

# Analisis Data Zomato Menggunakan Python

Terakhir Diperbarui : 28 Jul, 2025

2.

Memahami preferensi pelanggan dan tren restoran penting untuk membuat keputusan bisnis yang tepat di industri makanan. Pada artikel ini, kita akan menganalisis kumpulan data restoran Zomato menggunakan Python untuk menemukan wawasan yang bermakna. Kami bertujuan untuk menjawab pertanyaan seperti:

- Apakah lebih banyak restoran yang menyediakan pengiriman online dibandingkan dengan layanan offline?
- Jenis restoran apa yang paling disukai oleh masyarakat umum?
- Berapa kisaran harga yang disukai pasangan untuk makan di luar?

## Implementasi untuk Analisis Data Zomato menggunakan Python.

Langkah-langkah di bawah ini diikuti untuk implementasinya.

### Langkah 1: Mengimpor pustaka Python yang diperlukan.

We will be using [Pandas](#), [Numpy](#), [Matplotlib](#) and [Seaborn](#) libraries.

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

### Langkah 2: Membuat bingkai data.

Anda dapat mengunduh himpunan data.

```
dataframe = pd.read_csv("/content/Zomato-data-.csv")
print(dataframe.head())
```

### Langkah 3: Pembersihan dan Persiapan Data

Sebelum melangkah lebih jauh kita perlu membersihkan dan mengolah data.

1. Konversi kolom tarif menjadi float dengan menghapus karakter penyebut.

- **dataframe['rate']=dataframe['rate'].apply(handleRate):**  
Menerapkan fungsi **handleRate** untuk membersihkan dan mengonversi setiap nilai rating di kolom 'rate'.

```
def handleRate(value):  
    value=str(value).split('/')  
    value=value[0];  
    return float(value)
```

```
dataframe['rate']=dataframe['rate'].apply(handleRate)  
print(dataframe.head())
```

2. Mendapatkan ringkasan penggunaan dataframe `df.info()`.

```
dataframe.info()
```

3. Memeriksa nilai yang hilang atau null untuk mengidentifikasi kesenjangan data.

```
print(dataframe.isnull().sum())
```

### Langkah 4: Menjelajahi Jenis Restoran

1. Mari kita lihat **kolom listed\_in (jenis)** untuk mengidentifikasi kategori restoran populer.

```
sns.countplot(x=dataframe['listed_in(type)'])  
plt.xlabel("Type of restaurant")
```

,

## 2. Informasi berdasarkan Jenis Restoran

Di sini kita mendapatkan hitungan suara untuk setiap kategori.

```
grouped_data = dataframe.groupby('listed_in(type)')['votes'].sum()
result = pd.DataFrame({'votes': grouped_data})
plt.plot(result, c='green', marker='o')
plt.xlabel('Type of restaurant')
plt.ylabel('Votes')
```

## Langkah 5: Identifikasi Restoran yang Paling Banyak Dipilih

Temukan restoran dengan jumlah suara tertinggi.

```
max_votes = dataframe['votes'].max()
restaurant_with_max_votes = dataframe.loc[dataframe['votes'] ==
max_votes, 'name']
```

```
print('Restaurant(s) with the maximum votes:')
print(restaurant_with_max_votes)
```

## Langkah 6: Ketersediaan Pesanan Online

Menjelajahi kolom online\_order untuk melihat berapa banyak restoran yang menerima pesanan online.

```
sns.countplot(x=dataframe['online_order'])
```

## Langkah 7: Analisis Peringkat

Memeriksa distribusi peringkat dari **kolom** tarif.

```
plt.hist(dataframe['rate'],bins=5)
plt.title('Ratings Distribution')
plt.show()
```

## Langkah 8: Perkiraan Biaya untuk Pasangan

Analisis kolom **approx\_cost (untuk dua orang)** untuk menemukan kisaran harga yang disukai.

```
couple_data=dataframe['approx_cost(for two people)']  
sns.countplot(x=couple_data)
```

## Langkah 9: Perbandingan Peringkat - Pesanan Online vs Offline

Bandingkan peringkat antara restoran yang menerima pesanan online dan restoran yang tidak.

```
plt.figure(figsize = (6,6))  
sns.boxplot(x = 'online_order', y = 'rate', data = dataframe)
```

## Langkah 10: Preferensi Mode Pesanan berdasarkan Jenis Restoran

Temukan hubungan antara mode pesanan (**online\_order**) dan jenis restoran (**listed\_in(jenis)**).

- **pivot\_table = dataframe.pivot\_table(index='listed\_in(type)', columns='online\_order', aggfunc='size', fill\_value=0)**: Membuat tabel pivot yang menghitung restoran berdasarkan jenis dan ketersediaan pesanan online.

```
pivot_table = dataframe.pivot_table(index='listed_in(type)',  
columns='online_order', aggfunc='size', fill_value=0)  
sns.heatmap(pivot_table, annot=True, cmap='YlGnBu', fmt='d')  
plt.title('Heatmap')  
plt.xlabel('Online Order')  
plt.ylabel('Listed In (Type)')  
plt.show()
```

Dengan ini kita dapat mengatakan bahwa restoran makan terutama menerima pesanan offline sedangkan kafe terutama menerima pesanan online. Ini menunjukkan bahwa klien lebih suka melakukan pemesanan secara langsung di restoran tetapi lebih suka memesan online di kafe.