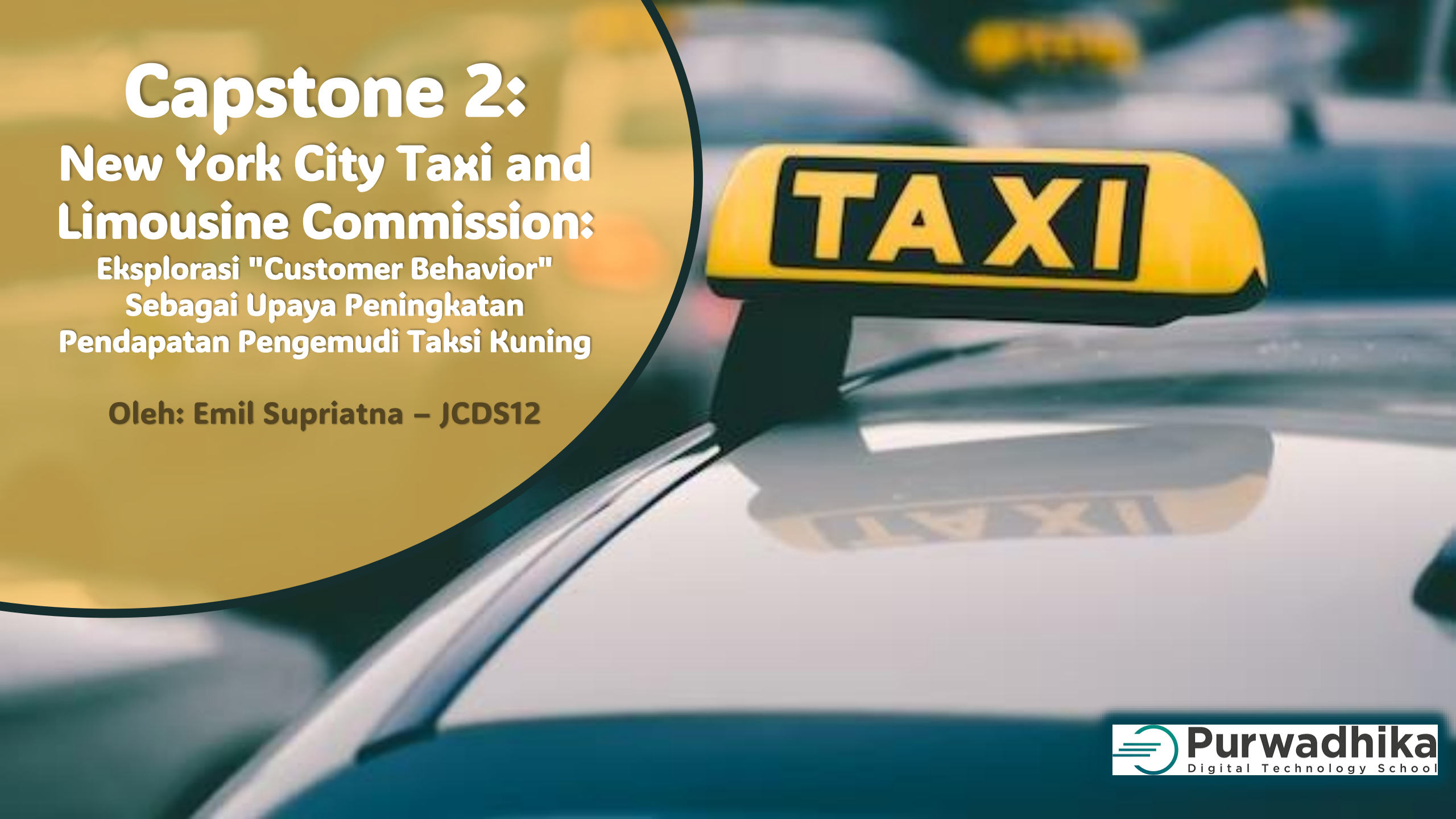
Data Exploration

Purwadhika
Digital Technology School

Capstone 2:
New York City Taxi and Limousine Commission:
Eksplorasi "Customer Behavior" Sebagai Upaya Peningkatan Pendapatan Pengemudi Taksi Kuning

Oleh: Emil Supriatna – JCDS12



Purwadhika
Digital Technology School

Konteks Permasalahan:

Taxi Kuning adalah ikon kota New York. Popularitasnya bahkan terkenal hingga ke Masyarakat global.

Sayangnya, pertumbuhan mode transportasi berbasis teknologi seperti Uber dan Lyft telah mengikis keberadaan Taxi Kuning di New York. Banyak dari penumpang Taksi Kuning telah beralih mode transportasi.

Di sisi lain, sebagian dari pengemudi Taksi Kuning juga terikat dengan utang akibat pembelian Medali sebagai hak eksklusif mengangkut penumpang di jalanan kota New York.

Dengan kata lain, pengemudi Taksi Kuning kesulitan mendapatkan penumpang karena persaingan, di sisi lain tidak dapat berhenti dari pekerjaan karena ada utang yang harus dibayarkan.

Projek ini berusaha mengekskplorasi pertanyaan seputar “Bagaimana meningkatkan daya resistansi pengemudi taksi kuning di Tengah persaingan dan impitan utang melalui eksplorasi data analysis?

TAXI SERVICE

Rumusan Masalah:

1. **Bagaimana pengemudi taksi melakukan manajemen waktu kerja untuk mendapatkan penumpang lebih banyak?** Pertanyaan ini bertujuan untuk menemukan rekomendasi waktu kerja terbaik dengan mengeksplorasi waktu kepadatan penumpang dan rata-rata pendapatan tertinggi saat weekdays dan weekend.
2. **Bagaimana pengemudi taksi bisa memperoleh pendapatan optimal?** Tujuan dari pertanyaan ini adalah mengevaluasi faktor-faktor yang mempengaruhi pendapatan pengemudi taksi dan menyusun strategi untuk membantu mereka mencapai pendapatan optimal, termasuk pemilihan rute, kriteria penumpang dan sebagainya.
3. **Bagaimana tips mendapatkan tip sebagai peluang memperoleh pendapatan tambahan?** Pertanyaan ini ditujukan untuk melihat peluang-peluang yang dapat meningkatkan perolehan uang tip dengan eksplorasi korelasi tip dengan faktor-faktor lainnya.

Metodologi Penelitian:
(Exploratory Data Analysis)

- Data Collection
- Data Cleansing and Preprocessing
- Feature Engineering
- Data Exploration

EDA 1: Data Collection

In [51]:
```
# Libraries
import pandas as pd
import warnings
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import kstest, norm
warnings.filterwarnings('ignore')
```

In [52]:
```
# Load Dataset
df = pd.read_csv('NYC TLC Trip Record.csv')
df.head()
```

Out[52]:

	VendorID	lpep_pickup_datetime	lpep_dropoff_datetime	store_and_fwd_flag	RatecodeID	PULocationID	DOLocationID	passenger_count	trip_distance	fare_a
0	2	2023-01-01 00:26:10	2023-01-01 00:37:11	N	1.0	166	143	1.0	2.58	
1	2	2023-01-01 00:51:03	2023-01-01 00:57:49	N	1.0	24	43	1.0	1.81	
2	2	2023-01-01 00:35:12	2023-01-01 00:41:32	N	1.0	223	179	1.0	0.00	
3	1	2023-01-01 00:13:14	2023-01-01 00:19:03	N	1.0	41	238	1.0	1.30	
4	1	2023-01-01 00:33:04	2023-01-01 00:39:02	N	1.0	41	74	1.0	1.10	

EDA 2: Missing Values

1

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 68211 entries, 0 to 68210
Data columns (total 20 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   VendorID                68211 non-null  int64
 1   lpep_pickup_datetime    68211 non-null  object
 2   lpep_dropoff_datetime   68211 non-null  object
 3   store_and_fwd_flag      63887 non-null  object
 4   RatecodeID              63887 non-null  float64
 5   PULocationID            68211 non-null  int64
 6   DOLocationID            68211 non-null  int64
 7   passenger_count         63887 non-null  float64
 8   trip_distance           68211 non-null  float64
 9   fare_amount             68211 non-null  float64
 10  extra                   68211 non-null  float64
 11  mta_tax                 68211 non-null  float64
 12  tip_amount              68211 non-null  float64
 13  tolls_amount            68211 non-null  float64
 14  ehail_fee               0 non-null      float64
 15  improvement_surcharge   68211 non-null  float64
 16  total_amount            68211 non-null  float64
 17  payment_type            63887 non-null  float64
 18  trip_type               63877 non-null  float64
 19  congestion_surcharge    63887 non-null  float64
dtypes: float64(14), int64(3), object(3)
memory usage: 10.4+ MB
```

2

```
# Menghitung berapa persen missing value

print(round(df.isna().mean()*100,2))
```

```
VendorID                  0.00
lpep_pickup_datetime      0.00
lpep_dropoff_datetime     0.00
store_and_fwd_flag        6.34
RatecodeID                6.34
PULocationID              0.00
DOLocationID              0.00
passenger_count           6.34
trip_distance             0.00
fare_amount               0.00
extra                     0.00
mta_tax                   0.00
tip_amount                0.00
tolls_amount              0.00
ehail_fee               100.00
improvement_surcharge     0.00
total_amount              0.00
payment_type              6.34
trip_type                 6.35
congestion_surcharge      6.34
dtype: float64
```

3

```
# Menghitung total varian data unik,
# selanjutnya diputuskan missing handling dengan modus/median/mean

columns = ['store_and_fwd_flag', 'RatecodeID', 'passenger_count',
           'payment_type', 'trip_type', 'congestion_surcharge']

for i in columns:
    print(f"unique value for {i}:  {df[i].nunique()}")
```

```
unique value for store_and_fwd_flag:  2
unique value for RatecodeID:  6
unique value for passenger_count:  10
unique value for payment_type:  5
unique value for trip_type:  2
unique value for congestion_surcharge:  4
```

4

```
# Handling Missing Value with modus

for column in columns:
    df[column] = df[column].fillna(df[column].mode().iloc[0])

df = df.drop('ehail_fee', axis=1)

# Periksa kembali dan pastikan semua missing values telah selesai diatasi
df.info()
```

EDA 3: Outliers

1

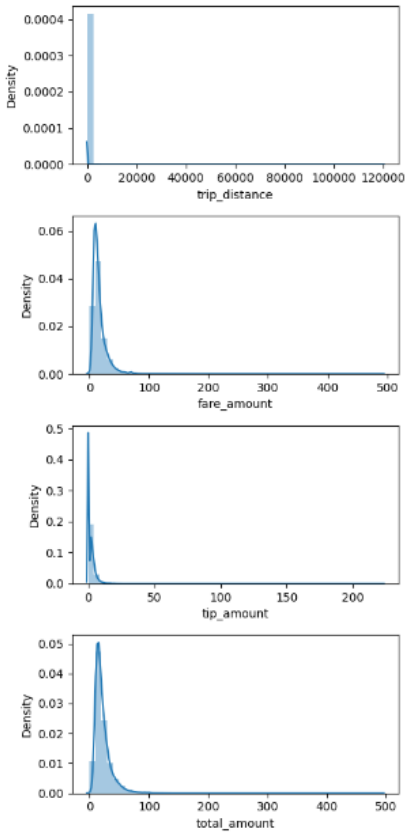
```
# Dengan mengecualikan data tanggal (datetime),
# identifikasi outliers dengan visualisasi distplot dari seaborn

columns = ['trip_distance', 'fare_amount', 'tip_amount', 'total_amount']

plt.figure(figsize=(5, 10))

for i, column in enumerate(columns, start=1):
    plt.subplot(4, 1, i)
    sns.distplot(df[column])

plt.tight_layout()
plt.show()
```



2

```
# Menghitung jumlah dan persentase outliers dengan metode IQR

columns_outliers = ['trip_distance', 'fare_amount', 'tip_amount', 'total_amount']

for column in columns_outliers:
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1

    lower_bound = Q1 - 1.5*IQR
    upper_bound = Q3 + 1.5*IQR

    outliers = df[(df[column] < lower_bound) | (df[column] > upper_bound)][column]
    print(f'Outliers pada {column} adalah {outliers.count()} atau sebesar {round(outliers.count()/df.shape[0]*100,2)}%')
```

```
Outliers pada trip_distance adalah 5774 atau sebesar 8.46%
Outliers pada fare_amount adalah 4304 atau sebesar 6.31%
Outliers pada tip_amount adalah 2045 atau sebesar 3.0%
Outliers pada total_amount adalah 3968 atau sebesar 5.82%
```

3

```
# Menghapus Outliers

for column in columns_outliers:
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1

    lower_bound = Q1 - 1.5*IQR
    upper_bound = Q3 + 1.5*IQR

    df = df[(df[column] >= lower_bound) & (df[column] <= upper_bound)]

df.reset_index(drop=True, inplace=True)
```

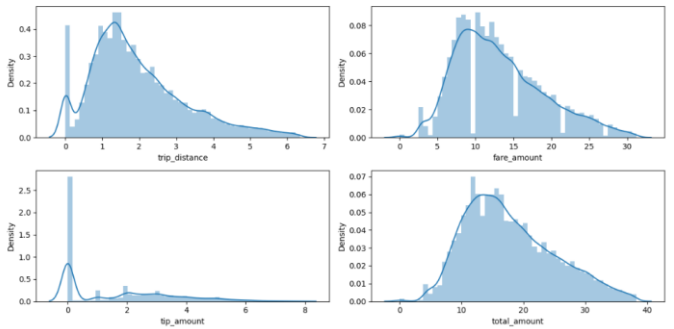
4

```
# Tampilkan kembali visualisasi distplot, maka data terlihat mendekati distribusi normal
# Namun, untuk mengatakan data terdistribusi normal atau tidak, perlu dilakukan uji normalitas

plt.figure(figsize=(12, 6))

for i, column in enumerate(columns_outliers, start=1):
    plt.subplot(2, 2, i)
    sns.distplot(df[column])

plt.tight_layout()
plt.show()
```

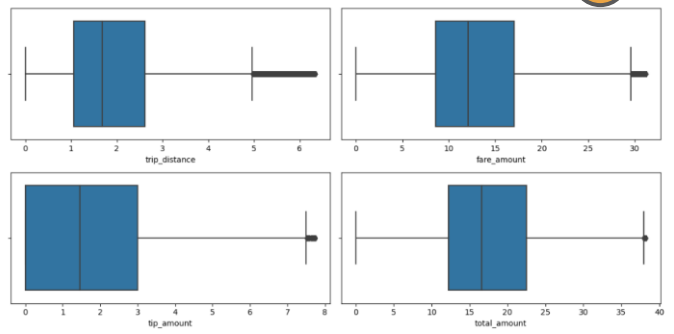


5

```
# Bandingkan dengan menggunakan boxplot
# Meskipun masih terdapat data outlier di luar 'upper
plt.figure(figsize=(12, 6))

for i, column in enumerate(columns_outliers, start=1):
    plt.subplot(2, 2, i)
    sns.boxplot(x = df[column])

plt.tight_layout()
plt.show()
```



EDA 4: Formatting & Add Column

1

```python
# Mengubah tipe data tanggal dari 'object' menjadi 'datetime'
df['lpep_pickup_datetime'] = pd.to_datetime(df['lpep_pickup_datetime'])
df['lpep_dropoff_datetime'] = pd.to_datetime(df['lpep_dropoff_datetime'])

# Mengubah sebagian tipe data float menjadi integer
df['RatecodeID'] = df['RatecodeID'].astype(int)
df['passenger_count'] = df['passenger_count'].astype(int)
df['payment_type'] = df['payment_type'].astype(int)
df['trip_type'] = df['trip_type'].astype(int)

# Mengubah sebagian tipe data integer menjadi string
df['PULocationID'] = df['PULocationID'].astype(str)
df['DOLocationID'] = df['DOLocationID'].astype(str)
```

2

```python
# Menambah beberapa kolom baru
df['pickup_date'] = pd.to_datetime(df['lpep_pickup_datetime']).dt.date
df['days'] = df['lpep_pickup_datetime'].dt.day_name()
df['day_number'] = df['lpep_pickup_datetime'].dt.dayofweek + 1
df['days_category'] = df['lpep_pickup_datetime'].dt.day_name()\
    .apply(lambda x: 'weekend' if x in ['Saturday', 'Sunday'] else 'weekdays')
df['pickup_hour'] = df['lpep_pickup_datetime'].dt.hour
df['time_duration'] = df['lpep_dropoff_datetime'] - df['lpep_pickup_datetime']
df['time_duration'] = df['time_duration'].dt.total_seconds() / 60
```

3

```python
# Tampak bahwa penambahan kolom berhasil dilakukan.
# Namun, urutan kolom terlihat tidak terstruktur.
# Dengan demikian, pengurutan ulang daftar kolom perlu dilakukan.

df = df[[

    # ID
    'VendorID',

    # Time Category
    'lpep_pickup_datetime', 'lpep_dropoff_datetime', 'days_category',
    'day_number', 'days', 'pickup_date', 'pickup_hour', 'time_duration',

    # Other Information
    'passenger_count', 'store_and_fwd_flag', 'RatecodeID', 'PULocationID',
    'DOLocationID', 'trip_type', 'trip_distance', 'payment_type',

    # Amount Category
    'fare_amount',  'extra', 'mta_tax', 'tip_amount', 'tolls_amount',
    'improvement_surcharge',  'congestion_surcharge', 'total_amount'

]]
```

Data Cleansing and Preprocessing

```
df.info()   # before preprocessing
```

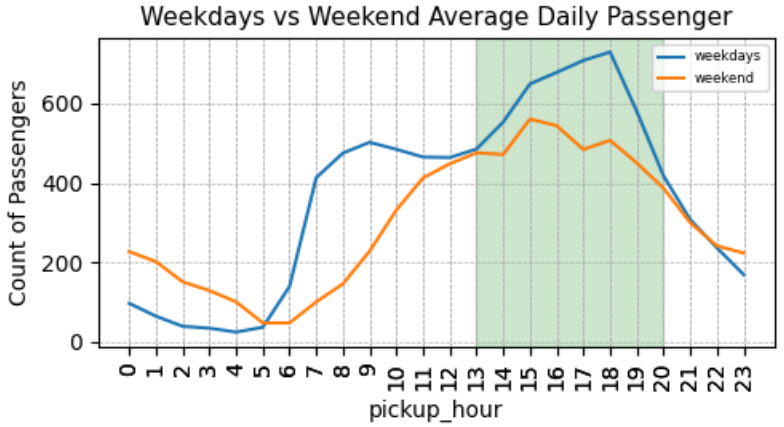
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 68211 entries, 0 to 68210
Data columns (total 20 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   VendorID                68211 non-null  int64
 1   lpep_pickup_datetime    68211 non-null  object
 2   lpep_dropoff_datetime   68211 non-null  object
 3   store_and_fwd_flag      63887 non-null  object
 4   RatecodeID              63887 non-null  float64
 5   PULocationID            68211 non-null  int64
 6   DOLocationID            68211 non-null  int64
 7   passenger_count         63887 non-null  float64
 8   trip_distance           68211 non-null  float64
 9   fare_amount             68211 non-null  float64
 10  extra                   68211 non-null  float64
 11  mta_tax                 68211 non-null  float64
 12  tip_amount              68211 non-null  float64
 13  tolls_amount            68211 non-null  float64
 14  ehail_fee               0 non-null      float64
 15  improvement_surcharge   68211 non-null  float64
 16  total_amount            68211 non-null  float64
 17  payment_type            63887 non-null  float64
 18  trip_type               63877 non-null  float64
 19  congestion_surcharge    63887 non-null  float64
dtypes: float64(14), int64(3), object(3)
memory usage: 10.4+ MB
```

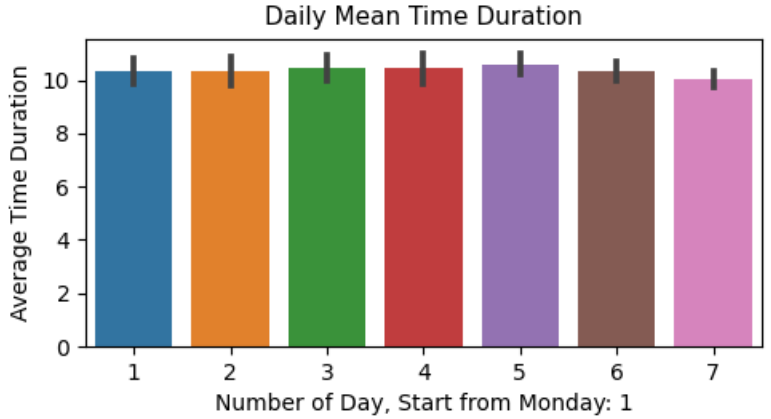
```
df.info()   # after preprocessing
```

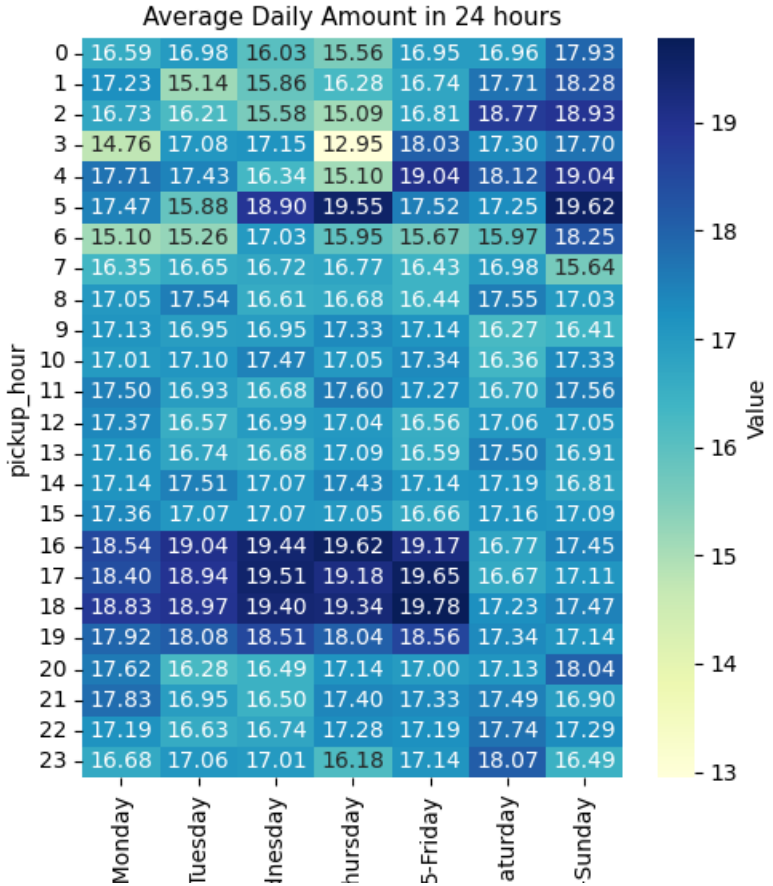
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58238 entries, 0 to 58237
Data columns (total 25 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   VendorID                58238 non-null  int64
 1   lpep_pickup_datetime    58238 non-null  datetime64[ns]
 2   lpep_dropoff_datetime   58238 non-null  datetime64[ns]
 3   store_and_fwd_flag      58238 non-null  object
 4   RatecodeID              58238 non-null  int32
 5   PULocationID            58238 non-null  object
 6   DOLocationID            58238 non-null  object
 7   passenger_count         58238 non-null  int32
 8   trip_distance           58238 non-null  float64
 9   fare_amount             58238 non-null  float64
 10  extra                   58238 non-null  float64
 11  mta_tax                 58238 non-null  float64
 12  tip_amount              58238 non-null  float64
 13  tolls_amount            58238 non-null  float64
 14  improvement_surcharge   58238 non-null  float64
 15  total_amount            58238 non-null  float64
 16  payment_type            58238 non-null  int32
 17  trip_type               58238 non-null  int32
 18  congestion_surcharge    58238 non-null  float64
 19  pickup_date             58238 non-null  object
 20  days                    58238 non-null  object
 21  day_number              58238 non-null  int64
 22  days_category           58238 non-null  object
 23  pickup_hour             58238 non-null  int64
 24  time_duration           58238 non-null  float64
dtypes: datetime64[ns](2), float64(10), int32(4), int64(3),
memory usage: 10.2+ MB
```

Data Exploration

Analisis 1: Mengoptimalkan Penumpang dan Pendapatan Melalui Manajemen Waktu Kerja







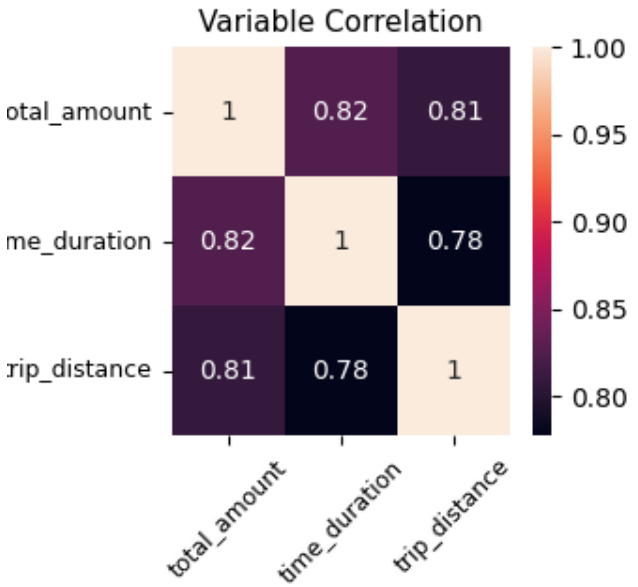
Overviews: Tidak seperti pekerja kantor yang bekerja dengan rentang waktu yang tetap, pengemudi taksi kuning merupakan pekerja lepas di mana mereka menentukan sendiri waktu kerjanya. Penentuan waktu kerja yang tidak tepat dapat berpotensi memperoleh sedikit pelanggan. Oleh sebab itu, manajemen waktu kerja yang baik berdasarkan data diperlukan. Grafik di atas menunjukkan kapan waktu kerja terbaik bagi pengemudi Taksi Kuning agar mendapatkan penumpang potensial dengan rata-rata amount tertinggi, serta perbedaan pola waktu antara *weekdays* dan *weekend*.

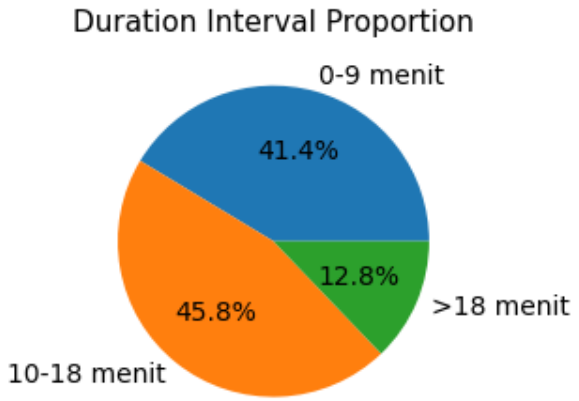
Analisis 2:

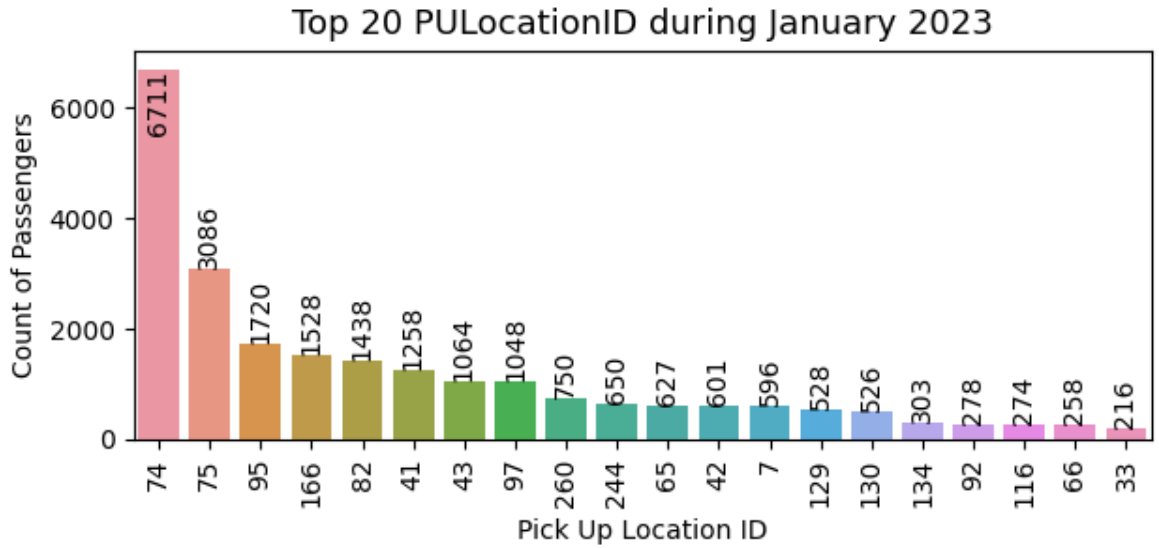
Korelasi Faktor-Faktor Penunjang Pendapatan Maksimal

Overviews: Sebagai pekerja tentunya ingin mendapatkan penghasilan semaksimal mungkin. Demikian juga dengan pengemudi Taksi Kuning. Grafik di samping adalah eksplorasi terhadap faktor-faktor yang berpotensi memaksimalkan penghasilan pengemudi taksi kuning.

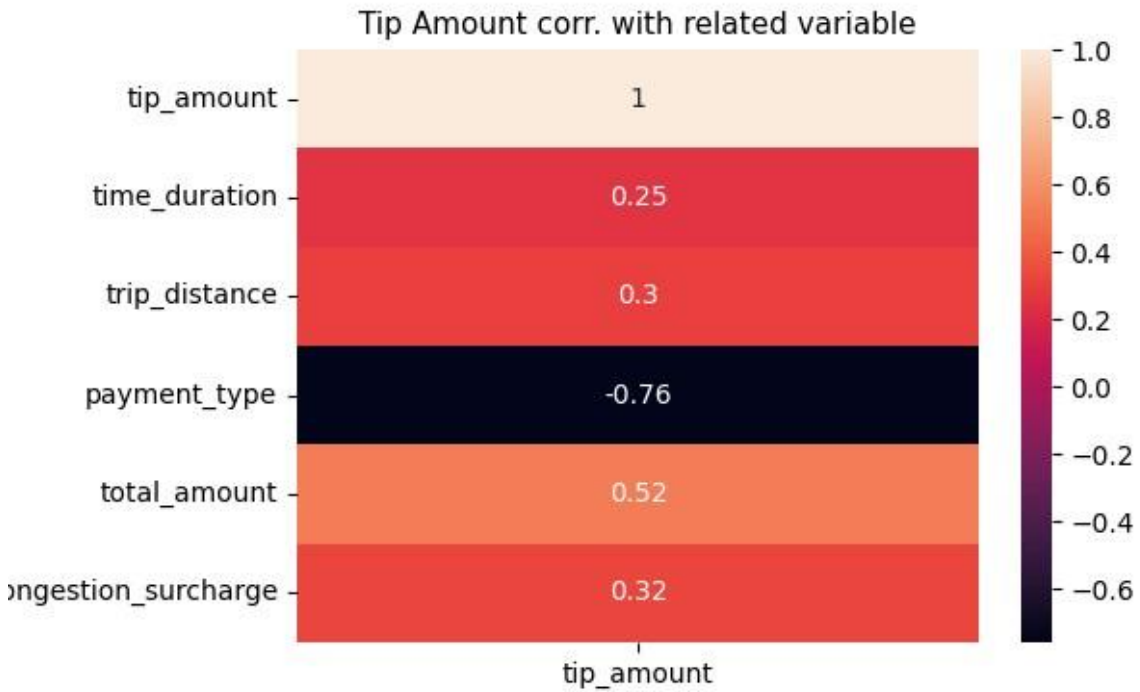
Berdasarkan uji korelasi (method. *spearman*), terdapat hubungan positif yang cukup tinggi pada variabel durasi dan jarak perjalanan terhadap besarnya bayaran yang diterima pengemudi taksi. Sayangnya, perjalanan dengan durasi lebih dari 18 menit sangat sedikit, sehingga pengemudi mungkin dapat membidik pelanggan dengan durasi perjalanan di bawah 18 menit dengan memahami karakteristik pelanggan, juga lokasi-lokasi yang memiliki potensi pelanggan lebih banyak.







Analisis 3: Credit Card, Peluang Besar Mendapatkan Tip



Total Amount vs Tip Amount

payment_type	customer	giving_tip	perc.(%)
Credit card	37,848	33,529	88.59
Cash	19,849	1	0.01
No charge	448	16	3.57
Dispute	90	0	0.0
Unknown	2	0	0.0

Overviews: Tip merupakan sumber penghasilan selain penghasilan utama yang didapatkan pengemudi Taksi Kuning. Oleh sebab itu, selain memaksimalkan penghasilan utama, pengemudi taksi juga penting untuk memaksimalkan tip. Berdasarkan grafik, sebagian besar pelanggan yang membayar dengan Credit Card cenderung memberikan tip kepada pengemudi taksi. Oleh sebab itu, vendor taksi perlu mendorong pelanggan untuk menggunakan Credit Card sebagai metode pembayaran untuk membantu pengemudi taksi mendapatkan uang tip.

Rekomendasi:

1. Optimalkan Waktu Berkendara

- Pahami jam-jam sibuk berdasarkan ID lokasi
- Kenali area dengan permintaan tinggi dan sedikit antrean
- Analisis rute secara optimal, hindari kemacetan

2. Manajemen Waktu Lainnya

- Tentukan waktu istirahat yang tepat
- Pilih tempat istirahat yang strategis
- Atur jadwal pemeliharaan kendaraan secara teratur

3. Pahami Spasial Demografi Penumpang

- Lakukan pemetaan penumpang dengan durasi 10-18 menit
- Perhatikan lokasi potensial
- Pada grafik ditampilkan 20 id lokasi strategis

4. Vendor bekerja sama dengan penyedia layanan kartu kredit

- Rekomendasi kerja sama: point reward dan cashback
- Pastikan EDC berfungsi baik
- Berikan layanan ramah dan menyenangkan selama perjalanan

5. Pengelolaan Keuangan Pribadi

- Analisis pendapatan dan pengeluaran secara teratur
- Tetapkan target pendapatan bulanan yang realistis
- Kelola pengeluaran dengan bijak

Kesimpulan

Eksplorasi data yang tersaji dalam tulisan ini telah memberikan gambaran dan rekomendasi yang bersifat operasional kepada pengemudi taksi kuning.

Namun penting diingat, insight akan senantiasa berubah seiring perkembangan aktifitas taksi kuning setiap waktunya.

Oleh sebab itu, kegiatan penelitian berkala penting dilakukan untuk dapat memperbaharui wawasan, sehingga kegiatan bisnis bisa terus resistan, adaptif dan bertumbuh.