Analisis Data COVID-19 Malaysia

Disediakan oleh: Muhammad Khairul Izdihar
Tujuan: Projek portfolio data analyst menggunakan Python, Pandas, Matplotlib & Seaborn.

Pengenalan

Data ini diambil daripada [MOH Malaysia Open Data](#).
Projek ini bertujuan:

- Menunjukkan kebolehan pembersihan & analisis data (Data Cleaning, Pandas).
- Membina visualisasi (Matplotlib, Seaborn).
- Menyediakan insight berguna berkaitan trend COVID-19.

1. Import Libraries

Gunakan `pandas` untuk manipulasi data & `matplotlib/seaborn` untuk visualisasi.

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

# styling utk graf
sns.set(style="whitegrid")
```

```
from google.colab import files
```

```
# pilih file dari laptop
uploaded = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving covid_cases (2).csv to covid_cases (2) (1).csv
```

2. Muat Naik Data

Dataset COVID-19 Malaysia dengan kolum:

- `date` : Tarikh laporan
- `state` : Negeri
- `cases_new` : Kes baharu
- `cases_recovered` : Kes sembuh
- `cases_active` : Kes aktif

```
# 3. Pastikan nama file sama dengan yang upload tadi
df = pd.read_csv("covid_cases (2).csv")
```

```
df.head()
```

	date	state	cases_new	cases_import	cases_recovered	cases_active	cases_cluster
0	2020-01-25	Malaysia	4	4	0	4	0
1	2020-01-26	Malaysia	0	0	0	4	0
2	2020-01-27	Malaysia	0	0	0	4	0
3	2020-01-28	Malaysia	0	0	0	4	0
4	2020-01-29	Malaysia	3	3	0	7	0

```
# Info asas dataset
df.info()

# Statistik (mean, min, max, std)
df.describe()

# Check nama columns
df.columns
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 33218 entries, 0 to 33217
Data columns (total 7 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   date             33218 non-null  object
 1   state            33218 non-null  object
 2   cases_new        33218 non-null  int64
 3   cases_import     33218 non-null  int64
 4   cases_recovered  33218 non-null  int64
 5   cases_active     33218 non-null  int64
 6   cases_cluster    33218 non-null  int64
dtypes: int64(5), object(2)
memory usage: 1.8+ MB
```

```
Index(['date', 'state', 'cases_new', 'cases_import', 'cases_recovered',
       'cases_active', 'cases_cluster'],
      dtype='object')
```

3. Data Cleaning

- Tukar tarikh ke format datetime.
- Buang missing values.
- Exclude data "Malaysia" supaya fokus pada negeri.

```
# Tukar column 'date' jadi format tarikh
df['date'] = pd.to_datetime(df['date'])

# Buang rows yang ada missing value
df = df.dropna()

#Exclude Malaysia
df = df[df['state'] != "Malaysia"]

# Confirmkan takde missing lagi
df.isnull().sum()
```

	0
date	0
state	0
cases_new	0
cases_import	0
cases_recovered	0
cases_active	0
cases_cluster	0

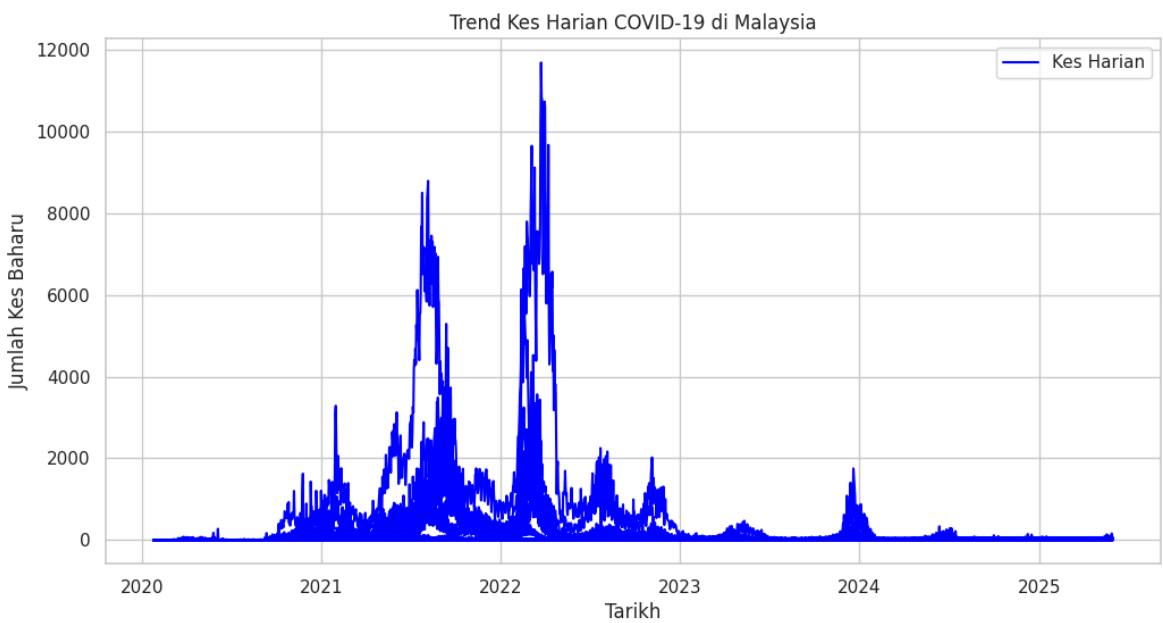
dtype: int64

```
# Total keseluruhan kes
total_cases = df['cases_new'].sum()
print("Jumlah Kes Baharu:", total_cases)

# Kes harian tertinggi
print("Kes Harian Tertinggi:", df['cases_new'].max())
```

```
Jumlah Kes Baharu: 5346653
Kes Harian Tertinggi: 11692
```

Double-click (or enter) to edit

```
plt.figure(figsize=(12,6))   # Saiz figure grafik (panjang=12, tinggi=6 inci)
plt.plot(df['date'], df['cases_new'], color='blue', label='Kes Harian')
# → Buat graf garis: x-axis = tarikh, y-axis = jumlah kes baru
# → Warna garis biru, label "Kes Harian"

plt.title("Trend Kes Harian COVID-19 di Malaysia")  # Tajuk graf
plt.xlabel("Tarikh")  # Label paksi X
plt.ylabel("Jumlah Kes Baharu")  # Label paksi Y
plt.legend()  # Tunjukkan legenda
plt.show()  # Paparkan graf
```

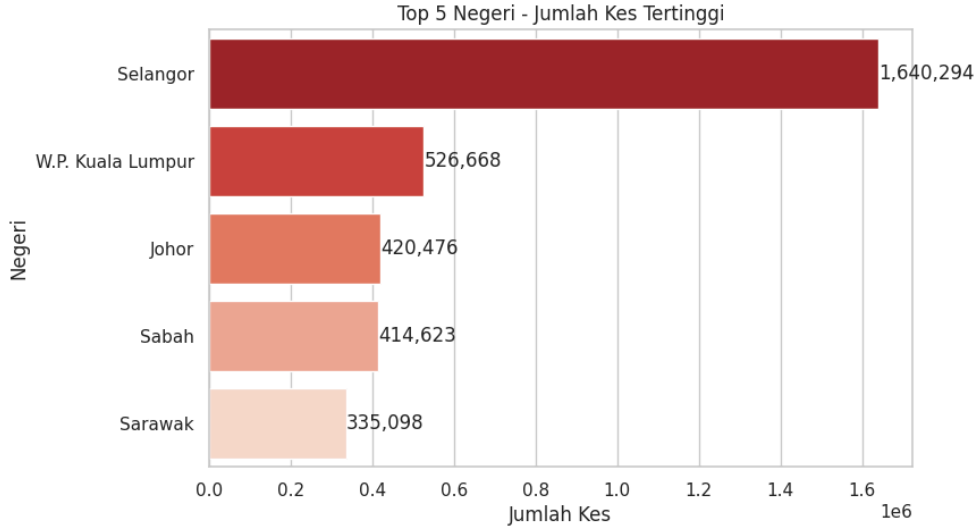


Top 5 Negeri dengan Kes Tertinggi

Bar chart menunjukkan negeri dengan jumlah kes kumulatif paling tinggi.

```
# 1. Kira jumlah kes untuk setiap negeri
top_states = df.groupby('state')['cases_new'].sum()
# → Kumpulkan data ikut negeri, tambah semua 'cases_new' untuk setiap negeri

top_states = top_states.sort_values(ascending=False).head(5)
# → Susun dari paling banyak → paling sedikit, ambil hanya 5 teratas



plt.figure(figsize=(8,5))  # Buat figure saiz 8x5 inci
sns.barplot(x=top_states.values, y=top_states.index, palette="Reds_r")
# → Buat bar chart: nilai (jumlah kes) di X-axis, nama negeri di Y-axis
# → Warna guna tema "Reds_r" (merah → pink pudar)

plt.title("Top 5 Negeri - Jumlah Kes Tertinggi")  # Tajuk graf
plt.xlabel("Jumlah Kes")  # Label paksi X
plt.ylabel("Negeri")      # Label paksi Y

# Tunjuk nilai di hujung bar
for i, v in enumerate(top_states.values):
    plt.text(v, i, f"{v:,}", va='center')  # format guna koma, contoh 1,234,567

plt.show()                # Paparkan graf
```

```
/tmp/ipython-input-4106865078.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same

  sns.barplot(x=top_states.values, y=top_states.index, palette="Reds_r")
```



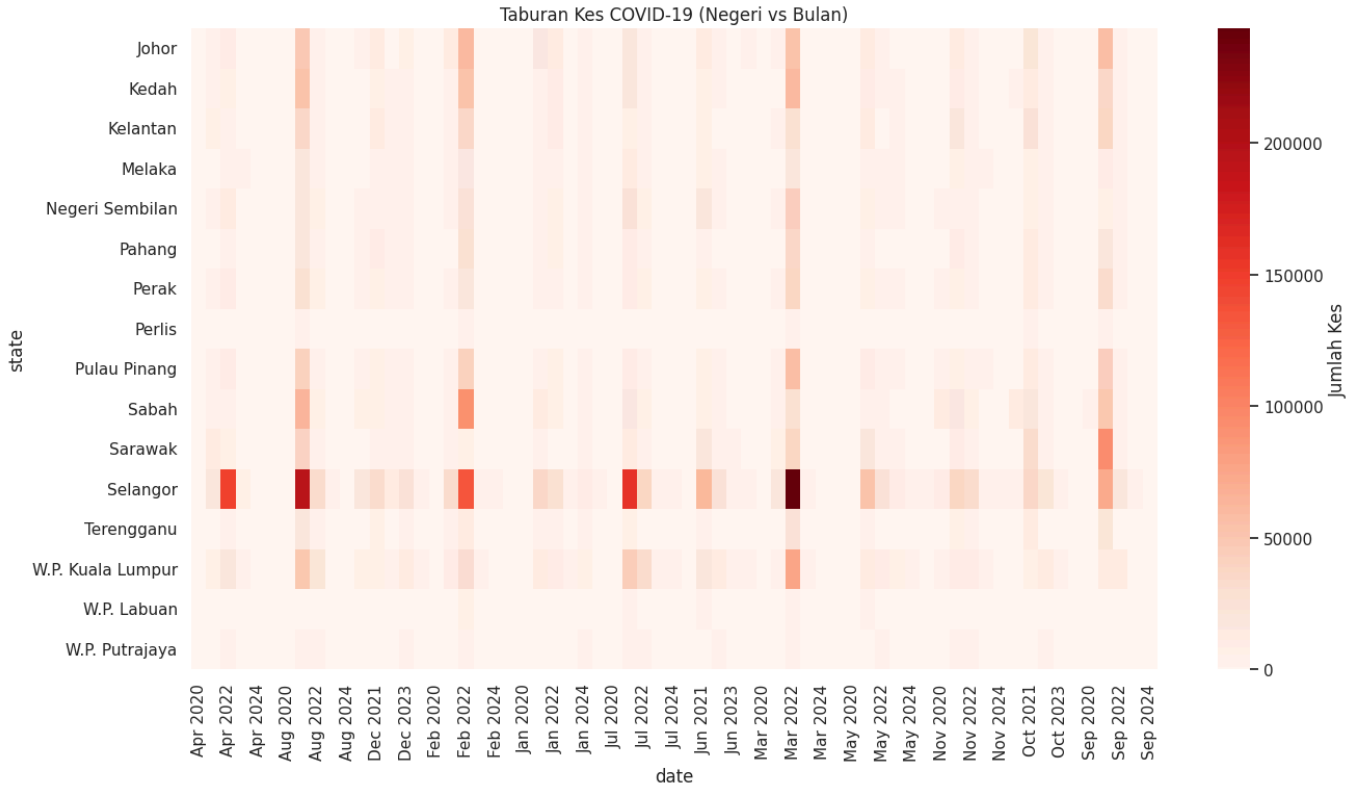

```
total_cases = df['cases_new'].sum()
total_recovered = df['cases_recovered'].sum()
avg_last30 = df.tail(30)['cases_new'].mean()

print(f"Jumlah Kes: {total_cases:,}")
print(f"Jumlah Kes Sembuh: {total_recovered:,}")
print(f"Purata 30 Hari Terakhir: {avg_last30:,.0f}")
```

```
Jumlah Kes: 5,346,653
Jumlah Kes Sembuh: 5,294,509
Purata 30 Hari Terakhir: 2
```

Heatmap Negeri vs Tarikh

Heatmap untuk melihat taburan kes mengikut negeri dan tarikh.

```
# Tukar column date jadi datetime (jika belum)
df['date'] = pd.to_datetime(df['date'])

# Simpan balik hasil format (contoh Jan 2021)
df['date'] = df['date'].dt.strftime('%b %Y')

# Pivot table ikut state & bulan
pivot = df.pivot_table(index="state", columns="date", values="cases_new", aggfunc="sum")

# Plot heatmap
plt.figure(figsize=(15,8))
sns.heatmap(pivot, cmap="Reds", cbar_kws={'label': 'Jumlah Kes'})
plt.title("Taburan Kes COVID-19 (Negeri vs Bulan)")
plt.show()
```




```
df.groupby("state")["cases_new"].describe()
```

state	count	mean	std	min	25%	50%	75%	max
Johor	1954.0	215.187308	476.021907	0.0	7.0	22.0	136.5	3238.0
Kedah	1954.0	170.566018	449.286034	0.0	4.0	16.0	98.0	3243.0
Kelantan	1954.0	137.784545	322.467249	0.0	1.0	9.0	68.0	2135.0
Melaka	1954.0	84.290686	147.755713	0.0	4.0	16.0	102.0	1120.0
Negeri Sembilan	1954.0	125.378199	275.160706	0.0	5.0	17.0	122.0	2115.0
Pahang	1954.0	98.136643	239.934486	0.0	2.0	10.0	56.0	2006.0
Perak	1954.0	128.903275	269.604169	0.0	3.0	17.0	132.0	1713.0
Perlis	1954.0	10.925793	32.207264	0.0	0.0	1.5	7.0	321.0
Pulau Pinang	1954.0	166.564483	397.665628	0.0	6.0	25.0	140.0	2773.0
Sabah	1954.0	212.191914	567.163112	0.0	2.0	10.0	189.0	5565.0
Sarawak	1954.0	171.493347	480.970830	0.0	3.0	12.0	89.0	5291.0
Selangor	1954.0	839.454452	1673.420779	0.0	29.0	132.0	879.0	11692.0
Terengganu	1954.0	73.443193	174.997468	0.0	0.0	8.0	40.0	1283.0
W.P. Kuala Lumpur	1954.0	269.533265	478.745426	0.0	19.0	67.0	322.0	4527.0
W.P. Labuan	1954.0	12.293245	42.853667	0.0	0.0	1.0	6.0	499.0
W.P. Putrajaya	1954.0	20.114125	33.447834	0.0	1.0	5.0	26.0	231.0

```
total_cases = df['cases_new'].sum()
avg_daily = df['cases_new'].mean()
top_state = df.groupby("state")["cases_new"].sum().idxmax()

print(f"Jumlah keseluruhan kes: {total_cases:,}")
print(f"Purata kes harian: {avg_daily:,.0f}")
print(f"Negeri dengan kes tertinggi: {top_state}")
```

```
Jumlah keseluruhan kes: 5,346,653
Purata kes harian: 171
Negeri dengan kes tertinggi: Selangor
```

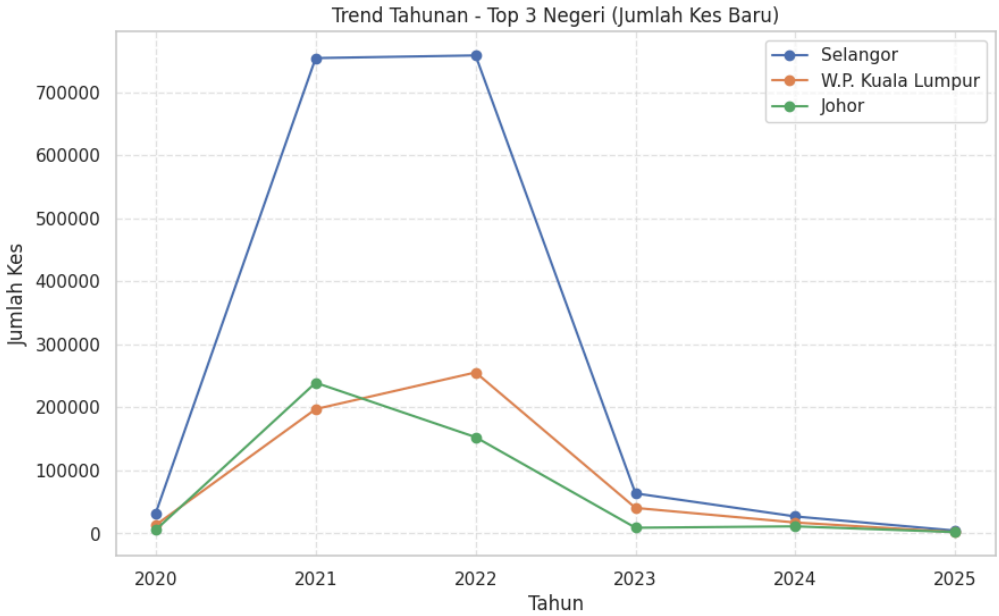
Perbandingan Top 3 Negeri

Line chart untuk banding trend tahunan 3 negeri dengan jumlah kes tertinggi.

Double-click (or enter) to edit

```
# Ambil top 3 negeri ikut jumlah keseluruhan
top3 = df.groupby("state")["cases_new"].sum().sort_values(ascending=False).head(3).index

# Convert tarikh ke datetime
df['date'] = pd.to_datetime(df['date'], errors='coerce')

# Buat column tahun
df['year'] = df['date'].dt.year


# Plot line chart untuk top 3 negeri ikut tahun
plt.figure(figsize=(10,6))
for state in top3:
    subset = df[df['state'] == state].groupby("year")["cases_new"].sum()
    plt.plot(subset.index, subset.values, marker="o", label=state)

plt.title("Trend Tahunan - Top 3 Negeri (Jumlah Kes Baru)")
plt.xlabel("Tahun")
plt.ylabel("Jumlah Kes")
plt.legend()
plt.grid(True, linestyle="--", alpha=0.5)
plt.show()
```

```
/tmp/ipython-input-2691785609.py:5: UserWarning: Could not infer format, so each element will be parsed individually, falling back to `dateutil`. To ensure
  df['date'] = pd.to_datetime(df['date'], errors='coerce')
```



Dapatan

1. Negeri Selangor konsisten mencatatkan kes tertinggi sepanjang pandemik.
2. Lonjakan besar berlaku sekitar Ogos–September 2021.
3. Negeri Sabah & Johor juga antara penyumbang utama kes kumulatif.

Kesimpulan

Projek ini menunjukkan bagaimana Python boleh digunakan untuk:

- Membersihkan & memproses data (ETL ringkas).
- Menghasilkan visual yang jelas.
- Memberikan pandangan dari sudut praktikal untuk memahami situasi COVID-19.

Start coding or generate with AI.