```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from google.colab import files
import uuid



# Membaca file CSV
path = ('/content/hotel_booking.csv')
data = pd.read_csv(path)


# Menampilkan 5 baris pertama untuk memeriksa data
print("5 Baris Pertama dari Dataset (Sebelum Pra-pemprosesan):")
print(data.head()) # Corrected: Using 'data' instead of 'df'
```

```
5 Baris Pertama dari Dataset (Sebelum Pra-pemprosesan):
          hotel  is_canceled  lead_time  arrival_date_year arrival_date_month  \
0  Resort Hotel          0.0      342.0             2015.0               July
1  Resort Hotel          0.0      737.0             2015.0               July
2  Resort Hotel          0.0        7.0             2015.0               July
3  Resort Hotel          0.0       13.0             2015.0               July
4  Resort Hotel          0.0       14.0             2015.0               July

   arrival_date_week_number  arrival_date_day_of_month  \
0                      27.0                        1.0
1                      27.0                        1.0
2                      27.0                        1.0
3                      27.0                        1.0
4                      27.0                        1.0

   stays_in_weekend_nights  stays_in_week_nights  adults  ...  customer_type  \
0                      0.0                   0.0     2.0  ...      Transient
1                      0.0                   0.0     2.0  ...      Transient
2                      0.0                   1.0     1.0  ...      Transient
3                      0.0                   1.0     1.0  ...      Transient
4                      0.0                   2.0     2.0  ...      Transient

    adr required_car_parking_spaces total_of_special_requests  \
0   0.0                         0.0                       0.0
1   0.0                         0.0                       0.0
2  75.0                         0.0                       0.0
3  75.0                         0.0                       0.0
4  98.0                         0.0                       1.0

  reservation_status reservation_status_date            name  \
0          Check-Out              2015-07-01   Ernest Barnes
1          Check-Out              2015-07-01    Andrea Baker
2          Check-Out              2015-07-02  Rebecca Parker
3          Check-Out              2015-07-02    Laura Murray
4          Check-Out              2015-07-03     Linda Hines

                         email  phone-number       credit_card
0  Ernest.Barnes31@outlook.com  669-792-1661  ************4322
1       Andrea_Baker94@aol.com  858-637-6955  ************9157
2   Rebecca_Parker@comcast.net  652-885-2745  ************3734
3            Laura_M@gmail.com  364-656-8427  ************5677
4           LHines@verizon.com  713-226-5883  ************5498

[5 rows x 36 columns]
```

```
# Menampilkan informasi dataset
print("\nInformasi Dataset (Sebelum Pra-pemrosesan):")
print(data.info())
```

```
Informasi Dataset (Sebelum Pra-pemrosesan):
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 89656 entries, 0 to 89655
Data columns (total 36 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   hotel                           89656 non-null  object
 1   is_canceled                     89655 non-null  float64
 2   lead_time                       89655 non-null  float64
 3   arrival_date_year               89655 non-null  float64
 4   arrival_date_month              89655 non-null  object
 5   arrival_date_week_number        89655 non-null  float64
 6   arrival_date_day_of_month       89655 non-null  float64
 7   stays_in_weekend_nights         89655 non-null  float64
 8   stays_in_week_nights            89655 non-null  float64
 9   adults                          89655 non-null  float64
 10  children                        89651 non-null  float64
 11  babies                          89655 non-null  float64
 12  meal                            89655 non-null  object
```

```
 13  country                         89168 non-null  object
 14  market_segment                  89655 non-null  object
 15  distribution_channel            89655 non-null  object
 16  is_repeated_guest               89655 non-null  float64
 17  previous_cancellations          89655 non-null  float64
 18  previous_bookings_not_canceled  89655 non-null  float64
 19  reserved_room_type              89655 non-null  object
 20  assigned_room_type              89655 non-null  object
 21  booking_changes                 89655 non-null  float64
 22  deposit_type                    89655 non-null  object
 23  agent                           76235 non-null  float64
 24  company                         5390 non-null   float64
 25  days_in_waiting_list            89655 non-null  float64
 26  customer_type                   89655 non-null  object
 27  adr                             89655 non-null  float64
 28  required_car_parking_spaces     89655 non-null  float64
 29  total_of_special_requests       89655 non-null  float64
 30  reservation_status              89655 non-null  object
 31  reservation_status_date         89655 non-null  object
 32  name                            89655 non-null  object
 33  email                           89655 non-null  object
 34  phone-number                    89655 non-null  object
 35  credit_card                     89655 non-null  object
dtypes: float64(20), object(16)
memory usage: 24.6+ MB
None
```

```
#  Menangani Missing Values
print("\nJumlah Nilai yang Hilang per Kolom (Sebelum):")
# Menggunakan 'data' alih-alih 'df'
print(data.isnull().sum())
```

```
Jumlah Nilai yang Hilang per Kolom (Sebelum):
hotel                               0
is_canceled                         1
lead_time                           1
arrival_date_year                   1
arrival_date_month                  1
arrival_date_week_number            1
arrival_date_day_of_month           1
stays_in_weekend_nights             1
stays_in_week_nights                1
adults                              1
children                            5
babies                              1
meal                                1
country                           488
market_segment                      1
distribution_channel                1
is_repeated_guest                   1
previous_cancellations              1
previous_bookings_not_canceled      1
reserved_room_type                  1
assigned_room_type                  1
booking_changes                     1
deposit_type                        1
agent                           13421
company                         84266
days_in_waiting_list                1
customer_type                       1
adr                                 1
required_car_parking_spaces         1
total_of_special_requests           1
reservation_status                  1
reservation_status_date             1
name                                1
email                               1
phone-number                        1
credit_card                         1
dtype: int64
```

```
# Mengisi missing values
data['children'].fillna(0, inplace=True)  # Mengisi NaN di 'children' dengan 0
data['country'].fillna('Unknown', inplace=True)  # Mengisi NaN di 'country' dengan 'Unknown'
data['agent'].fillna(0, inplace=True)  # Mengisi NaN di 'agent' dengan 0
data['company'].fillna(0, inplace=True)  # Mengisi NaN di 'company' dengan 0
```

```
<ipython-input-6-b0271925e803>:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained ass
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[co

  data['children'].fillna(0, inplace=True)  # Mengisi NaN di 'children' dengan 0
```

```
<ipython-input-6-b0271925e803>:3: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained ass
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[co


  data['country'].fillna('Unknown', inplace=True)  # Mengisi NaN di 'country' dengan 'Unknown'
<ipython-input-6-b0271925e803>:4: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained ass
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[co


  data['agent'].fillna(0, inplace=True)  # Mengisi NaN di 'agent' dengan 0
<ipython-input-6-b0271925e803>:5: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained ass
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[co


  data['company'].fillna(0, inplace=True)  # Mengisi NaN di 'company' dengan 0
```

```
# Memeriksa kembali missing values setelah pengisian
print("\nJumlah Nilai yang Hilang per Kolom (Setelah Pengisian):")
print(data.isnull().sum())
```

```
Jumlah Nilai yang Hilang per Kolom (Setelah Pengisian):
hotel                           0
is_canceled                     1
lead_time                       1
arrival_date_year               1
arrival_date_month              1
arrival_date_week_number        1
arrival_date_day_of_month       1
stays_in_weekend_nights         1
stays_in_week_nights            1
adults                          1
children                        0
babies                          1
meal                            1
country                         0
market_segment                  1
distribution_channel            1
is_repeated_guest               1
previous_cancellations          1
previous_bookings_not_canceled  1
reserved_room_type              1
assigned_room_type              1
booking_changes                 1
deposit_type                    1
agent                           0
company                         0
days_in_waiting_list            1
customer_type                   1
adr                             1
required_car_parking_spaces     1
total_of_special_requests       1
reservation_status              1
reservation_status_date         1
name                            1
email                           1
phone-number                    1
credit_card                     1
dtype: int64
```

```
# 2. Menangani Nilai yang Tidak Sesuai
# Memastikan 'children', 'babies', dan 'adults' tidak negatif
data['children'] = data['children'].apply(lambda x: max(0, x))
data['babies'] = data['babies'].apply(lambda x: max(0, x))
data['adults'] = data['adults'].apply(lambda x: max(0, x))
```

```
# Memastikan 'adr' (Average Daily Rate) tidak negatif
data['adr'] = data['adr'].apply(lambda x: max(0, x))
```

```
# Memeriksa apakah ada baris dengan 'adults', 'children', dan 'babies' semuanya 0
invalid_guests = data[(data['adults'] == 0) & (data['children'] == 0) & (data['babies'] == 0)]
if not invalid_guests.empty:
    print(f"\nDitemukan {len(invalid_guests)} baris dengan jumlah tamu tidak valid (adults, children, dan babies = 0). Menghapus baris i
    data = data[~((data['adults'] == 0) & (data['children'] == 0) & (data['babies'] == 0))]
```

```
Ditemukan 78 baris dengan jumlah tamu tidak valid (adults, children, dan babies = 0). Menghapus baris ini.
```

```
# Memastikan tipe data yang sesuai
data['children'] = data['children'].astype(int) # Changed df to data
data['babies'] = data['babies'].astype(int)   # Changed df to data
data['adults'] = data['adults'].astype(int)    # Changed df to data
data['reservation_status_date'] = pd.to_datetime(data['reservation_status_date']) # Changed df to data
```

```
# 3. Menghapus Duplikasi Data
print("\nJumlah Baris Duplikat (Sebelum):", data.duplicated().sum())
data = data.drop_duplicates()
print("Jumlah Baris Duplikat (Setelah):", data.duplicated().sum())
```

```
Jumlah Baris Duplikat (Sebelum): 0
Jumlah Baris Duplikat (Setelah): 0
```

```
# Menampilkan informasi dataset setelah pra-pemrosesan
print("\nInformasi Dataset (Setelah Pra-pemrosesan):")
# Mengganti 'df' dengan 'data'
print(data.info())
```

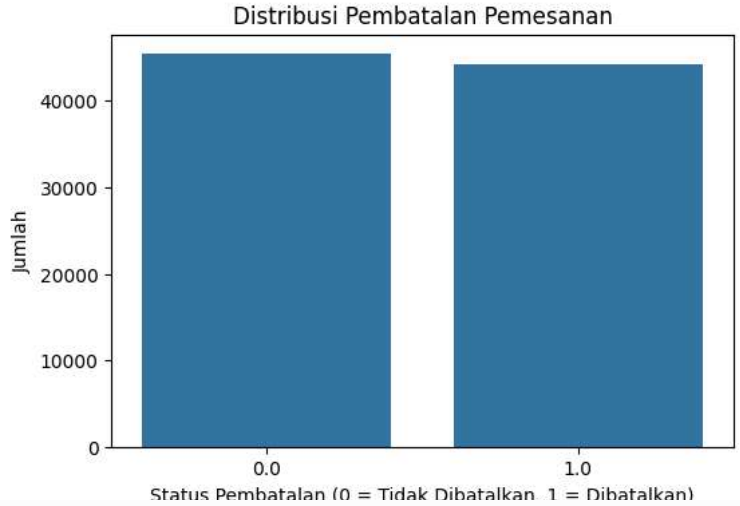
```
Informasi Dataset (Setelah Pra-pemrosesan):
<class 'pandas.core.frame.DataFrame'>
Index: 89578 entries, 0 to 89654
Data columns (total 36 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   hotel                           89578 non-null  object
 1   is_canceled                     89578 non-null  float64
 2   lead_time                       89578 non-null  float64
 3   arrival_date_year               89578 non-null  float64
 4   arrival_date_month              89578 non-null  object
 5   arrival_date_week_number        89578 non-null  float64
 6   arrival_date_day_of_month       89578 non-null  float64
 7   stays_in_weekend_nights         89578 non-null  float64
 8   stays_in_week_nights            89578 non-null  float64
 9   adults                          89578 non-null  int64
 10  children                        89578 non-null  int64
 11  babies                          89578 non-null  int64
 12  meal                            89578 non-null  object
 13  country                         89578 non-null  object
 14  market_segment                  89578 non-null  object
 15  distribution_channel            89578 non-null  object
 16  is_repeated_guest               89578 non-null  float64
 17  previous_cancellations          89578 non-null  float64
 18  previous_bookings_not_canceled  89578 non-null  float64
 19  reserved_room_type              89578 non-null  object
 20  assigned_room_type              89578 non-null  object
 21  booking_changes                 89578 non-null  float64
 22  deposit_type                    89578 non-null  object
 23  agent                           89578 non-null  float64
 24  company                         89578 non-null  float64
 25  days_in_waiting_list            89578 non-null  float64
 26  customer_type                   89578 non-null  object
 27  adr                             89578 non-null  float64
 28  required_car_parking_spaces     89578 non-null  float64
 29  total_of_special_requests       89578 non-null  float64
 30  reservation_status              89578 non-null  object
 31  reservation_status_date         89578 non-null  datetime64[ns]
 32  name                            89578 non-null  object
 33  email                           89578 non-null  object
 34  phone-number                    89578 non-null  object
 35  credit_card                     89578 non-null  object
dtypes: datetime64[ns](1), float64(17), int64(3), object(15)
memory usage: 25.3+ MB
None
```

```
# Analisis 1: Distribusi Pembatalan
print("\nDistribusi Pembatalan:")
cancel_counts = data['is_canceled'].value_counts() # Changed df to data
print(cancel_counts)
```

```
Distribusi Pembatalan:
is_canceled
0.0    45416
1.0    44162
Name: count, dtype: int64
```

```
# Visualisasi distribusi pembatalan
plt.figure(figsize=(6, 4))
```

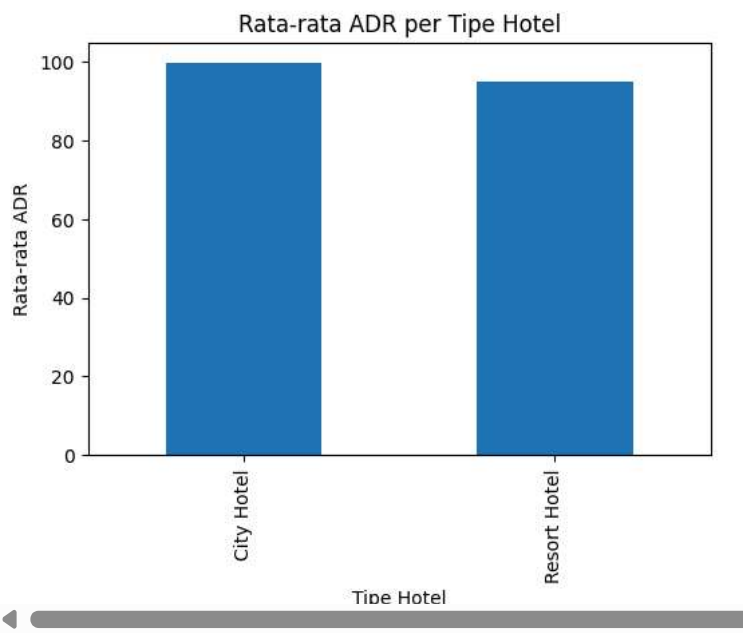
```
sns.countplot(x='is_canceled', data=data) # Changed df to data
plt.title('Distribusi Pembatalan Pemesanan')
plt.xlabel('Status Pembatalan (0 = Tidak Dibatalkan, 1 = Dibatalkan)')
plt.ylabel('Jumlah')
plt.savefig('cancellation_distribution.png')
plt.show()
```




```
# Analisis 2: Rata-rata ADR per Tipe Hotel
print("\nRata-rata ADR per Tipe Hotel:")
adr_by_hotel = data.groupby('hotel')['adr'].mean() # Mengganti 'df' dengan 'data'
print(adr_by_hotel)
```

```
Rata-rata ADR per Tipe Hotel:
hotel
City Hotel      99.827936
Resort Hotel    94.983213
Name: adr, dtype: float64
```

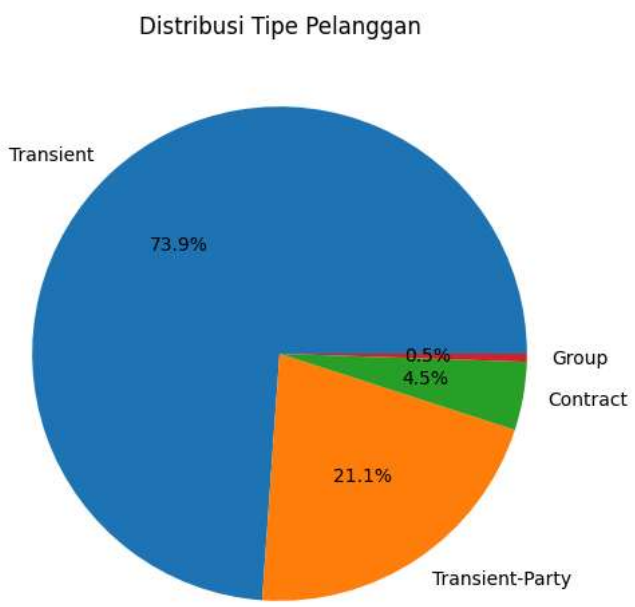
```
# Visualisasi rata-rata ADR per tipe hotel
plt.figure(figsize=(6, 4))
adr_by_hotel.plot(kind='bar')
plt.title('Rata-rata ADR per Tipe Hotel')
plt.xlabel('Tipe Hotel')
plt.ylabel('Rata-rata ADR')
plt.savefig('adr_by_hotel.png')
plt.show()
```




```
# Analisis 3: Distribusi Tipe Pelanggan
print("\nDistribusi Tipe Pelanggan:")
# Mengganti 'df' dengan 'data'
customer_type_counts = data['customer_type'].value_counts()
print(customer_type_counts)
```

```
Distribusi Tipe Pelanggan:
customer_type
Transient          66176
Transient-Party    18932
Contract            3998
Group                472
Name: count, dtype: int64
```

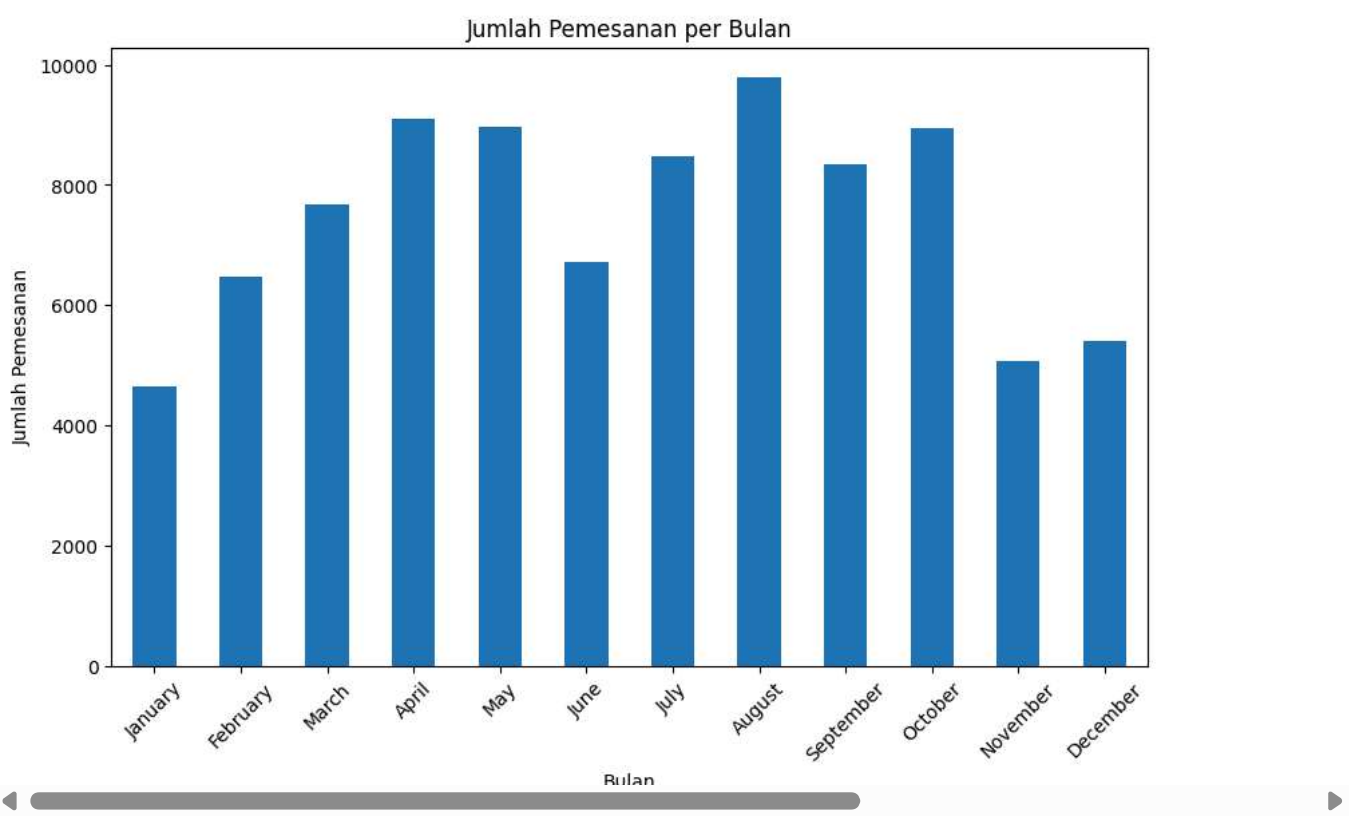
```
# Visualisasi distribusi tipe pelanggan
plt.figure(figsize=(8, 6))
customer_type_counts.plot(kind='pie', autopct='%1.1f%%')
plt.title('Distribusi Tipe Pelanggan')
plt.ylabel('')
plt.savefig('customer_type_distribution.png')
plt.show()
```




```
# Analisis 4: Jumlah Pemesanan per Bulan
print("\nJumlah Pemesanan per Bulan:")
data['arrival_date_month'] = pd.Categorical(data['arrival_date_month'], # Changed df to data
                                            categories=['January', 'February', 'March', 'April', 'May', 'June',
                                                        'July', 'August', 'September', 'October', 'November', 'December'],
                                            ordered=True)
monthly_bookings = data['arrival_date_month'].value_counts().sort_index() # Changed df to data
print(monthly_bookings)
```

```
Jumlah Pemesanan per Bulan:
arrival_date_month
January      4638
February     6469
March        7682
April        9099
May          8962
June         6721
July         8471
August       9800
September    8333
October      8937
November     5065
December     5401
Name: count, dtype: int64
```

```
# Visualisasi jumlah pemesanan per bulan
plt.figure(figsize=(10, 6))
monthly_bookings.plot(kind='bar')
plt.title('Jumlah Pemesanan per Bulan')
plt.xlabel('Bulan')
plt.ylabel('Jumlah Pemesanan')
plt.xticks(rotation=45)
plt.savefig('monthly_bookings.png')
plt.show()
```




```
import pandas as pd
import numpy as np # Import numpy to check for inf

# Membaca file CSV
# Change the path to the correct file name
path = '/content/hotel_booking.csv'
data = pd.read_csv(path)

# Mengisi NaN
data['country'].fillna('Unknown', inplace=True)
data['agent'].fillna(0, inplace=True)
data['company'].fillna(0, inplace=True)
data['children'].fillna(0, inplace=True) # Pastikan children diisi NaN sebelum diubah ke int

# Debugging: Check for NaN/inf before casting
print(f"NaN values in 'children' before casting: {data['children'].isnull().sum()}")
print(f"Inf values in 'children' before casting: {np.isinf(data['children']).sum()}")


# Mengubah tipe data
# Use errors='coerce' initially to see if conversion fails and results in NaN,
# then fill any new NaNs created by coercion.
# Or, try converting to a float first if the original data might be float with NaNs.

# Option 1: Convert to float first if original data is float with NaNs
# data['children'] = data['children'].astype(float) # Ensure it's float first if coming from CSV potentially as float with NaNs
# data['children'].fillna(0, inplace=True) # Fill again just in case (redundant if fillna above worked, but safe)
# data['children'] = data['children'].astype(int)

# Option 2: Use errors='coerce' (less ideal as it hides the original issue)
# data['children'] = pd.to_numeric(data['children'], errors='coerce')
# data['children'].fillna(0, inplace=True) # Fill any NaNs created by coercion
# data['children'] = data['children'].astype(int) # Now convert to int

# Let's stick to the original intent and re-verify the fillna logic
# data['children'] = data['children'].astype(int) # Original line causing error

# A safer approach might be to ensure the column is numeric first, handle NaNs, then cast to int
data['children'] = pd.to_numeric(data['children'], errors='coerce') # Ensure it's numeric, turn unparseable values into NaN
data['children'].fillna(0, inplace=True) # Fill any NaNs (original ones and those from coerce)
# Check again after filling from coerce
print(f"NaN values in 'children' after to_numeric and fillna: {data['children'].isnull().sum()}")
# Now, convert to int. If there are still non-integer-like floats, this might still error
# Or, if the original data type was problematic even for pd.to_numeric initially, though less likely.
# Let's assume fillna(0) on the numeric column should make it safe for int conversion.
data['children'] = data['children'].astype(int)


data['babies'] = pd.to_numeric(data['babies'], errors='coerce')
```

```
data['babies'].fillna(0, inplace=True)
data['babies'] = data['babies'].astype(int)

data['adults'] = pd.to_numeric(data['adults'], errors='coerce')
data['adults'].fillna(0, inplace=True)
data['adults'] = data['adults'].astype(int)


# Ensure reservation_status_date is also handled safely
data['reservation_status_date'] = pd.to_datetime(data['reservation_status_date'], errors='coerce')
# You might want to handle NaT values if errors='coerce' created them in the date column
# data['reservation_status_date'].fillna(some_default_date, inplace=True) # Example


# Menghapus baris dengan jumlah tamu tidak valid (188 baris)
# (Diasumsikan Anda telah menghapus baris di mana adults, children, dan babies = 0)
# If this step is needed again after reading the data, add it here.
# invalid_guests = data[(data['adults'] == 0) & (data['children'] == 0) & (data['babies'] == 0)]
# if not invalid_guests.empty:
#     print(f"\nDitemukan {len(invalid_guests)} baris dengan jumlah tamu tidak valid (adults, children, dan babies = 0). Menghapus baris
#     data = data[~((data['adults'] == 0) & (data['children'] == 0) & (data['babies'] == 0))]


# Menghapus duplikasi
data = data.drop_duplicates()
```

```
<ipython-input-26-bd3b688d4474>:10: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained as
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[co


  data['country'].fillna('Unknown', inplace=True)
<ipython-input-26-bd3b688d4474>:11: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained as
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[co


  data['agent'].fillna(0, inplace=True)
<ipython-input-26-bd3b688d4474>:12: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained as
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[co


  data['company'].fillna(0, inplace=True)
<ipython-input-26-bd3b688d4474>:13: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained as
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[co


  data['children'].fillna(0, inplace=True) # Pastikan children diisi NaN sebelum diubah ke int
<ipython-input-26-bd3b688d4474>:40: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained as
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[co


  data['children'].fillna(0, inplace=True) # Fill any NaNs (original ones and those from coerce)
<ipython-input-26-bd3b688d4474>:50: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained as
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[co


  data['babies'].fillna(0, inplace=True)
<ipython-input-26-bd3b688d4474>:54: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained as
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[co


  data['adults'].fillna(0, inplace=True)
NaN values in 'children' before casting: 0
Inf values in 'children' before casting: 0
NaN values in 'children' after to_numeric and fillna: 0
```

```
# Menyimpan dataset ke file CSV
data.to_csv('/content/el_booking_processed.csv', index=False)
print("File CSV telah disimpan sebagai: el_booking_processed.csv")
```

```
File CSV telah disimpan sebagai: el_booking_processed.csv
```

```
from google.colab import files

# Mengunduh file CSV
files.download('/content/el_booking_processed.csv')
```

```
print(data.info())
print(data.head())
```

```
 24  company                         119390 non-null  float64
 25  days_in_waiting_list            119390 non-null  int64
 26  customer_type                   119390 non-null  object
 27  adr                             119390 non-null  float64
 28  required_car_parking_spaces     119390 non-null  int64
 29  total_of_special_requests       119390 non-null  int64
 30  reservation_status              119390 non-null  object
 31  reservation_status_date         119390 non-null  datetime64[ns]
 32  name                            119390 non-null  object
 33  email                           119390 non-null  object
 34  phone-number                    119390 non-null  object
 35  credit_card                     119390 non-null  object
dtypes: datetime64[ns](1), float64(3), int64(17), object(15)
memory usage: 32.8+ MB
None
          hotel  is_canceled  lead_time  arrival_date_year arrival_date_month  \
0  Resort Hotel            0        342               2015               July
1  Resort Hotel            0        737               2015               July
2  Resort Hotel            0          7               2015               July
3  Resort Hotel            0         13               2015               July
4  Resort Hotel            0         14               2015               July

   arrival_date_week_number  arrival_date_day_of_month  \
0                        27                          1
1                        27                          1
2                        27                          1
3                        27                          1
4                        27                          1

   stays_in_weekend_nights  stays_in_week_nights  adults  ...  customer_type  \
0                        0                     0       2  ...      Transient
1                        0                     0       2  ...      Transient
2                        0                     1       1  ...      Transient
3                        0                     1       1  ...      Transient
4                        0                     2       2  ...      Transient

    adr required_car_parking_spaces total_of_special_requests  \
0   0.0                           0                         0
1   0.0                           0                         0
2  75.0                           0                         0
3  75.0                           0                         0
4  98.0                           0                         1

  reservation_status reservation_status_date            name  \
0          Check-Out              2015-07-01   Ernest Barnes
1          Check-Out              2015-07-01    Andrea Baker
2          Check-Out              2015-07-02  Rebecca Parker
3          Check-Out              2015-07-02    Laura Murray
4          Check-Out              2015-07-03     Linda Hines

                         email  phone-number       credit_card
0  Ernest.Barnes31@outlook.com  669-792-1661  ************4322
1       Andrea_Baker94@aol.com  858-637-6955  ************9157
2    Rebecca_Parker@comcast.net  652-885-2745  ************3734
3            Laura_M@gmail.com  364-656-8427  ************5677
4           LHines@verizon.com  713-226-5883  ************5498

[5 rows x 36 columns]
```