# Assurance of Learning: Text Mining
# LA09


Disusun Oleh :
Agnes Calista - 2501980690
Matthew Aaron Sugiyarto - 2501983742
Sharon Zefanya Setiawan - 2501961022

**UNIVERSITAS BINA NUSANTARA**
**SCHOOL OF COMPUTER SCIENCE**
**DATA SCIENCE PROGRAM**
**Jl. Kebon Jeruk Raya No. 27, Jakarta Barat 11530**

# DAFTAR ISI

**Daftar Isi** ..... **2**
**BAB I** ..... **3**
**PENDAHULUAN** ..... **3**
1.1 Latar Belakang ..... 3
1.2 Rumusan Masalah ..... 5
1.3 Tujuan ..... 5
1.4 Manfaat ..... 5
**BAB II** ..... **6**
**Landasan Teori** ..... **6**
2.1 Text Mining ..... 6
2.2 Topic Modeling ..... 7
2.3 Latent Dirichlet Allocation ..... 7
2.4 Coherence Model ..... 8
**BAB III** ..... **9**
**METHODOLOGY** ..... **9**
3.1 Dataset Description ..... 9
3.2 Exploratory Data Analysis ..... 9
3.2.1 Load Libraries and Read Files ..... 9
3.2.2 Check Missing Value ..... 10
3.2.3 Check Duplicated Data ..... 10
3.2.5 Count Token ..... 11
3.2.6 Check Anomalies ..... 11
3.2.7 Average Length ..... 12
3.2.8 Check Distribution ..... 13
3.3 Text Preprocessing ..... 13
3.3.1 Import Libraries ..... 14
3.3.2 Cleaning text ..... 14
3.3.3 Labelling Ratings ..... 15
3.3.4 Rating Distribution ..... 17
3.3.5 Wordcloud ..... 18
3.3.6 Tokenization ..... 19
3.3.6.1 using BERT Tokenizer ..... 20
3.4 Modeling and Evaluation ..... 21
3.4.1 Modeling 1 ..... 21
3.4.2 Modeling 2 ..... 24
3.4.3 Grid Search Tuning ..... 26
3.4.4 Best Model ..... 27
3.4.5 Result Topic Name ..... 28
**BAB IV** ..... **30**
**PENUTUP** ..... **30**
4.1 Conclusion ..... 30

# BAB I

# PENDAHULUAN

## 1.1 Latar Belakang

Di era digital saat ini, industri perhotelan menghadapi tantangan yang semakin kompleks. Konsumen kini memiliki akses yang lebih luas untuk berbagi dan memperoleh informasi tentang hotel melalui platform online seperti TripAdvisor, membuat ulasan pelanggan menjadi faktor kritis dalam keberhasilan sebuah hotel. Ulasan ini tidak hanya mempengaruhi reputasi hotel, tetapi juga secara langsung berdampak pada keputusan pemesanan oleh calon tamu. Oleh karena itu, memahami dan menanggapi umpan balik pelanggan melalui analisis data menjadi sangat penting dalam industri yang sangat kompetitif ini.

Masalah utama yang dihadapi oleh banyak hotel adalah bagaimana mengelola dan memanfaatkan secara efektif data besar yang dihasilkan dari ulasan pelanggan. Data ini seringkali bersifat tidak terstruktur dan beragam, mencakup berbagai aspek dari pengalaman menginap, mulai dari kualitas layanan hingga fasilitas hotel. Tanpa metode yang tepat untuk menganalisis dan mengekstrak insight dari data ini, hotel dapat kehilangan kesempatan untuk memahami kebutuhan dan preferensi pelanggan, serta gagal mengidentifikasi area yang membutuhkan perbaikan.

Selain itu, era digital juga membawa perubahan dalam perilaku pelanggan, yang kini mengharapkan personalisasi dan respons yang cepat terhadap keluhan atau kebutuhan mereka. Hotel harus mampu menangkap dan merespon feedback secara real-time untuk mempertahankan kepuasan pelanggan. Hal ini menuntut kebutuhan akan sistem analisis yang tidak hanya akurat dan cepat, tetapi juga mampu memberikan wawasan yang dapat ditindaklanjuti untuk peningkatan layanan dan pengalaman pelanggan.

Industri perhotelan, yang sangat tergantung pada persepsi dan ulasan pelanggan, mengakui pentingnya analisis ulasan sebagai sumber informasi yang berharga. Dalam konteks ini, penelitian ini berfokus pada penggunaan teknik analisis data canggih seperti analisis sentimen, pemodelan prediksi, pemodelan topik, dan pengelompokan ulasan negatif untuk memperoleh pemahaman mendalam dari ulasan hotel. Tujuan utama adalah untuk mengekstrak wawasan berharga dari ulasan, memprediksi sentimen pelanggan, mengidentifikasi topik utama, dan mengelompokkan ulasan negatif yang serupa untuk memberikan kontribusi signifikan dalam meningkatkan kepuasan pelanggan dan proses pengambilan keputusan di industri perhotelan.

Penelitian ini muncul dari kebutuhan mendesak untuk memahami ulasan pelanggan secara lebih efektif dengan menggunakan pendekatan analisis data yang canggih. Ini termasuk penggunaan teknik analisis sentimen yang dapat secara akurat menggambarkan emosi pelanggan, apakah positif, negatif, atau netral. Dengan mengidentifikasi dan menganalisis pola dalam sentimen pelanggan, peneliti dapat menemukan aspek khusus dari layanan hotel yang paling mempengaruhi kepuasan atau ketidakpuasan pelanggan.

Selain itu, penggunaan pemodelan topik seperti Latent Dirichlet Allocation (LDA) memberikan kesempatan untuk menggali lebih dalam dan memahami topik utama yang dibahas dalam ulasan. Pemodelan topik ini memungkinkan identifikasi tema-tema yang sering muncul dalam ulasan, seperti kebersihan, layanan pelanggan, atau fasilitas, memberikan insight yang lebih berfokus tentang apa yang paling dihargai atau dikritik oleh pelanggan. Dengan demikian, penelitian ini tidak hanya mengungkapkan sentimen umum terkait hotel, tetapi juga memberikan pemahaman spesifik tentang elemen-elemen tertentu yang berkontribusi pada pengalaman pelanggan.

Pengelompokan ulasan negatif juga merupakan aspek penting dari penelitian ini, yang bertujuan untuk mengidentifikasi dan menganalisis keluhan dan masalah yang sering muncul. Dengan melakukan ini, hotel dapat mengidentifikasi area kritis yang memerlukan perhatian dan perbaikan. Pendekatan ini membantu dalam merumuskan strategi yang lebih efektif untuk menangani masalah yang sering dikeluhkan oleh pelanggan, dengan tujuan utama untuk meningkatkan standar layanan dan kepuasan pelanggan secara keseluruhan.

Penelitian ini menggunakan dataset yang terdiri dari 20,491 ulasan hotel dari TripAdvisor, memberikan basis data yang kaya untuk analisis mendalam. Dengan menggabungkan analisis sentimen, pemodelan topik, dan pengelompokan ulasan negatif, penelitian ini diharapkan dapat memberikan wawasan yang lebih komprehensif tentang preferensi dan harapan pelanggan terhadap hotel, memprediksi sentimen ulasan secara akurat, dan menyoroti aspek-aspek kunci yang mempengaruhi persepsi pelanggan terhadap sebuah akomodasi. Temuan dari penelitian ini diharapkan tidak hanya memberikan manfaat dalam meningkatkan kepuasan pelanggan, tetapi juga memperkaya proses pengambilan keputusan dan strategi pengembangan layanan hotel di masa depan.

## 1.2 Rumusan Masalah

1. Bagaimana membangun model prediksi yang dapat memproyeksikan peringkat ulasan secara tepat, sehingga dapat memberikan informasi prediktif terkait kualitas layanan dan fasilitas hotel?
2. Apakah terdapat pola tertentu dalam ulasan negatif yang dapat diidentifikasi untuk meningkatkan tindakan perbaikan layanan hotel?
3. Bagaimana informasi yang didapatkan dari model dapat digunakan untuk meningkatkan fokus dan kualitas layanan?
4. Sejauh mana *LDA* topik modeling efektif dalam mengidentifikasi dan memahami topik utama dari ulasan hotel di TripAdvisor?

## 1.3 Tujuan

1. Mengembangkan model prediksi yang canggih untuk memproyeksikan peringkat ulasan hotel secara akurat. Model ini bertujuan untuk memberikan insight prediktif mengenai kualitas layanan dan fasilitas hotel, berdasarkan data ulasan yang ada.
2. Mengidentifikasi pola dalam ulasan negatif yang diberikan kepada hotel. Tujuan ini mencakup analisis mendalam terhadap ulasan negatif untuk menemukan tren atau isu umum yang sering disebut, yang kemudian dapat dijadikan dasar untuk perbaikan layanan hotel.
3. Menyelidiki bagaimana informasi yang dihasilkan dari model prediksi dapat digunakan secara strategis untuk meningkatkan fokus dan kualitas layanan di sektor perhotelan. Ini termasuk penerapan insight dari model dalam praktik manajemen dan operasional hotel.
4. Menilai efektivitas Latent Dirichlet Allocation (LDA) Topic Modeling dalam mengidentifikasi dan memahami topik utama dari ulasan hotel di TripAdvisor. Penelitian ini akan fokus pada keakuratan dan relevansi topik yang dihasilkan oleh LDA dalam konteks ulasan hotel.

## 1.4 Manfaat

1. Mengkaji dampak implementasi informasi model dalam praktik layanan sehari-hari, termasuk penilaian perubahan dalam kepuasan pelanggan dan kinerja operasional hotel pasca-implementasi informasi model.
2. Menganalisis sejauh mana Latent Dirichlet Allocation (LDA) Topic Modeling efektif dalam mengidentifikasi tema dan topik relevan dalam ulasan hotel. Tujuan ini meliputi evaluasi kemampuan LDA dalam mengklasifikasikan ulasan ke dalam kategori yang bermakna dan berguna untuk analisis lebih lanjut.
3. Menentukan bagaimana hasil LDA dapat digunakan untuk memahami kebutuhan dan keluhan pelanggan secara lebih akurat, serta mengevaluasi kontribusinya dalam memandu strategi peningkatan layanan hotel berdasarkan feedback yang diterima.

# BAB II

# Landasan Teori

## 2.1 Text Mining

Text mining, juga dikenal sebagai analisis teks atau penambangan teks, adalah bidang ilmu yang menggunakan teknik komputasional untuk mengekstraksi informasi yang berharga dari data teks. Ini mencakup sejumlah konsep dan metode dari berbagai disiplin ilmu, termasuk ilmu komputer, statistika, dan linguistik.

Teks yang digunakan dalam text mining seringkali bersifat tidak terstruktur karena berasal dari bahasa alami manusia, yang cenderung memiliki tingkat kompleksitas dan variasi yang tinggi. Berbeda dengan basis data yang didesain untuk pemrosesan otomatis oleh program komputer, teks ditulis dengan tujuan komunikasi langsung antara manusia. Hal ini menimbulkan tantangan dalam ekstraksi informasi dan penemuan pola dari teks. Dalam konteks ini, Herast (2003) menyatakan bahwa teks lebih terfokus pada aspek humanistik dan kekreatifan, sementara basis data lebih terkait dengan kebutuhan otomatisasi dan pengolahan data oleh komputer. Berikut ada beberapa konsep landasan teori dalam text mining :

1. **Case Folding dan Tokenizing**

   Case Folding, juga dikenal sebagai penyeragaman kata, adalah proses mengubah seluruh kata menjadi huruf kecil (lowercase). Hanya karakter huruf a sampai z yang diterima, sementara karakter selain huruf dihilangkan. Meskipun demikian, beberapa kata khusus, seperti lembaga atau institusi yang selalu diawali huruf kapital, serta nama gelar seperti ST, M.Psi, dan sejenisnya, tetap mempertahankan penulisan awalnya. Praktik Case Folding sangat tergantung pada sumber data yang digunakan untuk diproses. Tokenizing, di sisi lain, merupakan tahap pemotongan string kata berdasarkan penyusunan kata tersebut.

2. **Filtering**

   Filtering melibatkan pengambilan kata-kata penting dari hasil Tokenizing atau pengeliminasi kata-kata sesuai dengan kaidah tertentu. Algoritma stop-word removal menjadi salah satu metode yang digunakan untuk tahapan filtering. Meskipun, perlu dicatat bahwa beberapa kata mungkin memiliki nilai signifikatif dalam konteks tertentu, dan pengambilan keputusan mengenai kata-kata yang dianggap "penting" dapat bervariasi tergantung pada kebutuhan analisis.

3. **Stemming**

   Stemming adalah proses memecah variasi kata menjadi kata dasar sesuai dengan aturan yang berlaku. Dalam konteks Bahasa Indonesia, di mana aturan Bahasa memiliki peran penting. Hal ini membantu mengurangi variasi dan menyederhanakan representasi kata, sehingga mempermudah analisis teks.

4. **Analyzing**
   Analyzing adalah tahapan di mana data teks yang telah diproses dianalisis untuk menentukan kemiripan antar dokumen teks. Salah satu metode yang umum digunakan dalam tahapan ini adalah cosine similarity. Metode ini mengukur kesamaan antara vektor representasi dokumen dalam ruang fitur, memberikan insight tentang sejauh mana dokumen teks memiliki kesamaan atau perbedaan satu sama lain.

## 2.2 Topic Modeling

Topic modeling, sebagai salah satu metode unggul di bidang Text Mining, efektif dalam mengungkap informasi yang tersembunyi dan menemukan keterkaitan antar teks dalam suatu korpus (Jelodar, dkk., 2018). Secara sederhana, konsep topic modeling dapat dijelaskan sebagai langkah untuk mengelompokkan data teks berdasarkan topik tertentu, menggunakan pendekatan serupa dengan teknik clustering untuk mengelompokkan dokumen berdasarkan kesamaan. Kategorinya sebagai unsupervised learning karena tidak memerlukan label pada data.

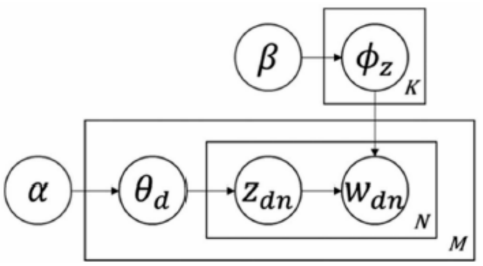
Dalam konsep dasarnya, topic modeling melibatkan tiga entitas utama, yaitu "kata," "dokumen," dan "corpora." "Kata" dianggap sebagai unit dasar dalam representasi data diskrit pada dokumen, diindeks sebagai elemen dari kosa kata untuk setiap kata unik dalam suatu dokumen. "Dokumen" adalah susunan N kata-kata, sementara corpus adalah kumpulan M dokumen, dan corpora merupakan bentuk jamak dari corpus. "Topik" merepresentasikan distribusi dari beberapa kosakata yang bersifat khusus. Secara sederhana, setiap dokumen dalam corpus memiliki proporsi sendiri dari topik-topik yang dibahas, sesuai dengan kata-kata yang terkandung di dalamnya.

Dalam dunia penelitian, topic modeling menarik perhatian para ilmuwan dari bidang Text Mining, Natural Language Processing, dan Machine Learning (Verma & Gahier, 2015). Dengan kemampuannya untuk mengidentifikasi pola dan topik yang tersembunyi dalam data teks, topic modeling menjadi alat yang sangat bermanfaat untuk analisis dan pemahaman konten teks secara lebih mendalam.

## 2.3 Latent Dirichlet Allocation

Salah satu pendekatan dalam Topic Modeling adalah dengan menggunakan metode Latent Dirichlet Allocation (LDA), yang diperkenalkan pertama kali oleh Blei, Ng, dan Jordan pada tahun 2003 dan telah menjadi metode yang sangat populer dalam pemodelan topik (Putra & Kusumawardani, 2017). LDA memiliki keunggulan yang memungkinkannya untuk merangkum, melakukan klasterisasi, menghubungkan, dan memproses data dalam skala besar. Metode ini menghasilkan daftar topik yang memiliki bobot tertentu untuk setiap dokumen. Hasil dari pengolahan data menggunakan LDA kemudian dapat dikenakan uji koherensi topik, suatu evaluasi yang bertujuan untuk mengevaluasi keterkaitan uraian probabilitas kata-kata dalam menyusun suatu topik, memberikan pemahaman yang lebih mendalam terhadap struktur dan signifikansi topik yang dihasilkan.

Dengan demikian, LDA tidak hanya menjadi alat yang efektif dalam mengelompokkan dokumen berdasarkan topik, tetapi juga memberikan wawasan yang kaya terkait struktur dan makna dari topik-topik tersebut melalui uji koherensi. Kombinasi dari pengantar topic modeling dan penerapan LDA memberikan landasan teoritis dan praktis yang kokoh untuk analisis yang mendalam terhadap data teks.




## 2.4 Coherence Model

Dalam penelitian ini, Topic Coherence digunakan sebagai metrik untuk mengevaluasi hasil dari proses topic modeling yang diterapkan pada data Tripadvisor, menggunakan metode Latent Dirichlet Allocation (LDA). Topic Coherence memberikan suatu ukuran yang akurat dalam menilai kualitas model topik yang dihasilkan (Annisa et al., 2019). Metrik ini mengukur tingkat kesamaan semantik antar kata-kata yang membentuk suatu topik, sebagaimana dijelaskan oleh Habibi et al. (2021). Dengan kata lain, Topic Coherence menciptakan penilaian terhadap sejauh mana kata-kata dalam suatu topik memiliki hubungan semantik yang kuat.

Menurut Habibi et al. (2021), penggunaan Topic Coherence sebagai metode evaluasi model topik merupakan suatu alternatif yang dapat memberikan analisis yang mendekati penilaian manusia, bahkan mungkin lebih tinggi. Syed and Spruit (2018) juga mengindikasikan bahwa Topic Coherence diusulkan sebagai pendekatan kualitatif untuk secara otomatis mengungkap koherensi dari suatu topik. Pendekatan pengukuran Topic Coherence ini bertujuan untuk menemukan metrik yang memiliki korelasi tinggi dengan penilaian topik yang dilakukan oleh manusia secara manual.

# BAB III

# METHODOLOGY

## 3.1 Dataset Description

Dataset ini merupakan kumpulan data yang diambil dari Kaggle, terdiri dari 20.000 ulasan hotel yang dikumpulkan dari platform TripAdvisor. Dalam dataset ini, terdapat dua kolom utama, yaitu "review" dan "rating". Kolom "review" memuat teks lengkap dari 20.000 ulasan hotel, mencerminkan berbagai pendapat dan pengalaman pelanggan selama menginap. Informasi ini mencakup aspek-aspek seperti pelayanan, fasilitas, dan suasana hotel, memberikan wawasan mendalam untuk analisis lebih lanjut. Sementara itu, kolom "rating" memberikan penilaian pelanggan dalam skala 1 hingga 5, menggambarkan tingkat kepuasan atau ketidakpuasan mereka.

Table I. Dataset Description

| **Attribute** | **Description** |
|---|---|
| Review | Dataset mencakup teks lengkap dari 20.000 ulasan hotel yang mencerminkan pendapat dan pengalaman pelanggan. |
| Rating | Terdapat penilaian dari skala 1-5 yang diberikan pelanggan untuk review hotel |

## 3.2 Exploratory Data Analysis

Project ini dimulai dengan proses mengimport dataset dan libraries ke dalam python, project ini menggunakan Google Colab sebagai platform. Tahapan Exploratory data analysis dimulai dari pengecekan missing value, pengecekan duplicates, mengitung token, mengecek anomalies, rata-rata word count dari rating.

### 3.2.1 Load Libraries and Read Files

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# read file
df = pd.read_csv('tripadvisor_hotel_reviews.csv')
df
```

### 3.2.2 Check Missing Value

```
# check missing value
df.isnull().sum()
```

Pada data kami tidak memiliki missing value dengan ini bisa mempermudah analisis statistik dari data tersebut.

### 3.2.3 Check Duplicated Data

```
# check duplicated data
print("Duplicated data:", df.duplicated().sum())
print("Duplicated text:", df['Review'].duplicated().sum())
```

Dalam dataset kami, tidak ada duplikasi dalam data.

### 3.2.4 Comparison of Rating Frequencies from Customers

```
# rating freqeuencies
plt.figure(figsize=(8, 5))
sns.countplot(x='Rating', data=df, palette='RdYlGn_r', order=df['Rating'].value_counts().index)
plt.title('Comparison of Rating Frequencies from Customers')
plt.tight_layout()
plt.show()
```

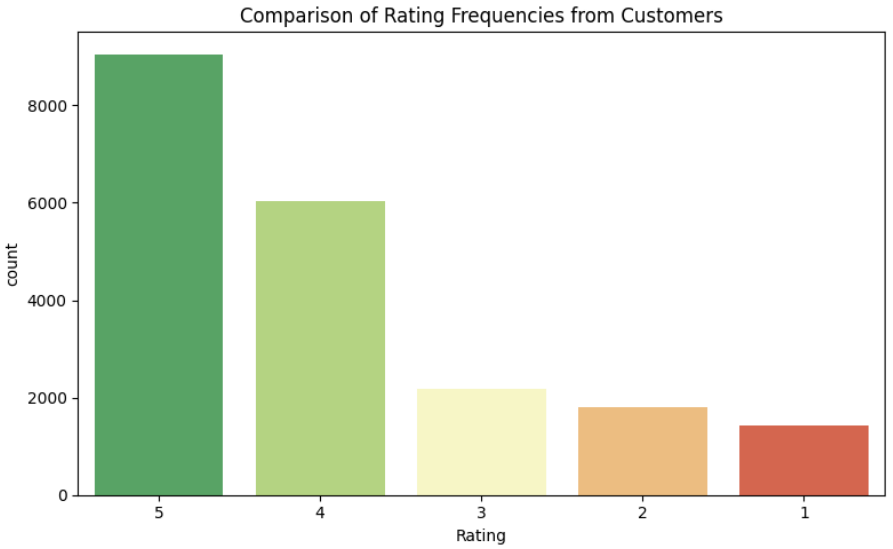



Image 1. Comparison of Rating Frequencies from Customers

Image I menunjukkan perbandingan frekuensi peringkat dari pelanggan, diurutkan mulai dari peringkat 5 (terbesar) hingga peringkat 1 (terkecil). Grafik ini memberikan gambaran visual tentang seberapa sering setiap peringkat diberikan oleh pelanggan, memungkinkan pemahaman cepat tentang preferensi dan kepuasan umum yang diekspresikan dalam ulasan.

Perhatikan bahwa sumbu vertikal mencerminkan frekuensi, sementara sumbu horizontal menunjukkan peringkat ulasan dari 5 hingga 1.

### 3.2.5 Count Token

```
# count token
df['Length'] = df['Review'].apply(len)
df.head()
```

| | Review | Rating | Length |
|---|---|---|---|
| 0 | nice hotel expensive parking got good deal sta... | 4 | 593 |
| 1 | ok nothing special charge diamond member hilto... | 2 | 1689 |
| 2 | nice rooms not 4* experience hotel monaco seat... | 3 | 1427 |
| 3 | unique, great stay, wonderful time hotel monac... | 5 | 600 |
| 4 | great stay great stay, went seahawk game aweso... | 5 | 1281 |

Count Token untuk menganalisis teks dengan bertujuan menghitung jumlah token dalam setiap dokumen. Pada gambar diatas terdapat length yang bervariasi tergantung barisnya.

### 3.2.6 Check Anomalies

```
# check anomalies
plt.figure(figsize=(8, 5))
sns.boxplot(x=df['Length'])
plt.title('Inconsistencies in the Word Count of Every Review')
plt.xlabel('Word Count')
plt.show()
```

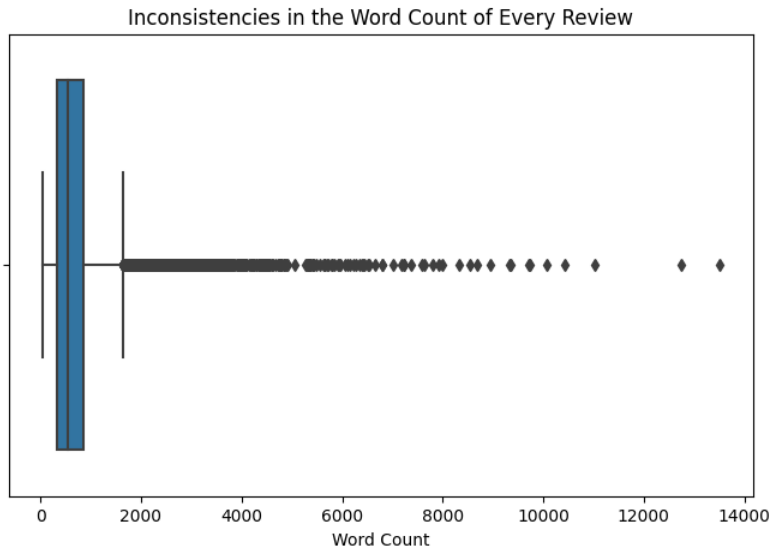



Image 2. Inconsistencies in the word Count of Every Review

Dalam gambar tersebut menjelaskan bahwa ketidaksesuaian dalam jumlah kata dari setiap ulasan. Gambar tersebut memberikan visualisasi tentang variasi panjang ulasan, menyoroti perbedaan dalam jumlah kata antara satu ulasan dan ulasan lainnya. Pada grafik ini, sumbu vertikal mewakili jumlah kata dalam ulasan, sementara sumbu horizontal menunjukkan setiap ulasan secara individual. Analisis ini dapat membantu dalam memahami tingkat rinci atau kedalaman informasi yang disampaikan oleh pelanggan dalam ulasan mereka, serta mengidentifikasi pola ketidaksesuaian yang mungkin memerlukan perhatian khusus.

### 3.2.7 Average Length

```
# average length
average_length_by_rating = df.groupby('Rating')['Length'].mean().reset_index()

plt.figure(figsize=(8, 5))
sns.pointplot(x='Rating', y='Length', data=average_length_by_rating, color='blue')
plt.title('Average Word Count Based on Ratings')
plt.xlabel('Rating')
plt.ylabel('Word Count (avg)')
plt.show()
```

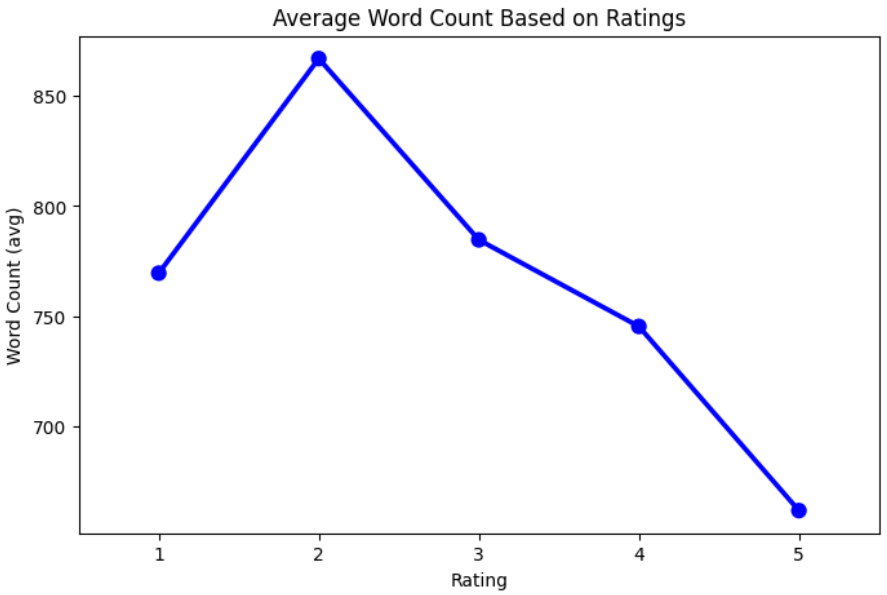



Image 3. Average Word Count Based on Ratings

Dari analisis rata-rata jumlah kata dalam ulasan berdasarkan tingkat peringkat pelanggan, tergambar pola menarik yang dapat memberikan wawasan mendalam tentang cara pelanggan menyampaikan ulasannya. Rating 1 menunjukkan bahwa pelanggan cenderung memberikan ulasan dengan rincian yang lebih tinggi, dengan rata-rata jumlah kata berkisar antara 750 hingga 800. Di sisi lain, rating 2 memiliki rata-rata jumlah kata paling tinggi, melebihi 850, menandakan bahwa pelanggan yang memberikan rating ini cenderung menyampaikan ulasan yang sangat rinci dan detail. Sementara rating 3 memiliki jumlah kata di atas rating 1, namun masih berada dalam rentang 750-800, mungkin mengindikasikan bahwa pelanggan memberikan feedback yang lebih ekstensif dibandingkan dengan rating 1, tetapi tidak sebanyak rating 2. Rating 4 menunjukkan kestabilan dengan rata-rata jumlah kata sekitar 750, menandakan bahwa pelanggan dengan kepuasan baik dapat menyampaikan ulasannya secara konsisten tanpa perlu memberikan detail yang berlebihan. Terakhir, rating 5 memiliki rata-rata jumlah kata di bawah 700, menunjukkan bahwa tingkat kepuasan yang tinggi dapat diungkapkan secara padat dan singkat. Analisis ini memberikan pemahaman lebih dalam tentang cara pelanggan mengartikan dan menyampaikan pengalaman mereka melalui ulasan, dengan variasi panjang ulasan yang mencerminkan tingkat kepuasan yang berbeda-beda.

### 3.2.8 Check Distribution

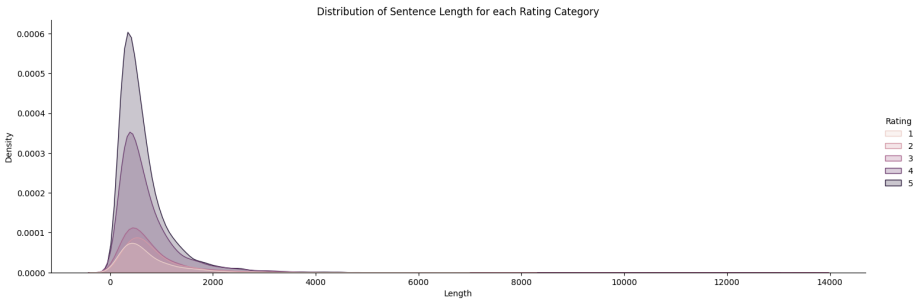



Image 4. Distribution of sentence Length for each Rating Category

Visualisasi distribusi panjang teks untuk setiap kategori rating menunjukan variasi yang signifikan, dengan panjang teks yang paling banyak berkisar antara 0 - 1500 karakter. Panjang teks tersebut memberikan gambaran tentang variasi dalam ekspresi pelanggan terhadap pengalaman mereka.  Panjang teks yang melibatkan rentang tersebut mungkin mencerminkan keragaman jenis ulasan, mulai dari tanggapan singkat hingga ulasan yang lebih rinci. Dalam konteks ulasan hotel, panjang teks yang bervariasi dapat mencerminkan kompleksitas pengalaman pelanggan, dimana beberapa mungkin hanya memberikan kesan umum semetara dan yang lain merinci detail tertentu seperti fasilitas, pelayanan, atau lokasi.

## 3.3 Text Preprocessing

Text preprocessing adalah langkah penting dalam mempersiapkan data teks mentah untuk analisis. Dalam proses ini, beberapa tugas kunci dilakukan untuk meningkatkan kualitas dan konsistensi teks. Awalnya, semua teks dikonversi menjadi huruf kecil untuk memastikan keseragaman. Selanjutnya, teks di-tokenisasi, dibagi menjadi kata-kata atau token. Tanda

baca kemudian dihapus untuk fokus pada konten alfabet. Kata-kata umum, dikenal sebagai stopwords (contohnya "the," "dan"), dikecualikan karena seringkali memiliki makna semantik yang sedikit. Selain itu, stemming atau lemmatization diterapkan untuk mengurangi kata-kata menjadi bentuk dasar atau akar, membantu dalam standarisasi. Langkah-langkah preprocessing ini bersama-sama menciptakan korpus teks yang disempurnakan dan terstandarisasi, lebih mudah untuk analisis atau pemodelan yang bermakna.

### 3.3.1 Import Libraries

```
import string
import re
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
nltk.download('stopwords')
nltk.download('omw-1.4')
nltk.download('wordnet')
nltk.download('punkt')
```

### 3.3.2 Cleaning text

```
def clean_text(text):
  text = text.lower() # lowercase
  text = re.sub(r'[\W_]+', ' ', text) # remove punctuation
  text = re.sub(r'\d+', '', text) # remove numbers
  text = re.sub(r'(.)\1\1+', r"\1\1", text) # reduce excessive character
  text = text.replace("\n", " ") # remove newlines
  text = re.sub(r'\s+', ' ', text) # remove multiple whitespaces
  return text

def preprocess_text(text):
  tokens = word_tokenize(text) # tokenization
  stop_words = set(stopwords.words('english')) # handle stop words
  tokens = [w for w in tokens if w.lower() not in stop_words]
  tokens = [w for w in tokens if len(w) > 2] # noise removal
  lemmatizer = WordNetLemmatizer() # lemmatization
  tokens = [lemmatizer.lemmatize(w) for w in tokens]
  text = ' '.join(tokens)
  return text
```

Dalam tahap preprocessing teks, sejumlah langkah kritis dilakukan untuk memastikan bahwa teks yang akan diolah dapat memberikan hasil analisis yang lebih baik. Pertama, dilakukan lower casing untuk mengubah semua huruf dalam teks menjadi huruf kecil guna memastikan konsistensi dan mengurangi kompleksitas. Selanjutnya, tanda baca seperti titik, koma, dan tanda tanya dihapus dari teks untuk mencegahnya mengganggu analisis dan menjaga fokus pada kata-kata kunci.

Selain itu, langkah-langkah seperti penghapusan angka dilakukan untuk mengurangi noise dalam teks, sedangkan reduksi karakter berlebih membantu menghilangkan varian yang tidak perlu. Pembersihan dari karakter newline dan penghapusan spasi berturut-turut juga dilakukan untuk memastikan konsistensi dalam format dan memudahkan proses tokenisasi.

Proses tokenisasi sendiri adalah langkah penting yang memecah teks menjadi token atau unit kecil, seperti kata atau frasa. Tokenisasi memungkinkan analisis lebih lanjut pada tingkat kata, membuat teks dapat diolah secara lebih terperinci. Selanjutnya, pengelolaan stopwords dilakukan untuk menghilangkan kata-kata yang tidak memberikan kontribusi signifikan pada makna teks, sehingga meningkatkan fokus pada kata-kata penting.

Dalam upaya menghilangkan gangguan atau elemen yang tidak diperlukan dalam analisis, noise removal dilakukan, seperti penghapusan karakter khusus atau informasi yang tidak relevan. Akhirnya, proses lemmatization mengubah kata-kata ke bentuk dasar atau kata dasar untuk memastikan variasi kata tidak mengganggu analisis. Penerapan langkah-langkah ini membantu menciptakan teks yang bersih, konsisten, dan siap untuk diolah dalam tahap analisis berikutnya, seperti ekstraksi topik menggunakan model seperti LDA. Setiap langkah memiliki peran penting dalam meningkatkan kualitas dan relevansi informasi yang dapat diperoleh dari teks tersebut.

### 3.3.3 Labelling Ratings

```
def label_rating(rating):
    if rating in [1, 2]:
        return 0
    elif rating == 3:
        return 1
    elif rating in [4, 5]:
        return 2

def label_name(label):
    if label == 0:
        return 'negative'
    elif label == 1:
        return 'neutral'
```

```
    elif label == 2:
        return 'positive'
```

```
# Create a new column 'rating_label' based on 'rating'
df['Rating_Label'] = df['Rating'].apply(lambda x: label_rating(x))

# Create a new column 'label_name' based on 'rating_label'
df['Label_Name'] = df['Rating_Label'].apply(lambda x: label_name(x))

pd.reset_option("max_colwidth")
df.head()
```

| | Review | Rating | Length | Rating_Label | Label_Name |
|---|---|---|---|---|---|
| 0 | nice hotel expensive parking got good deal sta... | 4 | 593 | 2 | positive |
| 1 | nothing special charge diamond member hilton d... | 2 | 1689 | 0 | negative |
| 2 | nice room experience hotel monaco seattle good... | 3 | 1427 | 1 | neutral |
| 3 | unique great stay wonderful time hotel monaco ... | 5 | 600 | 2 | positive |
| 4 | great stay great stay went seahawk game awesom... | 5 | 1281 | 2 | positive |

Image 5. Label Ratings

Tujuan dari proses label rating adalah memberikan pemahaman yang lebih dalam terhadap dataset dengan mengaitkan setiap nilai rating ulasan dengan label kategoris yang mencerminkan sentimen atau evaluasi umum terhadap produk atau layanan yang diulas.

- Jika label adalah 0, maka nama labelnya adalah 'negative'.
- Jika label adalah 1, maka nama labelnya adalah 'neutral'.
- Jika label adalah 2, maka nama labelnya adalah 'positive'.

### 3.3.4 Rating Distribution

```
# shows the count of each rating value
def plotdistribution(dataframe, rating_column, title, x_ticks=None):
    rating_counts = dataframe[rating_column].value_counts().sort_index()
    plt.bar(rating_counts.index, rating_counts)
    plt.xlabel('Rating')
    plt.ylabel('Count')
    plt.title(title)

    if x_ticks is not None:
        plt.xticks(x_ticks)

    plt.show()
```

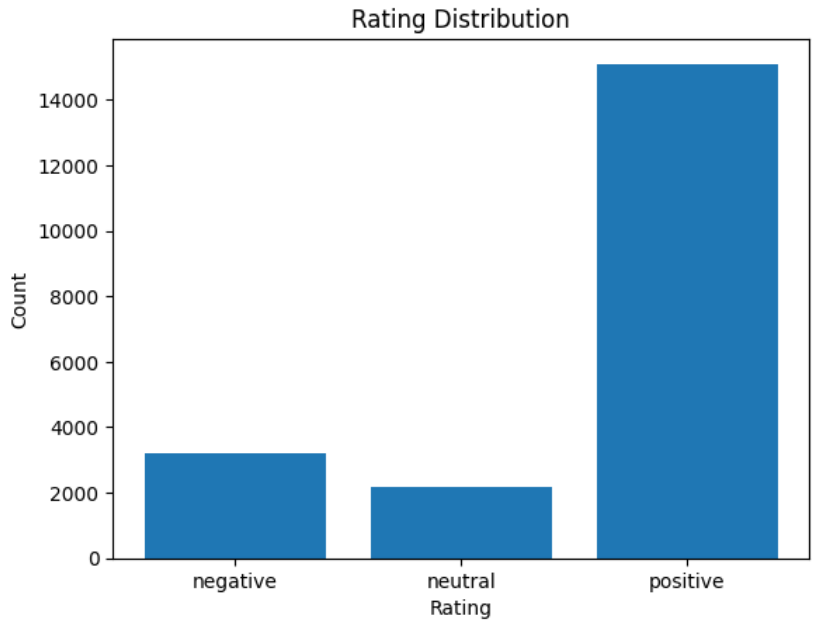



Image 6. Rating Distribution

Tujuan dari visualisasi distribusi nilai rating adalah memberikan gambaran yang komprehensif mengenai sebaran ulasan dalam kategori nilai tertentu. Dalam diagram tersebut, terlihat bahwa paling banyak terdapat rating positif, yaitu di atas 14.000, mengindikasikan bahwa mayoritas pengguna memberikan penilaian yang positif terhadap produk atau layanan. Sementara itu, distribusi untuk nilai netral terlihat lebih rendah dibandingkan dengan rating negatif. Analisis ini dapat memberikan wawasan yang berharga terkait dengan persepsi dan sentimen pengguna, serta menunjukkan kecenderungan dominan dalam penilaian yang dapat membimbing keputusan lebih lanjut terkait dengan perbaikan atau pengembangan produk atau layanan.

### 3.3.5 Wordcloud

```
from wordcloud import WordCloud
def wordcloud(data, title = None):
    wordcloud = WordCloud(
        background_color='black',
        max_words=150,
        max_font_size=50,
        scale=1,
        random_state=1
).generate(" ".join(data))

    fig = plt.figure(1, figsize=(15, 20))
    plt.axis('off')
    plt.imshow(wordcloud)
    plt.show()


wordcloud(df["Review"].values)
```


Image 7. Wordcloud

Langkah 3.3.5 melibatkan pembuatan Wordcloud, suatu visualisasi yang menarik dan informatif yang mewakili kata-kata yang paling sering muncul dalam ulasan produk atau layanan. Wordcloud dibentuk dengan mengonversi teks ulasan menjadi gambar, di mana ukuran dan intensitas kata-kata mencerminkan frekuensinya. Tujuan dari Wordcloud ini adalah untuk menyajikan gambaran visual yang cepat tentang kata-kata kunci atau tema utama yang sering muncul dalam ulasan.

Pada visualisasi ini, kata-kata yang lebih sering muncul akan memiliki ukuran yang lebih besar dan tampak lebih menonjol, sementara kata-kata yang muncul dengan frekuensi lebih rendah akan memiliki ukuran yang lebih kecil. Hal ini memberikan gambaran segera tentang fokus umum dari ulasan tersebut, memungkinkan analisis untuk mengidentifikasi aspek-aspek utama yang paling ditekankan oleh pengguna.

pada gambar wordcloud tersebut terdapat kata-kata seperti night, places, hotel, people, resort, nice, time, beach, good, room, place, area, restaurant, want, dan masih banyak lagi. Meskipun dalam word cloud terlihat beragam kata, terdapat kecenderungan netral, bahkan sebagian besar kata-kata bersifat positif.

Selain itu, apabila kita perhatikan histogram distribusi sentimen, tampak bahwa sentimen negatif hanya muncul dalam jumlah yang sedikit dibandingkan dengan sentimen positif yang mendominasi dengan frekuensi tertinggi. Hal ini menunjukkan bahwa umumnya, pandangan yang diungkapkan cenderung positif, dengan sedikit adanya sentimen negatif.

### 3.3.6 Tokenization

```
import pandas as pd
from nltk import word_tokenize  # You might need to install the nltk library: pip install nltk
```

```
# Define the tokenize function
def tokenize(text):
    return word_tokenize(text.lower())  # You can customize the tokenization process based on your needs

# Tokenize the 'Review' column
df["tokenized_review"] = df['Review'].apply(lambda x: tokenize(x))

# Calculate token count for each sentence
df["sent_token_length"] = df["tokenized_review"].apply(lambda x: len(x))

# Display the DataFrame
print(df)
```

Proses ini melibatkan tokenisasi, yaitu proses memecah teks menjadi unit kecil (token) seperti kata atau frasa. Fungsi word_tokenize dari pustaka Natural Language Toolkit (nltk) digunakan untuk melakukan tokenisasi pada kolom "Review". Hasilnya disimpan dalam kolom baru "tokenized_review", dan panjang token untuk setiap kalimat dihitung dan disimpan dalam kolom "sent_token_length". Tujuannya adalah mempersiapkan data untuk analisis statistik atau pembuatan model dengan menyajikan teks dalam bentuk token yang terpisah.

### 3.3.6.1 using BERT Tokenizer

```
# using BERT Tokenizer
from transformers import BertTokenizer
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased', do_lower_case=True)
```

```
# data tokenize with bert tokenizer
df["sent_bert_token_length"] = df["Review"].apply(lambda x: len(tokenizer.tokenize(x)))
```

```
# char count for each review
df['char_count'] = df['Review'].apply(lambda x: len(str(x)))

def plot_dist(df, feature, title):
    fig = plt.figure(constrained_layout=True, figsize=(18, 8))
    grid = gridspec.GridSpec(ncols=3, nrows=3, figure=fig)
    ax1 = fig.add_subplot(grid[0, :2])
    ax1.set_title('Histogram')
```

```
    sns.distplot(df.loc[:, feature],
                 hist=True,
                 kde=True,
                 ax=ax1,
                 )
    ax1.set(ylabel='Frequency')
    ax1.xaxis.set_major_locator(MaxNLocator(nbins=30))

    plt.suptitle(f'{title}', fontsize=24)
```

Langkah 3.3.6.1 menggunakan BERT Tokenizer untuk melakukan tokenisasi pada teks ulasan dalam kolom "Review". Jumlah token BERT yang dihasilkan untuk setiap ulasan disimpan dalam kolom "sent_bert_token_length". Selanjutnya, dihitung jumlah karakter untuk setiap ulasan dan disimpan dalam kolom "char_count". Tujuannya adalah memberikan gambaran mengenai kompleksitas teks dengan tokenisasi BERT dan menyajikan visualisasi distribusi panjang token serta jumlah karakter dalam dataset.

## 3.4 Modeling and Evaluation

Selama proses pemodelan, model yang digunakan untuk menganalisis dataset ini adalah Latent Dirichlet Allocation (LDA). Evaluasi kinerja model dilakukan dengan menggunakan Coherence Model dari library Gensim. Dua percobaan dilakukan, dengan percobaan pertama menggunakan model dengan 5 topik, dan percobaan kedua menggunakan model dengan hanya 3 topik. Hasil dari percobaan awal menunjukkan kinerja model yang masih kurang memuaskan. Sebagai upaya untuk meningkatkan performa model, dilakukan penyetelan parameter menggunakan metode grid search dengan rentang nilai dari 2 hingga 11 untuk jumlah topik yang mungkin. Setelah proses ini, ditemukan bahwa model terbaik mencapai performa optimal dengan menggunakan 8 topik. Hasil ini menunjukkan bahwa parameter melalui grid search berhasil meningkatkan kualitas model LDA, membuktikan keefektifan metode ini dalam mengoptimalkan analisis topik pada dataset ulasan hotel dari TripAdvisor.

### 3.4.1 Modeling 1

```
import numpy as np
import pyLDAvis
import pyLDAvis.gensim as gensimvis
import gensim

from gensim import corpora, models
from gensim.models import CoherenceModel, LdaModel
```

```
dictionary = corpora.Dictionary(reviews)
```

```
corpus = [dictionary.doc2bow(review) for review in reviews]

lda_model = LdaModel(corpus=corpus, id2word=dictionary, num_topics=5, random_state=42, passes=10)

for topic in lda_model.show_topics(num_topics=5):
    print('Topic', topic[0])
    print('Top words:', topic[1])
```

Dalam percobaan pertama dilakukan pemodelan topik menggunakan metode Latent Dirichlet Allocation (LDA) dengan menggunakan library Gensim. Pertama, dilakukan pembentukan dictionary dan corpus dari dataset ulasan ('reviews'). Dictionary dibuat untuk merepresentasikan kata-kata unik dalam dataset, sementara corpus adalah representasi vektor dokumen. Selanjutnya, model LDA dibentuk dengan menyertakan corpus, dictionary, jumlah topik (5 dalam kasus ini), nilai random_state, dan jumlah iterasi atau passes sebanyak 10.

```
coherence_model_lda = CoherenceModel(model=lda_model, texts=reviews, dictionary=dictionary, coherence='c_v')
coherence_lda = coherence_model_lda.get_coherence()

coherence_lda
```

**Hasil : 0.36315371515985817**

```
vis = gensimvis.prepare(lda_model, corpus, dictionary)
pyLDAvis.enable_notebook()
pyLDAvis.display(vis)
```

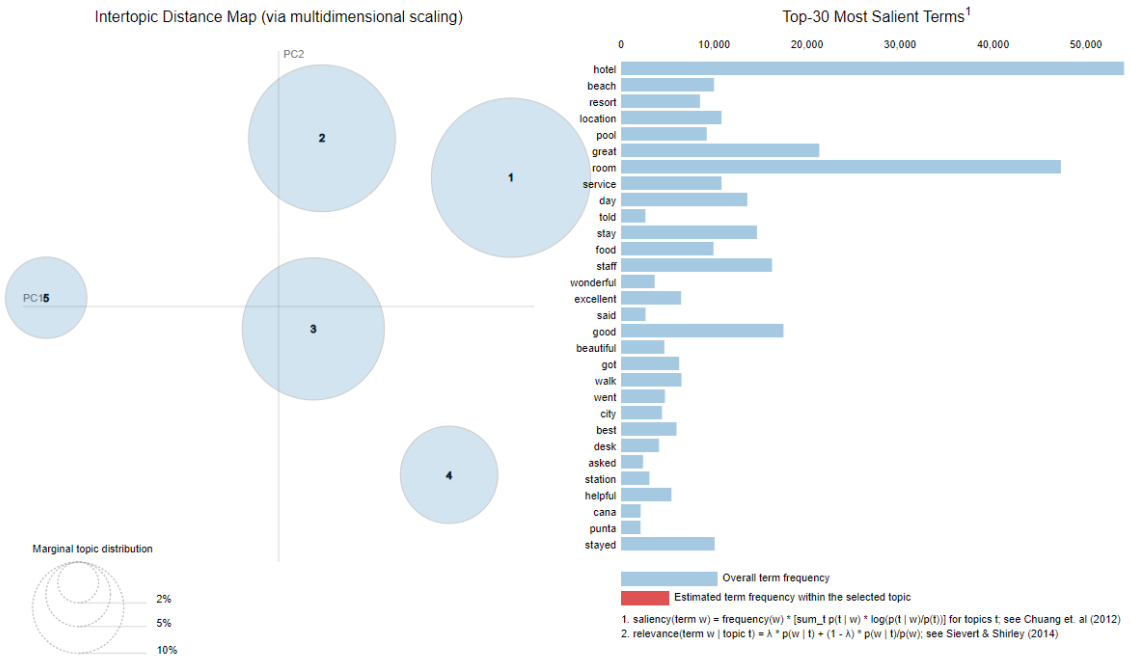



Image 8. Distance Map & Top 30 Most Salient Terms

**Output**

| | Sentence | Topic1 | Score1 | Topic2 | Score2 |
|---|---|---|---|---|---|
| 0 | nice hotel expensive parking got good deal sta... | 2 | 0.824494 | 1 | 0.153238 |
| 1 | nothing special charge diamond member hilton d... | 2 | 0.800261 | 4 | 0.113130 |
| 2 | nice room experience hotel monaco seattle good... | 2 | 0.871318 | 0 | 0.083890 |
| 3 | unique great stay wonderful time hotel monaco ... | 2 | 0.487465 | 1 | 0.474819 |
| 4 | great stay great stay went seahawk game awesom... | 2 | 0.356827 | 4 | 0.248631 |
| 5 | love monaco staff husband stayed hotel crazy w... | 2 | 0.375517 | 0 | 0.298446 |
| 6 | cozy stay rainy city husband spent night monac... | 1 | 0.711367 | 2 | 0.282257 |
| 7 | excellent staff housekeeping quality hotel cho... | 2 | 0.396998 | 1 | 0.325445 |
| 8 | hotel stayed hotel monaco cruise room generous... | 2 | 0.657541 | 1 | 0.247187 |
| 9 | excellent stayed hotel monaco past delight rec... | 1 | 0.737545 | 0 | 0.243809 |

Dari eksperimen awal, kami menerapkan model LDA untuk merancang model topik. Ketika kami mengevaluasi model dengan menggunakan metode kohesi, kami mencapai skor akurasi sebesar 0.36315371515985817. Skor ini mencerminkan tingkat kohesi atau keselarasan antar-topik dalam model yang berhasil dibentuk.

Namun, disayangkan bahwa hasil yang diperoleh masih belum memuaskan. Perlu dilakukan peningkatan lebih lanjut untuk meningkatkan kualitas model topik yang dihasilkan.

### 3.4.2 Modeling 2

```
dictionary = corpora.Dictionary(reviews)
corpus = [dictionary.doc2bow(review) for review in reviews]
lda_model2 = LdaModel(corpus=corpus, id2word=dictionary, num_topics=3, random_state=42, passes=10)

for topic in lda_model2.show_topics(num_topics=5):
    print('Topic', topic[0])
    print('Top words:', topic[1])
coherence_model_lda2 = CoherenceModel(model=lda_model2, texts=reviews, dictionary=dictionary, coherence='c_v')
coherence_lda2 = coherence_model_lda2.get_coherence()

coherence_lda2
```

Hasil : 0.3715842573258024

```
vis2 = gensimvis.prepare(lda_model2, corpus, dictionary)
pyLDAvis.enable_notebook()
pyLDAvis.display(vis2)
```

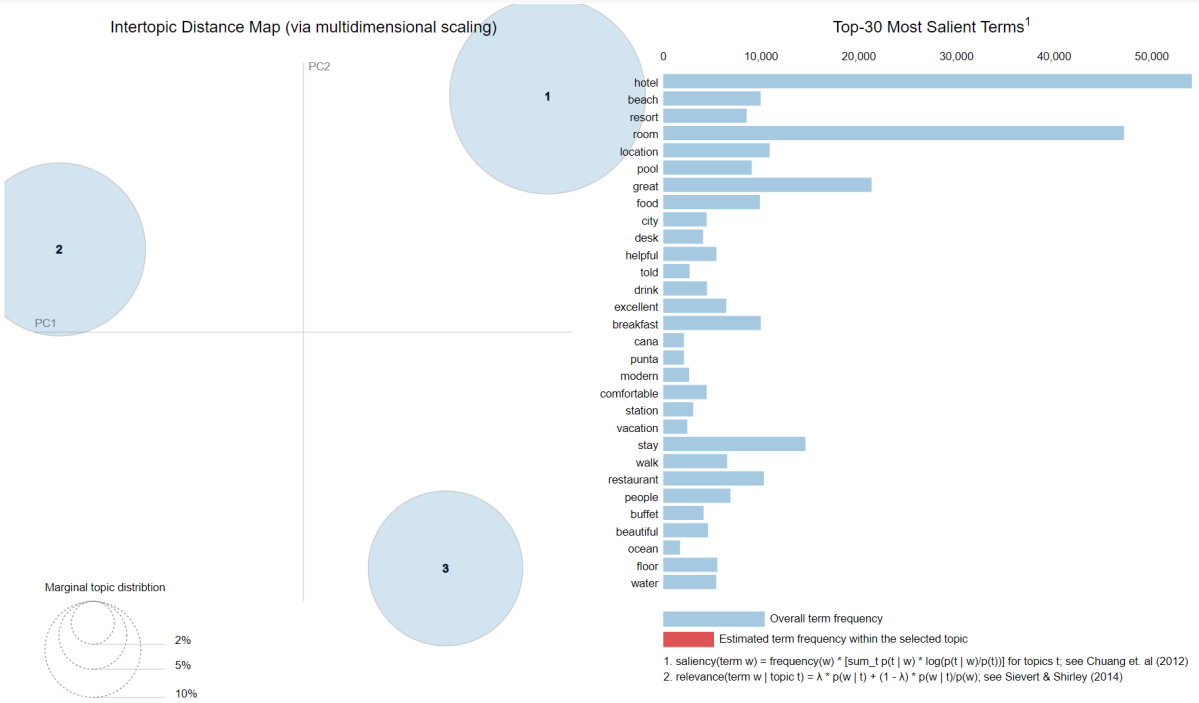



Image 9. Distance Map & Top 30 Most Salient Terms

```
test_shape = len(reviews)
topic1 = []
score1 = []
topic2 = []
score2 = []
sentence = []

for doc in range(test_shape):
    # Joining the tokenized words to form a sentence
    sentence.append(' '.join(reviews[doc]))

    # Get the topics and scores as strings for the given document
                        topics_and_scores      =      lda_model2.get_document_topics(corpus[doc],
minimum_probability=0.0)

    # Sort the topics based on probability and extract the top two topics and scores
    sorted_topics = sorted(topics_and_scores, key=lambda tup: -1*tup[1])[:2]

    # Extracting the top two topics and scores
    topic1.append(sorted_topics[0][0])
    score1.append(sorted_topics[0][1])

    topic2.append(sorted_topics[1][0])
    score2.append(sorted_topics[1][1])
```

```
data2 = {
    'Sentence': sentence,
    'Topic1': topic1,
    'Score1': score1,
    'Topic2': topic2,
    'Score2': score2
}

df2 = pd.DataFrame(data2)
df2.head(10)
```

**Output:**

| | Sentence | Topic1 | Score1 | Topic2 | Score2 |
|---|---|---|---|---|---|
| 0 | nice hotel expensive parking got good deal sta... | 2 | 0.583649 | 1 | 0.411772 |
| 1 | nothing special charge diamond member hilton d... | 2 | 0.831877 | 1 | 0.166372 |
| 2 | nice room experience hotel monaco seattle good... | 2 | 0.765523 | 1 | 0.227847 |
| 3 | unique great stay wonderful time hotel monaco ... | 1 | 0.687185 | 2 | 0.274859 |
| 4 | great stay great stay went seahawk game awesom... | 2 | 0.537129 | 1 | 0.280796 |
| 5 | love monaco staff husband stayed hotel crazy w... | 1 | 0.466896 | 2 | 0.342283 |
| 6 | cozy stay rainy city husband spent night monac... | 1 | 0.877579 | 2 | 0.118522 |
| 7 | excellent staff housekeeping quality hotel cho... | 1 | 0.497220 | 2 | 0.399133 |
| 8 | hotel stayed hotel monaco cruise room generous... | 1 | 0.615424 | 2 | 0.377961 |
| 9 | excellent stayed hotel monaco past delight rec... | 1 | 0.974169 | 2 | 0.014124 |

Image 10. Score & output LDA

Dalam percobaan kedua kami menggunakan metode yang sama yaitu LDA, tetapi sebagai perbandingan kami menggunakan tiga topik sebagai representasi utama dari kalimat yang muncul dalam ulasan tersebut. Hasil dari koherensi model sebesar 0.37 yang dimana menunjukan tingkat koherensi yang dapat diterima tetapi bisa lebih baik lagi. Kami melakukan analisis lebih lanjut untuk mengidentifikasi dua topik teratas beserta skor probabilitas masing - masing topik karena dalam setiap ulasan dikarenakan satu ulasan dapat mempunyai lebih dari satu topik atau *multilabel.* Contoh sepuluh baris pertama dari data frame menunjukan bagaimana setiap ulasan dikaitkan dengan dua topik utama beserta skornya dan memudahkan pemahaman pola - pola yang muncul. Bisa kita lihat bahwa hasil yang didapatkan cukup bagus pada sepuluh baris pertama dengan rata rata skor 0.67

### 3.4.3 Grid Search Tuning

```
from gensim.models import LdaModel
num_topics_range = list(range(2, 11))

# Function to calculate coherence score for a given number of topics
def calculate_coherence_score(corpus, dictionary, num_topics):
    lda_model = LdaModel(corpus, num_topics=num_topics, id2word=dictionary, passes=10, random_state=42)
                coherence_model = CoherenceModel(model=lda_model, texts=reviews, dictionary=dictionary, coherence='c_v')
    return coherence_model.get_coherence()
```

```
# Perform grid search
coherence_scores = []

for num_topics in num_topics_range:
    coherence_score = calculate_coherence_score(corpus, dictionary, num_topics)
    coherence_scores.append(coherence_score)

print("\nCoherence scores:", coherence_scores)

# Find the number of topics with the highest coherence score
best_coherence_score = max(coherence_scores)
best_num_topics = num_topics_range[np.argmax(coherence_scores)]

print("\nBest Coherence scores:", best_coherence_score)
print("Best number of topics:", best_num_topics)
```

```
Best Coherence scores: 0.49025794043523263
Best number of topics: 8
```

Image 11. Coherence score & number of topics

Dalam eksplorasi topik lebih lanjut menggunakan metode yang sama yaitu LDA, dilakukan penelusuran menggunakan grid search untuk menemukan jumlah topik optimal dimana mencari jumlah topik antara 2 hingga 10 dan diuji berdasarkan skor koherensi yang paling bagus. Hasil analisis menunjukan bahwa jumlah topik optimal untuk mencapai skor koherensi tertinggi adalah delapan topik dengan skor sebesar 0.49. Berdasarkan penemuan ini mendukung keputusan untuk menggunakan delapan topik sebagai representasi utama dari isu kalimat yang muncul dalam ulasan pelanggan.

### 3.4.4 Best Model

```
dictionary = corpora.Dictionary(reviews)
corpus = [dictionary.doc2bow(review) for review in reviews]
lda_model3 = LdaModel(corpus=corpus, id2word=dictionary, num_topics=8, random_state=42, passes=10)

for topic in lda_model3.show_topics(num_topics=8):
    print('Topic', topic[0])
    print('Top words:', topic[1])
```

```
coherence_model_lda3 = CoherenceModel(model=lda_model3, texts=reviews, dictionary=dictionary, coherence='c_v')
coherence_lda3 = coherence_model_lda3.get_coherence()

coherence_lda3
```

**Hasil : 0.49025794043523263**

```
vis3 = gensimvis.prepare(lda_model3, corpus, dictionary)
pyLDAvis.enable_notebook()
pyLDAvis.display(vis3)
```

Dalam menentukan model terbaik, kami menggunakan tuning grid search untuk menemukan prediksi topik yang optimal. Dalam konteks ini, kami mengatur rentang antara 2 hingga 11 untuk menemukan jumlah topik yang memberikan skor kohesi terbaik. Melalui proses ini, kami berhasil mendapatkan hasil terbaik dengan 8 topik, yang memiliki skor kohesi sebesar 0.49025794043523263. Skor kohesi ini mencerminkan rata-rata yang memuaskan, menandakan keberhasilan dalam menemukan model topik yang kohesif dan relevan dari data yang kami miliki.

### 3.4.5 Result Topic Name

| TOPIC NAME | | | | | | |
|---|---|---|---|---|---|---|
| | **Person 1** | **Person 2** | **Person 3** | | **RESULT** | |
| **Topic 0** | Luxurious | Hospitality | Experience | | **Topic 0** | **Facilities** |
| **Topic 1** | Highlights | Facilities | Hospitalities | | **Topic 1** | **Hospitality** |
| **Topic 2** | Room Service | Room Service | Room service | | **Topic 2** | **Room Service** |
| **Topic 3** | Resort | Resort | Resort | | **Topic 3** | **Resort** |
| **Topic 4** | Hotel Restaurant | Hotel Restaurant | Hotel Restaurant | | **Topic 4** | **Hotel Restaurant** |
| **Topic 5** | Strategic Location | City Exploration | Strategic location | | **Topic 5** | **Strategic Location** |
| **Topic 6** | Parking Area | Parking and Accessibility | Parking Area | | **Topic 6** | **Parking Area** |
| **Topic 7** | Adventures | Unique Experiences | Unique Experiences | | **Topic 7** | **Unique Experiences** |

Image 12. Topic Name

Dari nama topik diatas dapat disimpulkan bahwa kami menggunakan model LDA dengan 8 topik. Pengenalan nama topik kami lakukan berdasarkan kata-kata dengan skor tertinggi di setiap topik seperti berikut:

1. Topic 0 (Facilities):
    - Top Words: hotel, service, staff, great, pool, stay, time, wonderful, best, stayed.
    - Kesimpulan: Topik ini mencakup berbagai fasilitas hotel dan kualitas pelayanan, termasuk kolam renang dan pengalaman menginap yang luar biasa.

2. Topic 1 (Hospitality):
    - Top Words: hotel, room, great, location, good, staff, stay, breakfast, nice, clean.
    - Kesimpulan: Fokus utama topik ini tampaknya adalah pada keramahan staf, kualitas kamar, dan kebersihan hotel.
3. Topic 2 (Room Service):
    - Top Words: room, hotel, night, day, stay, bed, service, time, desk, floor.
    - Kesimpulan: Topik ini mungkin berbicara tentang pelayanan kamar, kenyamanan tempat tidur, dan pengalaman menginap secara umum.
4. Topic 3 (Resort):
    - Top Words: beach, resort, food, good, pool, day, time, great, people, restaurant.
    - Kesimpulan: Topik ini terfokus pada pengalaman menginap di resor, dengan penekanan pada pantai, makanan, dan kolam renang.
5. Topic 4 (Hotel Restaurant):
    - Top Words: fruit, coffee, breakfast, fresh, egg, juice, bread, cheese, cereal, hot.
    - Kesimpulan: Topik ini berkaitan dengan pengalaman sarapan dan makanan yang disajikan di restoran hotel.
6. Topic 5 (Strategic Location):
    - Top Words: san, car, westin, juan, boston, francisco, union, cable, bay, square.
    - Kesimpulan: Topik ini mungkin berbicara tentang lokasi strategis di beberapa kota, dengan menyebutkan beberapa nama tempat atau hotel.
7. Topic 6 (Parking Area):
    - Top Words: parking, car, free, valet, inn, downtown, block, street, area, park.
    - Kesimpulan: Topik ini terfokus pada pengalaman parkir, termasuk opsi parkir gratis dan valet, serta lokasi di pusat kota.
8. Topic 7 (Unique Experiences):
    - Top Words: anyways, disney, disneyland, stank, zero, thier, orientation, penthouse, repetitive, playground.
    - Kesimpulan: Topik ini tampaknya berkaitan dengan pengalaman unik atau situasi khusus, termasuk referensi ke Disney dan lokasi tertentu.

Dalam pemilihan nama topik, kita melibatkan interpretasi kata-kata kunci utama dari setiap topik dan mencoba memberikan label yang mencerminkan tema dominan atau fokus dari setiap topik. Nama topik tersebut diambil dari kata-kata dengan score terbanyak dalam setiap topik, sehingga mencerminkan esensi dari informasi yang diungkapkan oleh model LDA. Pemilihan nama topik juga dapat dipengaruhi oleh pemahaman domain dan konteks data yang lebih luas.

# BAB IV

# PENUTUP

## 4.1 Conclusion

Dalam penelitian ini, kami melakukan analisis perbandingan antara penilaian manusia dan hasil prediksi topik pada ulasan hotel. Hasilnya menunjukkan perbedaan signifikan antara keduanya. Kami mendapati bahwa label rating yang diberikan oleh pelanggan memiliki kesesuaian dengan hasil analisis sentimen berdasarkan penilaian manusia. Penyimpangan dalam hasil bisa disebabkan oleh jumlah input manusia yang terbatas dalam dataset dan potensi ketidakakuratan model dalam memahami konteks topik secara tepat. Analisis mendalam diperlukan untuk meningkatkan pemahaman tentang faktor-faktor yang mempengaruhi kinerja model dan hasil penilaian manusia.

Menggunakan teknik Latent Dirichlet Allocation (LDA) dan grid search, kami berhasil mengidentifikasi delapan topik relevan dalam ulasan hotel dari TripAdvisor. Model LDA, dengan skor koherensi 0.49, menunjukkan kinerja yang baik. Proses penelitian melibatkan pra-pemrosesan teks, pelabelan sentimen, dan analisis data eksploratif, yang memberikan dasar teoritis dan praktis untuk analisis teks yang komprehensif. Penelitian ini secara keseluruhan memberikan wawasan yang berharga mengenai apa yang dihargai oleh pelanggan, prediksi sentimen, dan faktor-faktor utama yang mempengaruhi penilaian terhadap hotel.

Analisis terhadap kata-kata teratas dalam setiap topik mengungkap tema-tema seperti pengalaman sarapan, lokasi strategis, dan fasilitas parkir. Penggunaan word cloud dan histogram distribusi sentimen menyoroti adanya kecenderungan sentimen positif dalam ulasan, mengindikasikan perspektif yang umumnya positif dari para pelanggan. Penelitian ini juga menjawab pertanyaan khusus tentang efektivitas LDA dalam memahami topik utama ulasan hotel di TripAdvisor dan bagaimana informasi dari model ini dapat digunakan untuk meningkatkan layanan hotel. Penggunaan teknik kesamaan kosinus dalam analisis teks dan penerapan stemming menggarisbawahi pendekatan analisis yang menyeluruh.

Kesimpulannya, penelitian ini memberikan kontribusi penting dalam memahami ulasan pelanggan dengan menggunakan teknik analisis data untuk mengungkap pola tersembunyi dalam ulasan, serta menyediakan informasi berharga bagi calon wisatawan dalam memilih akomodasi. Insight yang diperoleh diharapkan dapat membantu meningkatkan kepuasan pelanggan dan memperkuat proses pengambilan keputusan dalam industri perhotelan, serta memfasilitasi hotel dalam mengadaptasi dan mengoptimalkan strategi layanan mereka berdasarkan umpan balik langsung dari pelanggan.

# REFERENSI


Gaikwad, S. V., Chaugule, A., & Patil, P. (2014). Text mining methods and techniques. *International Journal of Computer Applications*, *85*(17).

Listari. (2019, 21 Juni). Topic Modeling Menggunakan Latent Dirichlet Allocation (Part 1): Pre-processing Data dengan Python. Medium. https://medium.com/@listari.tari/topic-modelling-menggunakan-latent-dirchlect-allocation-part-1-pre-processing-data-dengan-python-87bf5c580923

Tan, A. H. (1999, April). Text mining: The state of the art and the challenges. In *Proceedings of the pakdd 1999 workshop on knowledge disocovery from advanced databases* (Vol. 8, pp. 65-70).

Tong, Z., & Zhang, H. (2016, May). A text mining research based on LDA topic modelling. In *International conference on computer science, engineering and information technology* (pp. 201-210).

Vijayarani, S., Ilamathi, M. J., & Nithya, M. (2015). Preprocessing techniques for text mining-an overview. *International Journal of Computer Science & Communication Networks*, *5*(1), 7-16.