

**LAPORAN AKHIR DATA MINING:
Profiling Destinasi Wisata di Indonesia**



**Disusun oleh:
Kelompok 1**

Adytia Wiguna Gunawan	2602108262
Alvin Jusup	2602110720
Amadeus Reyno Adi N	2602112316
Aufa Ekananda Haydan P	2602151613

Dosen :

D6641 - Dr. Devi Fitrianah, S.Km, M . T . I

**PROGRAM STUDI COMPUTER SCIENCE
FAKULTAS SCHOOL OF COMPUTER SCIENCE
UNIVERSITAS BINA NUSANTARA
JAKARTA
Tahun Ajaran 2023/2024**

DAFTAR ISI

DAFTAR ISI.....	2
BAB 1 PENDAHULUAN.....	2
1.1. Latar Belakang.....	2
1.2. Tujuan.....	3
BAB 2 LANDASAN TEORI.....	3
2.1. Profiling Destinasi Wisata di Indonesia.....	3
2.2. Metode Clustering yang sering digunakan pada Destinasi Wisata.....	4
2.3. Haversine Method.....	5
2.4. Perbandingan metode clustering untuk Destinasi Wisata.....	5
2.5. Metode Robust Scaler dan Label Encoding.....	6
2.6. Metode PCA.....	6
BAB 3 METODE.....	7
3.1. Pemilihan Model Clustering.....	7
3.2. Pengumpulan Data.....	8
BAB 4 IMPLEMENTATION AND EVALUATION.....	9
.....	9
4.1.1. Data Cleaning.....	9
4.1.2. Data Integration.....	9
4. 3. Data Selection (Feature Selection).....	11
4.3.2. Categorical Features.....	11
4.3.3. Correlation Matrix.....	12
4. 4. Data Normalization.....	12
4.3.1. RobustScaler.....	12
4.3.2. LabelEncoder.....	12
4.3.3. Re-Integration.....	12
4. 5. Data Reduction.....	13
4.6. Outlier/Noise.....	13
4. 7. Data Mining.....	15
4. 8. Model Evaluation.....	17
BAB 5 CONCLUSION AND SUGGESTIONS.....	18
REFERENSI.....	19
LAMPIRAN.....	1

BAB 1 PENDAHULUAN

1.1. Latar Belakang

Indonesia merupakan negara kepulauan terbesar di dunia, terdiri dari belasan ribu pulau yang membentang luas dari Sabang sampai Merauke. Dengan bentuk geografis tersebut, Indonesia tidak hanya memiliki keindahan alam yang luar biasa, tetapi juga keberagaman tradisi, budaya, adat, warisan sejarah, maupun kekayaan alam yang unik di setiap wilayahnya.

Dengan keberagaman tersebut, Indonesia menjadi salah satu destinasi wisata yang diminati di dunia dan menjadi pertimbangan wisatawan dari dalam negeri maupun dari mancanegara. Wisatawan dalam negeri seringkali menjadikan destinasi Indonesia sebagai pilihan karena keindahan alam yang unik, keberagaman budaya maupun lokasi geografisnya. Sedangkan wisatawan internasional tertarik dengan suasana dan pesona eksotisme Indonesia dengan iklimnya yang tropis, serta keunikan yang terdapat pada tradisi dan kebudayaan

Namun, destinasi wisata yang melimpah tersebut seringkali menjadi tantangan bagi wisatawan yang ingin menjelajahi wisata di Indonesia. Wisatawan harus mempertimbangkan faktor seperti lokasi, jenis wisata, biaya, akomodasi seperti tempat penginapan, hotel, maupun moda transportasinya. Masalah ini menciptakan kebutuhan akan informasi dan analisis yang dapat membantu *profiling* destinasi wisata berdasarkan karakteristik yang ada.

Penelitian ini bertujuan untuk membantu mengatasi tantangan dan permasalahan tersebut dengan menggabungkan informasi berupa dataset destinasi wisata di Indonesia dan hotel di Indonesia. Dengan pendekatan dan metode *clustering* pada *data mining*, penelitian ini diharapkan dapat memahami pola dan karakteristik dari kombinasi informasi destinasi wisata dengan hotel terdekat, sehingga dapat mempermudah wisatawan untuk membuat pertimbangan dan memberikan wawasan kepada penggiat industri pariwisata di Indonesia agar lebih berkembang.

1.2. Tujuan

Tujuan utama analisis ini adalah:

- Menentukan metode clustering yang tepat untuk *profiling* destinasi wisata
- Menemukan *insight* dan informasi yang tepat sasaran untuk mendukung pengambilan keputusan.
- Membandingkan dan menentukan algoritma clustering untuk Profiling Destinasi Wisata

BAB 2 LANDASAN TEORI

2.1. Profiling Destinasi Wisata di Indonesia

Wisata adalah sebuah kegiatan yang dilakukan untuk mendapatkan pengetahuan, bersenang-senang dan menghibur diri. Indonesia sebagai negara kepulauan tentunya memiliki banyak wisata di dalamnya yang berbeda-beda. Profiling Destinasi Wisata memiliki peran penting dalam mewujudkan sebuah kegiatan wisata. Analisa karakteristik destinasi wisata dibutuhkan pola kunjungan wisata, fasilitas, cuaca, dan masih banyak lagi. Salah satu aspek terpenting terutama untuk turis dalam berwisata adalah hotel, sebagai tempat tinggal sementara. Tentunya pelayanan, harga, serta jarak wisata menjadi faktor penentu apakah turis akan memakai jasa hotel tersebut.

2.2. Metode Clustering yang sering digunakan pada Destinasi Wisata

Clustering adalah metode untuk pengelompokan data tanpa adanya label atau disebut metode unsupervised learning. Metode ini mengelompokkan apapun yang kita inginkan dan dapat memberikan insight dari pengelompokkan tersebut. Metode yang tidak menggunakan label ini dapat mengelompokkan kolom yang memiliki value serupa dan dikategorisasikan, sehingga tidak perlu banyak kerja seperti klasifikasi. Algoritma pada clustering yang biasanya digunakan pada profiling clustering adalah K-Means Clustering, Hierarchical Clustering, dan DBSCAN

K-Means adalah algoritma clustering yang sering digunakan sebagai unsupervised learning algorithm. Kerja algoritma ini adalah dengan membagi data menjadi beberapa

cluster dengan centroid sebagai pusat cluster. Data K-Means dikelompokkan melalui jarak yang dekat dengan centroid. Algoritma ini sering digunakan untuk demografi dan segmentasi wisatawan.

Hierarchical Clustering merupakan algoritma yang membentuk struktur layaknya pohon yang mengidentifikasi kelompok variabel berdasarkan tingkat kesamaan, karena pada Profiling Wisata maka bisa diberikan contoh variabel tersebut adalah wisatawan. Algoritma ini cocok untuk mengenali subkelompok seperti data wisatawan yang dapat dibagi menjadi wisatawan anak, muda, dan tua.

DBSCAN mengelompokkan data dengan berdasarkan mengidentifikasi outlier serta kepadatan spasial. Algoritma DBSCAN cocok untuk wisata karena dapat digunakan untuk mengenali lokasi dengan konsentrasi wisatawan tinggi atau juga dapat mengelola keramaian di tempat wisata.

2.3. Haversine Method

Metode Haversine merupakan rumus matematika yang digunakan dalam navigasi dan geografi untuk menghitung jarak antara dua titik koordinat di atas permukaan sebuah bola (*sphere*), seperti Bumi. Rumus Haversine dibuat berdasarkan hukum Haversine, yang berhubungan dengan sisi dan sudut dari segitiga bola. Rumus ini memperhitungkan kelengkungan dari permukaan (berbentuk bola) sehingga menghasilkan jarak yang lebih akurat dibandingkan dengan perhitungan jarak geometri sederhana (*Euclidean distance*). Rumus Haversine mengasumsikan bahwa bumi berbentuk bulat sempurna dengan jari-jari / Radius = 6.371 km. Persamaan Haversine dapat ditulis sebagai berikut:

$$a = \sin^2(\Delta\phi/2) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2(\Delta\lambda/2)$$

$$c = 2 \cdot \arctan2(\sqrt{a}, \sqrt{1-a})$$

$$d = R \cdot c$$

2.4. Perbandingan metode clustering untuk Destinasi Wisata

Metode	Struktur Data	Penanganan Noise	Hasil	Kekurangan	Kelebihan
--------	---------------	------------------	-------	------------	-----------

K-Means	Cluster yang memiliki ciri distribusi merata dan berbentuk bulat	Semua data harus memasuki cluster, karena K-Means tidak dapat menangani noise	Data dibagi ke K cluster dengan centroid yang sudah ditentukan	Bergantung pada K sehingga perlu ditentukan terlebih dahulu, serta rentan terhadap centroid yang buruk	Mudah dan sederhana untuk sebuah cluster
Hierarchical Clustering	Data yang kecil atau data dengan relasi hierarki	Tidak mempunyai konsep noise eksplisit	Menyusun dendrogram pada hierarki cluster	Sangat sensitif terhadap outlier serta komputasi yang lambat untuk big data	Tidak memerlukan parameter awal, serta fleksibel dengan linkage
DBSCAN	Cluster dengan bentuk arbitrer dan densitas bervariasi	Berbeda dengan algoritma 2 di atas, object di luar kerapatan dianggap noise oleh algoritma ini	Data dibagi dengan dense cluster melalui noise	Pada dataset yang memiliki dimensi tinggi menimbulkan ketidak optimalan algoritma DBSCAN	Dapat menangani noise. Cocok untuk cluster berbentuk arbitrer

2.5. Metode Robust Scaler dan Label Encoding

Robust Scaler merupakan metode feature scaling yang digunakan untuk menangani fitur numerik dengan noise atau outlier. Robust Scaler menggunakan pendekatan yang lebih tahan terhadap outlier dibandingkan metode scaling lainnya seperti Min-Max Scaling dan Standardization. Metode ini menggunakan estimasi statistik seperti median dan kuartil untuk mengubah skala dari fitur numerik pada data. Dengan menggunakan, median, kuartil, dan rentang antar kuartil (IQR), Robust Scaler memastikan bahwa skala data tidak terdistorsi oleh outlier.

Label encoding adalah teknik dalam machine learning untuk mengubah data kategorikal menjadi data numerik yang dapat dipahami oleh algoritma machine learning. Cara kerja Label Encoding adalah dengan melakukan identifikasi terhadap semua nilai unik yang ada pada suatu kolom. Selanjutnya nilai unik tersebut akan diberi label berupa angka yang dimulai dari 0 dan berurut hingga semua nilai unik terwakili.

2.6. Metode PCA

Principal Component Analysis (PCA) merupakan teknik reduksi dimensi yang umum digunakan dalam bidang data mining untuk mengatasi masalah data berdimensi tinggi. PCA bertujuan memproyeksikan data ke ruang berdimensi lebih rendah dengan tetap mempertahankan sebanyak mungkin informasi penting, yaitu variansi data. Proses ini dilakukan dengan menghitung komponen utama (principal components), yang merupakan kombinasi linier dari fitur asli yang menjelaskan variansi terbesar. Dalam analisis ini, PCA diterapkan untuk mereduksi dimensi data menjadi 5 komponen utama menggunakan fungsi `PCA(n_components=5)`. Reduksi ini menghasilkan representasi data yang lebih efisien dan optimal untuk analisis lebih lanjut, sekaligus mengurangi redundansi fitur dan beban komputasi tanpa kehilangan esensi informasi yang signifikan.

2.7. Metode Kaufman

Metode **Kaufman**, yang lebih dikenal dengan teknik *k-means++*, adalah pendekatan yang digunakan untuk meningkatkan pemilihan centroid awal dalam algoritma **K-Means Clustering**. Dalam K-Means, pemilihan centroid awal yang buruk dapat mengarah pada hasil clustering yang tidak optimal atau membutuhkan waktu konvergensi yang lebih lama. *K-means++* mengatasi masalah ini dengan memilih centroid pertama secara acak, lalu memilih centroid berikutnya berdasarkan jarak probabilistik dari data yang ada, di mana titik data yang lebih jauh dari centroid sebelumnya memiliki probabilitas lebih tinggi untuk dipilih. Metode ini memastikan centroid tersebar lebih merata di ruang data, mengurangi risiko terjebak pada solusi lokal dan mempercepat proses konvergensi. Dengan demikian, teknik ini meningkatkan efisiensi dan kualitas hasil clustering, terutama ketika diterapkan pada data berdimensi tinggi atau kompleks, yang menjadi masalah umum dalam data mining.

BAB 3 METODE

Project yang kami lakukan menggunakan metode machine learning K-Means dalam melakukan Profiling Destinasi Wisata di Indonesia. Kami menggunakan metode tersebut untuk memberikan *insight* dari clustering data wisata.

3.1. Pemilihan Model Clustering

K-Means adalah algoritma unsupervised learning yang mengelompokkan data ke dalam data cluster. Sesuai dengan namanya, unsupervised learning menandakan K-Means dapat menerima data tanpa adanya label kategori. Model ini bekerja dengan cara menentukan jumlah cluster, mendistribusikan data secara acak, dan menghitung centroid untuk setiap cluster. Tujuan dari model ini meminimalkan variasi dalam satu cluster dan memaksimalkan cluster satu dengan cluster lainnya. Metode berbasis jarak juga terdapat di model ini seperti Euclidean atau Manhattan distance. Destinasi wisata tepat menggunakan K-Means karena data yang diambil terdapat data lokasi geografis yang artinya data yang memiliki cluster alami, contoh data tersebut adalah latitude dan longitude.

DBSCAN merupakan algoritma clustering yang dapat menumbuhkan daerah yang mempunyai high density menjadi cluster dan memanfaatkan noise untuk menemukan cluster pada bentuk bebas. Konsep DBSCAN adalah banyaknya data (minPts) yang berada di radius Eps dari setiap data.

3.2. Pengumpulan Data

Kami menggunakan data dalam project ini yaitu data hotel & destinasi. Data hotel berjudul “New Year Hotel Rooms Availability from Traveloka” kami dapatkan melalui kaggle yang berisikan data reservasi hotel pada tahun baru 2023. Lalu terdapat data destinasi yang didapatkan melalui kaggle dengan berjudul “Indonesia Tourism Destination”. Data tersebut berisikan tempat destinasi turis di Indonesia. Chatgpt juga ikut serta dalam penambahan data untuk hotel dan destinasi agar data tersebut dapat digunakan di algoritma clustering.

BAB 4 IMPLEMENTATION AND EVALUATION

4.1.1. Data Cleaning

a. Handling Missing Value

- Membuang atribut `highRate`, `hotelFeatures` dan `userRatingInfo` dari data hotel karena kebanyakan valuenya kosong.

```
df_hotel.drop(['highRate', 'hotelFeatures'], axis=1,
inplace=True)

df_hotel.drop(["userRatingInfo"], axis=1,
inplace=True)
```

- Mengisi missing value pada atribut `userRating` dari data hotel dengan median (nilai tengah).

```
#Handle missing value di data hotel
#Isi dengan median (nilai tengah)

df_hotel['userRating'] =
df_hotel['userRating'].fillna(df_hotel['userRating'].median())
```

- Mengisi missing value pada atribut `numReviews` dari data hotel dengan mean (rata-rata).

```
#Handle missing value di data hotel
#Isi dengan mean (rata-rata)

df_hotel['numReviews'] =
df_hotel['numReviews'].fillna(df_hotel['numReviews'].mean())
```

4.1.2. Data Integration

- #### a.
- Membuat function Haversine Formula untuk menggabungkan data hotel dengan data destinasi wisata berdasarkan jarak terdekat antar titik koordinatnya (latitude & longitude).

- Function untuk menghitung jarak

```
#Hitung jarak antar dua koordinat tempat menggunakan Latitude dan Longitude,dalam konteks ini jarak setiap destinasi untuk setiap hotel
```

```
#Buat function untuk menghitung jarak, di sini kita menggunakan Haversine formula
def haversine(lat1, lon1, lat2, lon2):
    R = 6378 # Radius bumi (km)
    lat1, lon1, lat2, lon2 = map(np.radians, [lat1, lon1, lat2, lon2])
    dlat = lat2 - lat1
    dlon = lon2 - lon1
    a = np.sin(dlat / 2)**2 + np.cos(lat1) * np.cos(lat2) * np.sin(dlon / 2)**2
    c = 2 * np.arctan2(np.sqrt(a), np.sqrt(1 - a))
    return R * c
```

- Function untuk mencari destinasi terdekat

```
#Buat function untuk mencari destinasi terdekat untuk setiap hotel

def find_nearest_destination(hotel, destinations):
    distances = destinations.apply(
        lambda dest: haversine(hotel['latitude'], hotel['longitude'], dest['Lat'],
dest['Long']), axis=1
    )
    nearest_idx = distances.idxmin()
    return destinations.loc[nearest_idx, 'Place_Name'], distances[nearest_idx]
```

```
#Apply function yang sudah dibuat untuk menemukan destinasi terdekat untuk setiap
hotel

df_hotel[['Nearest_Destination', 'Distance_km']] = df_hotel.apply(
    lambda row: pd.Series(find_nearest_destination(row, df_destination)), axis=1
)
```

- Merging Dataset

```
#Merge data

df_merged = df_hotel.merge(
    df_destination[['Place_Name', 'City', 'Rating', 'Price']],
    left_on='Nearest_Destination',
    right_on='Place_Name',
    how='left'
)

df_merged.drop(columns=['Place_Name'], inplace=True)
```

4. 3. Data Selection (Feature Selection)

4.3.1. Numerical Features

Memilih fitur-fitur numerik yang penting dan relevan untuk clustering.

```
numerical_features = df_merged[
    ['starRating', 'userRating', 'numReviews', 'Distance_km',
    'lowRate', 'cheapestRate_perNight_totalFare', 'num_staying_nights']
]
```

4.3.2. Categorical Features

Memilih fitur-fitur kategorikal yang penting dan relevan untuk clustering.

```
categorical_features = df_merged[['region', 'city', 'Nearest_Destination']]
```

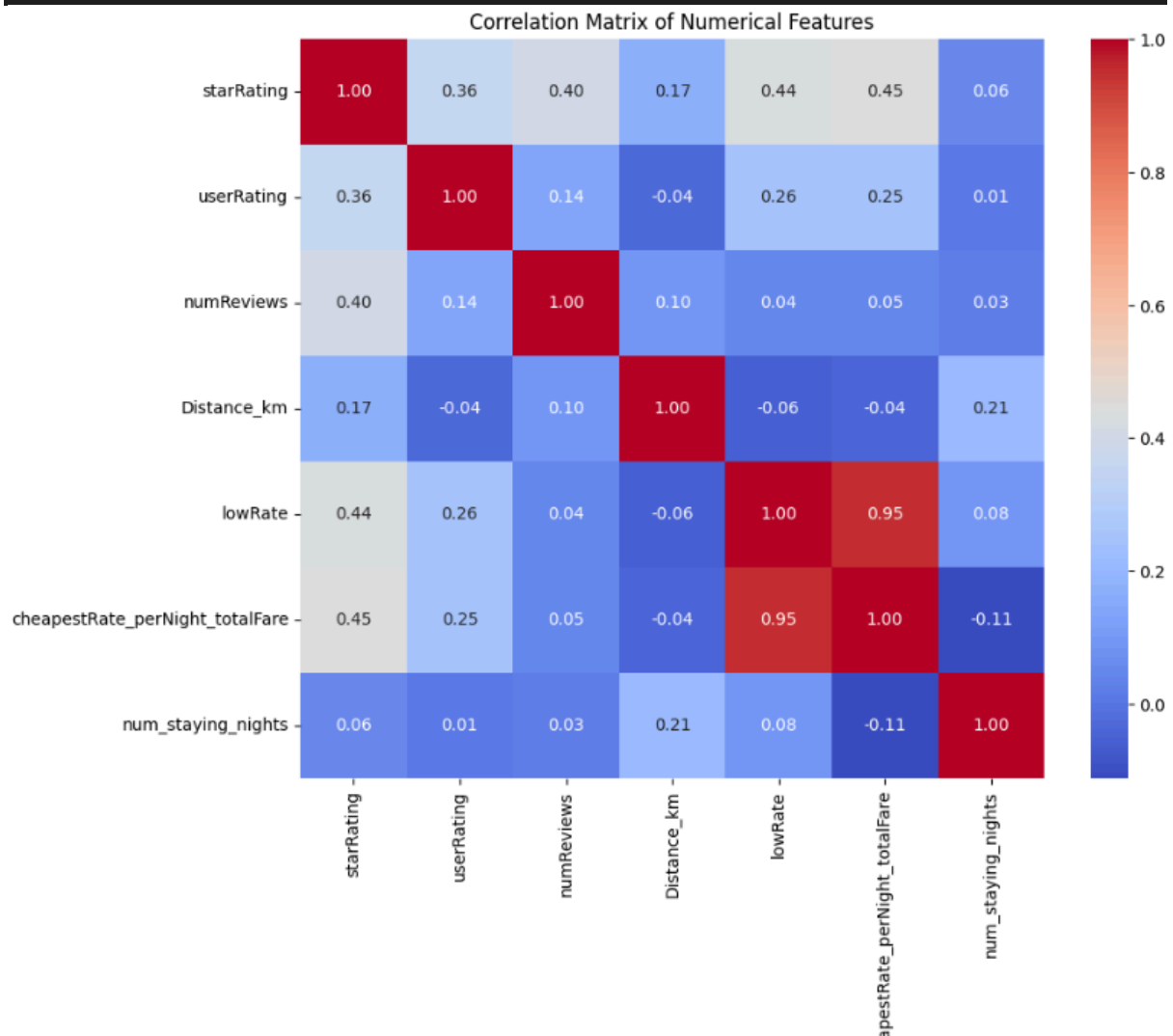
4.3.3. Correlation Matrix

Menghitung hubungan antar variabel dalam bentuk matriks.

```
# Feature Selection Matrix
correlation_matrix = numerical_features.corr()

print("Correlation Matrix:")
print(correlation_matrix)

plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix of Numerical Features')
plt.show()
```



4. 4. Data Normalization

4.3.1. RobustScaler

Melakukan normalisasi fitur numerik menggunakan RobustScaler.

```
scaler = StandardScaler()  
numerical_scaled = scaler.fit_transform(numerical_features)
```

4.3.2. LabelEncoder

Melakukan normalisasi fitur numerik menggunakan StandardScaler.

```
encoder = LabelEncoder()  
categorical_encoded = categorical_features.apply(encoder.fit_transform)
```

4.3.3. Re-Integration

Menggabungkan kembali fitur-fitur numerik dan kategorikal yang telah dinormalisasi.

```
#Gabungkan semua features yang ada setelah normalisasi  
processed_data = pd.concat([  
    pd.DataFrame(numerical_scaled, columns=numerical_features.columns),  
    categorical_encoded.reset_index(drop=True)  
], axis=1)
```

4. 5. Data Reduction

4.5.1. Dimension Reduction menggunakan PCA

```
#Apply PCA untuk mereduksi dimensi  
  
pca = PCA(n_components=5)  
pca_transformed = pca.fit_transform(processed_data)
```

4.6. Outlier/Noise

4.6.1. Check Outlier (Boxplot)

```
Q1 = processed_data.quantile(0.25)  
Q3 = processed_data.quantile(0.75)  
IQR = Q3 - Q1  
  
lower_bound = Q1 - 1.5 * IQR  
upper_bound = Q3 + 1.5 * IQR
```

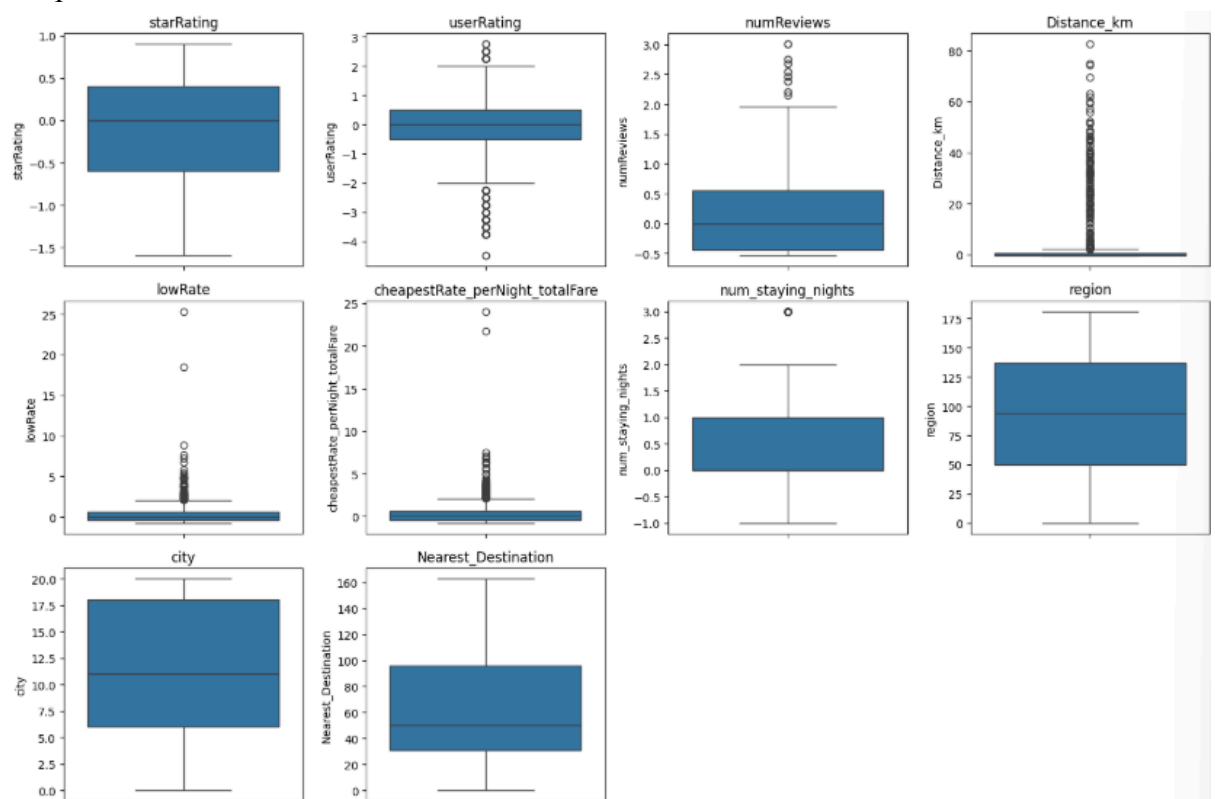
```

outliers = processed_data[(processed_data < lower_bound) |
                           (processed_data > upper_bound)].any(axis=1)

plt.figure(figsize=(15, 10))
for i, col in enumerate(processed_data.columns):
    plt.subplot(3, 4, i + 1)
    sns.boxplot(y=processed_data[col])
    plt.title(col)
plt.tight_layout()
plt.show()

```

Boxplot sebelum remove outlier



4.6.2. Handling Outlier

```

#Handle outlier

processed_data_no_outliers = processed_data[~outliers]

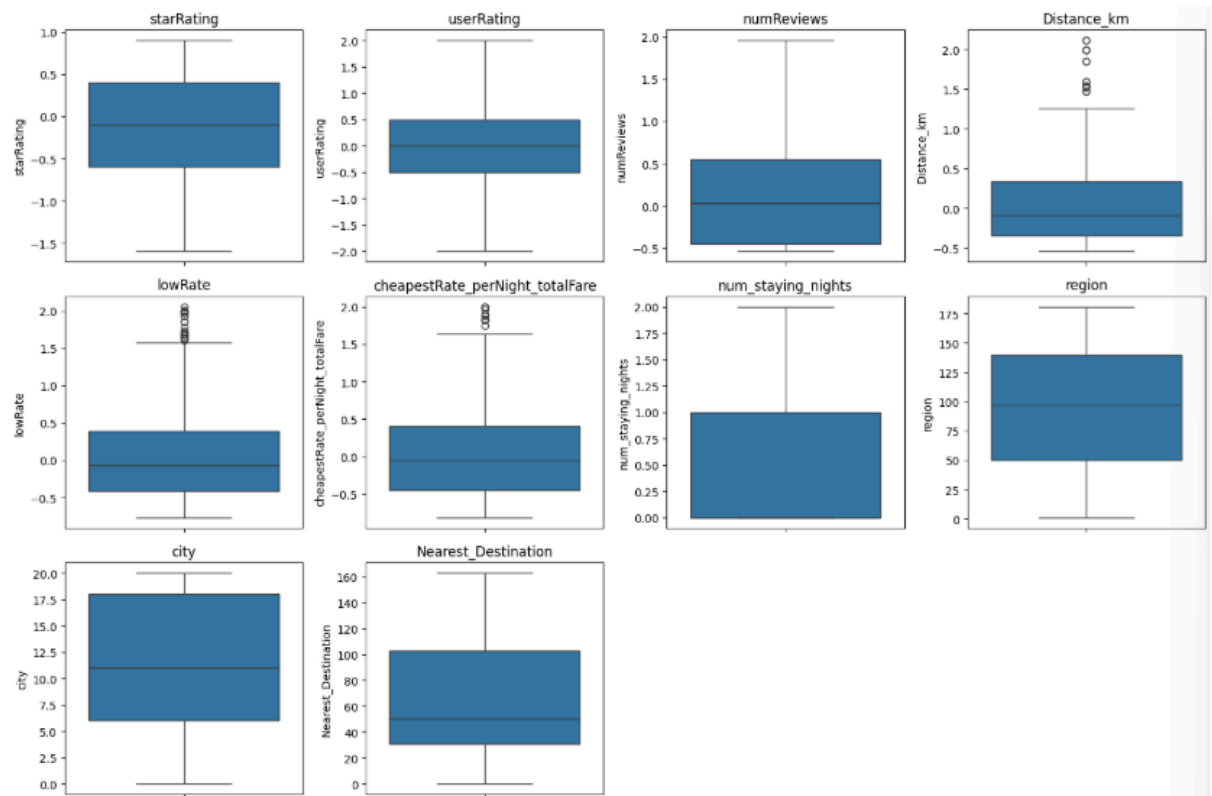
print("Shape of data before outlier removal:",
processed_data.shape)

print("Shape of data after outlier removal:",
processed_data_no_outliers.shape)

```

```
# Apply PCA lagi ke data tanpa outlier
pca_transformed_no_outliers =
pca.fit_transform(processed_data_no_outliers)
```

Boxplot sesudah remove outlier



4. 7. Data Mining

4.7.1. K-Means Clustering

```
#Apply clustering dengan KMeans

best_silhouette_score = -1
best_davies_bouldin_score = float('inf')
best_n_clusters = 0

for n_clusters in range(2, 11):
    kmeans = KMeans(n_clusters=n_clusters, random_state=42,
init='k-means++', n_init=10, max_iter=300)
    kmeans.fit(pca_transformed_no_outliers)
    clusters = kmeans.labels_

    silhouette = silhouette_score(pca_transformed_no_outliers,
clusters)
```

```

                                davies_bouldin      =
davies_bouldin_score(pca_transformed_no_outliers, clusters)

    if silhouette > best_silhouette_score and davies_bouldin <
best_davies_bouldin_score:
        best_silhouette_score = silhouette
        best_davies_bouldin_score = davies_bouldin
        best_n_clusters = n_clusters

    print(f"Best Silhouette Score: {best_silhouette_score:.3f} with
{best_n_clusters} clusters")
    print(f"Best          Davies-Bouldin          Score:
{best_davies_bouldin_score:.3f} with {best_n_clusters} clusters")

    # Gunakan parameter terbaik untuk model akhir
    kmeans_kaufman      =      KMeans(n_clusters=best_n_clusters,
random_state=42, init='k-means++', n_init=10, max_iter=300)
    kmeans_kaufman.fit(pca_transformed_no_outliers)
    clusters_kmeans = kmeans_kaufman.labels_

```

4.7.1. Dbscan

```

#Apply clustering dengan DBSCAN

best_silhouette_score = -1
best_davies_bouldin_score = float('inf')      # Initialize with
infinity
best_eps = 0
best_min_samples = 0

for eps in np.arange(0.1, 2.0, 0.1):
    for min_samples in range(2, 10):
        dbscan = DBSCAN(eps=eps, min_samples=min_samples)
                                clusters      =
dbscan.fit_predict(pca_transformed_no_outliers)
        if len(np.unique(clusters)) > 1:
                                silhouette      =
silhouette_score(pca_transformed_no_outliers, clusters)
                                davies_bouldin      =
davies_bouldin_score(pca_transformed_no_outliers, clusters)
        if silhouette > best_silhouette_score and
davies_bouldin < best_davies_bouldin_score:

```

```

best_silhouette_score = silhouette
best_davies_bouldin_score = davies_bouldin
best_eps = eps
best_min_samples = min_samples

```

4. 8. Model Evaluation

4.8.1 K-Means

```

print(f"Best Silhouette Score: {best_silhouette_score:.3f} with {best_n_clusters} clusters")
print(f"Best Davies-Bouldin Score: {best_davies_bouldin_score:.3f} with {best_n_clusters} clusters")

```

Elbow Method

```

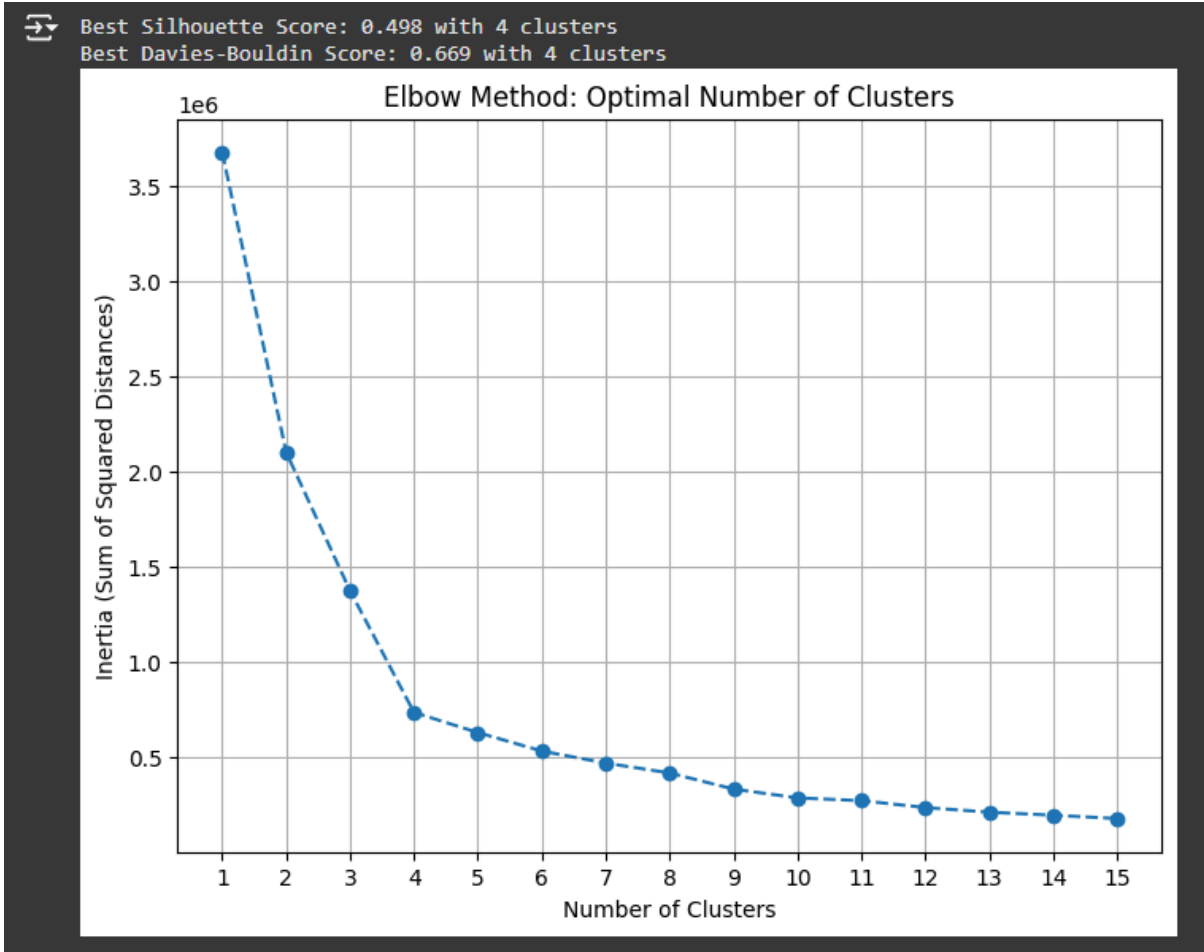
#Elbow method
cluster_range = range(1, 16)

inertia_values = []

for k in cluster_range:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(pca_transformed_no_outliers)
    inertia_values.append(kmeans.inertia_)

plt.figure(figsize=(8, 6))
plt.plot(cluster_range, inertia_values, marker='o',
linestyle='--')
plt.title("Elbow Method: Optimal Number of Clusters")
plt.xlabel("Number of Clusters")
plt.ylabel("Inertia (Sum of Squared Distances)")
plt.xticks(cluster_range)
plt.grid()
plt.show()

```

4.8.2. Dbscan

```
print(f"Best Silhouette Score with DBSCAN:
{best_silhouette_score:.3f} (eps={best_eps},
min_samples={best_min_samples})")

print(f"Best Davies-Bouldin Score with DBSCAN:
{best_davies_bouldin_score:.3f} (eps={best_eps},
min_samples={best_min_samples})")
```

Best Silhouette Score with DBSCAN: 0.722 (eps=1.1, min_samples=2)
Best Davies-Bouldin Score with DBSCAN: 1.170 (eps=1.1, min_samples=2)

BAB 5 CONCLUSION AND SUGGESTIONS

Berdasarkan hasil project clustering destinasi wisata, setiap kelompok hotel dirancang untuk memenuhi kebutuhan spesifik wisatawan. Cluster 0 menargetkan wisatawan kelas atas yang mencari kenyamanan mewah dengan lokasi strategis atau dekat wisata, ideal bagi

mereka yang memiliki anggaran besar dan menghargai akses cepat ke pusat aktivitas. Cluster 1, di sisi lain, dirancang untuk wisatawan dengan anggaran terbatas yang mencari akomodasi murah, terutama di area strategis pusat kota. Sementara itu, Cluster 2 cocok untuk wisatawan yang mencari pengalaman berkualitas tinggi dengan lokasi strategis dan fasilitas premium. Sebagai alternatif, Cluster 3 menawarkan kenyamanan premium yang sedikit lebih terjangkau dibandingkan Cluster 2, sehingga cocok bagi wisatawan yang menginginkan pengalaman tinggi tanpa membayar harga tertinggi. Melalui pendekatan ini, hotel dapat menyesuaikan layanan dan fasilitas untuk memenuhi kebutuhan setiap segmen wisatawan.

Hasil clustering ini diperoleh menggunakan algoritma K-Means, yang lebih optimal untuk dijadikan model dibandingkan DBSCAN. Walaupun K-Means tidak menghasilkan *silhouette score* sebesar DBSCAN, algoritma ini menunjukkan kinerja yang lebih baik dengan nilai Davies-Bouldin Score yang lebih rendah. Hal ini menunjukkan bahwa cluster yang dihasilkan memiliki pemisahan yang lebih jelas, sehingga cluster-cluster tersebut tidak saling overlap. Dengan demikian, model K-Means memberikan segmentasi yang lebih terdefinisi, memungkinkan hotel untuk memahami dan menargetkan segmen wisatawan dengan strategi yang lebih akurat dan terarah.

Saran dari hasil project ini adalah perlu banyak mencari data yang relevan dan bagus, sehingga dapat memberikan *insight* melalui clustering yang dibuat. Eksplorasi algoritma clustering lainnya untuk mendapatkan perbandingan mana algoritma yang sesuai untuk Destinasi Wisata

REFERENSI

- Dr. Mamta Tiwari, M. S. (2013). *Application of Clustering Algorithms on Tourism Industry*. Retrieved from www.ijraset.com:
<https://www.ijraset.com/research-paper/application-of-clustering-algorithms-on-tourism-industry>

Dwi Arman Prasetya, P. T. (2019). Resolving the Shortest Path Problem using the Haversine Algorithm. *Journal of Critical Reviews* , 64.

Meri Chindyana, L. A. (2021). Segmentation of Tourist Interest on Tourism Object Categories by Comparing PSO K. *Revue d'Intelligence Artificielle*, 37.

RUDIANTO, P. D. (2011). *Clustering Menggunakan Algoritma DBSCAN (Density-Based Spatial Clustering of Applications with Noise) untuk Data Hasil Produksi Potensi Pertanian Studi kasus : Kabupaten Gresik*. Retrieved from etd.repository.ugm.ac.id:
<https://etd.repository.ugm.ac.id/penelitian/detail/50239>

Stephanie Monica Setyana, F. N. (2018). Clustering Tourism Object in Bali Province Using K-Means and X-Means Clustering Algorithm. *ResearchGate*, 7.

LAMPIRAN

Link Collab:

- https://colab.research.google.com/drive/1-h8Tp_8W5ior_dAgK2GoS57wiS0-Woby?usp=sharing

Link Canva:

- https://www.canva.com/design/DAGYU0NFbIk/VwgMUPrO9F8Cn3IkfHy7Yw/edit?utm_content=DAGYU0NFbIk&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton