vektor singular kanan).
2. Normalisasi setiap vektor eigen yang didapatkan pada langkah sebelumnya dengan membagi dengan panjangnya. Hasil dari ini adalah vektor
3. Gabungkan semua vektor  $V$  menjadi matriks  $V$  yang disusun menurun berdasarkan nilai eigennya. Transpose matriks tersebut untuk menjadi matriks  $V^T$ .
4. Akarkan nilai eigen yang diperoleh sebelumnya untuk mendapatkan  $\sigma_i$ . Kemudian, gabungkan secara diagonal untuk membentuk matriks  $\Sigma$ .
5. Tentukan vektor singular kiri ( $u_i$ ) dengan persamaan,

$$u_i = \frac{Av_i}{|Av_i|} = \frac{1}{\sigma_i} Av_i$$

6. Normalisasi vektor  $u_i$  lalu gabungkan untuk membentuk matriks  $U$  sehingga terbentuk

$$A = U\Sigma V^T$$

- **Cara 2**

1. Lakukan langkah 1 hingga 4 dari cara 1.
2. Buat matriks  $AA^T$  lalu dicari nilai dan vektor eigennya. Vektor tersebut akan digunakan untuk membentuk matriks  $U$  (dengan vektor singular kiri  $u_i$ ).
3. Normalisasi vektor  $u_i$  seperti biasanya lalu gabungkan untuk menjadi matriks  $U$  yang terurut menurut nilai eigennya sehingga terbentuk  $A = U\Sigma V^T$ .

#### E. Singular Value Decomposition Tereduksi

Baris-baris dan kolom yang seluruhnya bernilai nol pada matriks  $\Sigma$  yang didapatkan dari dekomposisi  $SVD$  dapat dihilangkan. Hal ini disebut dengan  $SVD$  Tereduksi. Di sini, matriks  $A$  didekomposisikan seperti berikut:

$$A = U_1 \Sigma_1 V_1^T$$

dengan  $U_1$  berukuran  $m \times k$ ,  $\Sigma_1$  berukuran  $k \times k$ , dan  $V_1^T$  berukuran  $k \times n$ ,  $k$  sendiri adalah  $\text{rank}(A)$ .  $\Sigma_1$  memiliki balikan karena elemen diagonal utamanya bernilai positif. Hasil perkalian dari setiap elemen dari  $U_1$ ,  $\Sigma_1$ ,  $V_1^T$  dinamakan ekspansi  $SVD$  dari  $A$ .

#### F. Latent Semantic Analysis

*Latent Semantic Analysis (LSA)* adalah suatu metode pemrosesan bahasa alami (*Natural Language Process* atau

*NLP*) yang mengidentifikasi hubungan tersembunyi (laten) antarkata dalam dokumen teks dengan menganalisis kemunculan bersama (*co-occurrence*) kata-kata. *LSA* mampu menangani situasi. *LSA* sangat berkaitan dengan *Singular Value Decomposition (SVD)* karena *SVD* digunakan pada *LSA* untuk mereduksi dimensi pada matriks dokumen yang diprosesnya. Selanjutnya, *LSA* juga

Proses dalam *LSA* yang dilakukan pada makalah ini terdiri dari pemrosesan awal teks (*text preprocessing*), inialisasi matrix *Term Frequency-Inverse Document frequency (TF-IDF)*, proses *SVD* tereduksi, dan Interpretasi hasil dari topik. Pada Langkah awal, *text preprocessing* berupaya menormalisasi dan memfilterisasi kata pada teks agar informasi yang ada pada dokumen menjadi lebih padat dan ringkas. Hal ini dilakukan dengan proses berikut:

##### 1. Stopwords Removal

*Stopwords* adalah kata-kata yang sangat umum digunakan tetapi tidak memiliki berpengaruh pada makna kalimat. Umumnya, kata depan dan kata hubung termasuk dalam *stopwords*, contohnya adalah kata “yang, di, ke, dan pada”. Pada proses ini, teks akan difilter untuk mengeliminasi kata-kata yang tergolong dalam *stopwords* lalu seluruh huruf pada teks yang sudah difilter akan diubah menjadi huruf kecil.

##### 2. Lemmatization

Pada proses ini, kata-kata dalam teks akan dinormalisasi menjadi kata dasar pembentuk kata utuhnya dengan menghilangkan imbuhan yang melekat pada kata, contohnya “menyukai” diubah menjadi “suka”.

Selanjutnya, dilakukan inialisasi sebuah matriks yang berisi nilai relevansi setiap kata pada tiap teks dengan metode *Term Frequency-Inverse Document Frequency (TF-IDF)*. metode ini didefinisikan sebagai suatu metode statistic untuk mengevaluasi seberapa relevan suatu kata dalam suatu teks relatif terhadap kumpulan teks atau dokumen. *Term Frequency (TF)* merepresentasikan jumlah kemunculan kata tertentu dalam teks. *Term Frequency* dapat dihitung dengan menghitung bobot suatu kata pada teks dengan rumus berikut:

$$TF(t, d) = \frac{\text{Jumlah Kemunculan Kata } t \text{ dalam teks } d}{\text{Jumlah Kata dalam teks } d}$$

*Inverse Document Frequency (IDF)* mengukur seberapa penting suatu kata dalam seluruh Kumpulan teks. Kata yang sering muncul pada banyak teks dianggap kurang relevan dalam hal ini.

$$IDF(t) = \log\left(\frac{N}{DF(t)}\right)$$

Dengan  $N$  adalah jumlah total teks dan  $DF(t)$  adalah

jumlah teks yang mengandung kata t.

$TF-IDF$  dihitung dengan mengalikan hasil  $TF$  dengan  $IDF$  seperti berikut:

$$TF-IDF(t, d) = TF(t, d) \times IDF(t)$$

Pada matriks  $TF-IDF$ ,  $t$  yakni kata akan menjadi baris dan  $d$  yakni dokumen atau teks menjadi kolom dari matriks.

Tahap selanjutnya pada  $LSA$  adalah melakukan reduksi dimensi menggunakan *Singular Value Decomposition (SVD)* pada  $TF-IDF$  matrix.  $SVD$  dilakukan untuk mendekomposisi  $TF-IDF$  matrix menjadi tiga matriks, yakni

Matriks  $U$ : menggambarkan hubungan antara dokumen dan topik laten

Matriks  $\Sigma$ : nilai yang menunjukkan signifikansi topik

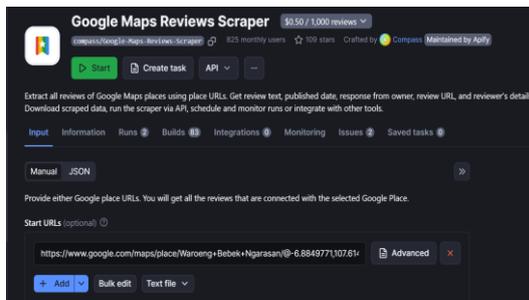
Matriks  $V^T$ : menggambarkan hubungan antara kata dan topik laten.

Setelah dilakukan proses  $SVD$ , Langkah selanjutnya dapat berupa pencarian *cosine similiarity* untuk menghitung kemiripan antar teks berdasarkan topik laten atau langsung mencocokkan hasil dari matriks dekomposisi  $V^T$  dengan kata pada setiap teks untuk mengetahui kata-kata yang memiliki bobot tertinggi dari setiap topik (kata dominan) yang menjadi tujuan akhir dari proses  $LSA$  dalam makalah ini.

### III. METODOLOGI

#### A. Pengambilan Data Ulasan

Data ulasan-ulasan pada Restoran Bebek Ngarasan yang dianalisis pada makalah ini didapatkan dengan melakukan *web scraping* yang memakai Google Maps Reviews Scraper yang tersedia pada website Apify. Proses ini dilakukan dengan menggunakan Apify Actor yang dikonfigurasi untuk mengekstrak informasi ulasan dari halaman Google maps yang berisi informasi terkait Restoran Bebek Ngarasan. Dengan memasukkan URL restoran Bebek Ngarasan yang didapatkan dari Google Maps ke dalam query, Apify Actor akan secara otomatis mengakses halaman Google Maps tersebut dan mengumpulkan semua data yang berisi semua ulasan yang diberikan pada Restoran Bebek Ngarasan.



**Gambar 7.** Tampilan Google Maps Scraper pada Apify  
Sumber : Hak Cipta Penulis

Data *scraping* yang dikumpulkan disimpan dalam file  $.Xlsx$  yang selanjutnya akan dianalisis lebih lanjut untuk mencari informasi mengenai topik atau komentar ulasan paling relevan dengan *Latent Semantic Analysis (LSA)*.

#	BX	BBJ	CA	CB	CC
1	stars	5	5	5	5
2	stars	5	5	5	5
3	5	V.JI.false	Pelayanan sangat OkeeeeeeeAyam/Bebeknya en	Service is very okayThe chicken/duck are all de	textTranslated
4	5	V.JI.false			
5	1	V.JI.false	Mknan enak.. Tp pesen via gopod lamaaaaa...	The food was delicious... But ordering via GoPc	
6	1	V.JI.false	Uta Hh		
7	5	V.JI.false	Tempatnya kecil, tapi ambience nya enak banget!	The place is small, but the ambience is really ni	
8	5	V.JI.false			
9	1	V.JI.false	Pelayanannya sangat2 judes,makanan enak,di	The service was very grumpy, the food was del	
10	1	V.JI.false	Pelayanan lelet kaya siput lagi mencret,lebih m	Service is slow like a snail with diarrhea, giving	
11	5	V.JI.false			
12	5	V.JI.false			
13	5	V.JI.false	Rumah makan bebek yg sudah lama dan masih	A duck restaurant that has been around for a l	
14	5	V.JI.false			
15	5	V.JI.false			
16	5	V.JI.false			
17	5	V.JI.false			
18	1	V.JI.false	Lama banget nunggunyaKaryawannya leha leha!	It's been a really long waitThe employees are r	
19	5	V.JI.false			
20	5	V.JI.false			
21	5	V.JI.false	Makanan oke. sambel kremes nya cukup oedes!	The food is okay. the Kremes sauce is ouite so	

**Gambar 8.** Dataset Ulasan pada Restoran Bebek Ngarasan

Sumber : Hak Cipta Penulis

#### B. Pembacaan Data Ulasan dan Preprocessing Teks

Bagian pertama dari analisis ulasan pada restoran Bebek Ngarasan adalah melakukan pembacaan pada data informasi ulasan yang sudah diekstrak pada proses sebelumnya lalu dilakukan pemrosesan awal data ulasan untuk menghasilkan data ulasan yang hanya mengandung teks ulasan yang memiliki informasi penting saja.

```
import pandas as pd
import numpy as np
import spacy
from sklearn.feature_extraction.text import TfidfVectorizer

# Pembacaan data ulasan
path = "Ulasan_Bebek_Ngarasan.xlsx"
dataUlasan = pd.read_excel(path)

# Preprocessing teks-teks ulasan
model = spacy.load("en_core_web_sm")

def textPreprocessing(text):
    if pd.isna(text):
        return ""
    processedText = model(text.lower())
    words = [token.lemma_ for token in processedText if not token.is_stop and not token.is_punct and len(token.text) > 3]
    return " ".join(words)

processedDataUlasan = dataUlasan['textTranslated'].dropna().apply(textPreprocessing)
```

**Gambar 9.** Kode proses pembacaan dan *preprocessing* dataset ulasan

Sumber : Hak Cipta Penulis

Pada bagian ini, dataset yang masih dalam berada *file excel (.xlsx)* dibaca lalu diubah menjadi sebuah dataframe pada variabel  $dataUlasan$  yang menyimpan seluruh informasi data ulasan.

Untuk preprocessing teks dari setiap ulasan, digunakan library *Natural Language Processing (NLP)* Spacy untuk melakukan pemrosesan bahasa alami pada kata-kata dalam data ulasan, yakni dengan tokenisasi (memecah teks menjadi unit kata-kata), lematisasi atau stemming (mengubah kata ke bentuk dasarnya), dan menghapus stopwords atau tanda baca dari suatu kata. Hal ini dilakukan dengan memuat model dari Spacy lalu dilakukan pemrosesan tokenisasi, lematisasi, penghapusan stopwords pada setiap kata dalam kolom  $textTranslated$ , yakni berisi kumpulan ulasan yang sudah ditranslasikan ke dalam bahasa Inggris, pada dataframe  $dataUlasan$  dengan

mengaplikasikan fungsi `textPreprocessing` lalu memasukkan hasilnya ke dalam variable `processedDataUlasan`.

### C. Pembentukan Matriks TF-IDF

Pada bagian ini, dilakukan pembentukan matriks *Term Frequency-Inverse Document Frequency (TF-IDF)* dari data ulasan yang sudah difilterisasi pada bagian preprocessing teks.

```
# Pembentukan matriks TF-IDF
vectorizer = TfidfVectorizer(
    max_features=400,
    ngram_range=(1, 2),
    max_df=0.75,
    min_df=0.025
)
matrixTFIDF = vectorizer.fit_transform(processedDataUlasan).toarray()
```

**Gambar 10.** Kode proses pembentukan matriks TF-IDF dari data ulasan yang telah diproses  
Sumber : Hak Cipta Penulis

Teks yang terdapat pada data ulasan diubah menjadi representasi numerik dalam bentuk matriks *TF-IDF* dengan objek vectorizer yang diinisialisasi dari kelas `TfidfVectorizer` yang berasal dari library `sklearn.feature_extraction.text`. Proses yang dilakukan vectorizer ini dibatasi dengan empat parameter, yakni `max_features` untuk memfilter jumlah kata dengan nilai bobot TF-IDF tertinggi, `ngram_range` untuk memfilter maksimal jumlah kata dalam frasa yang diproses, `max_df` akan mengabaikan kata yang muncul lebih dari persentasenya, dan `min_df` akan mengabaikan kata yang muncul kurang dari persentasenya. Selain itu, dilakukan `fit_transform` untuk menghitung dan mengubah teks ulasan menjadi matriks *TF-IDF* yang sesuai dengan parameter yang telah ditentukan, sehingga setiap dokumen (teks) akan diwakili sebagai vektor numerik berdasarkan nilai TF-IDF dari kata-kata yang ada. Hasil dari matriks *TF-IDF* yang terbentuk disimpan dalam variabel `matrixTFIDF` untuk dilakukan proses pada bagian selanjutnya.

### D. Proses SVD dan Interpretasi Topik dari Kata Dominan

```
# LSA menggunakan SVD
U, S, Vt = np.linalg.svd(matrixTFIDF, full_matrices=False)
jumlahTopic = 3
reducedVt = Vt[:,jumlahTopic, :]

# Kata-kata dari setiap topik
kata = vectorizer.get_feature_names_out()
```

**Gambar 11.** Kode proses pembacaan dan *preprocessing* dataset ulasan  
Sumber : Hak Cipta Penulis

Setelah terbentuk matriks *TF-IDF*, dilakukan proses dekomposisi matriks dengan *Singular Value Decomposition (SVD)* yang memakai fungsi bawaan dari library `numpy.linalg`. Dekomposisi *SVD* yang dilakukan pada proses ini bersifat tereduksi dengan mengatur `full_matrices` sebagai `false` untuk menghilangkan dimensi

yang tidak relevan. Pada proses *SVD* di sini, matriks *U* berukuran  $m \times m$  berisi informasi hubungan antara teks dan topik, matriks *S* berukuran  $m \times n$  berisi nilai singular yang mengukur pentingnya setiap dimensi pada data, dan matriks *Vt* berukuran  $n \times n$  berisi informasi hubungan antar kata-kata dengan topik.

Komponen matriks *Vt* diproses lebih lanjut untuk mengambil kata-kata yang dominan dari beberapa topik. Jumlah topik yang dipilih adalah 3 karena memiliki hasil yang paling optimum untuk hasil analisis. Matriks *Vt* akan dipotong berdasarkan jumlah topik untuk mengambil bagian kata yang relevan dengan jumlah topik yang ditentukan dan disimpan pada variabel `reducedVt`.

Untuk menginterpretasikan hasil analisis topik dari proses *LSA* yang dilakukan, semua kata yang terdapat pada data ulasan perlu diekstrak agar bisa dihubungkan dengan matriks `reducedVt` yang didapatkan pada proses sebelumnya. Hal ini dilakukan dengan fungsi `vectorizer.get_feature_names_out()`, yakni mengembalikan array berisi kata yang diproses pada matriks *TF-IDF* dari objek vectorizer.

```
print("\nTopik dan Kata-kata Dominan:")
for i, topic in enumerate(reducedVt):
    topKata = [kata[idx] for idx in np.argsort(topic)[-1][:-6]] # Top 6 kata per topik
    print(f"Topik (i + 1): {' '.join(topKata)}")
```

**Gambar 12.** Kode proses penampilan hasil topik dan kata-kata ulasan yang dominan pada setiap topik  
Sumber : Hak Cipta Penulis

Daftar kata yang diperoleh dari objek vectorizer dicocokkan dengan `reducedVt` yakni bobot kontribusi setiap kata terhadap setiap topik lalu ditampilkan 6 kata bobot tertinggi pada setiap topik secara berurutan.

## IV. HASIL ANALISIS DAN PEMBAHASAN

Dengan mengujikan teks-teks ulasan ulasan yang terdapat pada kolom `translatedText` pada dataset informasi ulasan Restoran Bebek Ngarasan dengan melakukan proses *LSA*, didapatkan hasil analisis Topik dan kata-kata yang memiliki bobot paling dominan relatif terhadap topik sebagai berikut:

```
Topik dan Kata-kata Dominan:
Topik 1: long time, wait, take, customer, restaurant, long
Topik 2: delicious, cheap, delicious cheap, food delicious, sauce delicious, food
Topik 3: good, price, taste, cheap, food, great
```

**Gambar 13.** Hasil Eksekusi program *LSA* terhadap data ulasan Restoran Bebek Ngarasan  
Sumber : Hak Cipta Penulis

Berdasarkan tampilan hasil eksekusi program *LSA*, dihasilkan kumpulan-kumpulan kata yang relatif terhadap pada setiap topik dengan batasan jumlah topik yang telah ditentukan. Setiap topik yang dihasilkan memiliki kecenderungan pada suatu tema ulasan. Pada topik pertama, informasi utama yang dapat diketahui adalah mengenai waktu pelayanan di Restoran Bebek Ngarasan. Hal ini dapat diketahui dengan kata "*long time*", "*wait*", dan "*take*" yang terdapat pada topik pertama. Kata-kata

yang muncul pada topik ini menunjukkan bahwa terdapat ketidakpuasan mengenai waktu tunggu yang lama dalam pemesanan makanan.

Untuk topik kedua, kata-kata sifat seperti “*delicious, cheap, sauce delicious*” dan kata benda “*food*” memberikan gambaran mengenai ulasan positif pada menu hidangan pada Restoran Bebek Ngarasan yang enak dan terjangkau. Hal ini menunjukkan tema pada topik kedua yang berkaitan dengan *worthiness* dari hidangan yang disajikan oleh Restoran Bebek Ngarasan.

Sementara pada topik ketiga, kata-kata dominannya memperlihatkan kecenderungan pada harga dari menu makanan Restoran Bebek Ngarasan. Kata “*good, price, cheap, food*” menggambarkan tema topik terkait dengan harga makanan yang murah.

Dari hasil analisis ini, dapat dibuktikan bahwa *Latent Semantic Analysis* (LSA) sebagai penerapan dari teknik *Singular Value Decomposition* (SVD) mampu mengungkap pola-pola topik yang diulas oleh konsumen berdasarkan kata-kata dominan yang muncul pada topik ulasan. Sayangnya, Latent Semantic Analysis tidak terlalu akurat dalam jumlah topik yang lebih besar dan diperlukan pengaturan parameter kemunculan kata pada topik yang berbeda.

Penulisan kata ulasan yang diberikan pengguna sebagai data ulasan yang dianalisis menjadi suatu tantangan sendiri karena penggunaan bahasa yang tidak baku sehingga bentuk kalimat ulasan sudah ditranslasi ke bahasa Inggris tetap tidak sepenuhnya sempurna dan koheren. Hal ini mempersulit dalam proses *preprocessing* pada teks dalam program LSA. Oleh karena itu, optimalisasi pada *preprocessing* teks dan penentuan parameter, filterisasi kata dengan kemunculannya pada pembuatan matriks *TF-IDF*, dan penentuan jumlah top yang tepat merupakan kunci dalam hasil analisis yang baik.

## V. KESIMPULAN

Dengan menggunakan *Latent Semantic Analysis* sebagai penerapan dari *Singular Value Decomposition*, Analisis pada data-data ulasan terhadap Restoran Bebek Ngarasan berhasil mendapatkan topik-topik utama yang menyangkut paut dengan ulasan yang diberikan, yakni ulasan konsumen berfokus pada topik waktu pelayanan, rasa makanan, dan harga makanan yang ditawarkan di Restoran Bebek Ngarasan. Walaupun begitu, penggunaan LSA dalam analisis masih dapat terus dioptimalkan untuk mendapatkan hasil informasi laten ulasan yang lebih relevan dan tidak abstrak.

## REFERENCES

- [1] “geeksforggeeks,” [Online]. Available: <https://www.geeksforggeeks.org/latent-semantic-analysis/> [Diakses 28 Desember 2024].
- [2] R. Munir, “Aljabar Linier dan Geometri,” 2023. [Online]. Available: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-21-Singular-value-decomposition-Bagian1-2023.pdf> [Diakses 28 Desember 2024].

- [3] D. Suhartono, “Penggunaan Latent Semantic Analysis (LSA) dalam Pemrosesan Teks,” 2015. [Online]. Available: <https://socs.binus.ac.id/2015/08/03/penggunaan-latent-semantic-analysis-lsa-dalam-pemrosesan-teks/> [Diakses 28 Desember 2024].
- [4] R. Munir, “Aljabar Linier dan Geometri,” 2023. [Online]. Available: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-01-Review-Matriks-2023.pdf> [Diakses 31 Desember 2024].
- [5] “geeksforggeeks,” [Online]. Available: <https://www.geeksforggeeks.org/matrices/> [Diakses 31 Desember 2024].
- [6] “Newcastle University” [Online]. Available: <https://www.ncl.ac.uk/webtemplate/ask-assets/external/math-resources/core-mathematics/pure-maths/matrices/matrix-determinant.html> [Diakses 31 Desember 2024].
- [7] R. Munir, “Aljabar Linier dan Geometri,” 2023. [Online]. Available: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-01-Review-Matriks-2023.pdf> [Diakses 1 Januari 2025].

## VI. UCAPAN TERIMA KASIH

Pertama-tama, penulis hendak ucapkan puji dan syukur kepada Tuhan Yang Maha Esa atas berkat, dan rahmat-Nya sehingga makalah ini bisa terselesaikan. penulis juga mengucapkan terima kasih kepada Bapak Ir. Rila Mandala, M.Eng., Ph.D. selaku dosen pengampu mata kuliah IF1220 Aljabar Linier dan Geometri kelas K-1 beserta Bapak Dr. Ir. Rinaldi Munir, M.T., yang telah membimbing dan memberikan materi-materi ilmu yang berharga sepanjang penulisan makalah ini. Tak lupa, penulis juga sampaikan terima kasih kepada keluarga, teman-teman, dan pihak lain yang turut mendukung penulis selama proses penulisan makalah ini. Semoga Tuhan Yang Maha Esa membalas semua kebaikan yang diberikan dengan kebaikan yang berlipat ganda.

## 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, 2 Januari 2025



Muhammad Aufa Farabi dan 13523023