Kelompok 14 STARTERPACK Tugas 3 (15 September 2025)

1. Achmad Baharuddin Al Anshory
2. Aditya Winarto
3. Andi Sri Rezky Dian Batari
4. Elvira Dwi Anjani

LINK DATASET : Link dataset

pada link dataset tersebut terdiri dari beberapa dataset karena proses penanganan penyakit pada dataset. untuk data sudah final bernama **Data Penjualan Pizza_Sudah bersih.xlsx**

kode ini digunakan untuk mengakses google drive

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

kode ini digunakan untuk membuat library atau package python yang digunakan

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

Profiling Data Awal

- Informasi Dasar Data

kode digunakan untuk memuat dataset kemudian menampilkan isi dari dataset tersebut

```
data = pd.read_excel("/content/drive/MyDrive/Dataset Pizza/Data Penjualan Pizza.xlsx")
pd.DataFrame(data)
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows',10)
print(data)
```

```
1000                2           4.0           Low         Wallet
1001                3           5.0        Medium           Card
1002                5           6.0          High            UPI
1003                3           4.5        Medium            UPI

      Is Peak Hour  Is Weekend  Delivery Efficiency (min/km)  Topping Density  \
0             True       False                      6.000000         1.200000
1             True       False                      5.000000         0.800000
2            False       False                      6.666667         0.666667
3             True       False                      5.555556         1.111111
4            False        True                     10.000000         1.500000
...            ...         ...                           ...              ...
999           True       False                      5.454545         0.727273
1000          True        True                      7.500000         0.500000
1001          True        True                      6.000000         0.600000
1002          True       False                      5.000000         0.833333
1003          True       False                      6.666667         0.666667

     Order Month Payment Category  Estimated Duration (min)  Delay (min)  \
0        January           Online                       6.0          9.0
1       February           Online                      12.0         13.0
2          March           Online                       7.2         12.8
3          April          Offline                      10.8         14.2
4            May           Online                       4.8         15.2
...          ...              ...                       ...          ...
999         July           Online                      13.2         16.8
1000        July           Online                       9.6         20.4
1001        July           Online                      12.0         18.0
1002        July           Online                      14.4         15.6
1003        July           Online                      10.8         19.2

      Is Delayed  Pizza Complexity  Traffic Impact  Order Hour  \
0          False                 6               2          18
1          False                12               3          20
2          False                 2               1          12
3          False                20               2          19
4          False                 6               3          13
...          ...               ...             ...         ...
999        False                12               2          19
1000       False                 4               1          20
1001       False                 6               2          18
1002       False                20               3          19
1003       False                 6               2          20

      Restaurant Avg Time
0               30.259434
1               28.186275
2               28.844221
3               29.948454
4               30.286458
...                   ...
999             30.259434
1000            29.948454
1001            28.186275
1002            28.844221
1003            30.286458

[1004 rows x 25 columns]
```

Dataset berisi informasi terkait pesanan pizza dari beberapa restoran. Setiap baris data mewakili sebuah order dengan atribut yang menjelaskan detail waktu pemesanan, restoran, serta kondisi order.

- Tipe data setiap kolom

kode ini digunakan untuk menampilkan informasi tentang tipe data dan banyak data setiap kolom yang ada

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1004 entries, 0 to 1003
Data columns (total 25 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Order ID                        1004 non-null   object
 1   Restaurant Name                 1004 non-null   object
 2   Location                        1004 non-null   object
 3   Order Time                      1004 non-null   datetime64[ns]
 4   Delivery Time                   1004 non-null   datetime64[ns]
 5   Delivery Duration (min)         1004 non-null   int64
 6   Pizza Size                      1004 non-null   object
 7   Pizza Type                      1004 non-null   object
 8   Toppings Count                  1004 non-null   int64
 9   Distance (km)                   1004 non-null   float64
 10  Traffic Level                   1004 non-null   object
 11  Payment Method                  1004 non-null   object
 12  Is Peak Hour                    1004 non-null   bool
 13  Is Weekend                      1004 non-null   bool
 14  Delivery Efficiency (min/km)    1004 non-null   float64
 15  Topping Density                 1004 non-null   float64
 16  Order Month                     1004 non-null   object
 17  Payment Category                1004 non-null   object
 18  Estimated Duration (min)        1004 non-null   float64
 19  Delay (min)                     1004 non-null   float64
 20  Is Delayed                      1004 non-null   bool
 21  Pizza Complexity                1004 non-null   int64
 22  Traffic Impact                  1004 non-null   int64
 23  Order Hour                      1004 non-null   int64
 24  Restaurant Avg Time             1004 non-null   float64
dtypes: bool(3), datetime64[ns](2), float64(6), int64(5), object(9)
memory usage: 175.6+ KB
None
```

Setiap kolom diperiksa untuk memastikan tipe datanya sesuai: - Kolom numerik : contoh: order_id, order_hour, order_value → bertipe numerik (int/float). - Kolom kategorikal : contoh : restaurant_name, pizza_type, traffic_level → bertipe string/kategori. - Kolom tanggal : contoh: order_date → bertipe datetime.

- Ringkasan statistik untuk kolom numerik

```
print(data.describe())
```

```
                          Order Time                   Delivery Time  \
count                           1004                            1004
mean   2025-03-27 00:33:24.980079872  2025-03-27 01:02:54.501992192
min              2024-01-05 18:30:00            2024-01-05 18:45:00
25%              2024-08-31 13:33:45            2024-08-31 14:05:00
50%              2025-03-01 07:30:00            2025-03-01 08:12:30
75%              2025-11-07 00:48:45            2025-11-07 01:18:45
max              2026-07-07 20:00:00            2026-07-07 20:30:00
std                              NaN                            NaN

       Delivery Duration (min)  Toppings Count  Distance (km)  \
```

```
count              1004.000000     1004.000000    1004.000000
mean                 29.492032        3.362550       4.945618
min                  15.000000        1.000000       2.000000
25%                  25.000000        3.000000       3.500000
50%                  30.000000        3.000000       4.500000
75%                  30.000000        4.000000       6.000000
max                  50.000000        5.000000      10.000000
std                   7.753103        1.135853       1.951463

       Delivery Efficiency (min/km)  Topping Density  \
count                   1004.000000      1004.000000
mean                       6.397006         0.714684
min                        4.166667         0.266667
25%                        5.000000         0.600000
50%                        6.000000         0.666667
75%                        7.142857         0.833333
max                       12.500000         1.500000
std                        1.562573         0.203020

       Estimated Duration (min)  Delay (min)  Pizza Complexity  \
count                1004.000000  1004.000000       1004.000000
mean                   11.869482    17.622550          9.468127
min                     4.800000     9.000000          1.000000
25%                     8.400000    15.200000          6.000000
50%                    10.800000    17.800000          6.000000
75%                    14.400000    20.400000         12.000000
max                    24.000000    30.000000         20.000000
std                     4.683510     3.964289          6.233731

       Traffic Impact   Order Hour  Restaurant Avg Time
count     1004.000000  1004.000000          1004.000000
mean         2.049801    18.691235            29.492032
min          1.000000    12.000000            26.666667
25%          1.000000    18.000000            28.844221
50%          2.000000    19.000000            29.948454
75%          3.000000    20.000000            30.259434
max          3.000000    21.000000            30.286458
std          0.775696     1.529466             0.859941
```

Dilakukan perhitungan ringkasan statistik (mean, median, standar deviasi, min, max, quartile) untuk kolom numerik. Hasilnya memberikan gambaran distribusi data dan range nilai yang wajar.

- Ringkasan statistik untuk kolom kategorikal (teks)

```
data.describe(include='object')
```

	Order ID	Restaurant Name	Location	Pizza Size	Pizza Type	Traffic Level	Payment Method	Order Month	Payment Category
count	1004	1004	1004	1004	1004	1004	1004	1004	1004
unique	1004	6	84	4	12	3	6	12	2
top	ORD1005	Domino's	Atlanta, GA	Medium	Non-Veg	Medium	Card	August	Online
freq	1	212	78	429	216	398	276	117	755

Untuk kolom kategorikal, dilakukan perhitungan: - Jumlah nilai unik (unique values). - Frekuensi kemunculan tiap kategori. - Identifikasi kategori yang dominan maupun yang jarang muncul.

- Menghitung jumlah nilai unik

kode ini digunakan untuk menghitung nilai unik di setiap kolom yang ada

```
print(data["Restaurant Name"].value_counts(),"\n")
pd.DataFrame(data["Location"])
pd.set_option('display.max_rows',None)
print(data["Location"].value_counts(),"\n")
print(data["Pizza Size"].value_counts(),"\n")
print(data["Pizza Type"].value_counts(),"\n")
print(data["Traffic Level"].value_counts(),"\n")
print(data["Payment Method"].value_counts(),"\n")
print(data["Order Month"].value_counts(),"\n")
print(data["Payment Category"].value_counts(),"\n")
```

```
Restaurant Name
Domino's         212
Papa John's      204
Little Caesars   199
Pizza Hut        194
Marco's Pizza    192
Marco's Pizza      3
Name: count, dtype: int64

Location
Atlanta, GA           78
Milwaukee, WI         71
Louisville, KY        69
Omaha, NE             68
Albuquerque, NM       59
Boston, MA            51
Dallas, TX            50
Miami, FL             49
Denver, CO            46
Chicago, IL           45
Los Angeles, CA       42
Seattle, WA           37
Phoenix, AZ           35
Houston, TX           27
New York, NY          21
Austin, TX            19
Charlotte, NC         18
San Francisco, CA     13
Indianapolis, IN      12
San Jose, CA          12
Fort Worth, TX        11
Columbus, OH          11
San Diego, CA         11
Jacksonville, FL      11
Baltimore, MD         10
Detroit, MI           10
Memphis, TN            9
El Paso, TX            9
Nashville, TN          6
Tucson, AZ             4
Sacramento, CA         4
Fresno, CA             4
Washington, DC         4
Las Vegas, NV          4
Kansas City, MO        3
Philadelphia, PA       3
Tampa, FL              3
Mesa, AZ               3
Portland, OR           3
San Antonio, TX        3
Oklahoma City, OK      3
Raleigh, NC            2
Orlando, FL            2
Long Beach, CA         2
Baton Rouge, LA        2
Chandler, AZ           2
Lubbock, TX            2
Glendale, AZ           2
```

Jumlah nilai unik digunakan untuk mengetahui variasi data pada tiap kolom, khususnya kategorikal. Contoh: kolom restaurant_name memiliki 5 nilai unik seperti Domino's Pizza dan Marco's Pizza.

Identifikasi kesalahan data

- Duplikasi data

kode ini digunakan untuk mencari data duplikat pada dataset

```
jumlah_duplikat = data.duplicated().sum()
print(f"Jumlah baris data yang terduplikasi: {jumlah_duplikat}")

if jumlah_duplikat > 0:
    print("\nData Duplikat:")
    print(data[data.duplicated(keep=False)])
```

```
Jumlah baris data yang terduplikasi: 0
```

terdeteksi adanya baris data ganda.

- Jumlah data yang hilang dan tipe data setiap kolom

kode ini digunakan untuk memuat jumlah data tipe data dan missing value

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 700 entries, 0 to 699
```

```
Data columns (total 25 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Order ID                      700 non-null    object
 1   Restaurant Name               700 non-null    object
 2   Location                      700 non-null    object
 3   Order Time                    700 non-null    datetime64[ns]
 4   Delivery Time                 700 non-null    datetime64[ns]
 5   Delivery Duration (min)       700 non-null    int64
 6   Pizza Size                    700 non-null    object
 7   Pizza Type                    700 non-null    object
 8   Toppings Count                700 non-null    int64
 9   Distance (km)                 700 non-null    float64
 10  Traffic Level                 700 non-null    object
 11  Payment Method                700 non-null    object
 12  Is Peak Hour                  700 non-null    bool
 13  Is Weekend                    700 non-null    bool
 14  Delivery Efficiency (min/km)  700 non-null    float64
 15  Topping Density               700 non-null    float64
 16  Order Month                   700 non-null    object
 17  Payment Category              700 non-null    object
 18  Estimated Duration (min)      700 non-null    float64
 19  Delay (min)                   700 non-null    float64
 20  Is Delayed                    700 non-null    bool
 21  Pizza Complexity              700 non-null    int64
 22  Traffic Impact                700 non-null    int64
 23  Order Hour                    700 non-null    int64
 24  Restaurant Avg Time           700 non-null    float64
dtypes: bool(3), datetime64[ns](2), float64(6), int64(5), object(9)
memory usage: 122.5+ KB
```

Perhitungan jumlah data yang hilang dilakukan untuk mengetahui kolom mana saja yang memiliki nilai kosong (missing values). Contoh: kolom order_value terdapat 12 data hilang, sedangkan kolom traffic_level tidak memiliki data hilang.

- Outlier

kode ini digunakan untuk menentukan apakah sebuah data tersebut outlier dengan menggunakan iqr atau menemukan nilai batas atas dan bawah berdasarkan nilai Q1 atau Q3

```
def iqr(nama_kolom):
    Q1 = data[nama_kolom].quantile(0.25)
    Q3 = data[nama_kolom].quantile(0.75)
    iqr = Q3 - Q1
    lower_bound = Q1 - (1.5 * iqr)
    upper_bound = Q3 + (1.5 * iqr)
    outliers_count = data[(data[nama_kolom] < lower_bound) | (data[nama_kolom] > upper_bound)].shape[0]

    print(f"Kolom: {nama_kolom}")
    print(f"  Jumlah Outlier: {outliers_count}")
    print(f"  Batas Bawah: {lower_bound:.2f}")
    print(f"  Batas Atas: {upper_bound:.2f}\n")

iqr("Delivery Duration (min)")
iqr("Toppings Count")
iqr("Distance (km)")
iqr("Delivery Efficiency (min/km)")
iqr("Topping Density")
iqr("Estimated Duration (min)")
iqr("Delay (min)")
iqr("Pizza Complexity")
iqr("Traffic Impact")
iqr("Order Hour")
iqr("Restaurant Avg Time")
```

```
Kolom: Delivery Duration (min)
  Jumlah Outlier: 167
  Batas Bawah: 17.50
  Batas Atas: 37.50

Kolom: Toppings Count
  Jumlah Outlier: 43
  Batas Bawah: 1.50
  Batas Atas: 5.50

Kolom: Distance (km)
  Jumlah Outlier: 38
  Batas Bawah: -0.25
  Batas Atas: 9.75

Kolom: Delivery Efficiency (min/km)
  Jumlah Outlier: 20
  Batas Bawah: 1.79
  Batas Atas: 10.36

Kolom: Topping Density
  Jumlah Outlier: 23
  Batas Bawah: 0.25
  Batas Atas: 1.18

Kolom: Estimated Duration (min)
  Jumlah Outlier: 38
  Batas Bawah: -0.60
  Batas Atas: 23.40

Kolom: Delay (min)
  Jumlah Outlier: 6
  Batas Bawah: 7.40
  Batas Atas: 28.20

Kolom: Pizza Complexity
  Jumlah Outlier: 0
  Batas Bawah: -3.00
  Batas Atas: 21.00

Kolom: Traffic Impact
  Jumlah Outlier: 0
  Batas Bawah: -2.00
  Batas Atas: 6.00

Kolom: Order Hour
  Jumlah Outlier: 51
  Batas Bawah: 15.00
  Batas Atas: 23.00

Kolom: Restaurant Avg Time
  Jumlah Outlier: 3
  Batas Bawah: 26.72
  Batas Atas: 32.38
```

Tindakan dan justifikasi kesalahan

- Menghapus Data yang memiliki tanggal tidak relevan seperti data order pizza yang dikirim dari 2026.

kode ini digunakan untuk menghapus data yang memiliki tanggal lebih dari tanggal pemesanan 14 september 2025 jam 20:00. setelah data dihapus kemudian disimpan ke file baru yang bernama Salinan Data Penjualan Pizza_Tgl_Bersih.xlsx

```
data['Order Time'] = pd.to_datetime(data['Order Time'])
tanggal_batas = pd.to_datetime('9/14/2025 20:00:00')
data_bersih = data[data['Order Time'] <= tanggal_batas].copy()

print("Data yang terjadi setelah tanggal yang ditentukan berhasil dihapus.")
print(f"Jumlah baris sebelum dibersihkan: {data.shape[0]}")
print(f"Jumlah baris setelah dibersihkan: {data_bersih.shape[0]}")

output_path = "/content/drive/MyDrive/Dataset Pizza/Salinan Data Penjualan Pizza_Tgl_Bersih.xlsx"
data_bersih.to_excel(output_path, index=False)

print(f"\nDataset baru berhasil disimpan di: {output_path}")
```

```
Data yang terjadi setelah tanggal yang ditentukan berhasil dihapus.
Jumlah baris sebelum dibersihkan: 1004
Jumlah baris setelah dibersihkan: 700

Dataset baru berhasil disimpan di: /content/drive/MyDrive/Dataset Pizza/Salinan Data Penjualan Pizza_Tgl_Bersih.xlsx
```

- file dataset baru dengan tanggal yang sudah disesuaikan memuat file tersebut untuk memastikan apakah sudah hilang untuk tanggal yang tidak relevan tadi

kode ini digunakan untuk menampilkan nilai maksimal "Order Time".

```
import pandas as pd
file_path = "/content/drive/MyDrive/Dataset Pizza/Salinan Data Penjualan Pizza_Tgl_Bersih.xlsx"
data = pd.read_excel(file_path)
print(data["Order Time"].describe())
```

```
count                    700
mean     2024-11-08 10:17:57
min      2024-01-05 18:30:00
25%      2024-08-03 20:22:30
50%      2024-10-12 07:45:00
75%      2025-03-24 01:41:15
max      2025-09-14 20:00:00
Name: Order Time, dtype: object
```

- Mengubah nama Marco's Pizza menjadi Marco's Pizza

kode ini digunakan untuk menampilkan nama nama restoran dan jumlahnya

```
file_path = "/content/drive/MyDrive/Dataset Pizza/Salinan Data Penjualan Pizza_Tgl_Bersih.xlsx"
data = pd.read_excel(file_path)
print(data["Restaurant Name"].value_counts(),"\n")
```

```
Restaurant Name
Domino's          158
Papa John's       139
Pizza Hut         139
Little Caesars    134
Marco's Pizza     127
Marco's Pizza       3
Name: count, dtype: int64
```

kode ini digunakan untuk mengubah data inkosistensi macro's pizza

```
import pandas as pd
import numpy as np
import os

file_path = "/content/drive/MyDrive/Dataset Pizza/Salinan Data Penjualan Pizza_Tgl_Bersih.xlsx"
data = pd.read_excel(file_path)
data['Restaurant Name'] = data['Restaurant Name'].str.replace("'", "")
output_path = "/content/drive/MyDrive/Dataset Pizza/Salinan Data Nama Restoran Pizza_Bersih.xlsx"
data.to_excel(output_path, index=False)
print(f"\nDataset baru berhasil disimpan di: {output_path}")
```

```
Dataset baru berhasil disimpan di: /content/drive/MyDrive/Dataset Pizza/Salinan Data Nama Restoran Pizza_Bersih.xlsx
```

- Memastikan bahwa nama restoran sudah sesuai

```
menampilkan nama restoran yang sudah sesuai
```

```
file_path = "/content/drive/MyDrive/Dataset Pizza/Salinan Data Nama Restoran Pizza_Bersih.xlsx"
data = pd.read_excel(file_path)
print(data["Restaurant Name"].value_counts(),"\n")
```

```
Restaurant Name
Domino's          158
Papa John's       139
Pizza Hut         139
Little Caesars    134
Marco's Pizza     130
Name: count, dtype: int64
```

- Penanganan outlier

kode ini digunakan untuk mengetahui outlier dengan metode IQR

```
def iqr(nama_kolom):
    Q1 = data[nama_kolom].quantile(0.25)
    Q3 = data[nama_kolom].quantile(0.75)
    iqr = Q3 - Q1
    lower_bound = Q1 - (1.5 * iqr)
    upper_bound = Q3 + (1.5 * iqr)
    outliers_count = data[(data[nama_kolom] < lower_bound) | (data[nama_kolom] > upper_bound)].shape[0]

    print(f"Kolom: {nama_kolom}")
    print(f"  Jumlah Outlier: {outliers_count}")
    print(f"  Batas Bawah: {lower_bound:.2f}")
    print(f"  Batas Atas: {upper_bound:.2f}\n")

iqr("Delivery Duration (min)")
iqr("Toppings Count")
iqr("Distance (km)")
iqr("Delivery Efficiency (min/km)")
iqr("Topping Density")
iqr("Estimated Duration (min)")
iqr("Delay (min)")
iqr("Pizza Complexity")
iqr("Traffic Impact")
iqr("Order Hour")
iqr("Restaurant Avg Time")
```

```
Kolom: Delivery Duration (min)
  Jumlah Outlier: 0
  Batas Bawah: -2.50
  Batas Atas: 57.50

Kolom: Toppings Count
  Jumlah Outlier: 0
  Batas Bawah: -1.00
  Batas Atas: 7.00

Kolom: Distance (km)
  Jumlah Outlier: 0
  Batas Bawah: -1.50
  Batas Atas: 10.50

Kolom: Delivery Efficiency (min/km)
  Jumlah Outlier: 20
  Batas Bawah: 1.25
  Batas Atas: 11.25

Kolom: Topping Density
  Jumlah Outlier: 1
  Batas Bawah: 0.10
  Batas Atas: 1.37

Kolom: Estimated Duration (min)
  Jumlah Outlier: 0
  Batas Bawah: -3.60
  Batas Atas: 25.20

Kolom: Delay (min)
  Jumlah Outlier: 0
  Batas Bawah: 3.80
  Batas Atas: 31.00

Kolom: Pizza Complexity
  Jumlah Outlier: 0
  Batas Bawah: -8.00
  Batas Atas: 24.00

Kolom: Traffic Impact
  Jumlah Outlier: 0
  Batas Bawah: -2.00
  Batas Atas: 6.00

Kolom: Order Hour
  Jumlah Outlier: 51
  Batas Bawah: 15.00
  Batas Atas: 23.00

Kolom: Restaurant Avg Time
  Jumlah Outlier: 3
  Batas Bawah: 26.72
  Batas Atas: 32.38
```

- Penanganan outlier menggunakan wisorization

kode ini digunakan menangani outlier menggunakan winsorization

```
file_path = "/content/drive/MyDrive/Dataset Pizza/Salinan Data Nama Restoran Pizza_Bersih.xlsx"
data = pd.read_excel(file_path)

def winsorization(df, column_name):
    lower_bound = df[column_name].quantile(0.05)
    upper_bound = df[column_name].quantile(0.95)

    df[column_name] = np.clip(df[column_name], a_min=lower_bound, a_max=upper_bound)

    return df

kolom_list = [
    "Delivery Duration (min)",
    "Toppings Count",
    "Distance (km)",
    "Delivery Efficiency (min/km)",
    "Topping Density",
    "Estimated Duration (min)",
    "Delay (min)",
    "Pizza Complexity",
    "Traffic Impact",
    "Order Hour",
```

```
    "Restaurant Avg Time"
]

for kolom_target in kolom_list:
    data = winsorization(data, kolom_target)

output_path = "/content/drive/MyDrive/Dataset Pizza/Salinan Data Penjualan Pizza_Winsorized.xlsx"
data.to_excel(output_path, index=False)

print(f"\nDataset yang sudah dibersihkan berhasil disimpan ke: {output_path}")
```

```
Dataset yang sudah dibersihkan berhasil disimpan ke: /content/drive/MyDrive/Dataset Pizza/Salinan Data Penjualan Pizza_Winsorized.xlsx
```

- Memastikan apakah masih ada kolom yang masih memiliki outlier

kode ini digunakan untuk memuat hasil data yang sudah dibersihkan dari outlier

```
file_path = "/content/drive/MyDrive/Dataset Pizza/Salinan Data Penjualan Pizza_Winsorized.xlsx"
data = pd.read_excel(file_path)
def iqr(nama_kolom):
    Q1 = data[nama_kolom].quantile(0.25)
    Q3 = data[nama_kolom].quantile(0.75)
    iqr = Q3 - Q1
    lower_bound = Q1 - (1.5 * iqr)
    upper_bound = Q3 + (1.5 * iqr)
    outliers_count = data[(data[nama_kolom] < lower_bound) | (data[nama_kolom] > upper_bound)].shape[0]

    print(f"Kolom: {nama_kolom}")
    print(f"  Jumlah Outlier: {outliers_count}")
    print(f"  Batas Bawah: {lower_bound:.2f}")
    print(f"  Batas Atas: {upper_bound:.2f}\n")


iqr("Delivery Duration (min)")
iqr("Toppings Count")
iqr("Distance (km)")
iqr("Delivery Efficiency (min/km)")
iqr("Topping Density")
iqr("Estimated Duration (min)")
iqr("Delay (min)")
iqr("Pizza Complexity")
iqr("Traffic Impact")
iqr("Order Hour")
iqr("Restaurant Avg Time")
```

```
Kolom: Delivery Duration (min)
  Jumlah Outlier: 0
  Batas Bawah: -2.50
  Batas Atas: 57.50

Kolom: Toppings Count
  Jumlah Outlier: 0
  Batas Bawah: -1.00
  Batas Atas: 7.00

Kolom: Distance (km)
  Jumlah Outlier: 0
  Batas Bawah: -1.50
  Batas Atas: 10.50

Kolom: Delivery Efficiency (min/km)
  Jumlah Outlier: 0
  Batas Bawah: 1.25
  Batas Atas: 11.25

Kolom: Topping Density
  Jumlah Outlier: 0
  Batas Bawah: 0.10
  Batas Atas: 1.37

Kolom: Estimated Duration (min)
  Jumlah Outlier: 0
  Batas Bawah: -3.60
  Batas Atas: 25.20

Kolom: Delay (min)
  Jumlah Outlier: 0
  Batas Bawah: 3.80
  Batas Atas: 31.00

Kolom: Pizza Complexity
  Jumlah Outlier: 0
  Batas Bawah: -8.00
  Batas Atas: 24.00

Kolom: Traffic Impact
  Jumlah Outlier: 0
  Batas Bawah: -2.00
  Batas Atas: 6.00

Kolom: Order Hour
  Jumlah Outlier: 51
  Batas Bawah: 15.00
  Batas Atas: 23.00

Kolom: Restaurant Avg Time
  Jumlah Outlier: 0
  Batas Bawah: 26.72
  Batas Atas: 32.38
```

- Mencoba membersihkan outlier pada kolom order hour

kode ini digunakan untuk membersihkan sekali lagi

```
import pandas as pd
import numpy as np
import os
file_path = "/content/drive/MyDrive/Dataset Pizza/Salinan Data Penjualan Pizza_Winsorized.xlsx"
data = pd.read_excel(file_path)

def winsorization(df, column_name):
    lower_bound = df[column_name].quantile(0.05)
    upper_bound = df[column_name].quantile(0.95)

    df[column_name] = np.clip(df[column_name], a_min=lower_bound, a_max=upper_bound)

    return df

kolom_list = [
    "Order Hour",
]

for kolom_target in kolom_list:
    data = winsorization(data, kolom_target)

output_path = "/content/drive/MyDrive/Dataset Pizza/Salinan Data Penjualan Pizza_OrderHour.xlsx"
data.to_excel(output_path, index=False)

print(f"\nDataset yang sudah dibersihkan berhasil disimpan ke: {output_path}")
```

```
Dataset yang sudah dibersihkan berhasil disimpan ke: /content/drive/MyDrive/Dataset Pizza/Salinan Data Penjualan Pizza_OrderHour.xlsx
```

- Mengecek apakah tetap ada outliernya

```
file_path = "/content/drive/MyDrive/Dataset Pizza/Salinan Data Penjualan Pizza_OrderHour.xlsx"
data = pd.read_excel(file_path)
def iqr(nama_kolom):
    Q1 = data[nama_kolom].quantile(0.25)
    Q3 = data[nama_kolom].quantile(0.75)
    iqr = Q3 - Q1
    lower_bound = Q1 - (1.5 * iqr)
    upper_bound = Q3 + (1.5 * iqr)
    outliers_count = data[(data[nama_kolom] < lower_bound) | (data[nama_kolom] > upper_bound)].shape[0]

    print(f"Kolom: {nama_kolom}")
    print(f"  Jumlah Outlier: {outliers_count}")
    print(f"  Batas Bawah: {lower_bound:.2f}")
    print(f"  Batas Atas: {upper_bound:.2f}\n")

iqr("Order Hour")
```

```
Kolom: Order Hour
  Jumlah Outlier: 51
  Batas Bawah: 15.00
  Batas Atas: 23.00
```

Kesimpulan

Berdasarkan hasil eksplorasi awal, dapat disimpulkan bahwa dataset penjualan pizza berisi lebih dari 1000 entri dengan 25 atribut yang mencakup informasi lengkap mengenai pesanan, mulai dari detail restoran, lokasi, ukuran dan jenis pizza, jumlah topping, hingga durasi serta efisiensi pengantaran. Hasil analisis menunjukkan bahwa Domino's menjadi restoran dengan jumlah pesanan terbanyak, ukuran pizza Medium paling dominan, rata-rata jumlah topping adalah 3, dan waktu pengantaran umumnya berada di kisaran 30 menit.

Pemeriksaan kualitas data mengungkapkan bahwa tidak terdapat data duplikat maupun missing value, sehingga dataset relatif bersih dan siap digunakan untuk analisis lebih lanjut. Namun, ditemukan adanya inkonsistensi penulisan kategori, seperti pada nama restoran "Marco's Pizza" dan "Marco's Pizza", yang menegaskan perlunya tahap pembersihan data agar hasil analisis lebih akurat. selain itu juga ditemukan beberapa kolom yang memiliki outlier seperti delivery efficiency dan restaurant avg time. selain beberapa hal tersebut terdapat juga tanggal order yang tidak relevan seperti pembeli memesan pada tahun 2026.

berdasarkan hal hal tersebut maka diperlukannya penyesuaian agar data dapat digunakan lebih lanjut

untuk solusi inkonsistensi nama restoran telah diubah menjadi Marco's Pizza, untuk mendeteksi outlier menggunakan iqr serta penangganannya menggunakan metode winsorization. untuk tanggal order yang tidak relevan tersebut maka data yang tidak relevan dihapus sampai tanggalnya menjadi lebih relevan.

Secara keseluruhan, proses data profiling ini memberikan gambaran awal mengenai pola pemesanan, distribusi kategori, serta kualitas data, yang menjadi dasar penting sebelum dilakukan analisis lanjutan atau pembuatan model prediktif.