

LAPORAN TUGAS BESAR 2
KU1102 PENGENALAN KOMPUTASI
SEMESTER I 2023/2024

ANALISIS DAN VISUALISASI DATA

“Pendekatan Informatif Menggunakan Python untuk Analisis dan Visualisasi Data COVID-19 Global”



Disusun oleh:

Kelompok 1 K-20

William Anthony	(16523109)
Muhammad Jafar Fadli	(16523137)
Zulfaqqar Nayaka Athadiansyah	(19623116)
Abdullah Farhan	(19623305)

PROGRAM TAHAP PERSIAPAN BERSAMA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
NOVEMBER 2023

PRAKATA

Dengan memanjatkan puja dan puji syukur ke hadirat Tuhan Yang Maha Esa, kami dapat menyelesaikan laporan ini. Laporan ini adalah hasil dari tugas besar 2 KU1102 yang kami susun untuk memenuhi kewajiban akademik dalam mata kuliah KU1102 Pengenalan Komputasi pada semester I 2023/2024.

Pelaksanaan penulisan laporan ini tidak lepas dari bantuan dan arahan dari berbagai pihak. Oleh karena itu, kami mengucapkan terima kasih kepada semua pihak yang telah mendukung penyusunan laporan ini, terutama dosen pengampu mata kuliah KU1102 Pengenalan Komputasi untuk kelas 20: Ibu Dr. Fazat Nur Azizah, S.T., M.Sc. dan Bapak Yuda Sukmana, S.Pd., M.T.

Akhir kata, kami berharap laporan ini dapat menjadi penunjang hasil IP KU1102 kami dan bermanfaat bagi pembaca serta dapat menjadi bahan pelajaran untuk kami selaku mahasiswa Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.

Dengan penuh kesadaran, kami telah berupaya dengan sepenuh hati untuk mencapai kesempurnaan dalam penyusunan laporan ini. Namun, kami menyadari bahwa masih ada kekurangan dalam penulisan laporan ini. Oleh karena itu, kami sangat menghargai dan berharap untuk menerima saran serta kritik yang konstruktif untuk meningkatkan kualitas penulisan laporan kami di masa mendatang.

DAFTAR ISI

1 Pendahuluan	4
1.1 Latar Belakang	4
1.2 Tujuan	4
1.3 Deskripsi Data dan File	4
2 Analisis Data	5
2.1 Karakteristik Data	5
2.2 Statistik.....	7
3 Visualisasi Data	8
3.1 Perbandingan Kategori	10
3.2 Perubahan terhadap Waktu.....	12
3.3 Hierarki dan Hubungan Keseluruhan-Bagian.....	12
3.4 <i>Plotting Relationships</i>	14
4 <i>Data Cleansing</i>	15
5 Kesimpulan dan Saran	16
6 Lampiran.....	17

1 PENDAHULUAN

1.1 Latar Belakang

Pandemi COVID-19 telah menjadi salah satu tantangan terbesar bagi kesehatan dan kesejahteraan manusia di abad ke-21. Pandemi ini diakibatkan virus SARS-CoV-2 yang muncul pada akhir tahun 2019 di Wuhan, Republik Rakyat Tiongkok. Sejak saat itu, COVID-19 telah menyebar ke lebih dari 200 negara dan wilayah, menginfeksi lebih dari 300 juta orang, menyebabkan lebih dari 5 juta kematian, serta meninggalkan dampak besar terhadap perekonomian dunia. Upaya global untuk mengendalikan dan mengatasi pandemi ini membutuhkan pemahaman yang mendalam dan terkini tentang situasi epidemiologis, dampak sosial-ekonomi, dan respons kebijakan dari berbagai pihak terkait.

Kami ingin melakukan kilas balik menuju masa awal pandemi serta melakukan analisis dan visualisasi data terhadap kasus, kematian, dan informasi lainnya terkait pandemi COVID-19. Dengan demikian, kami dapat lebih memahami bagaimana analisis dan visualisasi data dapat digunakan sebagai dasar pengambilan keputusan.

Salah satu sumber data yang dapat digunakan untuk mempelajari pandemi COVID-19 adalah Data Science Dojo, sebuah organisasi yang menyediakan data dan kode terbuka tentang berbagai topik global, salah satunya adalah data kasus COVID-19. Data ini mencakup berbagai indikator, seperti kasus, kematian, pembatasan perjalanan, sumber infeksi, dan masih banyak lagi.

Kami menggunakan Python untuk melakukan analisis pada data kasus COVID-19 yang kami dapatkan. Python adalah salah satu bahasa pemrograman yang paling populer dan serbaguna, yang dapat digunakan untuk berbagai tujuan, termasuk analisis data dan visualisasi. Python memiliki banyak *library* dan modul yang dapat membantu pengguna untuk mengolah, memanipulasi, dan menampilkan data dengan mudah dan efisien. Beberapa *library* dan modul Python yang berguna untuk analisis dan visualisasi data COVID-19 adalah *pandas*, *numpy*, *matplotlib*, dan *seaborn*.

1.2 Tujuan

- 1.2.1 Melakukan analisis pada data kasus COVID-19 menggunakan Python dengan *library pandas*.
- 1.2.2 Menjelaskan *insight* yang didapat dan keputusan yang bisa dibuat berdasarkan analisis data.
- 1.2.3 Membuat visualisasi data kasus COVID-19 menggunakan Python dengan *library pandas* dan *matplotlib*.

1.3 Metode

Berangkat dari tujuan, kami menerapkan metode statistika deskriptif (*descriptive statistics*) dalam melakukan analisis dan visualisasi data yang bertumpu pada *library pandas*, *matplotlib*, dan *seaborn* dalam bahasa pemrograman Python. Adapun pembersihan data dilakukan dengan Microsoft Excel. Source code dari Jupyter Notebook kami dapat diakses di www.github.com/NayakaZNA/ku1102_tubes2.

1.4 Deskripsi Data dan File

Data set yang dianalisis adalah data set kasus COVID-19 global hingga April 2020 yang berukuran 19,7 KB. Data set ini berasal dari Data Science Dojo¹ yang dikompilasikan dari data set milik Kaggle dan John Hopkins University. Kedua sumber tersebut menggabungkan laporan harian World Health Organization (WHO), European Centre for Disease Prevention and Control (ECDC), US Department of State Travel Advisories, dan banyak sumber lainnya.

Langkah pertama yang dilakukan adalah mengunduh data dalam format .csv dari laman Data Science Dojo (lihat catatan kaki 1) lalu membersihkan dan menyimpannya dengan nama file “data_corona.csv”. Data set data_corona.csv yang sudah dibersihkan berukuran 7,91 KB serta tersusun atas 167 baris dan 10 kolom/atribut. Pembersihan data (*data cleansing*) dilakukan dengan Microsoft Excel, sedangkan analisis data dilakukan dengan bahasa pemrograman Python melalui IDE Visual Studio Code. Selanjutnya, *library* yang digunakan kemudian di-*import* dengan *command* berikut:

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

Data kemudian di-*import* sebagai DataFrame dan disimpan dalam variabel df:

```
df = pd.read_csv("data_corona.csv")
```

Variabel df akan digunakan sepanjang proses analisis dan visualisasi data terkait kasus COVID-19 global.

2 ANALISIS DATA

2.1 Karakteristik Data

Karakteristik dari 10 atribut pada data set disajikan pada tabel berikut:

Tabel 2.1.1 Karakteristik Atribut pada data_corona.csv

No	Atribut	Rincian	TIPE ATRIBUT	TIPE DATA
1	Country	Negara dengan kasus COVID-19	Kategorikal-Nominal	str/object
2	Cases	Banyak kasus COVID-19 secara kumulatif	Kuantitatif-Diskrit	int/int64
3	New_cases	Kasus baru COVID-19	Kuantitatif-Diskrit	int/int64
4	Deaths	Banyak kematian akibat COVID-19 secara kumulatif	Kuantitatif-Diskrit	int/int64

¹

https://code.datasciencedojo.com/datasciencedojo/datasets/blob/master/Coronavirus/coronavirus_merged_sources_Apr2020.csv

5	New_deaths	Kematian baru akibat COVID-19	Kuantitatif-Diskrit	int/int64
6	Transmission	Penyebaran	Kategorikal-Biner	str/object (domestik/eksternal)
7	Days_last_reported	Banyak hari sejak pelaporan kasus terakhir	Kuantitatif-Diskrit	int/int64
8	Level	Tingkat pembatasan mobilitas: 1. <i>Exercise Normal Precautions</i> (Waspada) 2. <i>Exercise Increased Caution</i> (Siaga) 3. <i>Reconsider Travel</i> (Awas) 4. <i>Do Not Travel</i> (Bahaya)	Kategorikal-Ordinal ²	int/int64 (1/2/3/4)
9	Date_last_reported	Tanggal terakhir pelaporan kasus	Kategorikal-Ordinal	int/int64 (dd/mm/yyyy)
10	Travel_restriction	Pembatasan perjalanan	Kategorikal-Biner	Bool (True/False)

Beragam informasi ini didapatkan dengan menjalankan kode berikut:

```
for col in df.columns:
    # Jika tipe data adalah objek, maka itu adalah data kategorikal
    if df[col].dtype == 'object':
        print(f"{col} adalah atribut kategorikal dengan
{len(df[col].unique())} nilai unik, yaitu: {df[col].unique()}")
    # Jika bukan objek, maka itu adalah data kuantitatif
    else:
        print(f"{col} adalah atribut kuantitatif dengan tipe data
{df[col].dtype}, memiliki range nilai dari {df[col].min()} sampai
{df[col].max()}")
    # Cek data yang hilang
    print(f"{col} tidak memiliki data yang hilang, yaitu:
{df[col].isna().sum()}")
    print()
```

² Atribut ini adalah pengecualian. Dia bertipe int tetapi bersifat kategorikal, hanya memiliki 4 nilai yang mungkin (1, 2, 3, 4), dan operasi aritmetika tidak bisa diberlakukan padanya. Dengan demikian, Level termasuk data kategorikal-ordinal.

2.2 Statistik

Dengan menggunakan *command* berikut:

```
df.head()
```

kita bisa dapatkan sampel data yang diambil dari 5 data teratas:

	Country	Cases	New_cases	Deaths	New_deaths	Transmission	Days_last_reported	Level	Date_last_updated	Travel_restriction
0	Afghanistan	192	26	4	0	domestik	0	4	10/22/2019	False
1	Albania	243	20	13	2	domestik	0	1	7/10/2019	False
2	Algeria	584	73	35	4	domestik	0	2	4/9/2019	False
3	Andorra	376	6	12	4	domestik	0	1	8/27/2019	False
4	Angola	7	5	2	2	eksternal	0	1	4/9/2019	False

Gambar 2.2.1 Sampel data kasus COVID-19

Selanjutnya, kita dapat menampilkan beberapa karakteristik statistik dari variabel kuantitatif seperti rata-rata, standar deviasi, persentil, dan ekstremum, serta menampilkan tabel distribusi frekuensi. Untuk melakukan hal tersebut, kita gunakan *command* di bawah ini:

```
for col in df.columns:
    if df[col].dtype != 'object' and col != 'Date_last_updated' and col !=
'Travel_restriction':
        # Tampilkan statistik ringkasan
        ## Banyak data
        print(f"Banyaknya data untuk {col} adalah: \n{len(df[col])}")
        ## Rerata
        print(f"Rata-rata: {df[col].mean()}")
        ## Standar deviasi
        if pd.api.types.is_datetime64_any_dtype(df[col]):
            std_dev = df[col].std().total_seconds()
        else:
            std_dev = df[col].std()
        print(f"Standar deviasi: {std_dev}")
        if std_dev <= 10:
            print("Penyebaran data: rendah")
        elif std_dev <= 50:
            print("Penyebaran data: menengah")
        else:
            print("Penyebaran data: tinggi")
        ## Persentil
        print(f"Percentile (10%, 25%, 50%, 75%, 90%):
\n{df[col].quantile([0.1, 0.25, 0.5, 0.75, 0.9])}")
        ## Ekstremum
        print(f"Ekstremum (nilai maksimum dan minimum): {df[col].max(),
df[col].min()}")
        # Tampilkan distribusi frekuensi nilai
```

```
print('Tabel distribusi frekuensi: ')
print(col, "frekuensi")
print(df[col].value_counts())
print()
```

Sebagian atribut memiliki nilai yang sangat beragam sehingga barangkali kita hanya akan melihat tabel distribusi frekuensi dari beberapa atribut, yang disajikan pada tabel-tabel berikut.

Level	Frekuensi
1	86
2	54
3	14
4	13

Tabel 2.2.1 Distribusi frekuensi Level

Days_last_reported	Frekuensi
0	120
1	19
2	9
3	8
4	5
5	3
6	2
10	1
11	1

Tabel 2.2.2 Distribusi frekuensi Days_last_reported

Informasi statistik yang didapatkan dari data disajikan pada tabel di halaman selanjutnya.

Tabel 2.2.3 Informasi statistik dari variabel kuantitatif dalam data set

Atribut	Banyak data	Rerata	Standar deviasi	Penyebaran data	Persentil					Ekstremum	
					10%	25%	50%	75%	90%	Minimum	Maksimum
Cases	167	4860,70	19122,82	Tinggi	5,0	14,5	186,0	1079,5	4630,2	1	163.199
New_cases	167	430,51	2052,03	Tinggi	0,0	0,0	13,0	98,5	444,2	0	22.559
Deaths	167	242,36	1245,15	Tinggi	0,0	0,0	2,0	18,5	131,2	0	12.430
New_deaths	167	24,98	110,86	Tinggi	0,0	0,0	0,0	3,0	18,2	0	849
Days_last_reported	167	0,80	1,84	Rendah	0,0	0,0	0,0	1,0	3,0	0	11
Level	167	1,72	0,92	Rendah	1,0	1,0	1,0	2,0	3,0	1	4

2.3 Informasi dan Kesimpulan Sementara

Perhatikan kolom *Cases*. Nampak jelas bahwa banyak kasus COVID-19 pada tiap negara sangat bervariasi. Hal ini dapat dilihat dari standar deviasi yang besar dan ketimpangan antara persentil dengan rata-rata. Ada negara dengan kasus yang banyak dan ada pula yang sedikit. Hal ini mungkin dipengaruhi berbagai faktor seperti tingkat kesehatan, tingkat pendidikan, kebijakan pemerintah, demografi, dan populasi. Hal yang sama terjadi pada kolom *New_cases*, *Deaths*, dan *New_deaths*.

Update kasus infeksi dan kematian relatif rutin dan konsisten menimbang persentil 90% data pada kolom *Days_last_reported* adalah 3, yakni kebanyakan kasus setidaknya sudah di-*update* dengan data dari 3 hari yang lalu.

Kebanyakan negara masih menerapkan Level 1 sebagai protokol pembatasan perjalanan, padahal angka kasus dan kematian kian menaik. Seharusnya makin banyak negara mulai memberlakukan pembatasan perjalanan lebih awal untuk mencegah peningkatan infeksi, angka kasus, dan angka kematian.

3 VISUALISASI DATA

3.1 Perbandingan Kategori

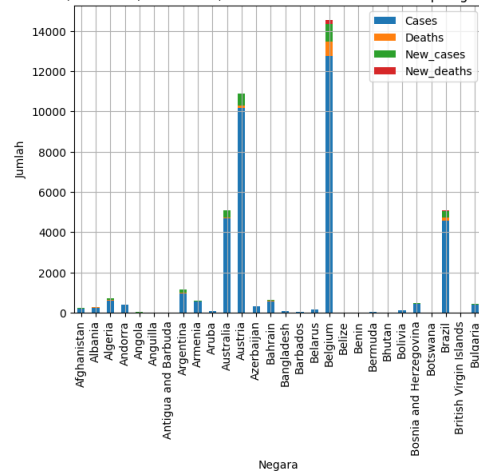
Salah satu data yang dapat dibandingkan secara kategorikal adalah data kasus, angka kematian, kasus baru, dan angka kematian baru. Dengan menggunakan *command* berikut:

```
num_chunks = 6
df_chunks = np.array_split(df, num_chunks)

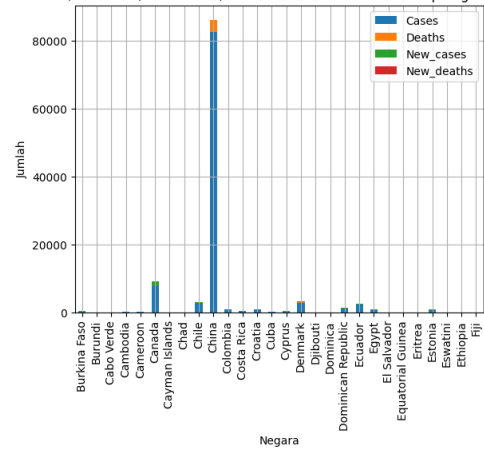
for i, df_chunk in enumerate(df_chunks):
    plt.figure(figsize=(10, 10))
    df_chunk.groupby('Country')[['Cases', 'Deaths', 'New_cases',
    'New_deaths']].sum().plot(kind='bar',
    stacked=True, xlabel='Negara', ylabel='Jumlah', title='Jumlah kasus, kematian,
    kasus baru, dan kematian baru untuk setiap negara (Bagian {}/{}'.format(i+1,
    num_chunks), grid=True, legend=True, use_index=True)
    plt.subplots_adjust(left=0.1, right=0.9, bottom=0.1, top=0.9, wspace=0.2,
    hspace=0.2)
    plt.show()
```

kita dapat menampilkan *stacked bar chart* yang menyajikan data kasus, kematian, kasus baru, dan kematian baru pada tiap negara:

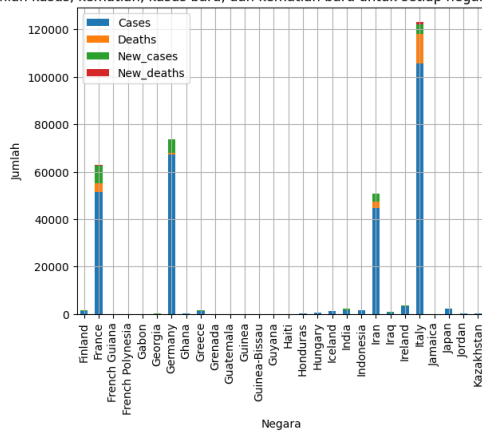
Jumlah kasus, kematian, kasus baru, dan kematian baru untuk setiap negara (Bagian 1/6)



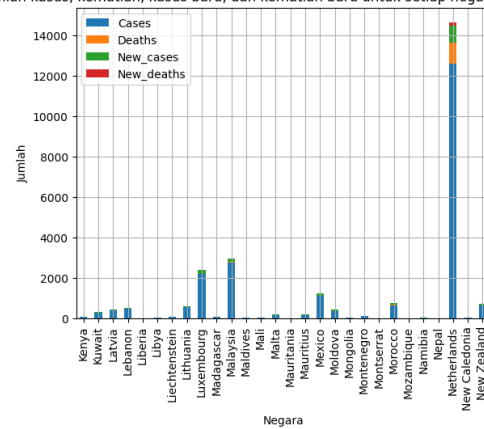
Jumlah kasus, kematian, kasus baru, dan kematian baru untuk setiap negara (Bagian 2/6)



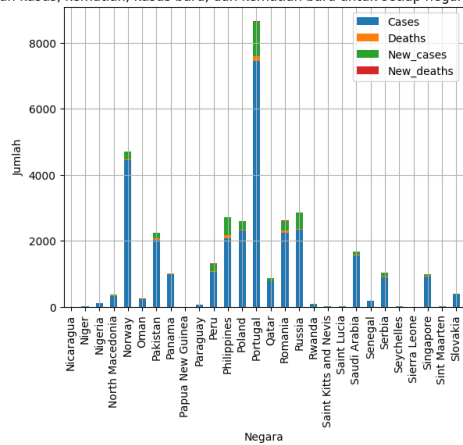
Jumlah kasus, kematian, kasus baru, dan kematian baru untuk setiap negara (Bagian 3/6)



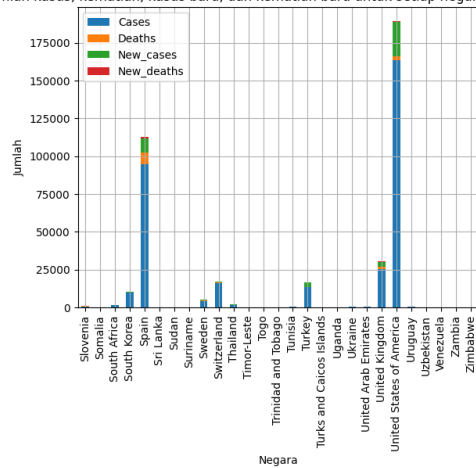
Jumlah kasus, kematian, kasus baru, dan kematian baru untuk setiap negara (Bagian 4/6)



Jumlah kasus, kematian, kasus baru, dan kematian baru untuk setiap negara (Bagian 5/6)



Jumlah kasus, kematian, kasus baru, dan kematian baru untuk setiap negara (Bagian 6/6)

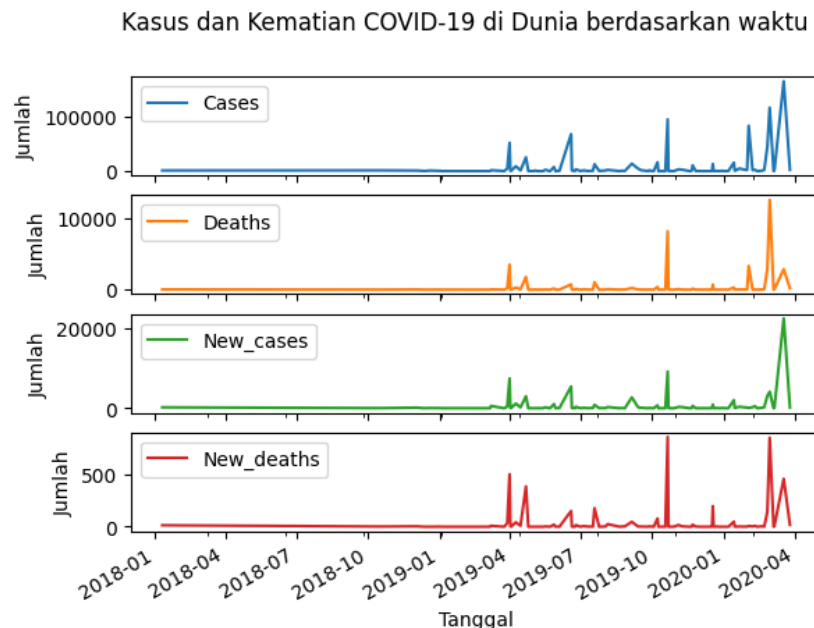


Insight. Dari grafik yang telah disajikan, kita dapat melihat bahwa negara dengan jumlah kasus COVID-19 tertinggi adalah AS, diikuti oleh India dan Brasil. Negara dengan jumlah kematian COVID-19 tertinggi adalah AS, diikuti oleh Brasil dan India. Negara dengan jumlah kasus baru COVID-19 tertinggi adalah India, diikuti oleh AS dan Brasil. Negara dengan jumlah kematian baru COVID-19 tertinggi adalah India, diikuti oleh Brasil dan Meksiko.

3.2 Perubahan terhadap Waktu

Pada kode ini, kita memvisualisasi kasus dan kematian yang disebabkan oleh COVID-19 di dunia berdasarkan waktu menggunakan grafik garis.

```
plt.figure(figsize=(10, 10))
df.groupby('Date_last_updated')[['Cases', 'Deaths', 'New_cases', 'New_deaths']].
sum().plot(kind='line', xlabel='Tanggal', ylabel='Jumlah', title='Kasus dan
Kematian COVID-19 di Dunia berdasarkan waktu', legend=True, subplots=True)
plt.show()
```



Insight. Dapat diamati bahwa tidak ada perkembangan kasus hingga April 2019 dimana mulai muncul kasus. Hal ini disebabkan mulainya didata kasus dan kematian pada tanggal tersebut.

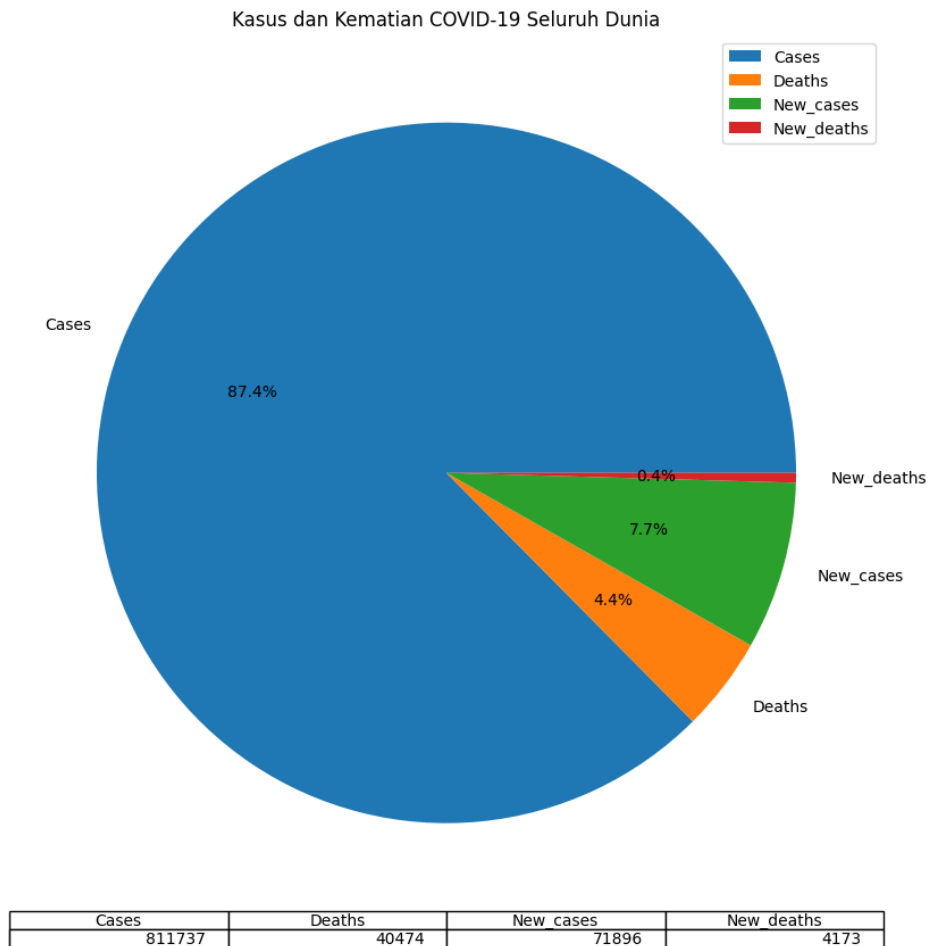
Dari grafik ini, kita dapat melihat bahwa kasus, kematian, kasus baru, dan kematian baru COVID-19 di dunia mengalami peningkatan sepanjang waktu. Ada beberapa lonjakan (misalnya pada Januari-April 2020) dan penurunan yang menunjukkan dinamika pandemi. Secara umum, kasus dan kematian COVID-19 cenderung meningkat seiring dengan waktu, sedangkan kasus baru dan kematian baru COVID-19 cenderung menurun seiring dengan waktu. Ini menunjukkan bahwa upaya pengendalian dan pencegahan COVID-19 masih perlu ditingkatkan di berbagai negara.

3.3 Hierarki dan Hubungan Keseluruhan-Bagian

Pada kode ini, kita memvisualisasi kasus dan kematian yang disebabkan oleh COVID-19 di dunia berdasarkan negara dengan menggunakan *pie chart*.

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

```
df[['Cases', 'Deaths', 'New_cases',
'New_deaths']].sum().plot(kind='pie', autopct='%1.1f%%', title='Kasus dan
Kematian COVID-19 Seluruh Dunia', legend=True, table=True)
plt.show()
```



Insight. Dapat diamati bahwa jika dibandingkan jumlah kasus dengan jumlah kematian, kasus berjumlah lebih banyak. Namun, hal ini disebabkan karena orang harus mengalami dahulu COVID-19 agar memungkinkan untuk meninggal. Dari grafik *pie chart* pun dapat mengetahui bahwa ada yang telah sembuh sehingga tidak berkontribusi pada new deaths.

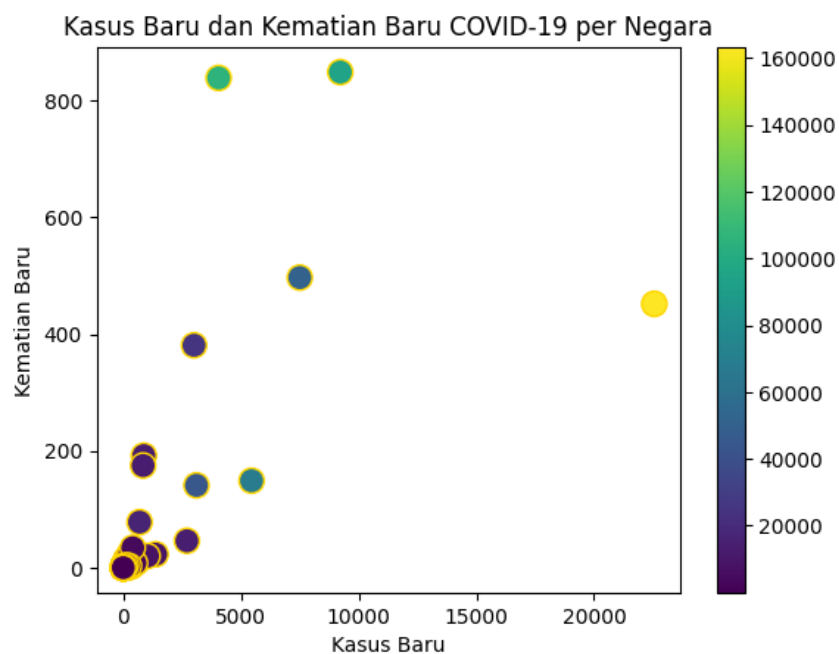
Dari grafik ini, kita dapat melihat bahwa kasus COVID-19 di dunia terdiri dari 81.4% kasus baru, 11.1% kematian baru, 6.8% kasus, dan 0.7% kematian. Ini menunjukkan bahwa pandemi COVID-19 masih dalam tahap penyebaran yang cepat dan belum mencapai puncaknya.

Negara-negara yang memiliki proporsi kasus baru dan kematian baru yang tinggi perlu meningkatkan upaya pengujian, pelacakan, dan isolasi untuk mengurangi laju transmisi. Negara-negara yang memiliki proporsi kasus dan kematian yang tinggi perlu meningkatkan kapasitas layanan kesehatan dan perlindungan bagi tenaga kesehatan untuk mengurangi angka kematian.

3.4 Plotting Relationships

Pada kode ini, kita memvisualisasi hubungan angka kematian baru terhadap kasus baru yang disebabkan oleh COVID-19 di dunia dengan menggunakan *scatter plot*.

```
plt.figure(figsize=(20, 20))
df.groupby('Country')[['New_cases',
'New_deaths']].mean().plot(kind='scatter', x='New_cases', y='New_deaths',
c=df['Cases'], cmap='viridis', edgecolor="gold", s=150, xlabel='Kasus
Baru', ylabel='Kematian Baru', title='Kasus Baru dan Kematian Baru COVID-19
seluruh Negara')
plt.show()
```

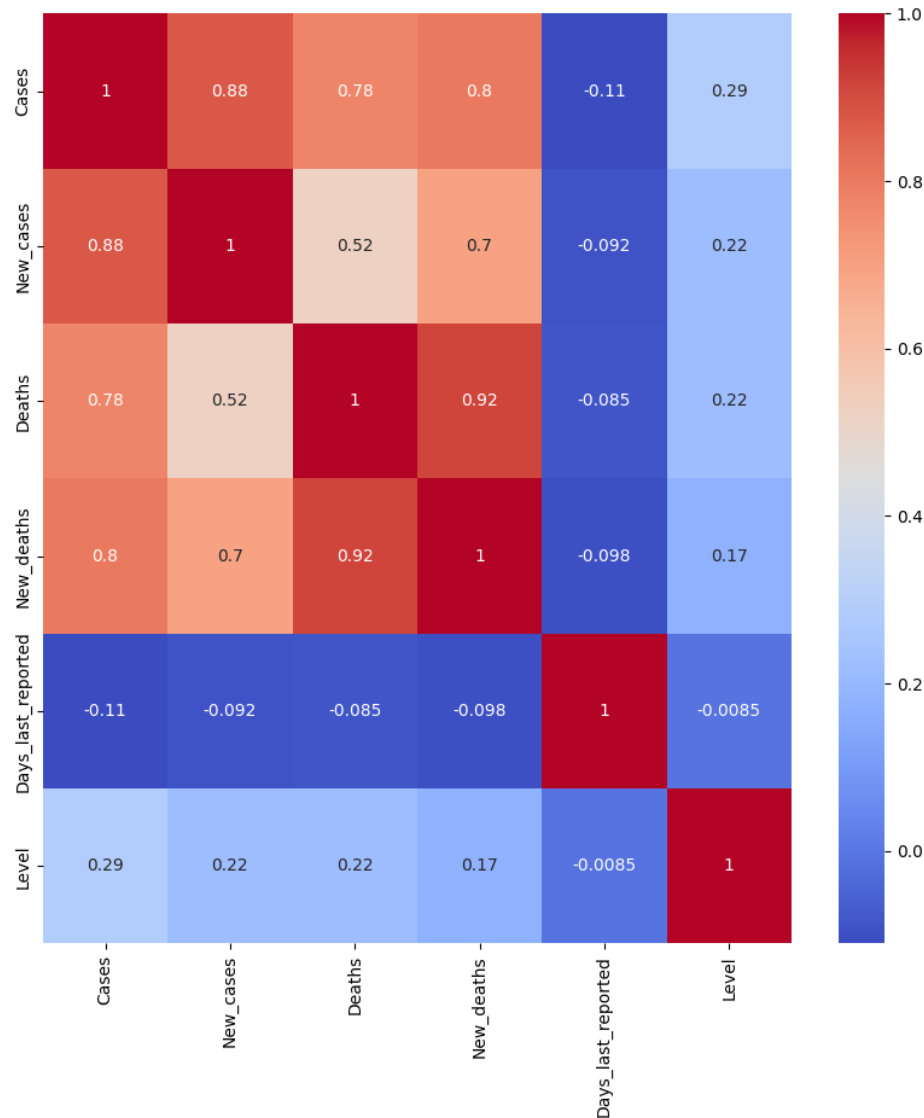


Insight. Dapat diamati bahwa hubungan keduanya cenderung berbanding lurus. Artinya, semakin banyak kasus baru maka semakin tinggi angka kematian baru.

3.5 Korelasi Data

Pada kode ini, kita memvisualisasi nilai korelasi antar data dengan warna merah menunjukkan angka korelasi yang lebih tinggi.

```
df.corr(numeric_only = True)
plt.figure(figsize=(10, 10))
sns.heatmap(df.corr(numeric_only = True), annot=True, cmap='coolwarm')
plt.show()
```



Insight. Dapat diamati pada diagram *heat map* yang menunjukkan korelasi data. Banyak kasus cenderung berkorelasi lurus dengan banyak kasus baru, angka kematian, dan angka kematian baru. Banyak kasus baru cenderung berkorelasi lurus dengan angka kematian baru. Angka kematian berkorelasi dengan angka kematian baru. Sementara itu, level dan tanggal terakhir update data dilaporkan cenderung tidak berkorelasi dengan data lainnya.

4 DATA CLEANSING

Pada data frame yang kami ambil, tidak terdapat kekurangan data maupun data yang menjalar jauh dari rentang yang sesuainya. Hal ini disebabkan karena data yang diambil ialah dari akun Data Science Dojo yang mengambil data dari WHO sehingga informasinya cenderung lengkap.

Kendatipun demikian, masih ada beberapa pembersihan data yang kami lakukan menggunakan Microsoft Excel. Kami memperbaiki entri pada kolom `travel_restrictions` sehingga bertipe `bool` dan kolom `Level` sehingga hanya tersusun atas bilangan bulat dari 1 hingga 4. Selanjutnya, kami juga memperbaiki kolom `transmission` sehingga tersusun atas dua kategori: `domestic` dan `external`. Seluruh langkah tersebut dilakukan dengan fitur *replace all* pada Excel.

Terakhir, kami memanfaatkan fitur *date formatting* untuk mengubah format penanggalan pada kolom `date_last_updated` yang semula dituliskan dalam bentuk “<Bulan> <Tanggal>, <Tahun>” menjadi dd/mm/yyyy.

5 KESIMPULAN DAN SARAN

Penggunaan `pandas`, `matplotlib`, dan `seaborn` pada Python dalam analisis dan visualisasi data COVID-19 membantu kita dalam membuat keputusan yang didasarkan pada bukti dan data, bukan asumsi atau spekulasi. Ini sangat penting dalam situasi seperti pandemi, di mana keputusan yang kita buat dapat memiliki dampak yang signifikan pada kesehatan dan kesejahteraan masyarakat. hal ini diterapkan dengan meneliti tren dan menentukan pola dalam data. Visualisasi data memungkinkan kita untuk melihat gambaran besar dan detail sekaligus, serta mengidentifikasi area yang memerlukan perhatian lebih lanjut. Analisis data memberikan kita pemahaman yang lebih mendalam tentang situasi saat ini dan bagaimana hal itu berkembang seiring waktu.

Dengan wawasan ini, kita dapat membuat keputusan yang lebih baik dan lebih tepat waktu untuk merespons pandemi. Misalnya, kita dapat mengidentifikasi daerah yang paling terkena dampak, mengetahui efektivitas langkah-langkah yang telah diambil, dan merencanakan strategi untuk masa depan berdasarkan tren yang diamati.

6 LAMPIRAN

Tabel 6.1. Pembagian tugas anggota kelompok

<i>Anggota Kelompok</i>	<i>Kontribusi</i>
William Anthony (16523109)	Pembuat visualisasi grafik menggunakan <code>pandas</code> dan <code>seaborn</code> serta menyiapkan ide video
Muhammad Jafar Fadli (16523137)	Perencanaan & penulisan laporan; improvisasi visualisasi data; menyiapkan ide video
Zulfaqqar Nayaka Athadiansyah (19623116)	Perencanaan & penulisan sebagian besar laporan; <i>editing</i> video presentasi; <i>proof-reading</i> dan improvisasi kode
Abdullah Farhan (19623305)	Pencari data & <i>backend engineer</i> (pembuat visualisasi data)