

Data Science Report

Analisis dan Prediksi Tingkat Kejahatan di Kota Los Angeles: Tren 2020-2023



PRADITA
University

Hansel Ricardo (2110101020)

Mandalan (2110101028)

Desfantio Wuidjaja (2110101014)

Ryan Christensen Wang (2110101024)

Daftar Isi

A. Pendahuluan

- 1) Latar Belakang
- 2) Motivasi
- 3) Tujuan

B. Metode

- 1) Deskripsi Data
- 2) Teknik/Algoritma/Analisis

C. Hasil dan Pembahasan

- 1) Output dan Analisis Python Programming

D. Kesimpulan

E. Daftar Pustaka

A. Pendahuluan

Latar Belakang

Kejahatan merupakan suatu masalah yang telah manusia alami sejak lama, sehingga dapat dikatakan bahwa di mana ada manusia, disitu ada kejahatan. Kejahatan adalah ancaman serius terhadap rasa aman manusia dalam perubahan nilai-nilai sosial, dan rasa aman merupakan hak asasi manusia, sehingga perlu kerja sama pemerintah dan masyarakat untuk mengurangi kejahatan di Indonesia. (Putra, Martha, Fikram & Yuhan, 2020)

Kriminalitas adalah masalah yang signifikan di banyak kota besar di seluruh dunia, termasuk Kota Los Angeles, California. Sebagai salah satu kota terbesar di Amerika Serikat dengan populasi yang mencapai jutaan penduduk, Los Angeles memiliki tantangan serius terkait keamanan masyarakat dan pengendalian tingkat kejahatan. Kriminalitas dapat mencakup berbagai jenis pelanggaran, mulai dari pencurian kecil hingga kejahatan serius seperti perampokan, pemerkosaan, dan pembunuhan.

Los Angeles merupakan kota yang besar dan memiliki tingkat kriminalitas yang tinggi bahkan pada saat pandemi COVID-19 tingkat kriminalitas pada kota Los Angeles tidak mengalami penurunan yang signifikan sehingga kita harus tetap waspada karena Amerika seperti yang diketahui memiliki hukum dimana warga negara nya dapat mempunyai senjata secara legal.

Kota Los Angeles memiliki data yang melimpah terkait kriminalitas. Data ini mencakup laporan polisi, catatan kejahatan, dan informasi terkait yang dikumpulkan selama bertahun-tahun. Data ini merupakan sumber daya berharga untuk menganalisis dan memahami pola kriminalitas.

Data science dapat memberikan wawasan yang mendalam tentang faktor-faktor yang mempengaruhi kriminalitas. Analisis data dapat membantu dalam mengidentifikasi pola dan tren yang dapat digunakan untuk mengembangkan strategi yang lebih efektif dalam menangani kriminalitas. Penggunaan teknologi data science dan kecerdasan buatan (AI) dapat menghasilkan pemahaman yang lebih dalam tentang kriminalitas dan mendorong inovasi dalam penanganan kejahatan.

Penggunaan bahasa pemrograman Python, bersama dengan berbagai library dan alat analisis data, memungkinkan kita untuk merumuskan solusi yang cerdas dalam menangani masalah kriminalitas. Matplotlib, sebagai salah satu library Python yang kuat, juga memainkan peran penting dalam visualisasi data yang informatif. Melalui penggunaan Python dan Matplotlib dalam analisis dan visualisasi data, proyek ini diharapkan dapat memberikan pandangan yang lebih baik tentang masalah kriminalitas di Kota Los Angeles dan membantu dalam perumusan kebijakan-kebijakan yang lebih efektif untuk meningkatkan keamanan masyarakat.

Beberapa sumber yang dijelaskan dalam daftar referensi memiliki latar belakang yang sangat relevan dalam pemahaman kriminalitas dan analisis data. "Faktor-Faktor yang Mempengaruhi Tingkat Kriminalitas di Indonesia Tahun 2018" memberikan wawasan tentang faktor-faktor yang mempengaruhi kriminalitas di Indonesia, sementara "Crime and Criminology: an Introduction" adalah buku teks yang memberikan dasar-dasar teoritis dalam studi kriminologi. "Crime Rate Inference with Big Data" mencerminkan pentingnya big data dalam menginterpretasikan tingkat kriminalitas. "Impact of social distancing during COVID-19 pandemic on crime in Los Angeles and Indianapolis" menyoroti dampak peristiwa eksternal seperti pandemi pada tingkat kriminalitas. "Tingkat Kriminalitas (Studi Pada 31 Provinsi Di Indonesia Tahun)" memberikan data kriminalitas di tingkat regional, yang relevan untuk pemahaman situasi lokal. Selanjutnya, buku "Introduction to Data Science" dan referensi terkaitnya tentang Python, seperti "Pandas" dan "Matplotlib," penting dalam pemrosesan dan visualisasi data kriminalitas. "Big Data and Visuality" dan "Criminology in the age of data explosion" memberikan wawasan tentang dampak ledakan data pada kriminologi dan bagaimana analisis data menjadi alat penting dalam pemahaman masalah kriminalitas. Keseluruhan, referensi ini membantu membentuk latar belakang yang kuat untuk penelitian dan analisis data kriminalitas yang komprehensif.

Motivasi

Project ini sangat relevan dan bermakna dalam hal keselamatan dan kualitas hidup penduduk Los Angeles. Menganalisis data kejahatan kota antara tahun 2020 dan 2023 dapat memberikan kontribusi yang signifikan bagi masyarakat, pemerintah, dan penegakan hukum.

Motivasi utama project ini adalah untuk memberikan wawasan berharga yang akan membantu meningkatkan keselamatan dan kualitas hidup penduduk Los Angeles. Dengan menyelidiki data kejahatan selama tiga tahun terakhir, kita mempunyai peluang untuk mengidentifikasi pola dan tren yang mungkin terlewatkan. Hal ini tidak hanya membantu penegak hukum mengalokasikan sumber daya secara lebih efisien, namun juga membuka pintu untuk mengembangkan strategi pencegahan yang lebih cerdas dan efektif.

Hal ini merupakan peluang untuk mengembangkan dan menguji keterampilan analisis data, yang merupakan aset berharga di era informasi. Pengalaman ini tidak hanya akan memperkaya pemahaman tentang kompleksitas data kriminal, namun juga berfungsi sebagai batu loncatan untuk membangun karir di bidang ilmu data dan analitik.

Project ini juga dapat memotivasi masyarakat untuk terlibat dalam upaya pencegahan kejahatan. Laporan ini diharapkan dapat menjadi inspirasi untuk penelitian lanjutan, meletakkan dasar bagi pemahaman yang lebih mendalam tentang tantangan kejahatan di Los Angeles dan solusi berkelanjutan.

Project ini merupakan inisiatif proaktif untuk membawa perubahan positif dalam kehidupan sehari-hari penduduk Los Angeles. Ini semua menjadi motivasi utama kami untuk menjalankan project ini dengan komitmen dan antusiasme penuh.

Kami berharap agar project ini dapat diterapkan bukan hanya di Los Angeles namun bisa diterapkan di manapun khususnya Indonesia.

Tujuan

Ada beberapa tujuan yang ingin kami capai dari penelitian ini, antara lain:

- Membuat visualisasi data yang jelas dan informatif untuk memudahkan pemahaman hasil analisis. Visualisasi ini dapat membantu masyarakat dan pemangku kepentingan lainnya dalam menginterpretasikan temuan.
- Membuat bar chart untuk menampilkan jenis kejahatan tertentu,
- Membuat pie chart untuk jumlah kejahatan per tahun,
- pie chart untuk gender distribution, linechart untuk melihat tren kejahatan selama periode 2020-2023,
- bar chart untuk menampilkan jumlah kejahatan per area,
- menampilkan jumlah kejahatan yang sudah dicluster,
- menampilkan tiap kejahatan yang terjadi di setiap daerah (sementara)
- Membuat prediksi jumlah kejahatan yang akan terjadi pada bulan-bulan selanjutnya dengan menggunakan time series.

B. Metode

Deskripsi Data

No	Nama Kolom	Tipe Data	Keterangan
1	DATE OCC	String	Tanggal dan waktu sering direpresentasikan dalam format string untuk memudahkan pembacaan.
2	TIME OCC	Integer	Waktu dalam format 24 jam dapat direpresentasikan dengan tipe data integer karena tidak memerlukan desimal atau komponen fraksional.
3	AREA	Integer	Kode area adalah nilai numerik yang mengidentifikasi wilayah tertentu. Representasi sebagai bilangan bulat memungkinkan pengindeksan dan pengelompokan wilayah.
4	AREA NAME	String	Nama wilayah diwakili sebagai string karena merupakan teks deskriptif dan tidak memerlukan operasi matematis atau pemrosesan lain yang kompleks.
5	Rpt Dist No	Integer	Nomor distrik laporan adalah nilai numerik yang mengidentifikasi distrik laporan. Representasi sebagai bilangan bulat memungkinkan pengindeksan dan pengelompokan distrik.
6	Part 1-2	Integer	Kategori kejahatan biasanya direpresentasikan dengan angka atau kode numerik untuk kenyamanan pemrosesan.
7	Crm Cd	Integer	Kode kategori kejahatan adalah angka atau kode numerik yang mengidentifikasi jenis kejahatan tertentu.

8	Crm Cd Desc	String	Deskripsi kategori kejahatan direpresentasikan sebagai string karena berisi teks deskriptif tentang jenis kejahatan.
9	Mocodes	String	Kode modus operandi mungkin berisi kombinasi angka atau teks yang menggambarkan cara kejahatan dilakukan. Representasi sebagai string memungkinkan fleksibilitas dalam format.
10	Vict Age	Integer	Usia korban adalah nilai numerik yang direpresentasikan sebagai bilangan bulat untuk memungkinkan operasi matematis.
11	Vict Sex	String	Jenis kelamin korban direpresentasikan sebagai string karena merupakan teks deskriptif.
12	Vict Descent	String	Kode etnis atau ras korban direpresentasikan sebagai string karena berisi teks deskriptif tentang karakteristik etnis atau ras.
13	Premis Cd	Integer	Kode jenis tempat kejadian adalah nilai numerik yang mengidentifikasi jenis tempat kejadian tertentu.
14	Premis Desc	String	Deskripsi tempat kejadian direpresentasikan sebagai string karena berisi teks deskriptif tentang jenis tempat.
15	Weapon Used Cd	Integer	Kode senjata yang digunakan adalah nilai numerik yang mengidentifikasi jenis senjata.
16	Weapon Desc	String	Deskripsi senjata yang digunakan direpresentasikan sebagai string karena berisi teks deskriptif tentang jenis senjata.

17	Status	String	Kode status kejadian adalah nilai yang mewakili status tertentu. Representasi sebagai string memungkinkan fleksibilitas dalam format.
18	Status desc	String	Deskripsi status kejadian direpresentasikan sebagai string karena berisi teks deskriptif tentang status tertentu.
19	Crm Cd 1, Crm Cd 2, Crm Cd 3, Crm Cd 4	Integer	Kategori kejahatan tambahan biasanya direpresentasikan dengan angka atau kode numerik.
20	LOCATION	String	Alamat tempat kejadian direpresentasikan sebagai string karena berisi teks deskriptif tentang lokasi.
21	Cross Street	String	Nama jalan lintas adalah teks deskriptif tentang jalan yang bersilangan (jika ada).
22	LAT	Float	Koordinat lintang adalah nilai numerik dengan komponen desimal yang mengidentifikasi lokasi secara geografis.
23	LON	Float	Koordinat bujur adalah nilai numerik dengan komponen desimal yang mengidentifikasi lokasi secara geografis.

Teknik dan Algoritma:

Time series dengan menggunakan

Penjelasan Coding:

- Import dan Membaca data:

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

```
df = pd.read_csv('Crime_Data.csv')
```

1.1 Code Import Library

Pada coding diatas kita melakukan import library pandas dan matplotlib.pyplot kemudian kita akan menggunakan Crime_Data.csv sebagai data utama dalam penelitian ini dengan menggunakan fungsi pada pandas yaitu pd.read_csv untuk membaca file csv..

- Battery - Simple Assault

```
# Convert the 'Date Rptd' column to datetime
data['Date Rptd'] = pd.to_datetime(data['Date Rptd'])

# Filter the data for the year 2020
data_2020 = data[data['Date Rptd'].dt.year == 2020]

# Filter data for 'BATTERY - SIMPLE ASSAULT'
battery_simple_assault = data_2020[data_2020['Crm Cd Desc'] == 'BATTERY - SIMPLE ASSAULT']

# Group the data by month and count occurrences
monthly_counts = battery_simple_assault.groupby(data_2020['Date Rptd'].dt.month)['Crm Cd Desc'].count()

# Define colors for each bar
colors = ['blue', 'green', 'red', 'purple', 'orange', 'pink', 'brown', 'gray', 'cyan', 'magenta', 'lime', 'yellow']

# Create a bar chart with different colors for each month
plt.figure(figsize=(10, 6))
ax = monthly_counts.plot(kind='bar', color=colors)
plt.xlabel('Month')
plt.ylabel('Count')
plt.title('BATTERY - SIMPLE ASSAULT Incidents in 2020 by Month')
plt.xticks(range(12), ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'], rotation=0)

# Add text annotations for each bar
for i, count in enumerate(monthly_counts):
    ax.text(i, count, str(count), ha='center', va='bottom')
plt.savefig("simple assault 2020.png")
plt.show()
```

1.2 Code BATTERY - SIMPLE ASSAULT

Pertama kita menggunakan data['Date Rptd'] = pd.to_datetime(data['Date Rptd']) untuk mengkonversi kolom 'Date Rptd' dalam DataFrame menjadi format datetime. Ini memungkinkan kita untuk bekerja dengan tanggal dalam analisis data.

battery_simple_assault = data_2020[data_2020['Crm Cd Desc'] == 'BATTERY - SIMPLE ASSAULT']: Baris ini menyaring data untuk jenis kejahatan 'BATTERY - SIMPLE ASSAULT'. Hasilnya adalah DataFrame yang berisi informasi tentang

kejahatan tersebut yang terjadi pada tahun 2020, kita dapat mengubah tahun yang diinginkan seperti 2021 - 2023.

```
monthly_counts = battery_simple_assault.groupby(data_2020['Date Rptd'].dt.month)['Crm Cd Desc'].count():
```

 Ini mengelompokkan data berdasarkan bulan dan menghitung jumlah kejadian. Hasilnya adalah Series yang berisi jumlah kejahatan "BATTERY - SIMPLE ASSAULT" yang terjadi setiap bulan dalam tahun 2020, kita dapat mengubah tahun yang diinginkan seperti 2021 - 2023

```
colors = ['blue', 'green', 'red', 'purple', 'orange', 'pink', 'brown', 'gray', 'cyan', 'magenta', 'lime', 'yellow']
```

 : kita menggunakan colors untuk mendefinisikan daftar warna yang akan digunakan untuk menggambar batang dalam grafik batang, 12 warna yang digunakan karena pada satu grafik terdapat 12 batang.

Baris `plt.figure(figsize=(10, 6))` dijalankan untuk menciptakan sebuah figur untuk grafik dengan ukuran 10 x 6 inci.

Kemudian baris `ax = monthly_counts.plot(kind='bar', color=colors)` berfungsi untuk membuat grafik batang dengan menggunakan data dari "monthly_counts" dan mengatur warna batang sesuai dengan daftar warna yang telah didefinisikan

```
plt.xlabel('Month')
```

```
plt.ylabel('Count')
```

 digunakan untuk menambahkan label sumbu x dan sumbu y ke grafik.

```
plt.title('BATTERY - SIMPLE ASSAULT Incidents in 2020 by Month')
```

 untuk menambahkan judul dan `plt.xticks(range(12), ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'], rotation=0)` untuk menambahkan label setiap bulan pada x

Terakhir kita gunakan `plt.show()` untuk menampilkan grafik pada output

- Rape Forcible

```
# Convert the 'Date Rptd' column to datetime
data['Date Rptd'] = pd.to_datetime(data['Date Rptd'])

# Filter the data for the year 2020
data_2020 = data[data['Date Rptd'].dt.year == 2020]

# Filter data for 'RAPE, FORCIBLE'
rape_forcible = data_2020[data_2020['Crm Cd Desc'] == 'RAPE, FORCIBLE']

# Group the data by month and count occurrences
monthly_counts = rape_forcible.groupby(data_2020['Date Rptd'].dt.month)['Crm Cd Desc'].count()

# Define colors for each bar
colors = ['blue', 'green', 'red', 'purple', 'orange', 'pink', 'brown', 'gray', 'cyan', 'magenta', 'lime', 'yellow']

# Create a bar chart with different colors for each month
plt.figure(figsize=(10, 6))
ax = monthly_counts.plot(kind='bar', color=colors)
plt.xlabel('Month')
plt.ylabel('Count')
plt.title('Rape, Forcible Incidents in 2020 by Month')
plt.xticks(range(12), ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'], rotation=0)

# Add text annotations for each bar
for i, count in enumerate(monthly_counts):
    ax.text(i, count, str(count), ha='center', va='bottom')
plt.savefig("rape forcible 2020.png")
plt.show()
```

1.3 Code Rape Forcible

Baris pertama kode mengonversi kolom 'Date Rptd' dalam dataset ke format datetime menggunakan fungsi `pd.to_datetime` dari pustaka Pandas. Ini penting untuk operasi berbasis tanggal selanjutnya.

Selanjutnya, data di filter untuk hanya mencakup catatan dari tahun 2020 dengan menggunakan variabel `data_2020`. Ini menggunakan atribut `dt.year` untuk mengekstrak tahun dari kolom 'Date Rptd'.

Selanjutnya, dilakukan operasi filtering lagi untuk mengisolasi kejadian 'RAPE, FORCIBLE' dengan menggunakan variabel `rape_forcible`. Ini dicapai dengan memilih baris di mana kolom 'Crm Cd Desc' sama dengan 'RAPE, FORCIBLE' dalam data untuk tahun 2020.

Kode kemudian menggunakan fungsi `groupby` untuk mengelompokkan kejadian 'RAPE, FORCIBLE' berdasarkan bulan, menggunakan kolom 'Date Rptd'. Hasilnya disimpan dalam variabel `monthly_counts`, yang mewakili jumlah kejadian untuk setiap bulan.

Yang menjadi pembeda pada kedua coding diatas adalah pada bagian `rape_forcible = data_2020[data_2020['Crm Cd Desc'] == 'RAPE, FORCIBLE']`, kita mengganti kolom yang digunakan menjadi 'RAPE, FORCIBLE'

- Crime Incident

```
# Convert the 'Date Rptd' column to datetime
data['Date Rptd'] = pd.to_datetime(data['Date Rptd'])

# Extract the year from the 'Date Rptd' column
data['Year'] = data['Date Rptd'].dt.year

# Group the data by year and count occurrences
yearly_counts = data['Year'].value_counts().sort_index()

# Create a pie chart
plt.figure(figsize=(8, 8))
plt.pie(yearly_counts, labels=yearly_counts.index, autopct='%1.1f%%', startangle=140)
plt.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.
plt.title('Crime Incidents in Los Angeles by Year')

plt.savefig("pie chart crime incidents.png")
plt.show()
```

1.4 Code Crime Incident

Pertama kita gunakan `data['Date Rptd'] = pd.to_datetime(data['Date Rptd'])` untuk mengonversi kolom 'Date Rptd' dalam Data Frame menjadi format datetime. Ini memungkinkan kita untuk bekerja dengan tanggal dalam analisis data.

Kemudian pada baris `data['Year'] = data['Date Rptd'].dt.year`: dijalankan untuk mengekstrak tahun dari kolom 'Date Rptd' dan menyimpannya dalam kolom baru bernama 'Year' dalam DataFrame.

`yearly_counts = data['Year'].value_counts().sort_index()`: Ini menghitung jumlah kejadian berdasarkan tahun dengan menggunakan kolom 'Year' dan kemudian mengurutkannya berdasarkan tahun. Hasilnya adalah Series yang berisi jumlah kejadian kejahatan per tahun.

Selanjutnya kita akan mendefinisikan pie chart menggunakan `plt.pie(yearly_counts, labels=yearly_counts.index, autopct='%1.1f%%', startangle=140)`: Ini membuat grafik pie dengan menggunakan data dari "yearly_counts". Parameter labels digunakan untuk menampilkan label tahun pada setiap potongan pie, autopct digunakan untuk menampilkan persentase di setiap potongan, dan startangle mengatur sudut awal dari pie chart.

- Distribution Gender

```
# Group the data by 'Vict Sex' and count occurrences
gender_counts = data['Vict Sex'].value_counts()

# Create a pie chart
plt.figure(figsize=(6, 6))
plt.pie(gender_counts, labels=gender_counts.index, autopct='%1.1f%%', startangle=140)
plt.axis('equal')
plt.title('Distribution of Victim Gender in Crime Incidents')

# Add a legend
plt.legend(title='Gender', labels=gender_counts.index, loc='upper right')
plt.savefig("pie chart victim gender.png")
plt.show()
```

1.5 Code Distribution Gender

Pertama, kita menggunakan `pd.read_csv('Crime_Data.csv')` untuk memuat dataset dari file 'Crime_Data.csv' ke dalam DataFrame data.

Kode ini mengelompokkan data berdasarkan kolom 'Vict Sex' (jenis kelamin korban). Kemudian, kita menggunakan `value_counts()` untuk menghitung jumlah munculnya setiap jenis kelamin (misalnya, berapa banyak korban laki-laki dan berapa banyak korban perempuan).

Kode ini membuat pie chart dengan `plt.pie()` untuk menggambarkan distribusi jenis kelamin korban.

`gender_counts` digunakan sebagai data yang digunakan dalam pie chart.

`labels=gender_counts.index` digunakan untuk memberikan label pada setiap sektor pie chart berdasarkan jenis kelamin.

`autopct='%1.1f%%'` digunakan untuk menampilkan persentase masing-masing sektor dalam format desimal dengan satu angka di belakang koma.

`startangle=140` mengatur sudut awal pie chart.

Kode ini menambahkan legend ke pie chart dengan menggunakan `plt.legend()`.

`title='Gender'` digunakan untuk memberi judul pada legend.

`labels=gender_counts.index` digunakan untuk menambahkan label jenis kelamin (laki-laki dan perempuan) ke legend.

`loc='upper right'` mengatur posisi legend di sudut kanan atas grafik.

- Frequency of Simple Assault

```
# Convert the 'Date Rptd' column to datetime
data['Date Rptd'] = pd.to_datetime(data['Date Rptd'])

# Filter the data for the year 2020 and 'Battery - Simple Assault' crime
data_2020_battery_assault = data[(data['Date Rptd'].dt.year == 2020) & (data['Crm Cd Desc'] == 'BATTERY - SIMPLE ASSAULT')]

# Group the data by month and count occurrences
monthly_counts = data_2020_battery_assault.groupby(data_2020_battery_assault['Date Rptd'].dt.month)['Crm Cd Desc'].count()

# Create a line chart to visualize the trend
plt.figure(figsize=(12, 6))
plt.plot(monthly_counts.index, monthly_counts, marker='o', linestyle='--')
plt.xlabel('Month')
plt.ylabel('Frequency')
plt.title('Battery - Simple Assault Incidents in Los Angeles (2020)')
plt.xticks(range(1, 13), ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'])
plt.savefig("linechart frequency simple assault 2020.png")
plt.show()
```

Gambar 1.6 Code Frequency of Simple Assault

Baris ini mengonversi kolom 'Date Rptd' ke tipe data datetime. Ini penting agar kita dapat mengambil tahun dari tanggal pelaporan kejahatan.

Kode ini menghasilkan DataFrame `data_2020_battery_assault` yang merupakan subset dari data awal.

Hanya data yang terjadi pada tahun 2020 (`data['Date Rptd'].dt.year == 2020`) dan jenis kejahatan "Battery - Simple Assault" (`data['Crm Cd Desc'] == 'BATTERY - SIMPLE ASSAULT'`) yang akan disertakan dalam analisis.

Kode ini mengelompokkan data berdasarkan bulan (dalam bentuk angka) dengan menggunakan `groupby`. Kemudian, kita menghitung jumlah kejahatan dengan menggunakan `.count()`.

Hasilnya adalah Series `monthly_counts` yang berisi frekuensi kejahatan "Battery - Simple Assault" per bulan.

Kode ini membuat line chart dengan `plt.plot()`.

`monthly_counts.index` digunakan sebagai sumbu x (bulan).

`monthly_counts` adalah data yang digunakan pada sumbu y (frekuensi kejahatan).

`marker='o'` mengatur penanda pada titik data, dan `linestyle='--'` mengatur jenis garis.

`.xlabel('Month')` dan `.ylabel('Frequency')` digunakan untuk memberi label sumbu x dan y.

`.title('Battery - Simple Assault Incidents in Los Angeles (2020)')` memberi judul grafik.

`.xticks()` mengatur label bulan pada sumbu x. Terakhir `plt.show()` untuk menampilkan line chart ke dalam jendela grafik.

- Autocorrelation:

```
crime_per_day = df.resample('M').size()

# Print beberapa baris pertama dari data jumlah kejahatan per hari
print(crime_per_day.head())

# Plot fungsi autocorrelation
from statsmodels.graphics.tsaplots import plot_acf
plot_acf(crime_per_day)
plt.title('Autocorrelation Plot')
plt.show()
```

Gambar 1.7 Code Autocorrelation

Kode ini adalah untuk melakukan analisis deret waktu pada DataFrame (df) yang berisi data kejahatan.

Baris pertama, `crime_per_day = df.resample('M').size()`, menghitung jumlah kejahatan per bulan dengan menggunakan metode resampling pada deret waktu. Fungsi `resample` digunakan untuk mengubah frekuensi data, dan dalam hal ini, 'M' menunjukkan bahwa kita ingin menghitung jumlah kejahatan per bulan. Metode `size()` menghitung jumlah kejadian dalam setiap periode bulanan. Hasilnya, yang merupakan deret waktu baru, disimpan dalam variabel `crime_per_day`.

Berikutnya, `print(crime_per_day.head())` digunakan untuk mencetak beberapa baris pertama dari data jumlah kejahatan per bulan. Ini memberikan gambaran awal tentang pola dan tren data.

Selanjutnya, dilakukan plotting terhadap fungsi autocorrelation (autokorelasi) dari deret waktu menggunakan `plot_acf` dari pustaka `statsmodels`. Autokorelasi adalah ukuran seberapa erat hubungan antara nilai dalam deret waktu dengan nilai-nilai sebelumnya. Grafik autokorelasi berguna untuk mengidentifikasi pola berulang dalam data deret waktu. Setelah plotting, sebuah judul ditambahkan menggunakan `plt.title`. Terakhir, diagram tersebut ditampilkan menggunakan `plt.show()`.

- Partial Autocorrelation:

```
# Import library yang diperlukan
from statsmodels.graphics.tsaplots import plot_pacf

# Plot Partial Autocorrelation Function (PACF)
plot_pacf(crime_per_day, lags=20) # 'lags' menunjukkan jumlah lag yang akan diplot
plt.title('Partial Autocorrelation Plot')

plt.show()
```

Gambar 1.8 Code Partial Autocorrelation

Kode ini menganalisis deret waktu lebih lanjut dengan fokus pada fungsi Autocorrelation Parsial (PACF).

Baris pertama adalah `from statsmodels.graphics.tsaplots import plot_pacf`, yang mengimpor fungsi `plot_pacf` dari pustaka `Statsmodels`. PACF digunakan untuk mengukur korelasi antara nilai dalam deret waktu dan nilai-nilai sebelumnya dengan mengontrol efek lag-lag di antaranya.

Selanjutnya, `plot_pacf(crime_per_day, lags=20)` menghasilkan plot fungsi Autocorrelation Parsial dari deret waktu `crime_per_day`. Parameter `lags` menentukan jumlah lag yang akan diplot pada grafik, dan dalam hal ini, diatur ke 20. Fungsi PACF membantu mengidentifikasi dampak lag-lag tertentu pada nilai dalam deret waktu.

Baris selanjutnya, `plt.title('Partial Autocorrelation Plot')`, menambahkan judul ke plot yang dihasilkan, memberikan informasi bahwa ini adalah grafik Autocorrelation Parsial dan kemudian ditampilkan menggunakan `plt.show()`.

- ARIMA

```
from statsmodels.tsa.arima.model import ARIMA
# ARIMA Model
# Choose the order based on ACF and PACF plot
order = (2, 0, 1)

# Create ARIMA model
model = ARIMA(crime_per_day, order=order)
results = model.fit()

# Print model summary
print(results.summary())
```

Gambar 1.9 Code ARIMA

Kode diatas merupakan implementasi dari model ARIMA untuk menganalisis deret waktu `crime_per_day`.

Baris pertama, `from statsmodels.tsa.arima.model import ARIMA`, mengimpor model ARIMA dari pustaka `Statsmodels`.

Berikutnya, komentar `# ARIMA Model` memberikan informasi bahwa kita akan membuat model ARIMA.

Baris selanjutnya, `order = (2, 0, 1)`, menentukan urutan model ARIMA yang akan dibuat. Angka dalam tupel (p, d, q) menunjukkan urutan komponen AR (Autoregressive), I (Integrated), dan MA (Moving Average) dari model ARIMA.

Dalam hal ini, nilai (2, 0, 1) dipilih berdasarkan analisis plot ACF dan PACF sebelumnya.

Baris model = ARIMA(crime_per_day, order=order) menciptakan model ARIMA menggunakan deret waktu crime_per_day dan urutan yang telah ditentukan.

Selanjutnya, results = model.fit() melatih model pada data dan menyimpan hasilnya dalam variabel results.

Baris terakhir, print(results.summary()), mencetak ringkasan hasil dari model ARIMA yang telah dilatih. Ringkasan ini berisi informasi penting tentang kinerja model, parameter yang diestimasi, dan sejumlah metrik evaluasi.

- Tabel Jumlah Kejahatan Pada Setiap Area

```
# Membersihkan data
df[['LAT', 'LON']] = df[['LAT', 'LON']].fillna(0)

# Menampilkan jumlah kejahatan di setiap area name
crime_counts_per_area_name = df['AREA NAME'].value_counts().reset_index()
crime_counts_per_area_name.columns = ['Area Name', 'Jumlah Kejahatan']

# Menampilkan tabel
print(crime_counts_per_area_name)
```

Gambar 1.10 Code Tabel Jumlah Kejahatan Pada Setiap Area

Baris pertama, df[['LAT', 'LON']] = df[['LAT', 'LON']].fillna(0), menggantikan nilai-nilai yang hilang (NaN) dalam kolom 'LAT' dan 'LON' dengan nilai 0. Hal ini dapat membantu dalam pemrosesan data geografis dengan memastikan bahwa setiap baris memiliki nilai yang valid.

Baris berikutnya, crime_counts_per_area_name = df['AREA NAME'].value_counts().reset_index(), menghitung jumlah kejahatan di setiap area dan menyimpan hasilnya dalam DataFrame baru crime_counts_per_area_name. Fungsi value_counts() digunakan untuk menghitung frekuensi setiap nilai unik dalam kolom 'AREA NAME', dan reset_index() digunakan untuk mengubah hasilnya menjadi DataFrame agar lebih mudah diolah.

Baris selanjutnya, crime_counts_per_area_name.columns = ['Area Name', 'Jumlah Kejahatan'], mengganti nama kolom dari DataFrame hasil perhitungan sebelumnya. Ini dilakukan untuk memberikan nama kolom yang lebih deskriptif.

Baris terakhir, print(crime_counts_per_area_name), mencetak tabel yang berisi jumlah kejahatan di setiap area. Tabel ini memberikan gambaran yang jelas tentang distribusi kejahatan di berbagai wilayah (area) yang terdapat dalam data.

- Perbandingan Tingkat Kejahatan di Setiap Area

```
import seaborn as sns

# Membersihkan data
df[['LAT', 'LON']] = df[['LAT', 'LON']].fillna(0)

# Menghitung jumlah kejahatan di setiap area name
crime_counts_per_area_name = df['AREA NAME'].value_counts().reset_index()
crime_counts_per_area_name.columns = ['Area Name', 'Jumlah Kejahatan']

# Mengurutkan berdasarkan jumlah kejahatan
crime_counts_per_area_name = crime_counts_per_area_name.sort_values(by='Jumlah Kejahatan', ascending=False)

# Visualisasi perbandingan tingkat kejahatan di setiap area name
plt.figure(figsize=(12, 6))
sns.barplot(x='Jumlah Kejahatan', y='Area Name', data=crime_counts_per_area_name, palette='viridis')
plt.title('Perbandingan Tingkat Kejahatan di Setiap Area Name')
plt.xlabel('Jumlah Kejahatan')
plt.ylabel('Area Name')
plt.show()
```

Gambar 1.11 Code Perbandingan Tingkat Kejahatan di Setiap Area

Kode di atas digunakan untuk membersihkan data, menghitung jumlah kejahatan di setiap area, mengurutkan hasil perhitungan, dan memvisualisasikan perbandingan tingkat kejahatan di setiap area.

Baris pertama, `import seaborn as sns`, mengimpor pustaka Seaborn, yang akan digunakan untuk visualisasi data.

Baris selanjutnya, `df[['LAT', 'LON']] = df[['LAT', 'LON']].fillna(0)`, mirip dengan kode sebelumnya, menggantikan nilai-nilai yang hilang (NaN) dalam kolom 'LAT' dan 'LON' dengan nilai 0.

Baris berikutnya, `crime_counts_per_area_name = df['AREA NAME'].value_counts().reset_index()`, menghitung jumlah kejahatan di setiap area dan menyimpan hasilnya dalam DataFrame baru `crime_counts_per_area_name`. Kemudian, dengan `crime_counts_per_area_name.columns = ['Area Name', 'Jumlah Kejahatan']`, nama kolom diubah untuk lebih deskriptif.

Baris selanjutnya, `crime_counts_per_area_name = crime_counts_per_area_name.sort_values(by='Jumlah Kejahatan', ascending=False)`, mengurutkan DataFrame berdasarkan jumlah kejahatan secara descending untuk memudahkan visualisasi.

Baris selanjutnya, `plt.figure(figsize=(12, 6))`, mengatur ukuran figur untuk plot yang akan dibuat.

Baris `sns.barplot(x='Jumlah Kejahatan', y='Area Name', data=crime_counts_per_area_name, palette='viridis')` memplot diagram batang

Seaborn dengan sumbu x sebagai jumlah kejahatan dan sumbu y sebagai nama area. Warna plot diatur menggunakan palet 'viridis'.

Baris `plt.title('Perbandingan Tingkat Kejahatan di Setiap Area Name')` menambahkan judul ke plot, sementara `plt.xlabel('Jumlah Kejahatan')` dan `plt.ylabel('Area Name')` menambahkan label sumbu x dan y, masing-masing.

Baris `plt.savefig('Perbandingan Tingkat Kejahatan di Setiap Area Name')` menyimpan plot sebagai gambar dengan nama "Perbandingan Tingkat Kejahatan di Setiap Area Name".

Baris terakhir, `plt.show()`, menampilkan plot secara interaktif atau dalam sebuah laporan.

- Jumlah Kejahatan per Cluster

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import KMeans # Don't forget to import KMeans
from sklearn.metrics.pairwise import linear_kernel
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer()
tfidf_matrix = vectorizer.fit_transform(df['Crm Cd Desc'])

# Melakukan clustering dengan KMeans
num_clusters = 15 # Ganti dengan jumlah cluster yang diinginkan
kmeans = KMeans(n_clusters=num_clusters, random_state=42).fit(tfidf_matrix)

# Menambahkan kolom cluster ke DataFrame
df['Cluster'] = kmeans.labels_

# Membuat kamus untuk memetakan cluster ke jenis kejahatan
cluster_names = df.groupby('Cluster')['Crm Cd Desc'].agg(lambda x: x.value_counts().index[0]).to_dict()

# Mengganti label cluster dengan nama jenis kejahatan
df['Cluster_Name'] = df['Cluster'].map(cluster_names)

# Menampilkan jumlah kejahatan per cluster dalam plot bar
plt.figure(figsize=(12, 6))
sns.countplot(x='Cluster_Name', data=df, palette='viridis')
plt.title('Jumlah Kejahatan per Cluster')
plt.xlabel('Cluster')
plt.ylabel('Jumlah Kejahatan')
plt.xticks(rotation=90)
plt.show()
```

Gambar 1.12 Code Jumlah Kejahatan per Cluster

Baris pertama dan kedua, `from sklearn.feature_extraction.text import TfidfVectorizer` dan `from sklearn.cluster import KMeans`, mengimpor kelas `TfidfVectorizer` untuk menghasilkan matriks TF-IDF dan `KMeans` untuk melakukan clustering.

Baris ketiga, `from sklearn.metrics.pairwise import linear_kernel`, mengimpor `linear_kernel` untuk menghitung kernel linear antara dua matriks.

Baris selanjutnya, `vectorizer = TfidfVectorizer()` dan `tfidf_matrix = vectorizer.fit_transform(df['Crm Cd Desc'])`, menggunakan `TfidfVectorizer` untuk mengubah kolom 'Crm Cd Desc' menjadi matriks TF-IDF. Ini adalah representasi numerik dari teks yang digunakan untuk mengukur seberapa penting sebuah kata dalam suatu dokumen relatif terhadap korpus keseluruhan.

Baris berikutnya, `num_clusters = 15` dan `kmeans = KMeans(n_clusters=num_clusters, random_state=42).fit(tfidf_matrix)`, menentukan jumlah cluster yang diinginkan (dalam contoh ini 15) dan melakukan clustering menggunakan algoritma `KMeans` pada matriks TF-IDF. Model `KMeans` disesuaikan dengan data menggunakan metode `fit`.

Baris selanjutnya, `df['Cluster'] = kmeans.labels_`, menambahkan kolom baru 'Cluster' ke `DataFrame` yang berisi label cluster dari setiap data.

Baris berikutnya, `cluster_names = df.groupby('Cluster')['Crm Cd Desc'].agg(lambda x: x.value_counts().index[0]).to_dict()`, membuat kamus yang memetakan setiap cluster ke jenis kejahatan yang paling umum dalam cluster tersebut.

Baris selanjutnya, `df['Cluster_Name'] = df['Cluster'].map(cluster_names)`, menambahkan kolom 'Cluster_Name' ke `DataFrame` yang berisi nama jenis kejahatan berdasarkan cluster.

Baris berikutnya, `plt.figure(figsize=(12, 6))`, mengatur ukuran figur untuk plot yang akan dibuat.

Baris `sns.countplot(x='Cluster_Name', data=df, palette='viridis')` memplot diagram batang Seaborn dengan sumbu x sebagai nama jenis kejahatan dan sumbu y sebagai jumlah kejahatan dalam setiap cluster.

Baris selanjutnya, `plt.title('Jumlah Kejahatan per Cluster')`, `plt.xlabel('Cluster')`, dan `plt.ylabel('Jumlah Kejahatan')`, menambahkan judul dan label sumbu pada plot.

Baris `plt.xticks(rotation=90)` digunakan untuk memutar label sumbu x agar mudah dibaca.

- Distribusi jenis kejahatan di setiap area

```
# Menambahkan kolom cluster ke DataFrame
df['Cluster'] = kmeans.labels_

# Membuat kamus untuk memetakan cluster ke jenis kejahatan
cluster_names = df.groupby('Cluster')['Crm Cd Desc'].agg(lambda x: x.value_counts().index[0]).to_dict()

# Mengganti label cluster dengan nama jenis kejahatan
df['Cluster_Name'] = df['Cluster'].map(cluster_names)

# Membuat tabel kontingensi antara Cluster_Name dan Area Name
contingency_table = pd.crosstab(df['Cluster_Name'], df['AREA NAME'])

# Menyajikan dalam grid 8x2
fig, axes = plt.subplots(nrows=8, ncols=2, figsize=(30, 45))

# Iterasi melalui setiap subplot
colors = plt.cm.tab20c.colors
for i, (ax, cluster_name) in enumerate(zip(axes.flatten(), contingency_table.index)):
    contingency_table.loc[cluster_name].plot(kind='bar', ax=ax, color=colors)
    ax.set_title(f'Cluster {i + 1}: {cluster_name}', fontsize=20)
    ax.set_xlabel('Area Name', fontsize=18)
    ax.set_ylabel('Jumlah Kejahatan', fontsize=18)

# Menyesuaikan label area agar lebih jelas
ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha='right', fontsize=14)

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

Gambar 1.13 Code Distribusi jenis kejahatan di setiap area

Baris pertama, `df['Cluster'] = kmeans.labels_`, menambahkan kolom 'Cluster' ke DataFrame yang berisi label cluster dari setiap data. Ini merupakan langkah yang sama dengan sebelumnya.

Baris selanjutnya, `cluster_names = df.groupby('Cluster')['Crm Cd Desc'].agg(lambda x: x.value_counts().index[0]).to_dict()`, membuat kamus yang memetakan setiap cluster ke jenis kejahatan yang paling umum dalam cluster tersebut. Ini serupa dengan langkah sebelumnya.

Baris berikutnya, `df['Cluster_Name'] = df['Cluster'].map(cluster_names)`, menambahkan kolom 'Cluster_Name' ke DataFrame yang berisi nama jenis kejahatan berdasarkan cluster. Ini adalah langkah yang sama dengan sebelumnya.

Baris selanjutnya, `contingency_table = pd.crosstab(df['Cluster_Name'], df['AREA NAME'])`, membuat tabel kontingensi antara 'Cluster_Name' dan 'AREA NAME'. Tabel kontingensi ini memberikan informasi tentang sebaran kejahatan di berbagai area untuk setiap cluster.

Baris berikutnya, `fig, axes = plt.subplots(nrows=8, ncols=2, figsize=(30, 45))`, membuat subplots dalam grid 8x2 untuk memvisualisasikan tabel kontingensi.

Baris selanjutnya, `for i, (ax, cluster_name) in enumerate(zip(axes.flatten(), contingency_table.index))`, melakukan iterasi melalui setiap subplot untuk menghasilkan visualisasi.

Baris `contingency_table.loc[cluster_name].plot(kind='bar', ax=ax, color=colors)` memplot diagram batang untuk setiap area dalam cluster tertentu.

Baris `ax.set_title(f'Cluster {i + 1}: {cluster_name}', fontsize=20)` menambahkan judul pada setiap subplot, mencakup informasi tentang cluster dan jenis kejahatan.

Baris `ax.set_xlabel('Area Name', fontsize=18)` dan `ax.set_ylabel('Jumlah Kejahatan', fontsize=18)` menambahkan label pada sumbu x dan y.

Baris selanjutnya, `ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha='right', fontsize=14)`, menyesuaikan label area agar lebih jelas dan mudah dibaca.

Baris terakhir, `plt.tight_layout()` dan `plt.show()`, menampilkan visualisasi tabel kontingensi dalam bentuk subplot.

- Predicted Crime

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA

# Assuming df is your DataFrame with a datetime index
crime_per_month = df[(df.index >= '2020-01-01') & (df.index <= '2023-08-31')].resample('M').size()

# Create ARIMA model
model = ARIMA(crime_per_month, order=(2, 0, 1))
results = model.fit()

# Make predictions for the next 12 months
forecast_steps = 12
forecast = results.get_forecast(steps=forecast_steps)

# Create forecast index and extract predicted mean
forecast_index = pd.date_range(crime_per_month.index[-1], periods=forecast_steps + 1, freq='M')[1:]
forecast_mean = forecast.predicted_mean

# Plot observations and predictions without confidence interval
plt.figure(figsize=(12, 6))
plt.plot(crime_per_month, label='Observations')
plt.plot(forecast_index, forecast_mean, color='red', label='Predictions')
plt.title('Predicted Number of Crimes per Month')
plt.xlabel('Date')
plt.ylabel('Number of Crimes')
plt.legend()
plt.savefig('Predicted Number of Crimes per Month.png')
plt.show()
```

Gambar 1.14 Code Predicted Crime

Kode di atas digunakan untuk membuat model ARIMA pada data kejahatan per bulan dan membuat prediksi untuk 12 bulan ke depan.

Baris pertama, `import pandas as pd, import matplotlib.pyplot as plt, dan from statsmodels.tsa.arima.model import ARIMA`, mengimpor pustaka yang diperlukan.

Baris berikutnya, `crime_per_month = df[(df.index >= '2020-01-01') & (df.index <= '2023-08-31')].resample('M').size()`, menghasilkan deret waktu yang berisi jumlah kejahatan per bulan dari tanggal 1 Januari 2020 hingga 31 Agustus 2023. Fungsi `resample('M').size()` digunakan untuk meresample data ke skala bulanan.

Baris selanjutnya, `model = ARIMA(crime_per_month, order=(2, 0, 1))`, membuat model ARIMA dengan urutan (2, 0, 1). Ini adalah nilai yang telah dipilih untuk parameter p, d, dan q berdasarkan analisis sebelumnya.

Baris berikutnya, `results = model.fit()`, melatih model pada data kejahatan bulanan.

Baris selanjutnya, `forecast_steps = 12` dan `forecast = results.get_forecast(steps=forecast_steps)`, menentukan jumlah langkah prediksi (12 bulan ke depan) dan menghasilkan prediksi menggunakan model yang telah dilatih.

Baris berikutnya, `forecast_index = pd.date_range(crime_per_month.index[-1], periods=forecast_steps + 1, freq='M')[1:]`, membuat indeks untuk prediksi dengan langkah-langkah 12 bulan ke depan.

Baris selanjutnya, `forecast_mean = forecast.predicted_mean`, mengambil nilai prediksi rata-rata dari objek hasil prediksi.

Baris `plt.plot(crime_per_month, label='Observations')` memplot data observasi kejahatan bulanan.

Baris selanjutnya, `plt.plot(forecast_index, forecast_mean, color='red', label='Predictions')`, memplot prediksi kejahatan bulanan dalam warna merah.

Baris `plt.title('Predicted Number of Crimes per Month')` menambahkan judul pada plot.

Baris `plt.xlabel('Date')` dan `plt.ylabel('Number of Crimes')` menambahkan label pada sumbu x dan y.

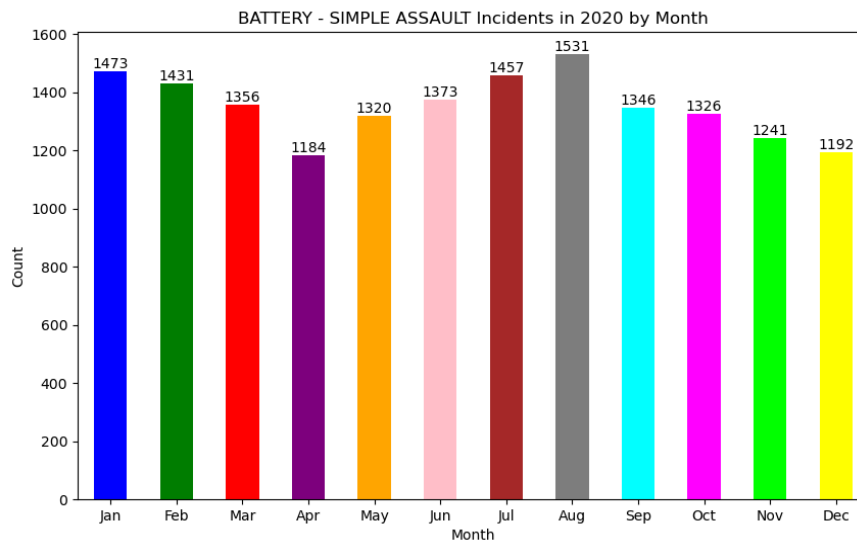
Baris `plt.legend()` menambahkan legenda pada plot.

Baris terakhir, `plt.show()`, menampilkan plot secara interaktif atau dalam sebuah laporan.

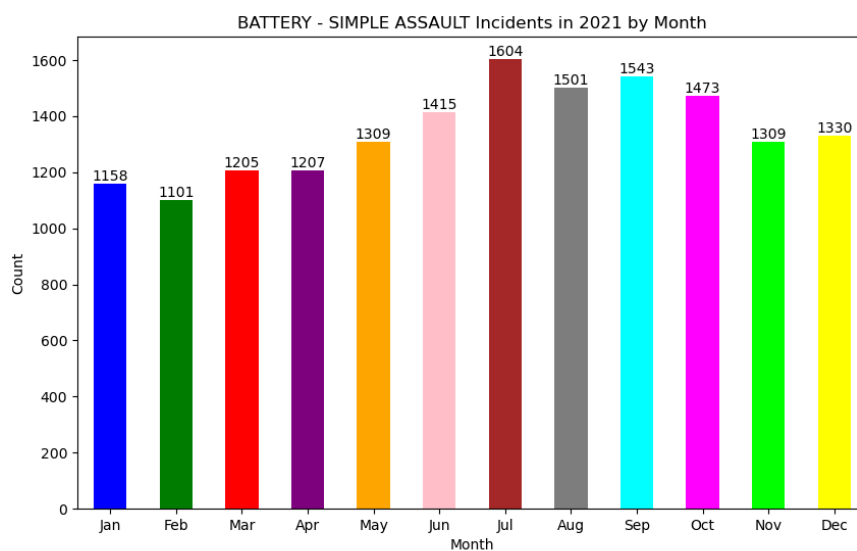
C. Hasil dan Pembahasan

Output dan Analisis

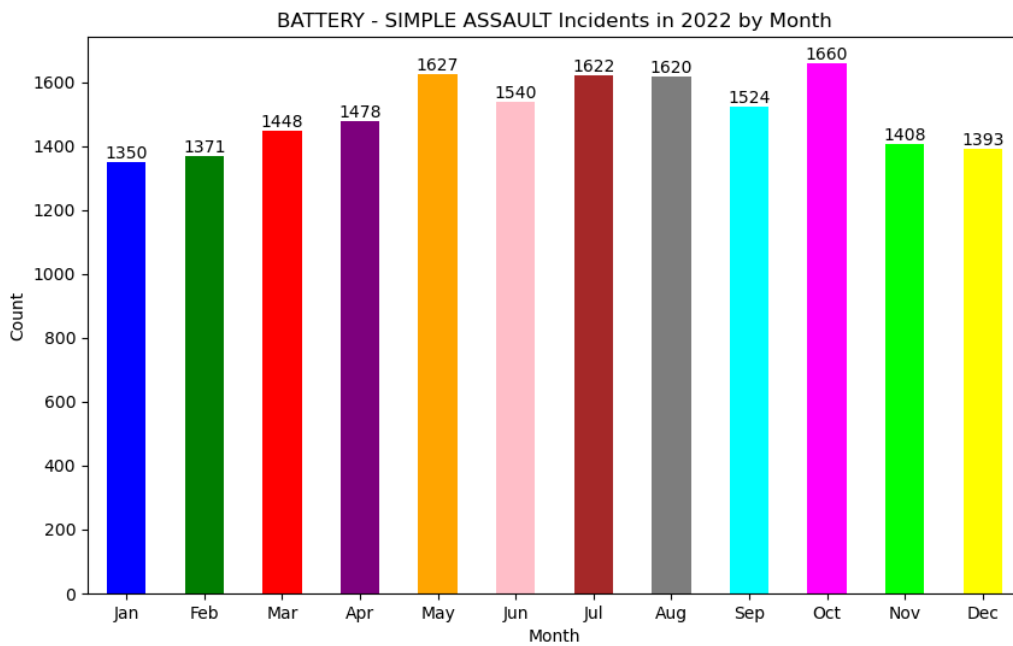
- **Battery - Simple Assault 2020-2023 by Month:**



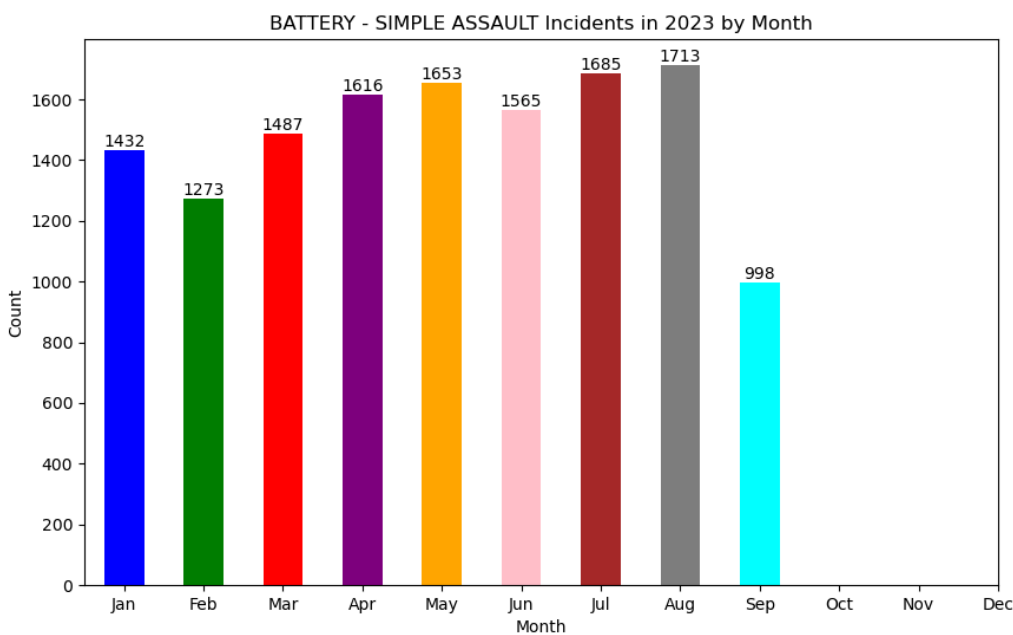
Graph 1.1.1 Tabel BATTERY SIMPLE-ASSAULT 2020



Graph 1.1.2 Tabel BATTERY SIMPLE-ASSAULT 2021



Graph 1.1.3 Tabel BATTERY SIMPLE-ASSAULT 2022

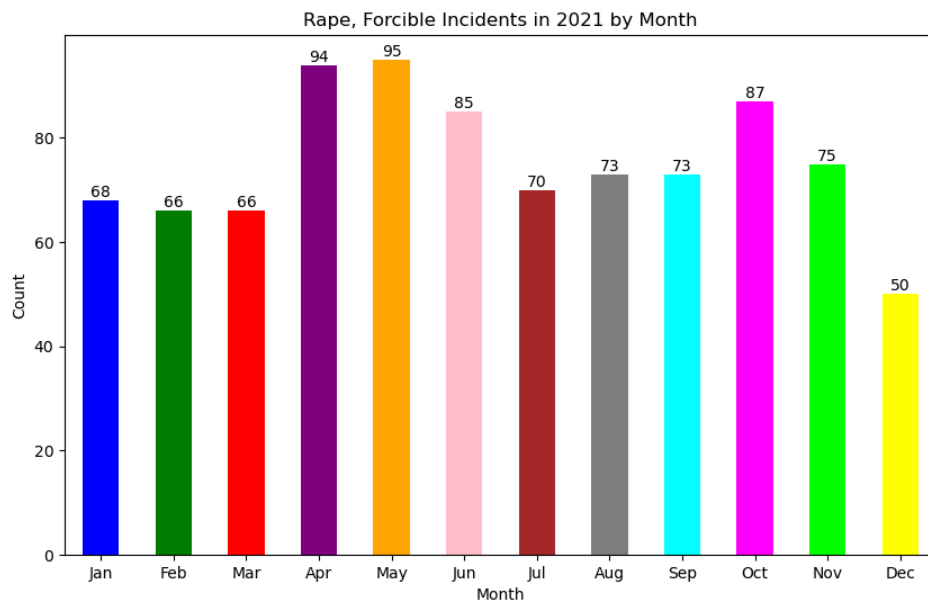


Graph 1.1.4 Tabel BATTERY SIMPLE-ASSAULT 2023

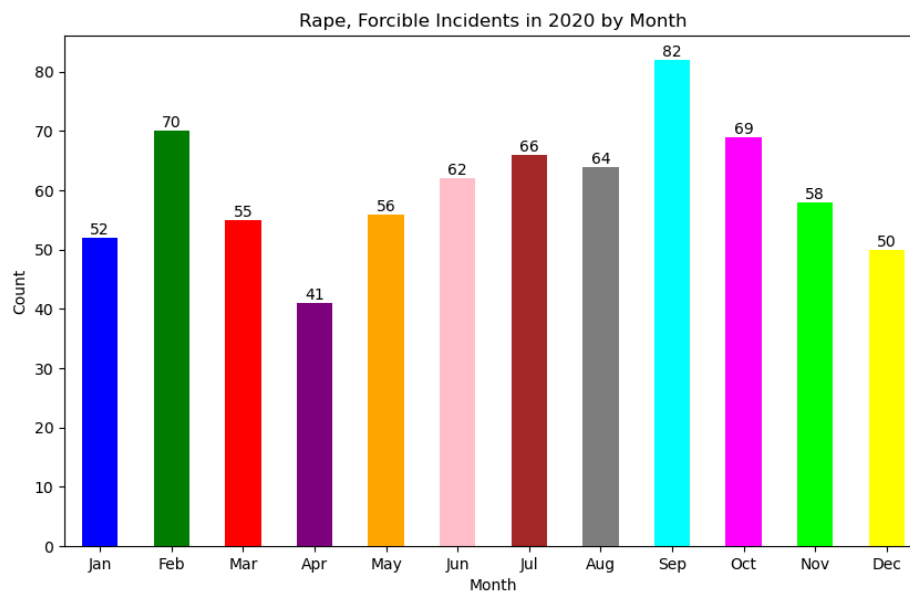
Data yang ditampilkan diatas adalah data yang menampilkan tentang Simple Assault yang terjadi pada Los Angeles yang dibagi menjadi 4 tabel berdasarkan tahun dan pada masing-masing tabel menampilkan bulan dan juga jumlah kriminalitas terjadi pada bulan tersebut.

Dari hasil plot yang ditampilkan pada tahun 2020, kejahatan BATTERY - SIMPLE ASSAULT terjadi paling banyak pada bulan Agustus dan paling rendah di bulan April. Pada tahun 2021 kejahatan tertinggi ada di bulan Juli dan paling rendah ada di bulan Februari. Pada tahun 2022, kejahatan tertinggi berada di bulan Oktober dan paling rendah berada di bulan Januari. Pada tahun 2023, kejahatan tertinggi ada di bulan Agustus dan paling rendah ada di bulan September.

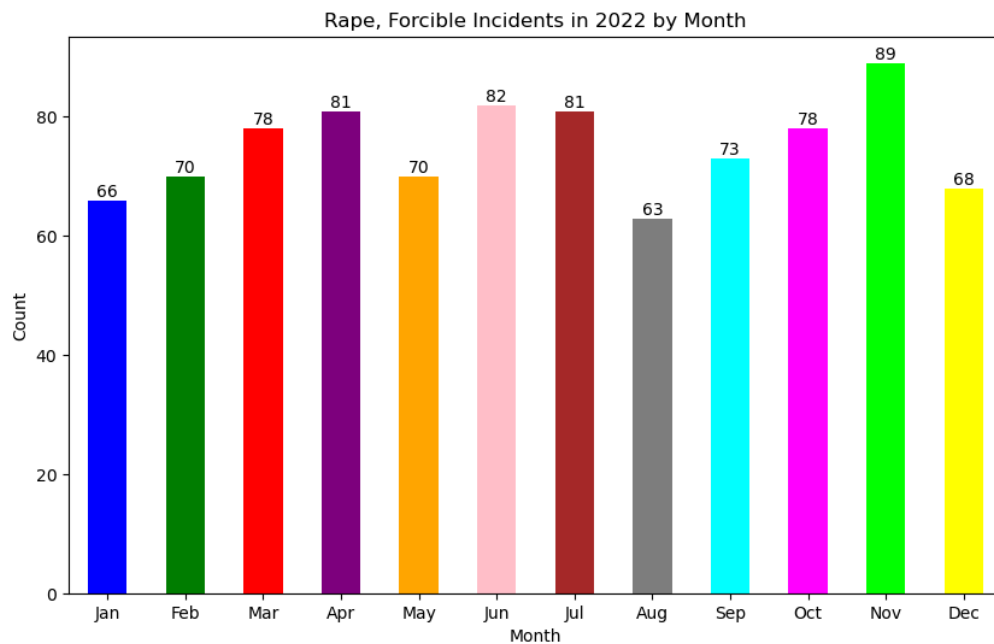
- **RapeForcible:**



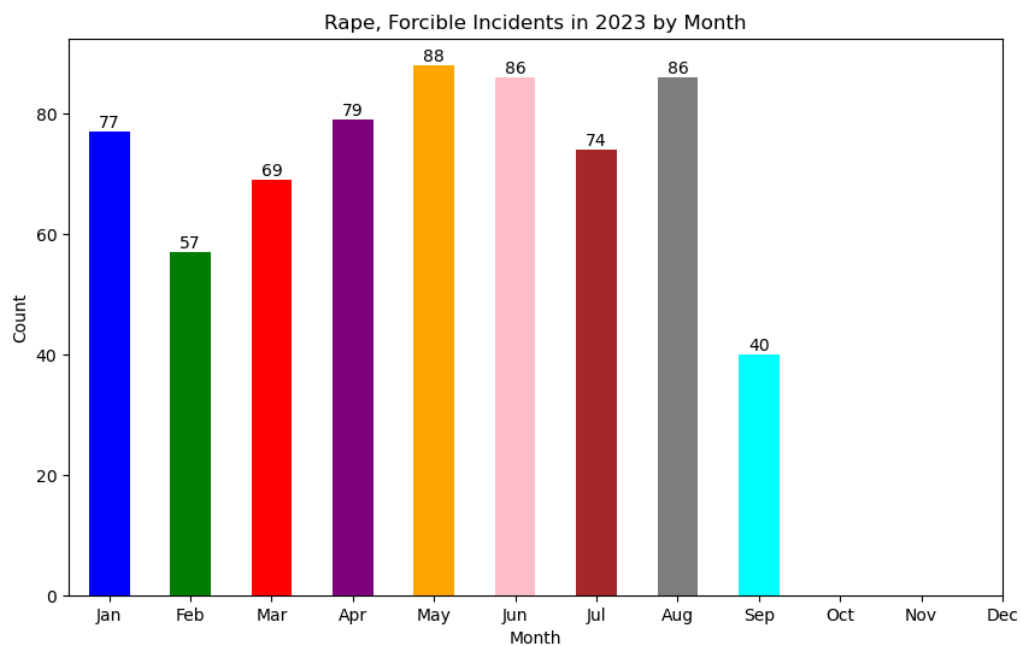
Graph 1.2.1 Tabel Rape Forcible 2020



Graph 1.2.2 Tabel Rape Forcible 2021



Graph 1.2.3 Tabel Rape Forcible 2022



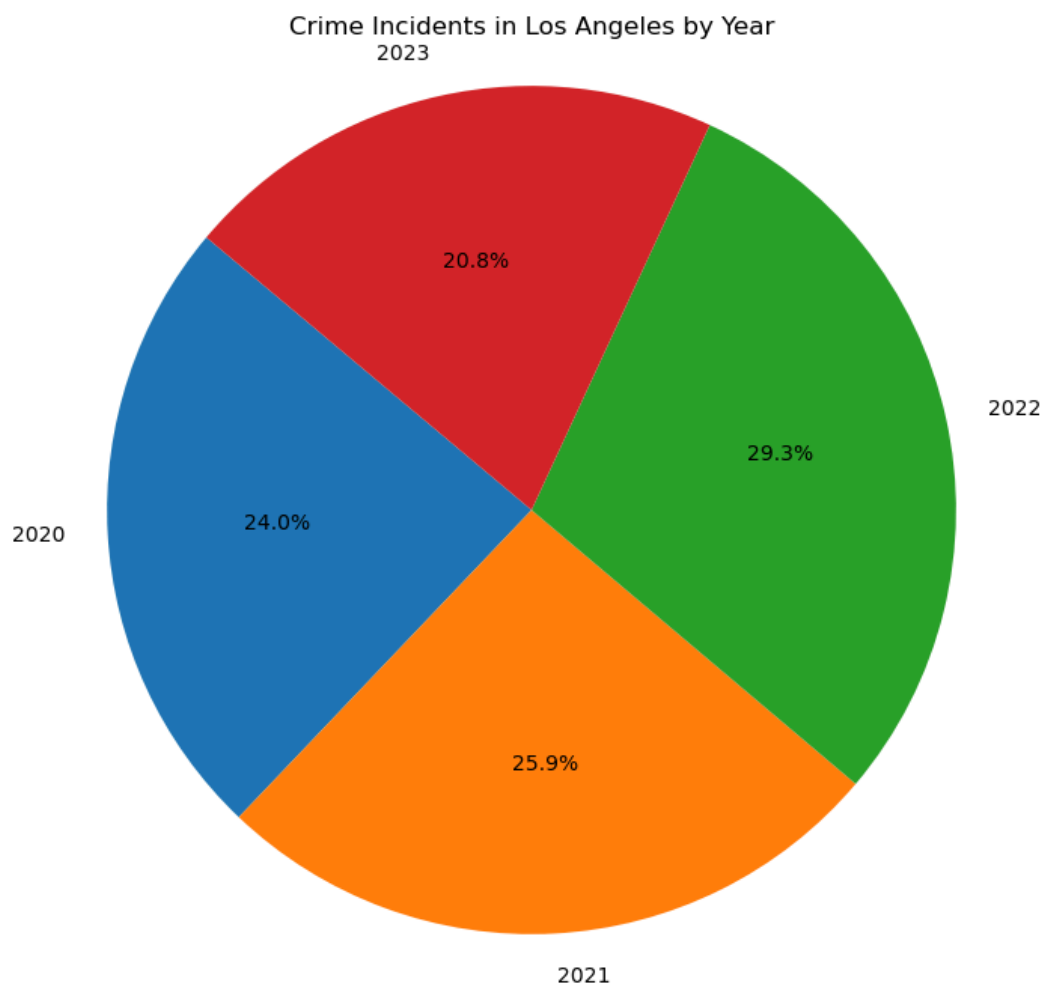
Graph 1.2.4 Tabel Rape Forcible 2023

Grafik ini sama dengan grafik sebelumnya, yang menjadi perbedaan adalah kategori kriminalitas yang ditampilkan

Dari hasil plot yang ditampilkan pada tahun 2020, kejahatan Rape -Forcible terjadi paling banyak pada bulan September dan paling rendah di bulan April. Pada tahun 2021 kejahatan tertinggi ada di bulan Mei dan paling rendah ada di bulan Desember. Pada tahun 2022,

kejahatan tertinggi berada di bulan November dan paling rendah berada di bulan Agustus. Pada tahun 2023, kejahatan tertinggi ada di bulan Mei dan paling rendah ada di bulan September.

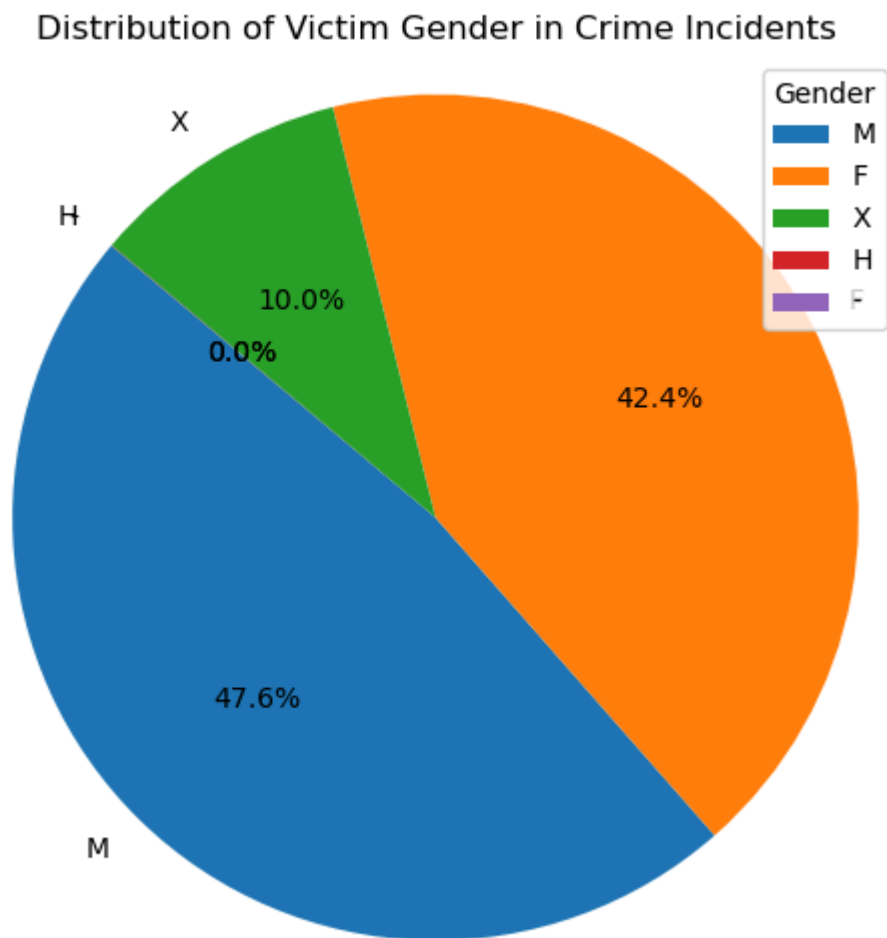
- **Crime Incident:**



Graph 1.3 Pie Chart Crime Incidents in Los Angeles by Year

Dari Pie chart diatas, dapat dilihat pada tahun 2020, terdapat 24% kejahatan yang terjadi selama periode 2020-2023 di Los Angeles, Amerika Serikat. Dilanjutkan pada tahun 2021 sebesar 25.9%, tahun 2022 sebesar 29.3% dan tahun 2023 sebesar 20.8.

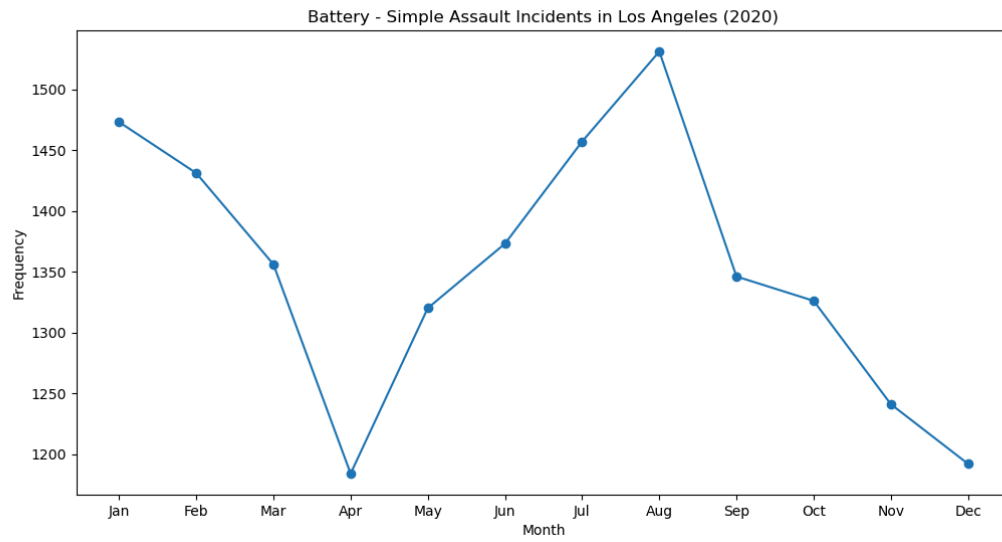
- **Distribution Gender:**



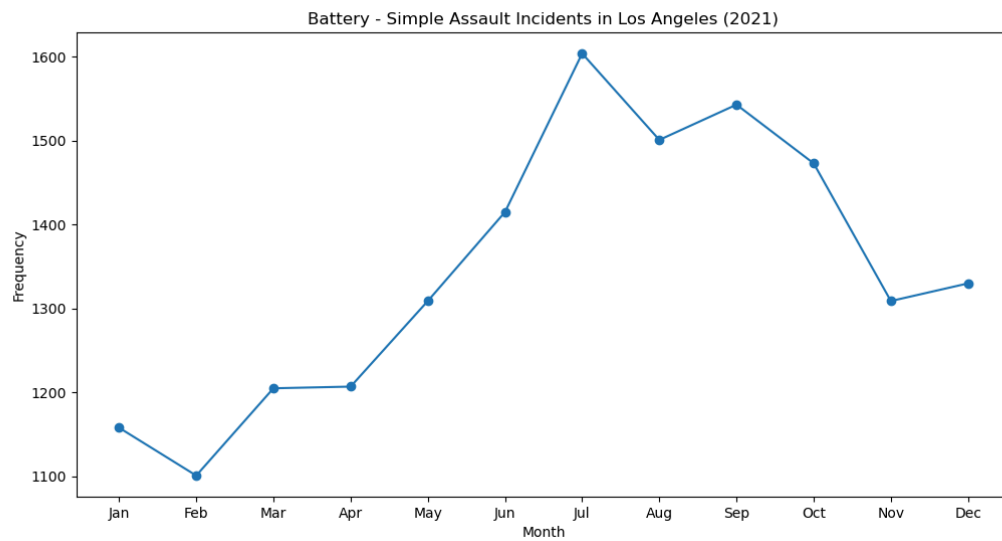
Graph 1.4 Pie Chart Distribution of Victim Gender in Crime Incidents

Dari pie chart diatas, dapat dilihat bahwa sebanyak 47.6% laki-laki sebagai korban kejahatan, sebanyak 42.4% perempuan sebagai korban kejahatan. Adapula yang tidak menyebutkan gendernya apa seperti “X” dan “-”. Sedangkan gender H tidak dapat diidentifikasi.

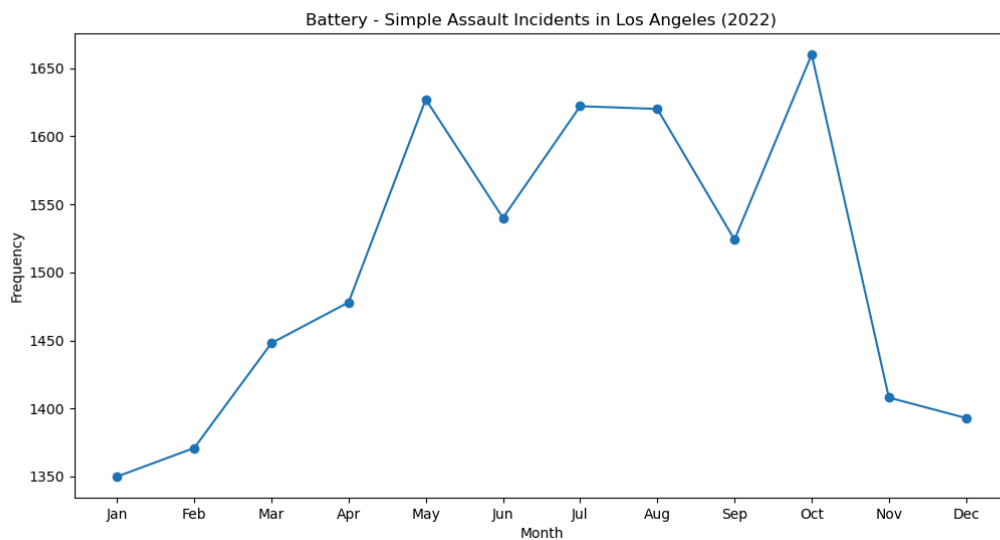
- **Frequency of Simple Assault:**



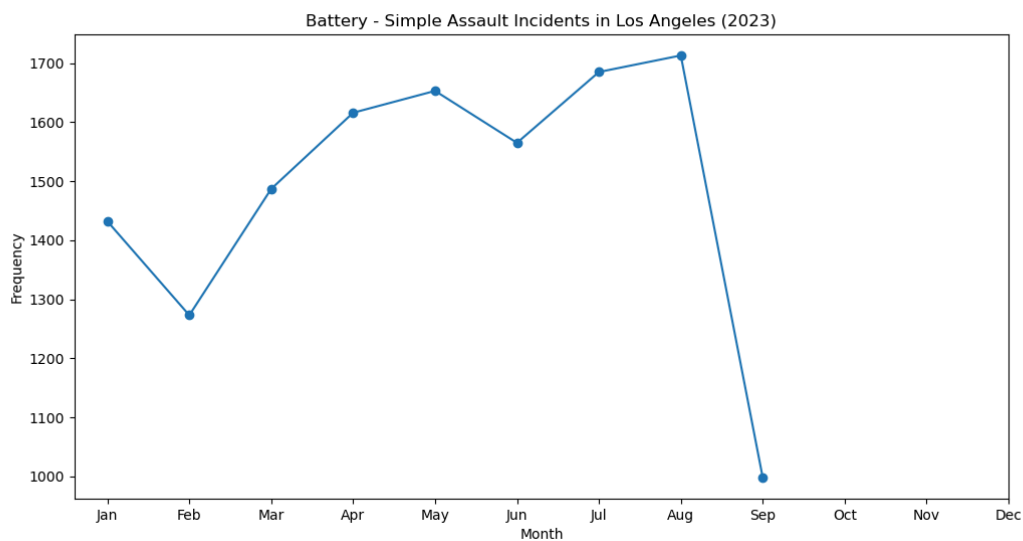
Graph 1.5.1 Tabel Frequency of Simple Assault 2020



Graph 1.5.2 Tabel Frequency of Simple Assault 2021



Graph 1.5.3 Tabel Frequency of Simple Assault 2022

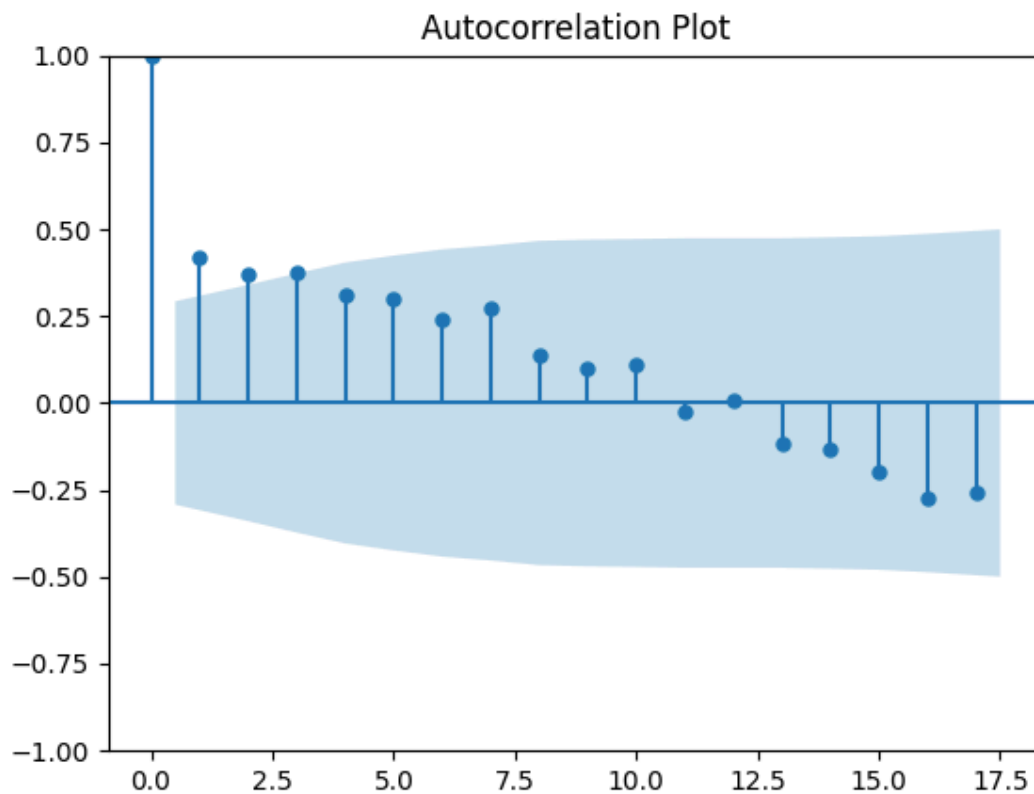


Graph 1.5.4 Tabel Frequency of Simple Assault 2023

Dari Tabel-tabel frekuensi diatas, dapat dilihat pada tahun 2020, tren kejahatan cenderung menurun dibulan April dan Desember dan mencapai puncak tertinggi pada bulan Agustus. Pada tahun 2021, kejahatan cenderung naik dari bulan Februari dan mencapai puncaknya pada bulan Juli dan mulai menurun di Bulan Agustus dan menurun lagi di bulan Oktober. Pada tahun 2022, Kejahatan cenderung naik dari bulan Januari sampai Mei dan terjadi fluktuatif selama beberapa bulan dan mencapai puncak tertinggi di bulan Oktober. Pada tahun 2023, Data Kejahatan menunjukkan kecenderungan naik dari bulan Februari sampai Bulan Agustus sebagai puncaknya, tetapi menurun sedikit di bulan Juli. Data bulan

September hanya sampai tanggal 10 Jadi saat divisualisasikan terlihat sangat menurun bagian paling bawah.

- **Autocorrelation:**

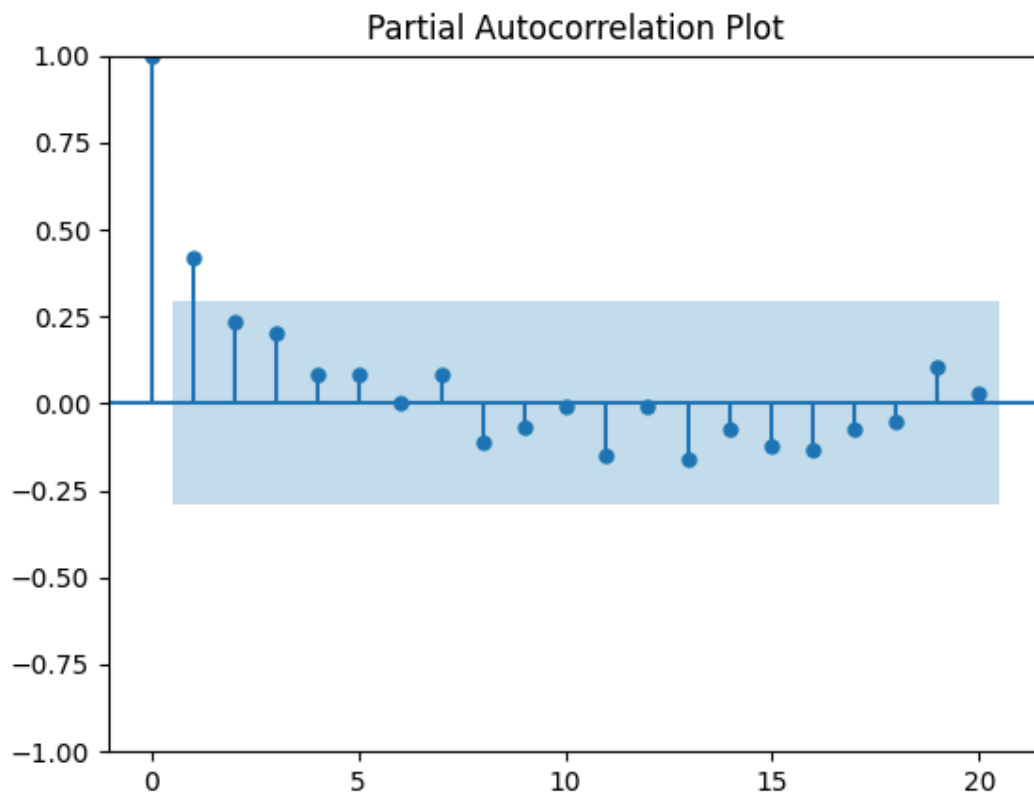


Graph 1.6 Autocorrelation Plot

Langkah dan penjelasan mengenai koding ini untuk menghitung pca dengan menghitung Jumlah Kejahatan per Bulan.

Data jumlah kejahatan diambil dari DataFrame (df) dengan meresample data pada frekuensi bulanan dan menghitung ukuran setiap grup. Beberapa baris pertama dari data jumlah kejahatan per bulan ditampilkan. Dengan menggunakan fungsi `plot_acf` dari modul `statsmodels.graphics.tsaplots` diimpor untuk membuat plot autocorrelation. Kemudian kita membuat Plot autocorrelation untuk data jumlah kejahatan per bulan.

- **Partial Autocorrelation:**



Graph 1.7 Partial Autocorrelation Plot

PACF yang dihasilkan akan menunjukkan korelasi parsial antara observasi pada bulan tertentu dengan observasi pada bulan-bulan sebelumnya hingga 20 bulan yang lalu. Nilai-nilai di luar interval keyakinan yang signifikan dapat memberikan indikasi pola autocorrelation parsial.

- **ARIMA:**

SARIMAX Results						
=====						
Dep. Variable:	y	No. Observations:	45			
Model:	ARIMA(2, 0, 1)	Log Likelihood	-395.503			
Date:	Sun, 10 Dec 2023	AIC	801.007			
Time:	16:31:53	BIC	810.040			
Sample:	01-31-2020	HQIC	804.374			
	- 09-30-2023					
Covariance Type:	opg					
=====						
	coef	std err	z	P> z	[0.025	0.975]

const	1.782e+04	1553.389	11.472	0.000	1.48e+04	2.09e+04
ar.L1	-0.1766	0.479	-0.368	0.713	-1.116	0.763
ar.L2	0.5745	0.345	1.668	0.095	-0.101	1.250
ma.L1	0.9999	0.155	6.462	0.000	0.697	1.303
sigma2	2.383e+06	0.001	4.04e+09	0.000	2.38e+06	2.38e+06
=====						
Ljung-Box (L1) (Q):	0.01	Jarque-Bera (JB):	761.36			
Prob(Q):	0.91	Prob(JB):	0.00			
Heteroskedasticity (H):	8.03	Skew:	-3.68			
Prob(H) (two-sided):	0.00	Kurtosis:	21.76			
=====						

Graph 1.8 Tabel ARIMA

Menampilkan model arima untuk melihat error.

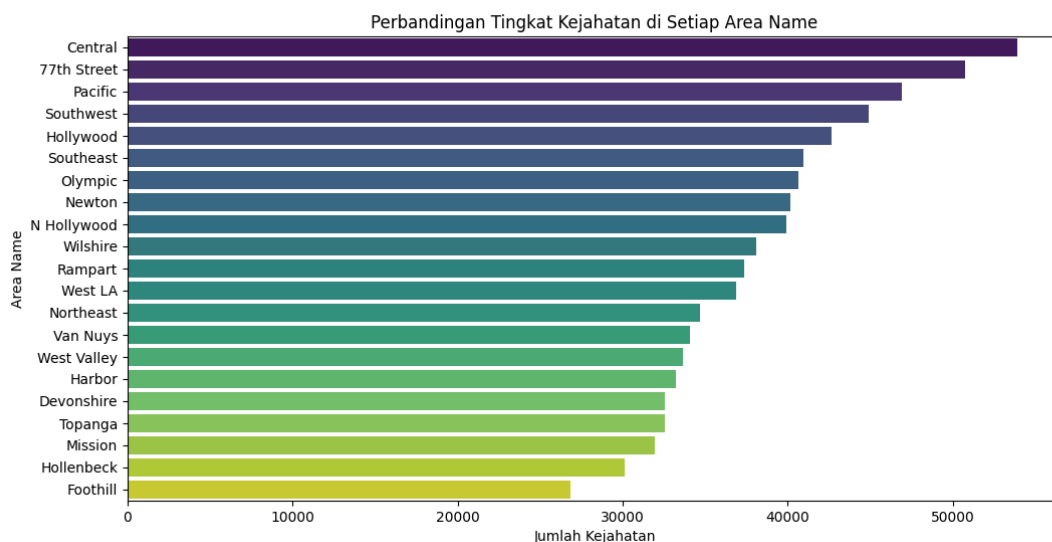
Menentukan orde model ARIMA dengan menggunakan nilai (2, 0, 1). Angka-angka ini mengacu pada orde AR, orde differencing, dan orde MA pada model ARIMA. Membuat model ARIMA menggunakan data `crime_per_day` dan orde yang telah ditentukan sebelumnya. Selanjutnya, model tersebut dilatih dengan memanggil metode `fit()`.

	Area Name	Jumlah Kejahatan
0	Central	53901
1	77th Street	50732
2	Pacific	46915
3	Southwest	44931
4	Hollywood	42658
5	Southeast	40971
6	Olympic	40623
7	Newton	40178
8	N Hollywood	39916
9	Wilshire	38100
10	Rampart	37392
11	West LA	36855
12	Northeast	34674
13	Van Nuys	34093
14	West Valley	33642
15	Harbor	33254
16	Devonshire	32588
17	Topanga	32588
18	Mission	31972
19	Hollenbeck	30134
20	Foothill	26839

Graph 1.8.1 Tabel Jumlah Kejahatan

Menghitung jumlah kejahatan di setiap area menggunakan fungsi `value_counts()` pada kolom 'AREA NAME'. Kemudian, mereset indeks dan memberikan nama kolom baru.

- **Perbandingan Tingkat Kejahatan di Setiap Area:**



Graph 1.9 Plot Perbandingan Tingkat Kejahatan di Setiap Area

Ini adalah grafik perbandingan tingkat kejahatan di setiap area dengan menggunakan bar plot dengan colormap viridis.

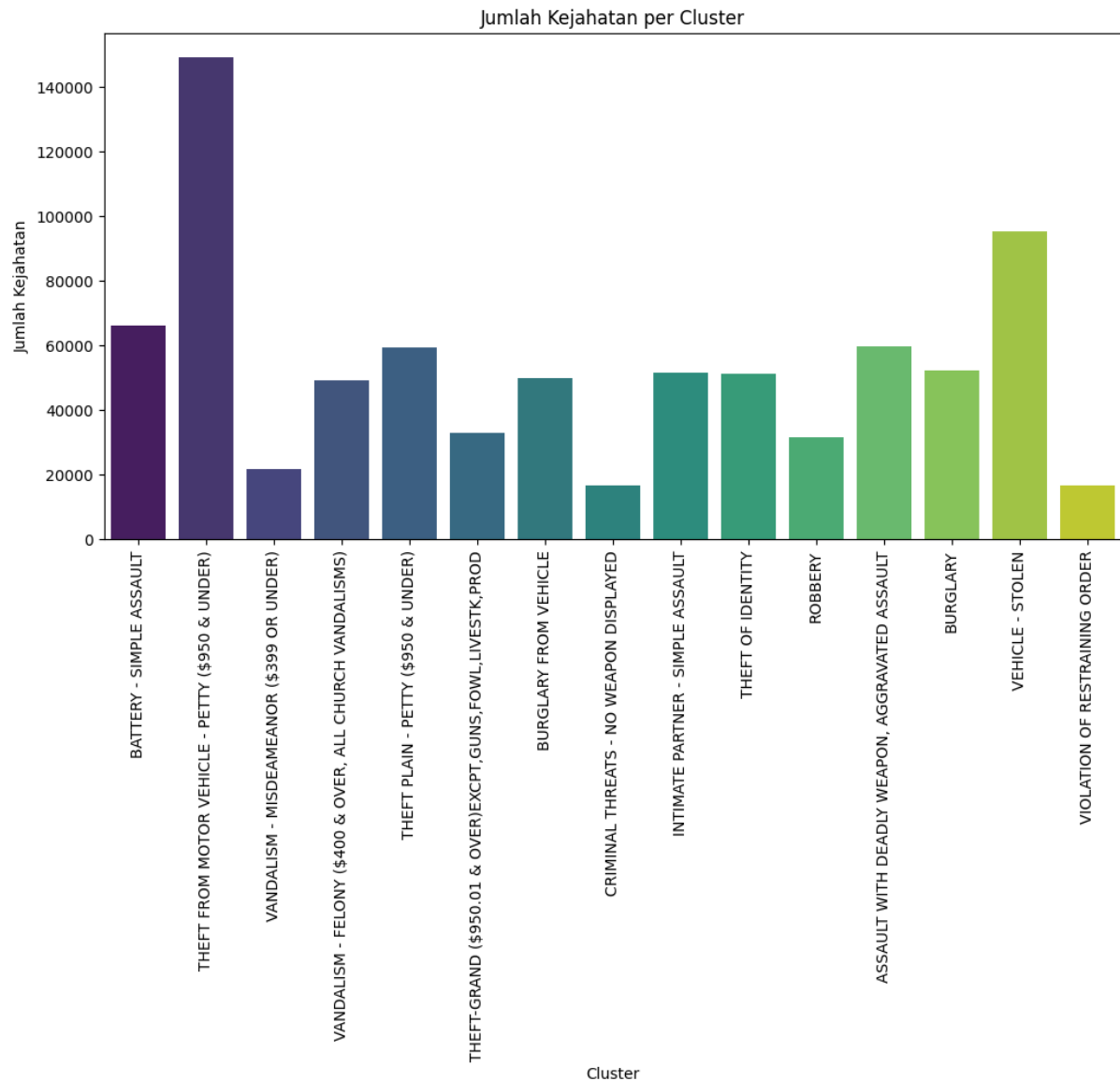
	Jenis Kejahatan	Jumlah Kejahatan
0	VEHICLE - STOLEN	85807
1	BATTERY - SIMPLE ASSAULT	63848
2	THEFT OF IDENTITY	51144
3	BURGLARY FROM VEHICLE	49330
4	VANDALISM - FELONY (\$400 & OVER, ALL CHURCH VA...	49055
..
133	FIREARMS EMERGENCY PROTECTIVE ORDER (FIREARMS ...	4
134	GRAND THEFT / AUTO REPAIR	4
135	FAILURE TO DISPERSE	3
136	DISHONEST EMPLOYEE ATTEMPTED THEFT	2
137	INCITING A RIOT	1

Graph 1.10 Jumlah dari Jenis Kejahatan

Menggunakan fungsi `value_counts()` pada kolom 'Crm Cd Desc' untuk menghitung jumlah setiap jenis kejahatan. Kemudian, mereset indeks dan memberikan nama kolom baru. Menampilkan tabel yang berisi daftar jenis kejahatan dan jumlah kejahatan untuk setiap jenisnya.

Tabel menunjukkan jumlah kejahatan untuk berbagai jenis kejahatan. Setiap baris menunjukkan jenis kejahatan (Jenis Kejahatan) dan jumlah kejahatan (Jumlah Kejahatan) yang terjadi. Data diurutkan secara otomatis berdasarkan frekuensi kejahatan secara descending. Sebagai contoh, "VEHICLE - STOLEN" memiliki jumlah kejahatan tertinggi, diikuti oleh "BATTERY - SIMPLE ASSAULT", "THEFT OF IDENTITY", dan seterusnya. Ada total 138 jenis kejahatan yang berbeda dalam dataset.

- **Jumlah Kejahatan per Cluster:**



Graph 1.11 Plot Jumlah Kejahatan per Cluster

Menggunakan TfidfVectorizer dari scikit-learn untuk mengonversi teks ke dalam representasi TF-IDF (Term Frequency-Inverse Document Frequency). Ini menghasilkan matriks di mana setiap baris mewakili suatu dokumen (dalam hal ini, jenis kejahatan) dan setiap kolom mewakili istilah dalam teks. Melakukan clustering menggunakan algoritma KMeans dengan jumlah cluster yang ditentukan (num_clusters). Model KMeans dilatih pada matriks TF-IDF.

Kemudian kami menambahkan kolom 'Cluster' ke DataFrame, yang berisi label cluster dari hasil clustering dan membuat kamus (cluster_names) yang memetakan label cluster ke jenis kejahatan yang paling mendominasi dalam cluster tersebut. Lalu kami menambahkan kolom 'Cluster_Name' yang berisi nama jenis kejahatan berdasarkan hasil pemetaan cluster. Selanjutnya kami membuat grafiknya dengan bar plot.

- **Distribusi Jenis Kejahatan di Setiap Area:**

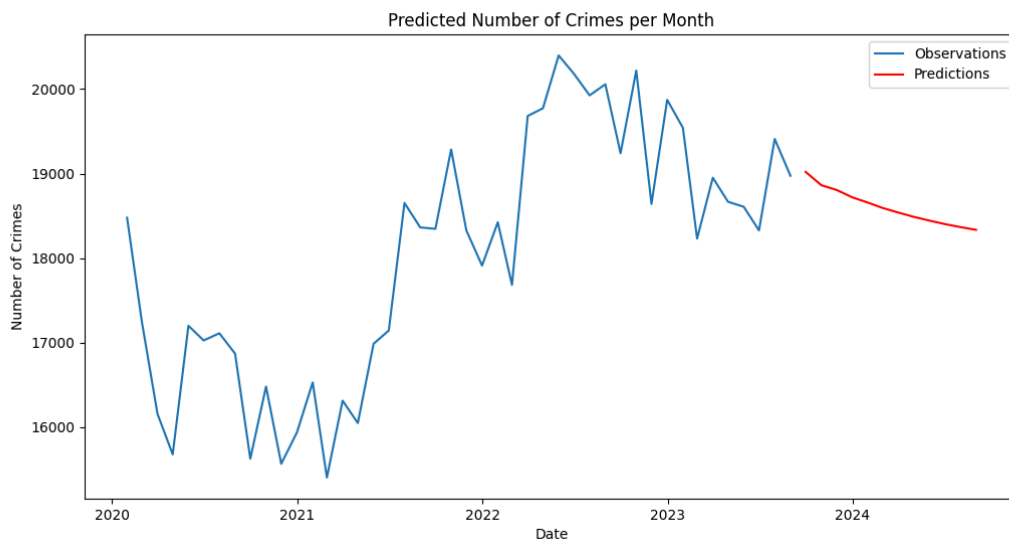


Graph 1.12 Plot Distribusi Jenis Kejahatan di Setiap Area

Visualisasi ini bertujuan untuk menunjukkan distribusi jenis kejahatan di setiap area untuk setiap cluster. Setiap subplot mewakili satu cluster, dan pada setiap subplot, terdapat visualisasi bar untuk menunjukkan jumlah kejahatan di setiap area dalam cluster tersebut. Menunjukkan jumlah membership setiap cluster. Ada satu slot yang kosong karena penulis ingin menampilkan plot dengan visual yang lebih enak dilihat, dengan mengatur layout untuk menampilkan plot sebagai 8 baris dan 2 kolom.

Menambahkan kolom 'Cluster' ke DataFrame yang berisi label cluster dari hasil clustering menggunakan KMeans. Membuat kamus (cluster_names) yang memetakan label cluster ke jenis kejahatan yang paling mendominasi dalam cluster tersebut. Menambahkan kolom 'Cluster_Name' yang berisi nama jenis kejahatan berdasarkan hasil pemetaan cluster. Membuat tabel kontingensi yang menunjukkan hubungan antara 'Cluster_Name' dan 'AREA NAME'. Membuat subplot dengan ukuran grid 8x2 untuk menampung visualisasi hasil. Iterasi melalui setiap subplot, dan menggunakan enumerate untuk mendapatkan indeks dan nilai cluster_name. Membuat plot bar untuk setiap cluster dalam subplot. Menambahkan judul, label sumbu x, dan label sumbu y untuk setiap subplot. Menyesuaikan label area agar lebih jelas dengan memutar label sebesar 45 derajat. Lalu ditampilkan dengan menggunakan plot.show.

- **Predicted Crime:**



Graph 1.13 Plot Predicted Crime

Berikut kami tampilkan grafik prediksi jumlah kejahatan untuk 12 bulan tahun kedepan. Cara untuk menampilkan dan menghitung prediksi diatas adalah sebagai berikut: Memilih data kejahatan per bulan dari Januari 2020 hingga Agustus 2023 menggunakan metode resampling untuk mendapatkan jumlah kejahatan per bulan. Membuat model ARIMA dengan urutan (2, 0, 1) dan melatih model tersebut menggunakan data jumlah kejahatan per bulan.

Menggunakan model yang telah dilatih untuk membuat prediksi kejahatan untuk 12 bulan ke depan. Membuat indeks untuk prediksi, dimulai dari bulan terakhir dalam data observasi dan melanjutkan selama 12 bulan ke depan. Mengambil nilai rata-rata dari prediksi yang dihasilkan. Membuat plot yang menunjukkan observasi kejahatan per bulan dan prediksi kejahatan untuk 12 bulan ke depan. Plot ini tidak termasuk interval kepercayaan.

Daftar Pustaka

Putra, Andrian Dwi, et al. "Faktor-Faktor yang Mempengaruhi Tingkat Kriminalitas di Indonesia Tahun 2018." *Indonesian Journal of Applied Statistics*, vol. 3, no. 2, 2020.

White, Rob and Fiona Haines. 1996. *Crime and Criminology: An Introduction*. Oxford University Press.

Wang, Hongjian, et al. "Crime Rate Inference with Big Data." *In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, 2016, pp. 635-644.

George Mohler, Andrea L. Bertozzi, Jeremy Carter, Martin B. Short, Daniel Sledge, George E. Tita, Craig D. Uchida, and P. Jeffrey Brantingham. "Impact of social distancing during COVID-19 pandemic on crime in Los Angeles and Indianapolis." *Journal of Criminal Justice*, Volume 68, 2020, 101692. ISSN 0047-2352, <https://doi.org/10.1016/j.jcrimjus.2020.101692>.

Anata, F. (2013). *Tingkat Kriminalitas (Studi Pada 31 Provinsi Di Indonesia Tahun*. Universitas Brawijaya.

Igual, Laura, et al. *Introduction to Data Science: A Python Approach to Concepts, Techniques and Applications*. Springer International Publishing, 2017. Accessed October 2023.

McKinney,W. (2018). *Pandas: Powerful Python Data Analysis Toolkit*. California: PyData Development Team

J. Hunter, "Matplotlib: Visualization with Python," 2012. [Online]. Available: <https://matplotlib.org/>.

Chan, J. (2017). Big Data and Visuality. *Oxford Research Encyclopedia of Criminology and Criminal Justice*. <https://doi.org/10.1093/acrefore/9780190264079.013.128>

Ozkan, T. (2019a). Criminology in the age of data explosion: New directions. *The Social Science Journal*, 56(2), 208–219. <https://doi.org/10.1016/j.soscij.2018.10.010>