

**LAPORAN PROJECT AKHIR
DATA VISUALIZATION (BA183)**



**VISUALISASI KASUS COVID-19 BERDASARKAN PROVINSI
DI INDONESIA**

Oleh:

PUTU ANDIKA TEDJA PERMANA (180030302)

**PROGRAM STUDI SISTEM INFORMASI
INSTITUT TEKNOLOGI DAN BISNIS STIKOM BALI
JUNI 2021**

RINGKASAN

Saat ini dunia sedang dihadapi oleh pandemi COVID-19. Indonesia adalah salah satu negara yang terdampak pandemi COVID-19. Setiap provinsi di Indonesia diharapkan melapor jumlah kasus yang terjadi per hari. Pemerintah pusat wajib mengetahui bagaimana persebaran kasus COVID-19 di setiap provinsi, untuk melakukan pengambilan keputusan kedepannya. Keputusan tersebut dapat berupa provinsi mana yang menjadi prioritas untuk di awasi baik itu dari jumlah kasus positif, jumlah sembuh, dan jumlah kematian. Untuk mengetahui hal tersebut, pemerintah pusat perlu melakukan visualisasi data untuk mempermudah dan mempercepat membaca data-data yang telah dikumpulkan oleh pemerintah setiap provinsi. Setelah melakukan visualisasi data, diharapkan pemerintah pusat dapat mengetahui provinsi mana yang perlu mendapatkan perhatian khusus.

KATA PENGANTAR

Puji syukur penulis panjatkan kehadapan Tuhan Yang Maha Esa atas berkat rahmat-Nya penulis dapat menyelesaikan laporan project Ujian Akhir Semester (UAS) mata kuliah Data Visualization yang berjudul “Visualisasi Kasus COVID-19 Berdasarkan Provinsi di Indonesia” sesuai dengan waktu yang telah direncanakan. Penulis ingin menyampaikan terima kasih kepada:

1. Bapak IGKG Puritan Wijaya ADH, S.Kom.,MMSI selaku dosen mata kuliah Data Visualization
2. Semua teman dan berbagai pihak yang telah memberikan dukungan dan bantuan kepada penulis.

Semoga penulisan laporan project UAS ini dapat bermanfaat bagi pihak yang berkepentingan.

Denpasar, 6 Juni 2021

Penulis

DAFTAR ISI

RINGKASAN	i
KATA PENGANTAR	ii
DAFTAR ISI.....	iii
DAFTAR TABEL.....	v
DAFTAR GAMBAR	vi
DAFTAR LAMPIRAN.....	vii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	2
BAB II TINJAUAN PUSTAKA	3
2.1 State of The Art.....	3
2.2 Visualisasi	3
2.3 COVID-19.....	4
2.4 Python	4
BAB III COVID-19 INDONESIA DATASET (CASE AND VACCINATION) ...	5
3.1 Deskripsi Teknis	5
3.2 Preprocessing Data	7
3.2.1 Data Total Kasus, Sembuh, Kematian, dan Rasio Per Provinsi.....	8
3.2.2 Data Pertumbuhan dan Penambahan Total Kasus Per Bulan	9
3.3 Dataset Akhir	10
3.3.1 Dataset Total Kasus, Sembuh, Kematian, dan Rasio Per Provinsi	10
3.3.2 Data Pertumbuhan Total Kasus Per Bulan.....	11
BAB IV VISUALISASI DATA	13
4.1 Analisa Permasalahan	13
4.1.1 Mengetahui 10 Provinsi Dengan Kasus Terbanyak.....	13
4.1.2 Mengetahui 10 Provinsi Dengan Rasio Kematian Tertinggi	13
4.1.3 Mengetahui 10 Provinsi Dengan Rasio Sembuh Tertinggi.....	13
4.1.4 Mengetahui Penambahan Total Kasus Per Bulan	13
4.1.5 Mengetahui Pertumbuhan Total Kasus Per Bulan	13

4.2 Rancangan Visualisasi	14
4.2.1 Rancangan 10 Provinsi Dengan Kasus Terbanyak	14
4.2.2 Rancangan 10 Provinsi Dengan Rasio Kematian Tertinggi	14
4.2.3 Rancangan 10 Provinsi Dengan Rasio Sembuh Terendah.....	15
4.2.4 Rancangan Penambahan Total Kasus Per Bulan	15
4.2.5 Rancangan Pertumbuhan Total Kasus Per Bulan	16
4.3 Implementasi dan Hasil Visualisasi	17
4.3.1 Visualisasi 10 Provinsi Dengan Kasus Terbanyak	17
4.3.2 Visualisasi 10 Provinsi Dengan Rasio Kematian Tertinggi.....	18
4.3.3 Visualisasi 10 Provinsi Dengan Rasio Sembuh Terendah.....	20
4.3.4 Visualisasi Penambahan Total Kasus Per Bulan.....	22
4.3.5 Visualisasi Pertumbuhan Total Kasus Per Bulan.....	24
BAB V KESIMPULAN.....	27
DAFTAR PUSTAKA	28
LAMPIRAN.....	29
SUMBER DATA	30

DAFTAR TABEL

Tabel 2.1 State of The Art.....	3
Tabel 3.1 Dataset Total Kasus per Provinsi	10
Tabel 3.2 Data Pertumbuhan dan Penambahan Total Kasus Per Bulan	12

DAFTAR GAMBAR

Gambar 3.1 Sepuluh (10) Data Teratas dari Dataset	6
Gambar 3.2 Tipe Data Setiap Kolom Pada Dataset	6
Gambar 3.3 Deskripsi Dataframe.....	7
Gambar 3.4 Gambar Pengecekan Value Null	7
Gambar 4.1 Rancangan 10 Provinsi Dengan Kasus Terbanyak	14
Gambar 4.2 Rancangan 10 Provinsi Dengan Rasio Kematian Tertinggi.....	14
Gambar 4.3 Rancangan 10 Provinsi Dengan Rasio Sembuh Terendah.....	15
Gambar 4.4 Rancangan Penambahan Total Kasus Per Bulan	16
Gambar 4.5 Rancangan Pertumbuhan Total Kasus Per Bulan.....	16
Gambar 4.6 Visualisasi 10 Provinsi Dengan Kasus Terbanyak.....	18
Gambar 4.7 Visualisasi 10 Provinsi Dengan Rasio Kematian Tertinggi.....	20
Gambar 4.8 Visualisasi 10 Provinsi Dengan Rasio Sembuh Terendah	22
Gambar 4.9 Visualisasi Penambahan Total Kasus Per Bulan.....	24
Gambar 4.10 Visualisasi Pertumbuhan Total Kasus Per Bulan.....	26

DAFTAR LAMPIRAN

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saat ini dunia sedang menghadapi pandemi COVID-19. Indonesia merupakan salah satu negara yang terkena pandemi COVID-19. Per 23 Mei 2021 menurut situs covid19.go.id terdapat 1.758.898 kasus positif. Pandemi ini menyebabkan berbagai kegiatan menjadi terhambat bahkan tidak berjalan sama sekali. Perekonomian juga menjadi menurun karena pembatasan sosial yang menyebabkan banyak pekerjaan menjadi harus dihentikan. Oleh karena itu, pandemi ini perlu mendapatkan perhatian khusus pemerintah terutama pemerintah pusat.

Pemerintah perlu mengetahui bagaimana persebaran virus COVID-19 yang terjadi di Indonesia. Setiap provinsi harus mencatat berapa kasus yang terjadi di provinsi masing-masing. Pemerintah pusat menganalisis data-data dari provinsi tersebut dan mengetahui provinsi mana yang perlu mendapatkan perhatian lebih. Aspek yang mempengaruhi perhatian pemerintah adalah seperti jumlah kasus per hari, jumlah kematian, dan jumlah sembuh.

Terdapat sebuah dataset berisi persebaran virus COVID-19 di Indonesia untuk setiap provinsi. Dataset ini didapat dari Kaggle dengan nama COVID-19 Indonesia Dataset (Case and Vaccination) [1]. Dataset ini dapat dijadikan sebagai acuan untuk mengetahui bagaimana persebaran pandemi COVID-19 baik itu jumlah kasus per hari, jumlah kematian, dan jumlah sembuh. Namun banyaknya data yang ada membuat sulit untuk membaca bagaimana persebaran pandemi COVID-19 di setiap provinsi.

Oleh karena itu penting adanya visualisasi data untuk mempercepat dan mempermudah membaca data-data yang ada, sehingga proses pengambilan keputusan menjadi lebih cepat. Sebelum melakukan visualisasi data, penting juga dilakukan proses pengolahan data. Data-data yang ada harus diproses menjadi data yang siap untuk divisualisasikan.

1.2 Perumusan Masalah

Dari latar belakang tersebut, dapat diuraikan perumusan masalah yang terjadi yaitu bagaimana pengolahan data hingga visualisasi untuk dataset persebaran COVID-19 di Indonesia untuk setiap provinsi.

1.3 Batasan Masalah

Adapun batasan masalah yang ada dalam pembuatan visualisasi data persebaran COVID-19 di Indonesia untuk setiap provinsi sebagai berikut.

1. Dataset yang digunakan adalah dataset kasus COVID-19 yang tanggal recordnya hingga tanggal 19 Maret 2021.
2. Visualisasi yang dilakukan adalah visualisasi data 10 provinsi teratas dengan total kasus terbanyak, 10 provinsi dengan rasio kematian tertinggi, 10 provinsi dengan rasio sembuh terendah, penambahan total kasus setiap bulan, dan pertumbuhan total kasus setiap bulan.
3. Bahasa pemrograman yang digunakan adalah Python.
4. Library yang digunakan adalah pandas, datetime, dan matplotlib.
5. Platform untuk visualisasi menggunakan Jupyter Notebook dan Google Colab.

BAB II

TINJAUAN PUSTAKA

2.1 State of The Art

State of The Art adalah kumpulan hasil-hasil terdahulu yang pembahasannya terkait dengan pembahasan yang akan dilakukan pada laporan ini. Berikut ini adalah state of the art dalam bentuk tabel yang telah penulis kumpulkan.

Tabel 2.1 State of The Art

Penulis	Judul	Hasil dan Pembahasna
Saepuloh, Dani [2]	Visualisasi Data Covid 19 Provinsi DKI Menggunakan Tableau	Hasil dari penelitian ini adalah visualisasi dari data COVID-19 di provinsi DKI Jakarta dengan menggunakan software Tableau. Data yang digunakan diperoleh dari website https://corona.jakarta.go.id . Hasil visualisasi cukup interaktif.

2.2 Visualisasi

Menurut [3] definisi viusalisasi adalah menggunakan teknologi komputer sebagai pendukung untuk melakukan penggambaran data visual yang interaktif untuk memperkuat pengamatan. Menurut Kamus Besar Bahasa Indonesia (KBBI), visualisasi adalah pengungkapan suatu gagasan atau perasaan dengan menggunakan bentuk gambar, tulisan (kata dan angka), peta, grafik, dan sebagainya [4]. Dari beberapa pengertian di atas, visualisasi adalah pengungkapan suatu gagasan atau perasaan dengan menggunakan bentuk gambar, tulisan (kata dan angka), peta, grafik, dan sebagainya menggunakan teknologi komputer untuk memperkuat pengamatan.

2.3 COVID-19

Coronavirus merupakan keluarga besar virus yang menyebabkan penyakit pada manusia dan hewan. Pada manusia biasanya menyebabkan penyakit infeksi saluran pernapasan, mulai flu biasa hingga penyakit yang serius seperti Middle East Respiratory Syndrome (MERS) dan Sindrom Pernafasan Akut Berat/ Severe Acute Respiratory Syndrome (SARS). Coronavirus jenis baru yang ditemukan pada manusia sejak kejadian luar biasa muncul di Wuhan Cina, pada Desember 2019, kemudian diberi nama Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-COV2), dan menyebabkan penyakit Coronavirus Disease-2019 (COVID-19) [5].

2.4 Python

Python adalah bahasa pemrograman general-purpose yang secara spesifik didesain untuk membuat program-program dapat dibaca dengan sangat mudah. Python juga memiliki pustaka (library) yang banyak sehingga memungkinkan untuk membangun aplikasi yang canggih menggunakan kode yang tampak relatif sederhana. Untuk itu, Python telah menjadi bahasa pengembangan aplikasi yang populer dan juga bahasa pemrograman "pertama" yang dipilih. Sedikit sejarah mengenai bahasa pemrograman Python, bahasa pemrograman ini dikembangkan pada akhir 1980-an oleh pemrogram Belanda Guido van Rossum saat bekerja di CWI (Centrum voor Wiskunde en Informatica di Amsterdam, Belanda). Bahasa pemrograman Python tidak dinamai dari suatu spesies ular besar yang ada, melainkan diambil dari serial komedi BBC Monty Python's Flying Circus [6].

BAB III

COVID-19 INDONESIA DATASET (CASE AND VACCINATION)

3.1 Deskripsi Teknis

Dataset “COVID-19 Indonesia Dataset (Case And Vaccination)” adalah dataset yang berisi data kasus harian COVID-19 dan vaksinasi di Indonesia. Penulis hanya menggunakan dataset kasus harian COVID-19. Terdapat 12316 baris pada dataset ini. Terdapat 10 kolom atau variabel pada dataset ini yaitu Date, Province, Daily_Case, Daily_Death, Daily_Recovered, Active_Case, Cumulative_Case, Cumulative_Recovered, Cumulative_Death, CumulativeActiveCase. Terdapat 34 provinsi di dataset ini. Record tiap lokasi (provinsi) memiliki perbedaan tanggal kasus pertama. Tanggal record terakhir pada dataset ini adalah tanggal 19 Maret 2021. Sumber dari dataset ini dari Kaggle [1].

Penjelasan Kolom:

- **Date**, yaitu tanggal terjadinya atau dicatatnya kasus baru (positif, sembuh, atau meninggal) yang terjadi.
- **Province**, yaitu provinsi terjadinya kasus.
- **Daily_Case**, yaitu banyaknya kasus positif baru yang terkonfirmasi pada tanggal tertentu yang terjadi di suatu provinsi.
- **Daily_Death**, yaitu banyaknya kasus kematian baru yang terkonfirmasi pada tanggal tertentu yang terjadi di suatu provinsi.
- **Daily_Recovered**, yaitu banyaknya angka yang terkonfirmasi sembuh pada tanggal tertentu yang terjadi di suatu provinsi.
- **Active_Case**, yaitu jumlah kasus aktif baru setiap hari seperti diisolasi atau dirawat di rumah sakit (masih belum sembuh atau meninggal).
- **Cumulative_Case**, yaitu total untuk setiap hari jumlah kasus yang dikonfirmasi dari baris tanggal, untuk baris provinsi tersebut.
- **Cumulative_Recovered**, yaitu total untuk setiap hari jumlah sembuh yang dikonfirmasi dari baris tanggal, untuk baris provinsi tersebut.

- **Cumulative_Death**, yaitu total untuk setiap hari jumlah kematian yang dikonfirmasi dari baris tanggal tersebut, untuk baris provinsi tersebut.
- **CumulativeActiveCase**, yaitu total untuk setiap hari jumlah kasus aktif dari baris tanggal, untuk baris provinsi tersebut.

Dataset di-load dengan menggunakan dataframe dari library pandas. Berikut ini adalah kode untuk loading dataset.

```
corona_df = pd.read_csv('Indonesia_coronavirus_daily_data.csv')
```

Berikut ini screenshot sepuluh (10) data teratas dari dataset:

	Date	Province	Daily_Case	Daily_Death	Daily_Recovered	Active_Case	Cumulative_Case	Cumulative_Recovered	Cumulative_Death	Cumulative_Active_Case
0	01/03/2020	DKI JAKARTA	2	0	0	2	2	0	0	2
1	02/03/2020	DKI JAKARTA	2	0	0	2	4	0	0	4
2	03/03/2020	DKI JAKARTA	2	0	0	2	6	0	0	6
3	04/03/2020	DKI JAKARTA	2	0	0	2	8	0	0	8
4	05/03/2020	DKI JAKARTA	0	1	0	-1	8	0	1	7
5	06/03/2020	DKI JAKARTA	0	0	0	0	8	0	1	7
6	07/03/2020	DKI JAKARTA	0	2	0	-2	8	0	3	5
7	08/03/2020	DKI JAKARTA	0	0	0	0	8	0	3	5
8	09/03/2020	DKI JAKARTA	0	1	0	-1	8	0	4	4
9	10/03/2020	DKI JAKARTA	0	0	0	0	8	0	4	4

Gambar 3.1 Sepuluh (10) Data Teratas dari Dataset

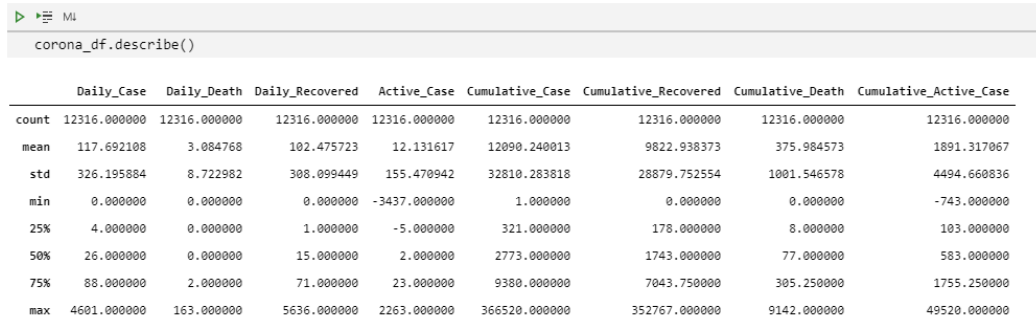
Berikut ini adalah tipe data dari setiap kolom pada dataset dengan menggunakan fungsi info().

```
corona_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12316 entries, 0 to 12315
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Date                  12316 non-null  object
1   Province              12316 non-null  object
2   Daily_Case            12316 non-null  int64
3   Daily_Death           12316 non-null  int64
4   Daily_Recovered       12316 non-null  int64
5   Active_Case           12316 non-null  int64
6   Cumulative_Case       12316 non-null  int64
7   Cumulative_Recovered  12316 non-null  int64
8   Cumulative_Death      12316 non-null  int64
9   Cumulative_Active_Case 12316 non-null  int64
dtypes: int64(8), object(2)
memory usage: 962.3+ KB
```

Gambar 3.2 Tipe Data Setiap Kolom Pada Dataset

Berikut ini adalah hasil setelah menjalankan fungsi `describe()`.

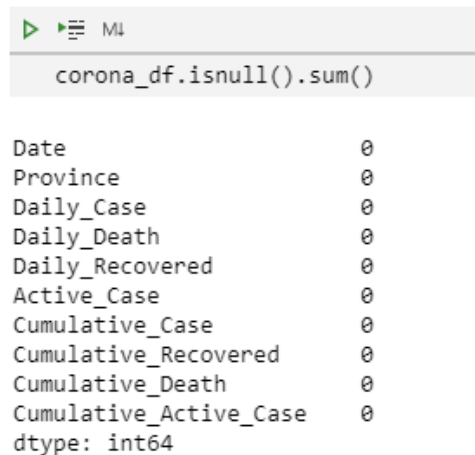


```
corona_df.describe()
```

	Daily_Case	Daily_Death	Daily_Recovered	Active_Case	Cumulative_Case	Cumulative_Recovered	Cumulative_Death	Cumulative_Active_Case
count	12316.000000	12316.000000	12316.000000	12316.000000	12316.000000	12316.000000	12316.000000	12316.000000
mean	117.692108	3.084768	102.475723	12.131617	12090.240013	9822.938373	375.984573	1891.317067
std	326.195884	8.722982	308.099449	155.470942	32810.283818	28879.752554	1001.546578	4494.660836
min	0.000000	0.000000	0.000000	-3437.000000	1.000000	0.000000	0.000000	-743.000000
25%	4.000000	0.000000	1.000000	-5.000000	321.000000	178.000000	8.000000	103.000000
50%	26.000000	0.000000	15.000000	2.000000	2773.000000	1743.000000	77.000000	583.000000
75%	88.000000	2.000000	71.000000	23.000000	9380.000000	7043.750000	305.250000	1755.250000
max	4601.000000	163.000000	5636.000000	2263.000000	366520.000000	352767.000000	9142.000000	49520.000000

Gambar 3.3 Deskripsi Dataframe

Untuk mengecek apakah ada data yang null, maka dilakukan pemanggilan perintah untuk mengecek data null sebagai berikut.



```
corona_df.isnull().sum()
```

```
Date      0
Province  0
Daily_Case      0
Daily_Death     0
Daily_Recovered  0
Active_Case     0
Cumulative_Case  0
Cumulative_Recovered  0
Cumulative_Death  0
Cumulative_Active_Case  0
dtype: int64
```

Gambar 3.4 Gambar Pengecekan Value Null

3.2 Preprocessing Data

Pada deskripsi teknis sebelumnya diketahui bahwa dataset tidak memiliki nilai null, sehingga tidak dilakukan preprocessing untuk mengisi nilai null. Selanjutnya, preprocessing data dilakukan untuk mengubah atau memperbaiki dataset awal menjadi dataset-dataset yang lebih kecil agar lebih mudah untuk diolah menjadi visualisasi. Berikut ini adalah beberapa tahapan preprocessing data untuk membuat dataset-dataset kecil.

3.2.1 Data Total Kasus, Sembuh, Kematian, dan Rasio Per Provinsi

Untuk mengetahui berapa banyak total kasus, sembuh, kematian, dan rasio per provinsi, perlu adanya preprosesing data dari dataset utama yang kemudian dibuat dataset baru. Pembuatan dataset atau dataframe baru ini bertujuan untuk mempermudah pengolahan data menjadi visualisasi, karena perlu melakukan grouping berdasarkan provinsi dan melakukan penjumlahan untuk kolom Daily_Case, Daily_Recovered, dan Daily_Death. Berikut ini adalah script untuk membuat dataset total kasus, sembuh, kematian, dan rasio per provinsi.

```
total_per_province = corona_df.groupby('Province')[['Daily_Case',
'Daily_Recovered', 'Daily_Death']].sum().reset_index().rename(columns={
    'Daily_Case': 'Total_Case',
    'Daily_Recovered': 'Total_Recovered',
    'Daily_Death': 'Total_Death',
})

# Membuat Kolom Rasio Kematian terhadap Kasus
total_per_province['Death_Case_Ratio'] = total_per_province['Total_Death'] / total_per_province['Total_Case'] * 100

# Membuat Kolom Rasio Sembuh terhadap Kasus
total_per_province['Recovered_Case_Ratio'] = total_per_province['Total_Recovered'] / total_per_province['Total_Case'] * 100
```

Penjelasan script atau kode:

- total_per_province adalah dataframe baru untuk menampung data total kasus, sembuh, dan kematian per provinsi.
- corona_df adalah dataframe dari dataset utama.
- Fungsi groupby untuk melakukan grouping berdasarkan Province
- Fungsi sum untuk menjumlahkan nilai pada kolom Daily_Case, Daily_Recovered, dan Daily_Death berdasarkan grouping sebelumnya.
- Fungsi reset_index untuk melakukan reset index pada dataframe.
- Fungsi rename untuk mengubah nama kolom-kolom yang ingin diubah seperti 'Daily_Case' diubah menjadi 'Total_Case', 'Daily_Recovered' diubah menjadi 'Total_Recovered', dan 'Daily_Death' diubah menjadi 'Total_Death'.

- `total_per_province['Death_Case_Ratio']` untuk membuat kolom baru bernama `Death_Case_Ratio` yang berisi nilai hasil dari nilai pada kolom `Total_Death / Total_Case * 100`.
- `total_per_province['Recovered_Case_Ratio']` untuk membuat kolom baru bernama `Recovered_Case_Ratio` yang berisi nilai hasil dari nilai pada kolom `Total_Recovered / Total_Case * 100`.

3.2.2 Data Pertumbuhan dan Penambahan Total Kasus Per Bulan

Untuk mengetahui bagaimana data pertumbuhan dan penambahan total kasus per bulan dari semua provinsi, perlu juga dilakukan preprosesing data untuk mempermudah visualisasi. Pertama-tama dibuat sebuah kolom baru yang bernama 'Month' pada dataset utama. Kemudian dilakukan grouping dan menjumlahkan kolom `Daily_Case`. Dibuat juga kolom baru yaitu 'Cumulative' untuk menghitung kasus kumulatif per bulan. Berikut ini adalah script untuk membuat data pertumbuhan total kasus per bulan.

```
# Membuat kolom baru 'Month'
corona_df['Month'] = corona_df['Date'].apply(lambda x: datetime.datetime.strptime(x, "%d/%m/%Y").strftime('%Y-%m'))
```

```
total_cases_per_month = corona_df.groupby('Month')['Daily_Case'].sum().reset_index(name='Total_Case')
total_cases_per_month['Cumulative'] = total_cases_per_month['Total_Case'].cumsum()
```

Penjelasan script atau kode:

- `corona_df['Month']` untuk membuat kolom baru bernama `Month` pada dataframe `corona_df`.
- Fungsi `apply` dan `strftime` untuk melakukan fungsi konversi tanggal dari "date/month/year" menjadi "year-month".
- `total_cases_per_month` adalah variabel untuk menyimpan dataframe baru.
- Fungsi `groupby` untuk melakukan grouping berdasarkan kolom `month`.
- Fungsi `sum` untuk melakukan penjumlahan pada kolom `Daily_Case`.
- Fungsi `reset_index` untuk mereset index dan membuat kolom baru bernama `Total_Case`.

- `total_cases_per_month['Cumulative']` untuk membuat kolom baru bernama Cumulative.
- Fungsi `cumsum` untuk melakukan penjumlahan kumulatif pada kolom Total_Case.

3.3 Dataset Akhir

Setelah melakukan preprosesing, maka hasil dari preprosesing adalah dataset-dataset kecil. Dataset yang dihasilkan merupakan dataframe dari library pandas. Berikut ini adalah hasil dataset-dataset setelah melakukan prosesing data.

3.3.1 Dataset Total Kasus, Sembuh, Kematian, dan Rasio Per Provinsi

Setelah melakukan preprosesing data untuk total kasus, sembuh, kematian, dan rasio per provinsi, maka hasilnya adalah sebuah dataset atau dataframe yang berisi nama provinsi serta jumlah total kasus, sembuh, kematian, rasio kematian, dan rasio sembuh. Berikut ini adalah hasil dari proses prosesing ini dalam bentuk tabel.

Tabel 3.1 Dataset Total Kasus per Provinsi

	Province	Total_Case	Total_Recovered	Total_Death	Death_Case_Ratio	Recovered_Case_Ratio
0	ACEH	9728	7971	387	3.978207	81.938734
1	BALI	37656	32655	923	2.451137	86.719248
2	BANTEN	34098	26149	605	1.774298	76.687782
3	BENGKULU	5232	4807	151	2.886086	91.876911
4	DAERAH ISTIMEWA YOGYAKARTA	31100	25207	746	2.398714	81.051447
5	DKI JAKARTA	366520	352767	6136	1.674124	96.247681
6	GORONTALO	5020	4687	141	2.808765	93.366534
7	JAMBI	5867	4709	82	1.397648	80.262485
8	JAWA BARAT	238604	203829	2850	1.194448	85.425642
9	JAWA TENGAH	164060	117526	6618	4.033890	71.635987
10	JAWA TIMUR	136092	119136	9142	6.717515	87.540781

11	KALIMANTAN BARAT	5484	4670	31	0.565281	85.156820
12	KALIMANTAN SELATAN	25172	21905	789	3.134435	87.021294
13	KALIMANTAN TENGAH	15736	13249	370	2.351296	84.195475
14	KALIMANTAN TIMUR	61134	54603	1436	2.348938	89.316910
15	KALIMANTAN UTARA	10658	8459	162	1.519985	79.367611
16	KEPULAUAN BANGKA BELITUNG	8366	7497	132	1.577815	89.612718
17	KEPULAUAN RIAU	8938	8535	224	2.506154	95.491161
18	LAMPUNG	13464	12038	712	5.288176	89.408794
19	MALUKU	7218	6662	110	1.523968	92.297035
20	MALUKU UTARA	4257	3540	117	2.748414	83.157153
21	NUSA TENGGARA BARAT	9168	7050	348	3.795812	76.897906
22	NUSA TENGGARA TIMUR	11472	8632	314	2.737099	75.244073
23	PAPUA	18897	10280	198	1.047785	54.400169
24	PAPUA BARAT	8016	7289	129	1.609281	90.930639
25	RIAU	33230	31110	805	2.422510	93.620223
26	SULAWESI BARAT	5352	4688	110	2.055306	87.593423
27	SULAWESI SELATAN	58913	55336	900	1.527676	93.928335
28	SULAWESI TENGAH	10771	9180	274	2.543868	85.228855
29	SULAWESI TENGGARA	10177	9501	203	1.994694	93.357571
30	SULAWESI UTARA	15247	12279	499	3.272775	80.533876
31	SUMATERA BARAT	30629	28770	673	2.197264	93.930589
32	SUMATERA SELATAN	16821	14588	788	4.684620	86.724927
33	SUMATERA UTARA	26399	22787	887	3.359976	86.317664

3.3.2 Data Pertumbuhan dan Penambahan Total Kasus Per Bulan

Setelah melakukan preprosesing data untuk data pertumbuhan total kasus per bulan, maka hasilnya adalah sebuah dataset atau dataframe yang berisi bulan - tahun serta jumlah total kasus dan jumlah kumulatif dari kasus tiap bulannya. Berikut ini adalah hasil dari proses prosesing ini dalam bentuk tabel.

Tabel 3.2 Data Pertumbuhan dan Penambahan Total Kasus Per Bulan

	Month	Total_Case	Cumulative
0	2020-03	943	943
1	2020-04	8478	9421
2	2020-05	16220	25641
3	2020-06	29621	55262
4	2020-07	52033	107295
5	2020-08	66448	173743
6	2020-09	112312	286055
7	2020-10	123327	409382
8	2020-11	128889	538271
9	2020-12	204542	742813
10	2021-01	334985	1077798
11	2021-02	256195	1333993
12	2021-03	115503	1449496

BAB IV

VISUALISASI DATA

4.1 Analisa Permasalahan

Terdapat permasalahan yang akan dipecahkan dalam pembuatan visualisasi ini. Berikut adalah permasalahan-permasalahan yang akan dipecahkan.

4.1.1 Mengetahui 10 Provinsi Dengan Kasus Terbanyak

Pemerintah pusat perlu mengetahui 10 provinsi dengan kasus terbanyak. Hal ini dilakukan untuk pengambilan keputusan terhadap provinsi mana yang perlu mendapatkan perhatian khusus, sehingga dapat dilakukan evaluasi dan pencegahan agar penambahan kasus di provinsi tersebut dapat berkurang.

4.1.2 Mengetahui 10 Provinsi Dengan Rasio Kematian Tertinggi

Pemerintah perlu mengetahui seberapa besar rasio kasus kematian yang terjadi di setiap provinsi. Jika rasio semakin besar artinya perlu diperiksa kenapa hal ini bisa terjadi.

4.1.3 Mengetahui 10 Provinsi Dengan Rasio Sembuh Tertinggi

Pemerintah perlu mengetahui seberapa besar rasio kesembuhan yang terjadi di setiap provinsi. Jika rasio semakin kecil artinya perlu dikaji apakah provinsi tersebut kekurangan bantuan medis atau bagaimana.

4.1.4 Mengetahui Penambahan Total Kasus Per Bulan

Pemerintah perlu mengetahui seberapa banyak total kasus yang dilaporkan dari semua provinsi setiap bulannya.

4.1.5 Mengetahui Pertumbuhan Total Kasus Per Bulan

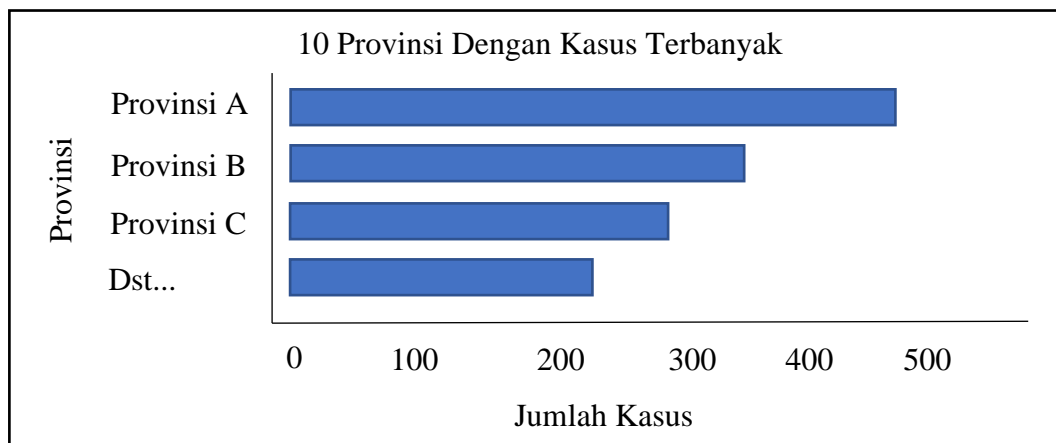
Pemerintah perlu mengetahui bagaimana pertumbuhan total kasus dari seluruh provinsi per bulan, atau dalam kata lain adalah total kasus di Indonesia.

4.2 Rancangan Visualisasi

Setelah mengetahui analisa permasalahan, maka dibuatlah rancangan visualisasi yang akan dibuat. Berikut ini adalah rancangan visualisasi berdasarkan setiap analisa permasalahan yang ada.

4.2.1 Rancangan 10 Provinsi Dengan Kasus Terbanyak

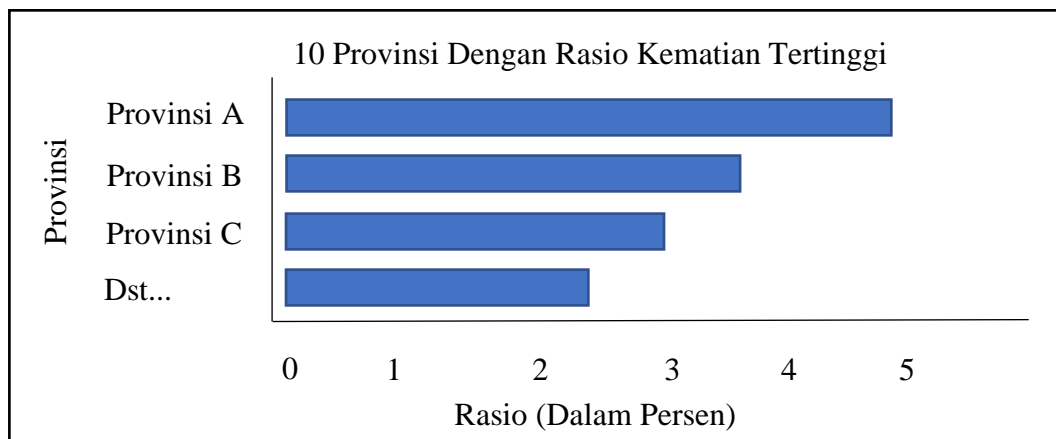
Rancangan visualisasi 10 Provinsi Dengan Kasus Terbanyak menggunakan bar chart horizontal. Berikut ini adalah rancangan terkait dengan 10 provinsi dengan kasus terbanyak.



Gambar 4.1 Rancangan 10 Provinsi Dengan Kasus Terbanyak

4.2.2 Rancangan 10 Provinsi Dengan Rasio Kematian Tertinggi

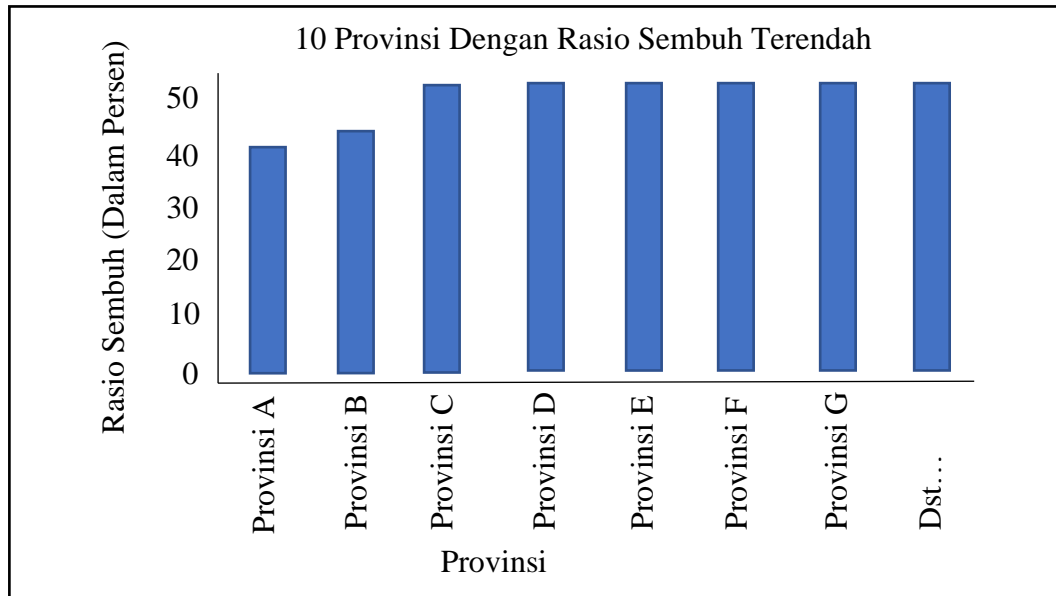
Rancangan visualisasi 10 Provinsi Dengan Rasio Kematian Tertinggi menggunakan bar chart horizontal. Berikut ini adalah rancangan terkait dengan 10 provinsi dengan rasio kematian tertinggi.



Gambar 4.2 Rancangan 10 Provinsi Dengan Rasio Kematian Tertinggi

4.2.3 Rancangan 10 Provinsi Dengan Rasio Sembuh Terendah

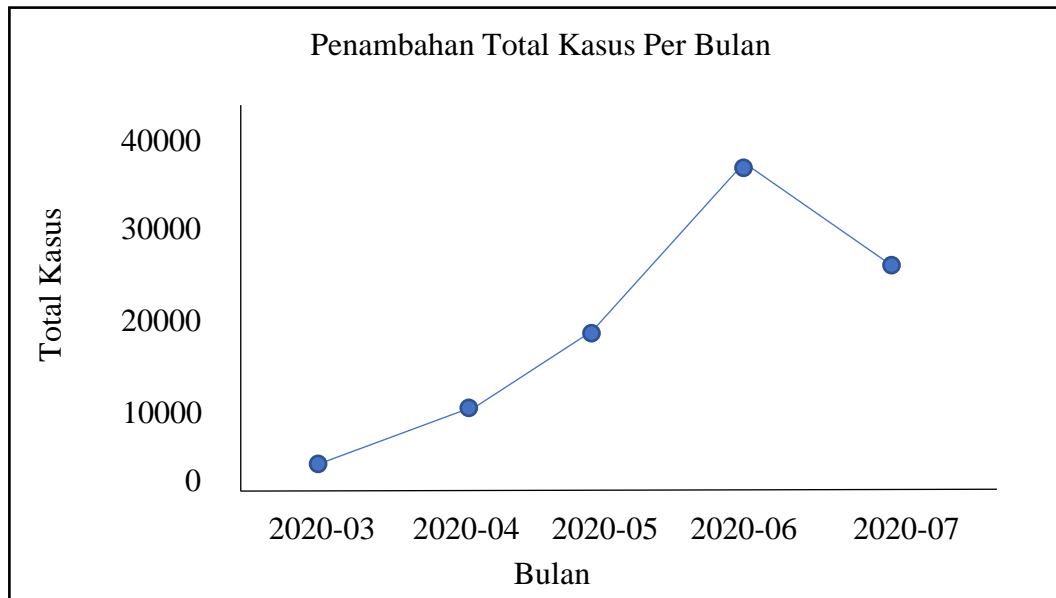
Rancangan visualisasi 10 Provinsi Dengan Rasio Sembuh Terendah menggunakan bar chart. Berikut ini adalah rancangan terkait dengan 10 provinsi dengan rasio sembuh terendah.



Gambar 4.3 Rancangan 10 Provinsi Dengan Rasio Sembuh Terendah

4.2.4 Rancangan Penambahan Total Kasus Per Bulan

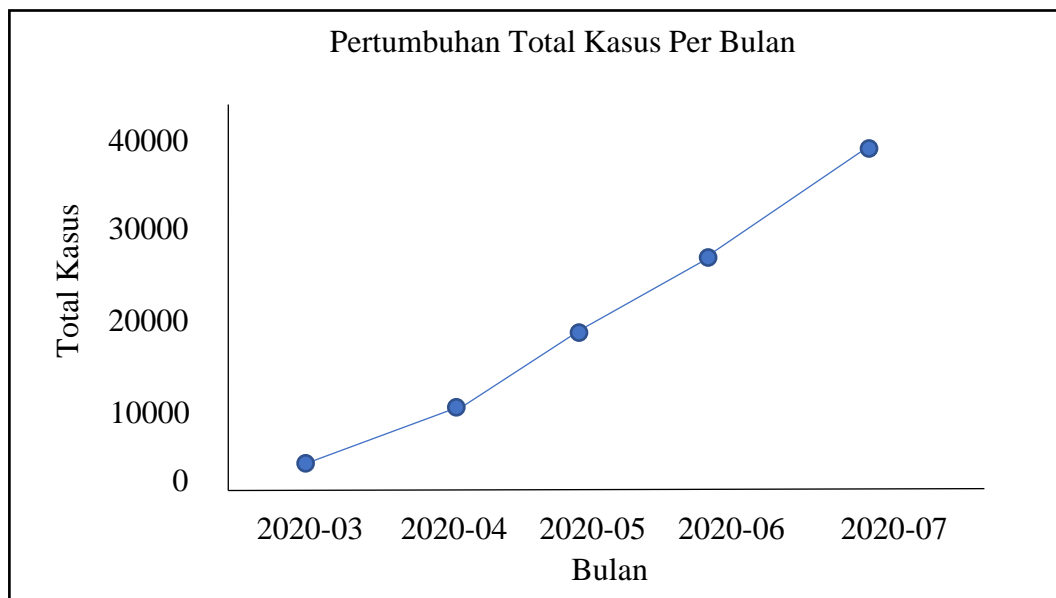
Rancangan visualisasi penambahan total kasus per bulan menggunakan line chart. Berikut ini adalah rancangan terkait dengan visualisasi penambahan total kasus per bulan.



Gambar 4.4 Rancangan Penambahan Total Kasus Per Bulan

4.2.5 Rancangan Pertumbuhan Total Kasus Per Bulan

Rancangan visualisasi pertumbuhan total kasus per bulan menggunakan line chart. Berikut ini adalah rancangan terkait dengan visualisasi pertumbuhan total kasus per bulan.



Gambar 4.5 Rancangan Pertumbuhan Total Kasus Per Bulan

4.3 Implementasi dan Hasil Visualisasi

Setelah melakukan pembuatan perancangan visualisasi, selanjutnya adalah implementasi dan hasil visualiasi. Implementasi dan hasil dilakukan dengan menggunakan bantuan dari library matplotlib. Berikut ini adalah hasil serta kode untuk melakukan visualisasi dari setiap rancangan.

4.3.1 Visualisasi 10 Provinsi Dengan Kasus Terbanyak

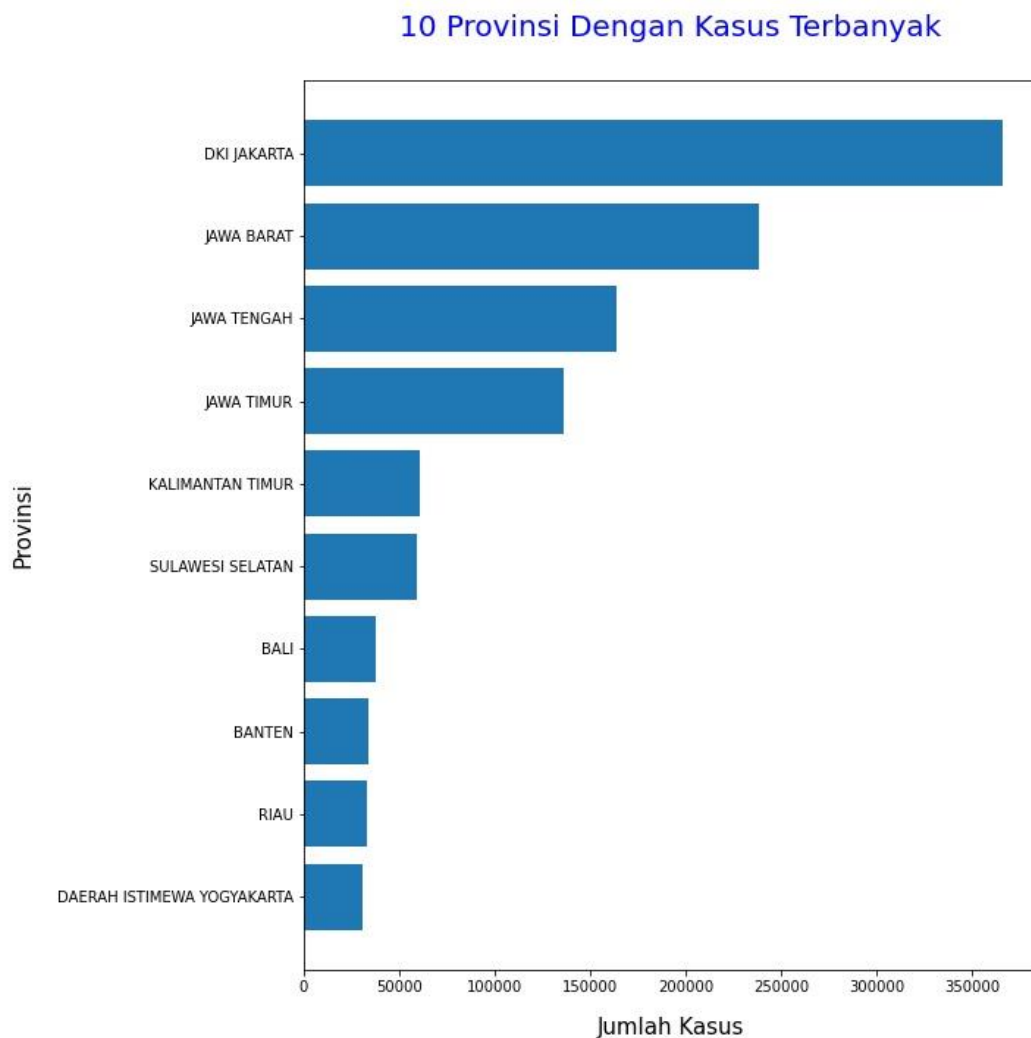
Berikut ini adalah kode untuk menampilkan visualisasi 10 provinsi dengan kasus terbanyak.

```
top_10_case = total_per_province.sort_values('Total_Case', ascending=True).tail(10)

plt.figure(figsize=(10,10))
plt.barh(top_10_case['Province'], top_10_case['Total_Case'])
plt.title('10 Provinsi Dengan Kasus Terbanyak', pad=30, fontsize=20, color='blue')
plt.xlabel('Jumlah Kasus', fontsize=15, labelpad=15)
plt.ylabel('Provinsi', fontsize=15, labelpad=15)
plt.tight_layout()
plt.show()
```

Penjelasan kode:

1. Variabel `top_10_case` untuk menyimpan 10 data terbawah dari `total_per_province` yang telah diurutkan berdasarkan kolom `Total_Case` secara ascending.
2. `plt.figure` untuk membuat figure baru dengan ukuran 10x10.
3. `plt.barh` untuk membuat bar chart horizontal, dimana x adalah provinsi dan y adalah total case.
4. `plt.title` untuk membuat judul dari visualisasi dengan ukuran font 20px, jarak 30, dan berwarna biru.
5. `plt.xlabel` untuk menambahkan label pada sumbu x dengan ukuran font 15 dan jarak 15.
6. `plt.ylabel` untuk menambahkan label pada sumbu y dengan ukuran font 15 dan jarak 15.
7. `plt.show` untuk menampilkan visualisasi.



Gambar 4.6 Visualisasi 10 Provinsi Dengan Kasus Terbanyak

Dari hasil visualisasi dapat dilihat bahwa provinsi DKI Jakarta memiliki jumlah kasus terbanyak urutan pertama, sehingga provinsi DKI Jakarta perlu mendapatkan perhatian khusus. Perhatian khusus tersebut dapat berupa menerbitkan peraturan untuk pembatasan sosial agar jumlah kasus dapat berkurang.

4.3.2 Visualisasi 10 Provinsi Dengan Rasio Kematian Tertinggi

Berikut ini adalah kode untuk menampilkan visualisasi 10 provinsi dengan rasio kematian tertinggi.

```

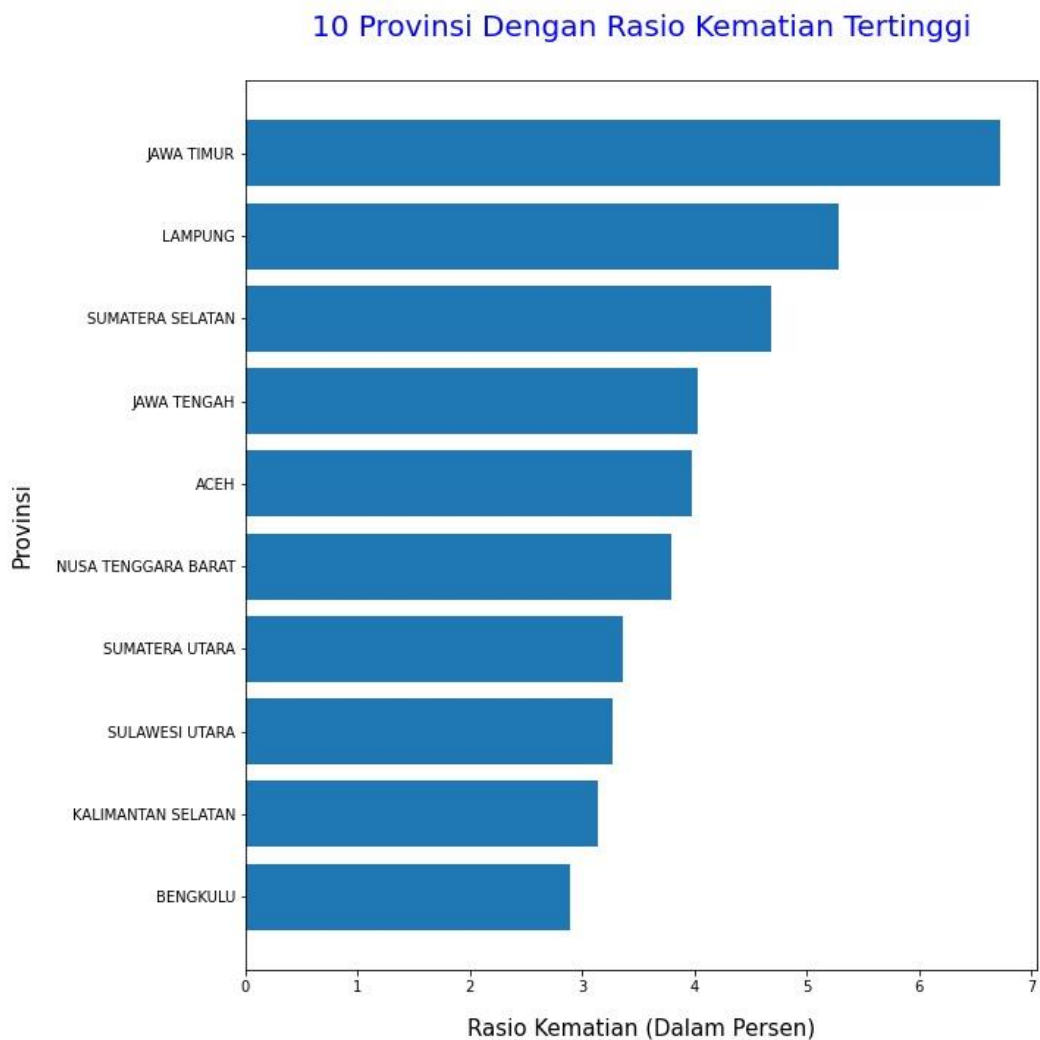
death_ratio = total_per_province.sort_values('Death_Case_Ratio',
ascending=True).tail(10)

plt.figure(figsize=(10,10))
plt.barh(death_ratio['Province'], death_ratio['Death_Case_Ratio']
)
plt.title('10 Provinsi Dengan Rasio Kematian Tertinggi', pad=30,
fontsize=20, color='blue')
plt.xlabel('Rasio Kematian (Dalam Persen)', fontsize=15, labelpad
=15)
plt.ylabel('Provinsi', fontsize=15, labelpad=15)
plt.tight_layout()
plt.show()

```

Penjelasan kode:

1. Variabel death_ratio untuk menyimpan data total_per_province yang telah diurutkan berdasarkan kolom Death_Case_Ratio secara ascending, kemudian diambil 10 data terbawah dengan menggunakan fungsi tail
2. plt.figure untuk membuat figure berukuran 10x10
3. plt.barh untuk membuat plot bar horizontal dengan x adalah nilai dari kolom Province dan y adalah nilai dari kolom Death_Case_Ratio
4. plt.title untuk membuat judul visualisasi dengan jarak 30, ukuran font 20, dan berwarna biru.
5. plt.xlabel untuk membuat label pada sumbu x dengan ukuran font 15 dan jarak 15
6. plt.ylabel untuk membuat label pada sumbu y dengan ukuran font 15 dan jarak 15
7. plt.tight_layout untuk memadatkan ukuran visualisasi
8. Plt.show untuk menampilkan visualisasi



Gambar 4.7 Visualisasi 10 Provinsi Dengan Rasio Kematian Tertinggi

Dari gambar di atas, dapat dilihat bahwa Provinsi Jawa Timur memiliki rasio kematian tertinggi. Dari visualisasi ini pemerintah dapat mengambil tindakan dengan terjun langsung ke Jawa Timur untuk mengetahui kenapa ini bisa terjadi.

4.3.3 Visualisasi 10 Provinsi Dengan Rasio Sembuh Terendah

Berikut ini adalah kode untuk menampilkan visualisasi 10 provinsi dengan rasio sembuh terendah

```

recovered_ratio = total_per_province.sort_values('Recovered_Case_Ratio', ascending=True).head(10)

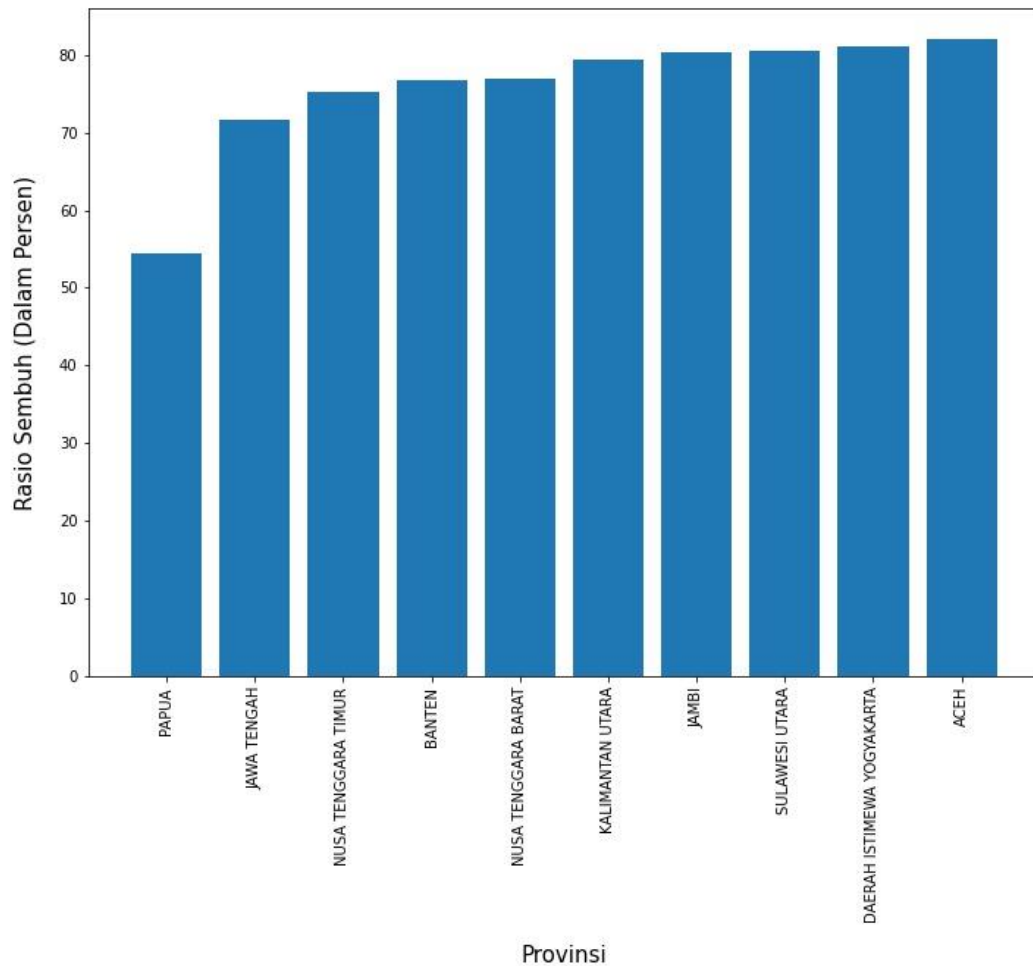
plt.figure(figsize=(10,10))
plt.bar(recovered_ratio['Province'], recovered_ratio['Recovered_Case_Ratio'])
plt.title('10 Provinsi Dengan Rasio Sembuh Terendah', pad=30, fontsize=20, color='blue')
plt.xlabel('Provinsi', fontsize=15, labelpad=15)
plt.xticks(rotation=90)
plt.ylabel('Rasio Sembuh (Dalam Persen)', fontsize=15, labelpad=15)
plt.tight_layout()
plt.show()

```

Penjelasan kode

1. Variabel `recovered_ratio` untuk menyimpan data `total_per_province` yang telah diurutkan berdasarkan kolom `Recovered_Case_Ratio` secara ascending, kemudian diambil 10 data terbawah dengan menggunakan fungsi `tail`
2. `plt.figure` untuk membuat figure berukuran 10x10
3. `plt.barh` untuk membuat plot bar horizontal dengan x adalah nilai dari kolom `Province` dan y adalah nilai dari kolom `Recovered_Case_Ratio`
4. `plt.title` untuk membuat judul visualisasi dengan jarak 30, ukuran font 20, dan berwarna biru.
5. `plt.xlabel` untuk membuat label pada sumbu x dengan ukuran font 15 dan jarak 15
6. `plt.xticks` untuk mengkonfigurasi values x yang dilakukan rotasi sebesar 90 derajat.
7. `plt.ylabel` untuk membuat label pada sumbu y dengan ukuran font 15 dan jarak 15
8. `plt.tight_layout` untuk memadatkan ukuran visualisasi
9. `plt.show` untuk menampilkan visualisasi

10 Provinsi Dengan Rasio Sembuh Terendah



Gambar 4.8 Visualisasi 10 Provinsi Dengan Rasio Sembuh Terendah

Dari gambar di atas, dapat dilihat bahwa Provinsi Papua memiliki rasio sembuh yang rendah. Pemerintah dapat mengambil tindakan untuk mengetahui kenapa hal tersebut bisa terjadi, apakah kekurangan tenaga medis, alat medis, dsb.

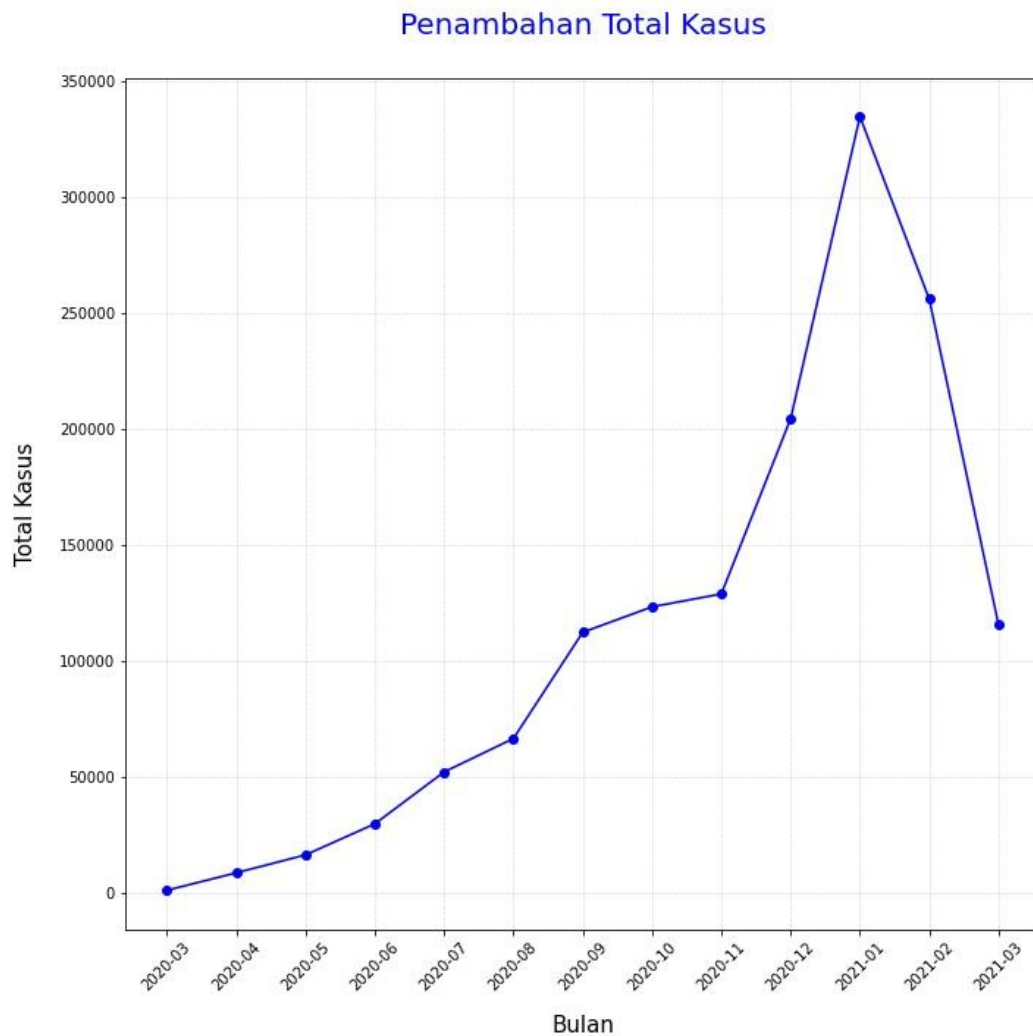
4.3.4 Visualisasi Penambahan Total Kasus Per Bulan

Berikut ini adalah kode untuk menampilkan visualisasi penambahan total kasus per bulan.

```
plt.figure(figsize=(10,10))
plt.plot(total_cases_per_month['Month'], total_cases_per_month['Total_Case'], 'bo-')
plt.title('Penambahan Total Kasus', pad=30, fontsize=20, color='blue')
plt.xlabel('Bulan', fontsize=15, labelpad=15)
plt.ylabel('Total Kasus', fontsize=15, labelpad=15)
plt.xticks(rotation=45)
plt.grid(color='darkgray', linestyle=':', linewidth=0.5)
plt.tight_layout()
plt.show()
```

Penjelasan kode:

1. plt.figure untuk membuat figure berukuran 10x10.
2. plt.plot untuk membuat plot line chart dengan x adalah nilai dari kolom Month dan y adalah nilai dari kolom Total_Case pada dataframe total_cases_per_month. Parameter 'bo-' untuk mengkonfigurasi line chart dimana b adalah blue yang artinya membuat line berwarna biru, o artinya membuat marker bulat, dan - artinya tampilkan garis.
3. plt.title untuk membuat judul visualisasi dengan jarak 30, ukuran font 20, dan berwarna biru.
4. plt.xlabel untuk membuat label pada sumbu x dengan ukuran font 15 dan jarak 15
5. plt.ylabel untuk membuat label pada sumbu y dengan ukuran font 15 dan jarak 15
6. plt.xticks untuk mengkonfigurasi values x yang dilakukan rotasi sebesar 45 derajat.
7. plt.grid untuk menampilkan grid dengan warna darkgray, linestyle :, dan ukuran garis sebesar 0.5
8. plt.tight_layout untuk memadatkan ukuran visualisasi
9. plt.show untuk menampilkan visualisasi



Gambar 4.9 Visualisasi Penambahan Total Kasus Per Bulan

Dari gambar di atas, dapat dilihat pada bulan Januari 2021 terjadi lonjakan kasus yang tinggi, kemudian pada bulan Februari 2021 terjadi penurunan penambahan kasus. Hal ini dapat disimpulkan bahwa kemungkinan pada saat tahun baru, banyak masyarakat yang berkumpul atau bepergian sehingga membuat lonjakan tinggi.

4.3.5 Visualisasi Pertumbuhan Total Kasus Per Bulan

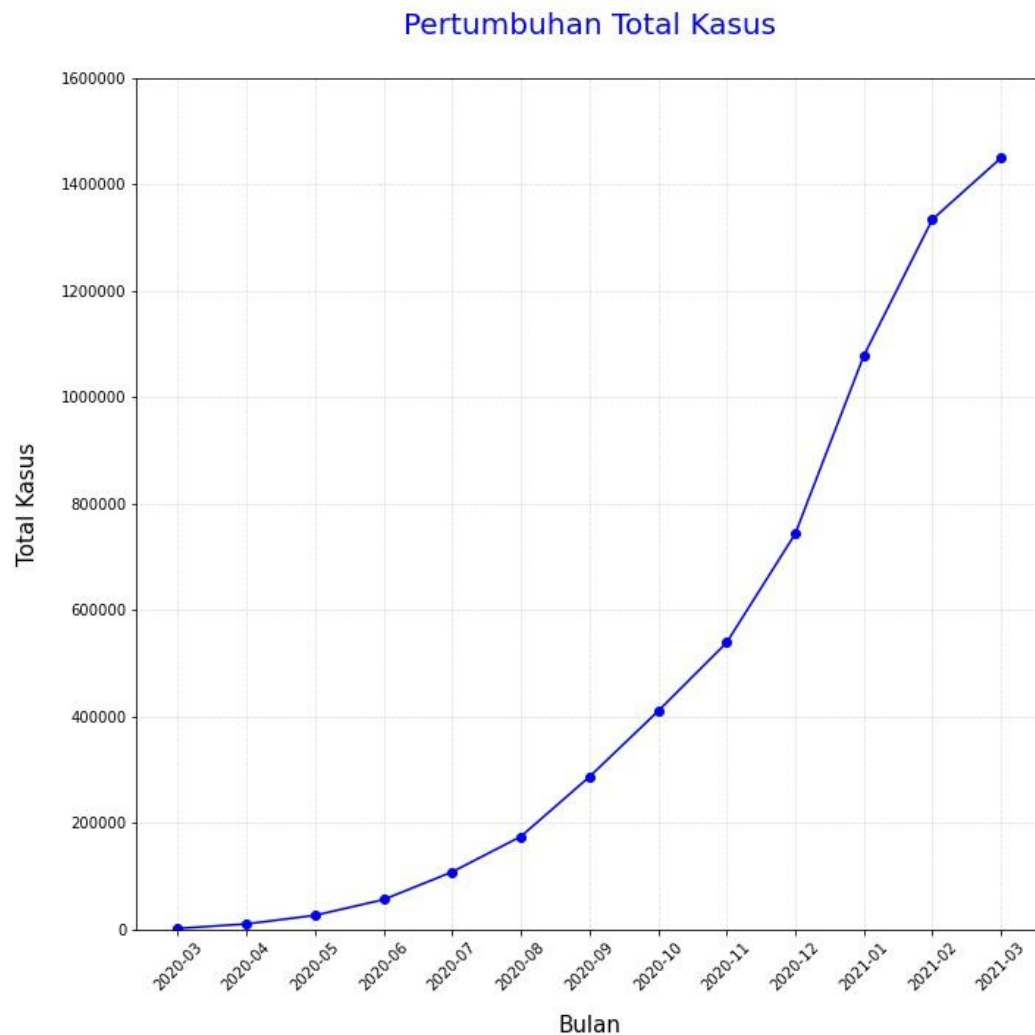
Berikut ini adalah kode untuk menampilkan visualisasi pertumbuhan total kasus per bulan.


```
plt.figure(figsize=(10,10))
plt.plot(total_cases_per_month['Month'], total_cases_per_month['Cumulative'], 'bo-')
plt.title('Pertumbuhan Total Kasus', pad=30, fontsize=20, color='blue')
plt.xlabel('Bulan', fontsize=15, labelpad=15)
plt.ylabel('Total Kasus', fontsize=15, labelpad=15)
plt.xticks(rotation=45)
plt.tight_layout()
plt.grid(color='darkgray', linestyle=':', linewidth=0.5)
labels, locations = plt.yticks()
plt.yticks(labels, labels.astype(int))
plt.ylim(0)
plt.tight_layout()
plt.show()
```

Penjelasan kode:

1. plt.figure untuk membuat figure berukuran 10x10.
2. plt.plot untuk membuat plot line chart dengan x adalah nilai dari kolom Month dan y adalah nilai dari kolom Cumulative pada dataframe total_cases_per_month. Parameter 'bo-' untuk mengkonfigurasi line chart dimana b adalah blue yang artinya membuat line berwarna biru, o artinya membuat marker bulat, dan - artinya tampilkan garis.
3. plt.title untuk membuat judul visualisasi dengan jarak 30, ukuran font 20, dan berwarna biru.
4. plt.xlabel untuk membuat label pada sumbu x dengan ukuran font 15 dan jarak 15
5. plt.ylabel untuk membuat label pada sumbu y dengan ukuran font 15 dan jarak 15
6. plt.xticks untuk mengkonfigurasi values x yang dilakukan rotasi sebesar 45 derajat.
7. plt.grid untuk menampilkan grid dengan warna darkgray, linestyle :, dan ukuran garis sebesar 0.5
8. labels, locations = plt.yticks() untuk mendapatkan label dan lokasi dari yticks
9. plt.yticks(labels, labels.astype(int)) untuk mengubah tipe data dari label pada yticks menjadi integer

10. plt.ylim(0) untuk mengatur nilai bawah dari y adalah 0
11. plt.tight_layout untuk memadatkan ukuran visualisasi
12. plt.show untuk menampilkan visualisasi



Gambar 4.10 Visualisasi Pertumbuhan Total Kasus Per Bulan

Dari gambar di atas, dapat dilihat bahwa pertumbuhan dari total kasus tiap bulan di Indonesia cenderung bertambah cukup tinggi.

BAB V

KESIMPULAN

Adapun kesimpulan yang dapat ditarik dari hasil visualisasi pada laporan ini adalah sebagai berikut ini

1. Didapatkan pengetahuan-pengetahuan seperti pada analisa permasalahan yang terjadi.
2. Dataset menjadi dapat dengan mudah dibaca dengan adanya visualisasi data.

Penulis juga memberikan saran terkait dengan laporan yang telah dibuat untuk pengembangan selanjutnya yaitu sebagai berikut.

1. Informasi atau pengetahuan lain dapat digali dari dataset yang ada.
2. Melakukan visualisasi terkait dengan informasi atau pengetahuan baru yang telah digali.
3. Menggunakan dataset yang lebih update.

DAFTAR PUSTAKA

- [1] T. Aziz, “COVID-19 Indonesia Dataset (Case and Vaccination) | Kaggle.” <https://www.kaggle.com/riqulaziz/case-vaccination-covid19-indonesia-dataset> (diakses Mei 23, 2021).
- [2] D. Saepuloh, “Visualisasi Data Covid 19 Provinsi DKI Menggunakan Tableau,” *J. Ris. Jakarta*, vol. 13, no. 2, hal. 55–64, Des 2020, doi: 10.37439/jurnaldrd.v13i2.37.
- [3] S. K. Card, J. D. Mackinlay, dan B. Shneiderman, Ed., *Readings in Information Visualization: Using Vision to Think*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999.
- [4] “Arti kata visualisasi - Kamus Besar Bahasa Indonesia (KBBI) Online.” <https://kbbi.web.id/visualisasi> (diakses Mei 23, 2021).
- [5] Kementerian Kesehatan Republik Indonesia, “Kementerian Kesehatan Republik Indonesia.” <https://www.kemkes.go.id/article/view/20031600011/pertanyaan-dan-jawaban-terkait-covid-19.html> (diakses Mei 23, 2021).
- [6] L. Perkovic, *Introduction to Computing Using Python: An Application Development Focus: An Application Development Focus*. Wiley, 2011.

LAMPIRAN

Link Video Presentasi

Link Google Colab

SUMBER DATA

Link Sumber Data Kaggle