

BAB 4

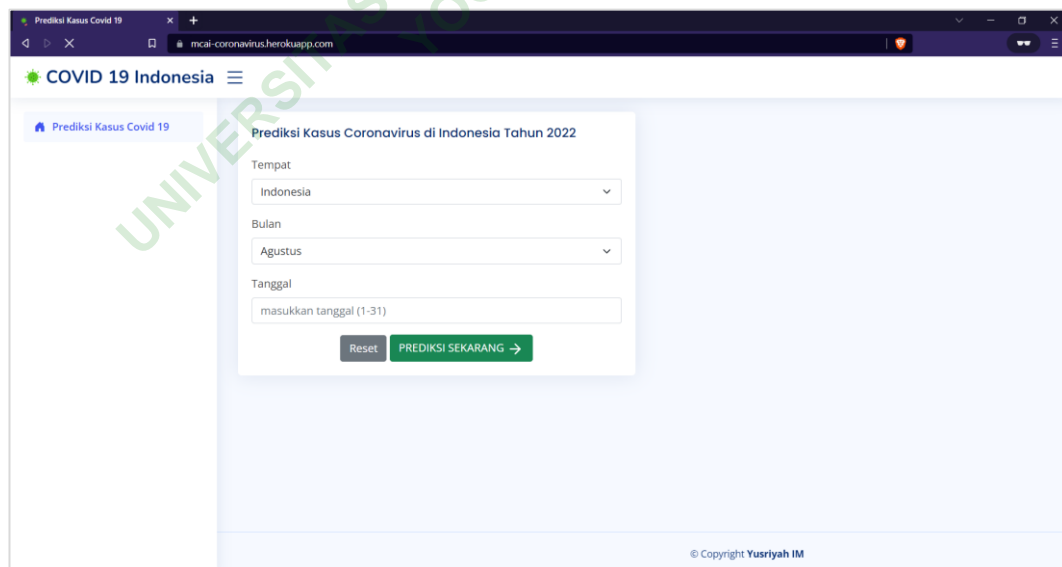
HASIL PENELITIAN

4.1 RINGKASAN HASIL PENELITIAN

Hasil penelitian berisi uraian rinci tentang hasil yang didapatkan dari penelitian analisis prediksi kasus COVID-19 di Indonesia dengan algoritma Linear Regression. Penelitian ini membutuhkan dataset yang berisi kasus terkonfirmasi positif, kasus kematian, dan kesembuhan penderita di Indonesia, selanjutnya dilakukan pengolahan data berupa *preprocessing* untuk melakukan pembersihan dan transformasi data. Data tersebut nantinya digunakan untuk melatih model yang akan digunakan untuk memprediksi kasus COVID-19. Kemudian membuat sistem sederhana yang memuat model dan masukan untuk memprediksi kasus positif COVID-19 selama 3 bulan ke depan.

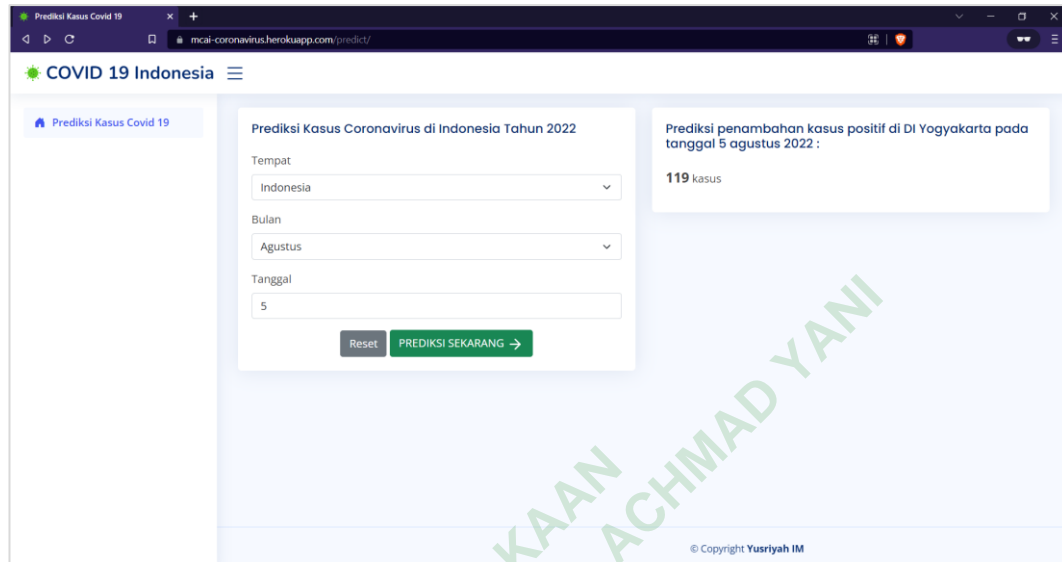
4.2 IMPLEMENTASI DISAIN INTERFACE

Tampilan UI halaman utama sistem prediksi kasus COVID-19 ditunjukkan pada gambar 4.2-1 pada laman <https://mcai-coronavirus.herokuapp.com/>.



Gambar 4.2-1 Tampilan halaman utama awal

Tampilan UI pada halaman prediksi dalam sistem prediksi kasus COVID-19 ditunjukkan pada gambar 4.2-2 pada laman <https://mcai-coronavirus.herokuapp.com/predict/>.



Gambar 4.2-2 Tampilan halaman prediksi

4.3 FITUR-FITUR SISTEM

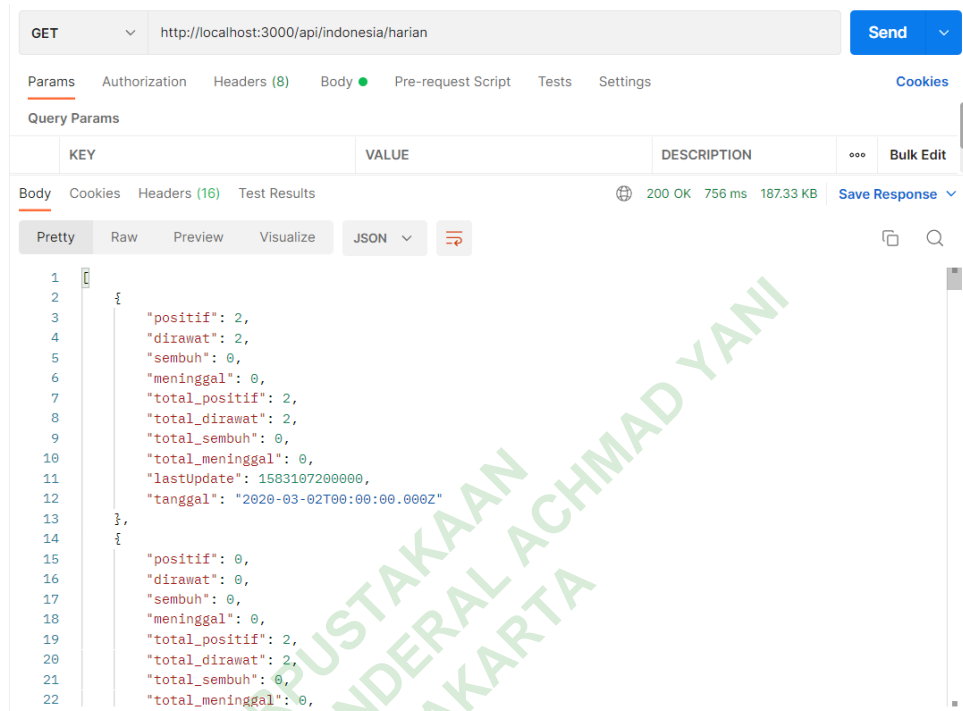
Terdapat 2 route, yaitu halaman home, dan halaman /predict. Halaman home / tampilan awal sistem ditunjukkan pada gambar 4.2-1 diatas. Dan pada halaman /predict ditunjukkan pada gambar 4.2-2 diatas. Hasil dari predict diperoleh dengan memasukkan input tempat, bulan, dan tanggal.

4.4 PEMBAHASAN

4.4.1 Pengambilan Data

Sesuai pada jalan penelitian sebelumnya, penelitian ini menggunakan dataset di website <https://covid19.go.id/>. Untuk mengambil data dari website tersebut secara *real-time* dengan menggunakan *Application Programming Interface* (API). Secara sederhana API dapat dipahami sebagai cara komputer untuk berkomunikasi dengan komputer lainnya sehingga dapat terjadi transaksi pertukaran data.

Karena data yang ada di portal tersebut banyak dan beberapa tidak terpakai, penelitian ini menggunakan modifikasi API agar data mudah dibaca. Gambar 4.4-1 menunjukkan hasil request data API.



Gambar 4.4-1 Respon API

Disini mengambil sample ekstrak dari Respon API dengan menggunakan fungsi `len()` dan method `keys()`.

```
print('Jumlah hari Indonesia terdampak kasus COVID-19 : %d'
      %len(covid_raw_ina), 'hari')
cov_id_update = covid_raw_ina[-1]
print(cov_id_update)
```

Output yang dihasilkan:

```
Jumlah hari Indonesia terdampak kasus COVID-19 : 884 hari
{'positif': 5827, 'dirawat': 1239, 'sembuh': 4564, 'meninggal':
24, 'total_positif': 6216621, 'total_dirawat': 49048,
'total_sembuh': 6010545, 'total_meninggal': 157028, 'lastUpdate':
1659398400000, 'tanggal': '2022-08-02T00:00:00.000Z'}
```

4.4.2 Exploratory Data Analysis (EDA)

Exploratory Data Analysis untuk menemukan pola, anomali, menguji hipotesis, dan memeriksa asumsi dengan bantuan statistik dan grafik.

Gambar 4.4-2 menunjukkan analisa penambahan kasus COVID-19 di Indonesia.

```

print('Tanggal pembaharuan data penambahan kasus : ', str(cov_id_raw[-1]['tanggal']))
print('Jumlah penambahan kasus sembuh :', cov_id_raw[-1]['sembuh'])
print('Jumlah penambahan kasus meninggal :', cov_id_raw[-1]['meninggal'])
print('Jumlah total kasus positif hingga saat ini :', cov_id_raw[-1]['total_positif'])
print('Jumlah total kasus meninggal hingga saat ini :', cov_id_raw[-1]['total_meninggal'])

```

[9] ✓ 0.1s

```

... Tanggal pembaharuan data penambahan kasus : 2022-08-02T00:00:00.000Z
Jumlah penambahan kasus sembuh : 4564
Jumlah penambahan kasus meninggal : 24
Jumlah total kasus positif hingga saat ini : 6216621
Jumlah total kasus meninggal hingga saat ini : 157028

```

Gambar 4.4-2 Analisa penambahan kasus

Informasi lebih detail mengenai struktur DataFrame dengan menggunakan fungsi `info()`. Gambar 4.4-3 di bawah ini menunjukkan hasil informasi detail data.

```

import pandas as pd

cov_ina = pd.DataFrame(cov_id_raw)
print('Info cov_ina:\n', cov_ina.info())

```

[11] ✓ 0.7s

```

... <class 'pandas.core.frame.DataFrame'>
RangeIndex: 884 entries, 0 to 883
Data columns (total 10 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   positif         884 non-null   int64
1   dirawat         884 non-null   int64
2   sembuh         884 non-null   int64
3   meninggal       884 non-null   int64
4   total_positif   884 non-null   int64
5   total_dirawat   884 non-null   int64
6   total_sembuh    884 non-null   int64
7   total_meninggal 884 non-null   int64
8   lastUpdate      884 non-null   int64
9   tanggal         884 non-null   object
dtypes: int64(9), object(1)
memory usage: 69.2+ KB
Info cov_ina:
None

```

Gambar 4.4-3 Informasi detail data

Gambar 4.4-4 mempresentasikan lima data teratas COVID-19 Indonesia.

```

print('\nLima data teratas COVID Indonesia :\n', cov_ina.head())

```

[12] ✓ 0.7s

...

Lima data teratas COVID Indonesia :

	positif	dirawat	sembuh	meninggal	total_positif	total_dirawat	\
0	2	2	0	0	2	2	
1	0	0	0	0	2	2	
2	0	0	0	0	2	2	
3	0	0	0	0	2	2	
4	2	2	0	0	4	4	

	total_sembuh	total_meninggal	lastUpdate	tanggal
0	0	0	1583107200000	2020-03-02T00:00:00.000Z
1	0	0	1583193600000	2020-03-03T00:00:00.000Z
2	0	0	1583280000000	2020-03-04T00:00:00.000Z
3	0	0	1583366400000	2020-03-05T00:00:00.000Z
4	0	0	1583452800000	2020-03-06T00:00:00.000Z

Gambar 4.4-4 Lima data teratas COVID-19 Indonesia

Gambar 4.4-5 mempresentasikan lima data terbaru COVID-19 Indonesia.

```

print('\nLima data terbaru COVID Indonesia :\n', cov_ina.tail())

```

[13] ✓ 0.1s

...

Lima data terbaru COVID Indonesia :

	positif	dirawat	sembuh	meninggal	total_positif	total_dirawat	\
879	5831	1333	4485	13	6197495	47988	
880	5398	1117	4268	13	6202893	49105	
881	4205	-402	4597	10	6207098	48703	
882	3696	-894	4579	11	6210794	47809	
883	5827	1239	4564	24	6216621	49048	

	total_sembuh	total_meninggal	lastUpdate	tanggal
879	5992537	156970	1659052800000	2022-07-29T00:00:00.000Z
880	5996805	156983	1659139200000	2022-07-30T00:00:00.000Z
881	6001402	156993	1659225600000	2022-07-31T00:00:00.000Z
882	6005981	157004	1659312000000	2022-08-01T00:00:00.000Z
883	6010545	157028	1659398400000	2022-08-02T00:00:00.000Z

Gambar 4.4-5 Lima data terbaru COVID-19 Indonesia

Setelah mengekstrak dan mengamati variabel `cov_ina`, berikut adalah beberapa anomali yang ditemukan pada data. Di antaranya adalah data untuk kolom tanggal dan representasi kolom yang tidak konsisten seperti kolom `lastUpdate`. Beberapa langkah diperlukan untuk membuat data lebih mudah diproses dan dianalisis. Tahapan pada normalisasi data `cov_ina` ini yaitu:

1. Menghapus kolom tanggal dan kolom nilai total.
2. Mengganti penulisan kolom menjadi huruf kecil

3. Merubah nama kolom kasus dengan kasus_baru
4. Memperbaiki tipe data kolom tanggal menggunakan pd.to_datetime

Gambar 4.4-6 menunjukkan data setelah normalisasi.

```

covid_ina_clean = (cov_ina.drop(
    columns=[
        item for item in cov_ina.columns
        if item.startswith('total') or item.startswith('tanggal')]).rename(columns={'lastUpdate': 'tanggal'}))
covid_ina_clean['tanggal'] = pd.to_datetime(covid_ina_clean['tanggal']*1e6, unit='ns')
print('Lima data teratas:\n', covid_ina_clean.head())

```

[28]

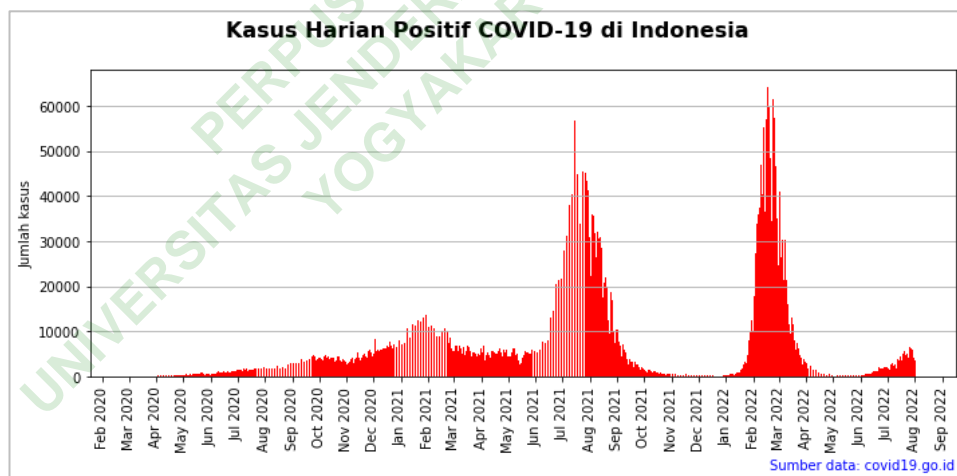
... Lima data teratas:

	positif	dirawat	sembuh	meninggal	tanggal
0	2	2	0	0	2020-03-02
1	0	0	0	0	2020-03-03
2	0	0	0	0	2020-03-04
3	0	0	0	0	2020-03-05
4	2	2	0	0	2020-03-06

Gambar 4.4-6 Data yang sudah dinormalisasi

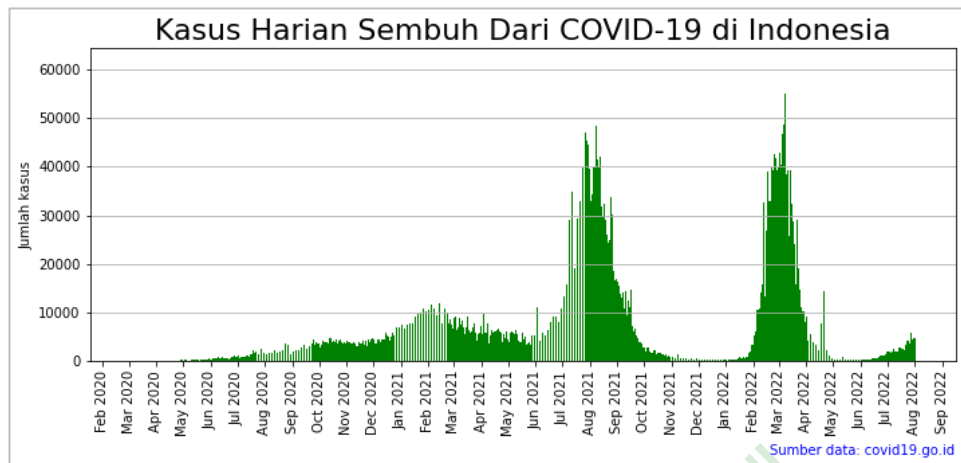
Visualisasi

Diagram batang kondisi kasus harian positif COVID-19 di Indonesia ditunjukkan pada gambar 4.4-7.

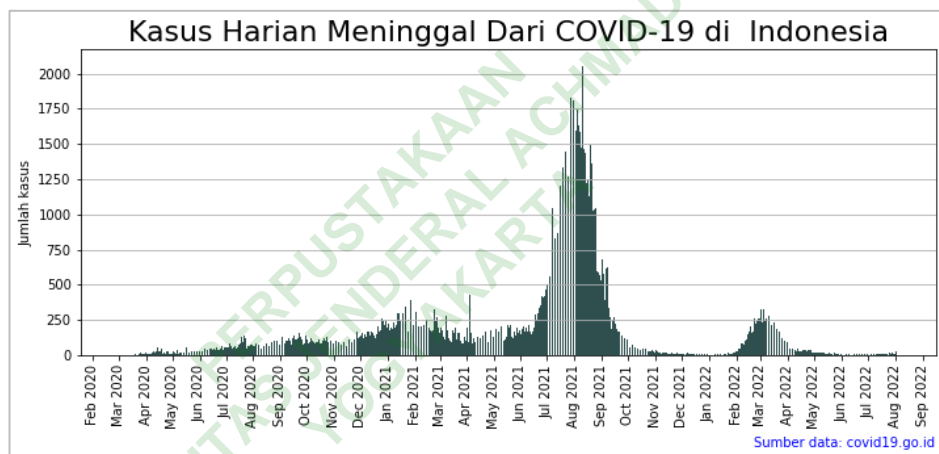


Gambar 4.4-7 Kasus Harian Positif COVID-19 di Indonesia

Diagram batang kondisi kasus harian sembuh COVID-19 di Indonesia ditunjukkan pada gambar 4.4-8 dan kasus harian meninggal COVID-19 di Indonesia ditunjukkan pada gambar 4.4-9.



Gambar 4.4-8 Kasus Harian Sembuh COVID-19 di Indonesia



Gambar 4.4-9 Kasus Harian Meninggal COVID-19 di Indonesia

Setelah melihat grafik perkembangan kasus, diketahui bahwa jumlah kasus harian berfluktuasi dari hari ke hari. Atas dasar itu, kemudian memantau perkembangan kasus tersebut dalam rentang waktu selama beberapa minggu ke depan (pekanan) pada gambar 4.4-10:

```

covid_ina_pekanan = (covid_ina_clean.set_index('tanggal')
                    ['positif'].resample('W').sum()
                    .reset_index().rename(
                        columns={'positif': 'jumlah'})
                    )
covid_ina_pekanan['tahun'] = covid_ina_pekanan['tanggal'].apply(
    Lambda x: x.year)
covid_ina_pekanan['pekan_ke'] = covid_ina_pekanan['tanggal'].apply(
    Lambda x: x.weekofyear)
covid_ina_pekanan = covid_ina_pekanan[['tahun', 'pekan_ke', 'jumlah']]

print('Info cov_ina_pekanan:')
print(covid_ina_pekanan.info())
print('\nLima data teratas cov_ina_pekanan:\n', covid_ina_pekanan.head())

```

```

[25]
... Info cov_ina_pekanan:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 127 entries, 0 to 126
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype
---  ---
 0   tahun      127 non-null    int64
 1   pekan_ke   127 non-null    int64
 2   jumlah     127 non-null    int64
dtypes: int64(3)
memory usage: 3.1 KB
None

```

Gambar 4.4-10 Info lima data pekanan teratas COVID-19 Indonesia

Sepuluh data teratas perkembangan kasus per minggu ditunjukkan pada gambar 4.4-11:

```

import numpy as np

covid_ina_pekanan['jumlah_pekanlalu'] = covid_ina_pekanan['jumlah'].shift().replace(np.nan, 0).astype(np.int)
covid_ina_pekanan['lebih_baik'] = covid_ina_pekanan['jumlah'] < covid_ina_pekanan['jumlah_pekanlalu']

print('Sepuluh data teratas:\n', covid_ina_pekanan.head(10))

```

```

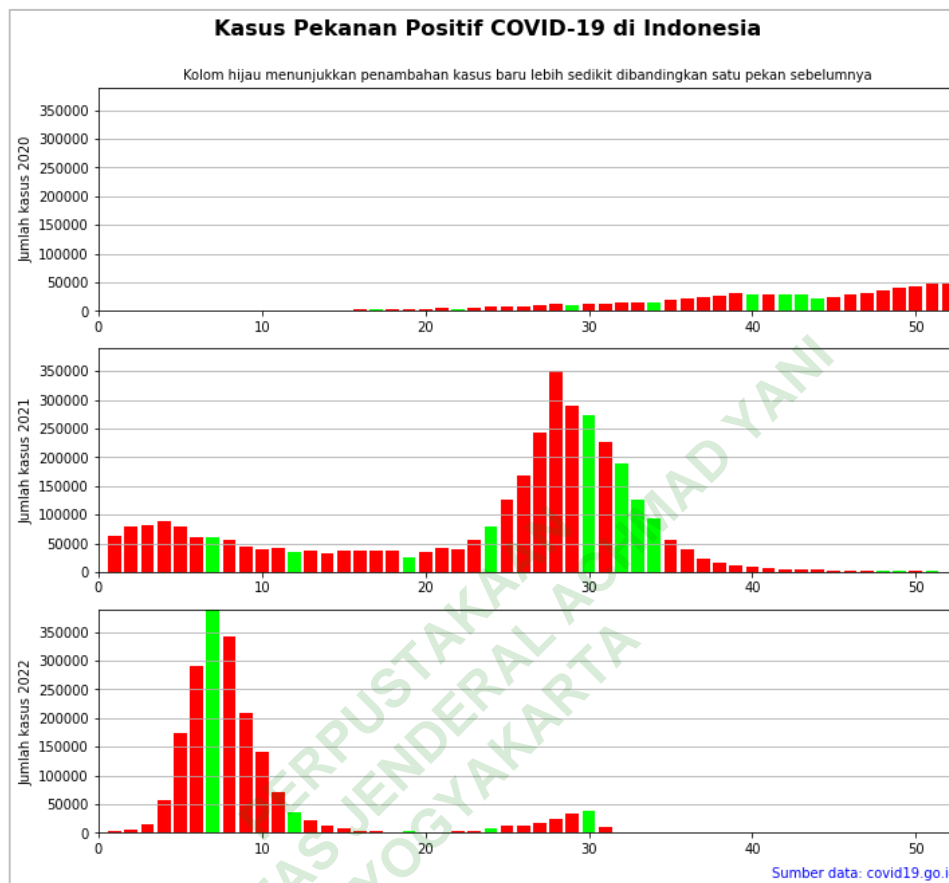
[27]
... Sepuluh data teratas:
   tahun  pekan_ke  jumlah  jumlah_pekanlalu  lebih_baik
0  2020     10      6              0             False
1  2020     11     111              6             False
2  2020     12     397             111            False
3  2020     13     771             397            False
4  2020     14     988             771            False
5  2020     15    1968             988            False
6  2020     16    2334            1968            False
7  2020     17    2307            2334             True
8  2020     18    2310            2307            False
9  2020     19    2840            2310            False

```

Gambar 4.4-11 Sepuluh data pekanan lebih baik teratas

Dengan menggunakan data yang perhitungan diatas, dapat dibuat diagram batang dari kasus yang ditambahkan berdasarkan minggu, seperti yang

ditunjukkan pada Gambar 4.4-12. Ini akan dilengkapi dengan informasi baru untuk menjawab perkembangan kasus selama jangka waktu mingguan.



Gambar 4.4-12 Grafik perkembangan kasus pekan COVID-19 di Indonesia

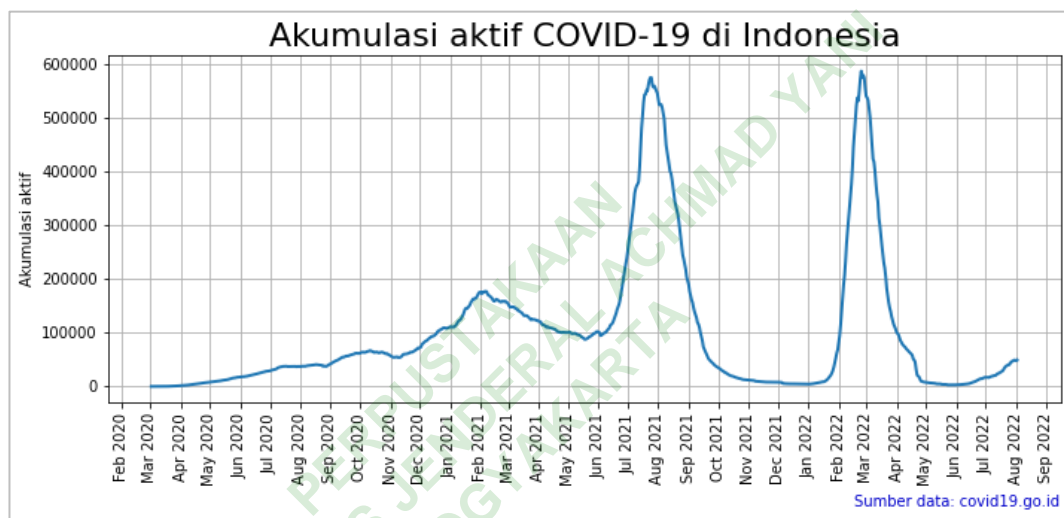
Mencermati grafik di atas, banyak orang yang telah sembuh, tetapi jumlah yang meninggal karena COVID-19 cukup signifikan, sementara kasus baru terus bertambah di Indonesia. Hal ini menimbulkan pertanyaan lain tentang berapa banyak kasus aktif yang masih berlangsung hari ini. Aktif mengenai pengobatan atau isolasi.

Jumlah kumulatif kasus aktif dapat dihitung dengan mengurangi total kasus positif dari total kesembuhan dan jumlah kematian. Gambar 4.4-13 menunjukkan hasil akumulasi aktif.

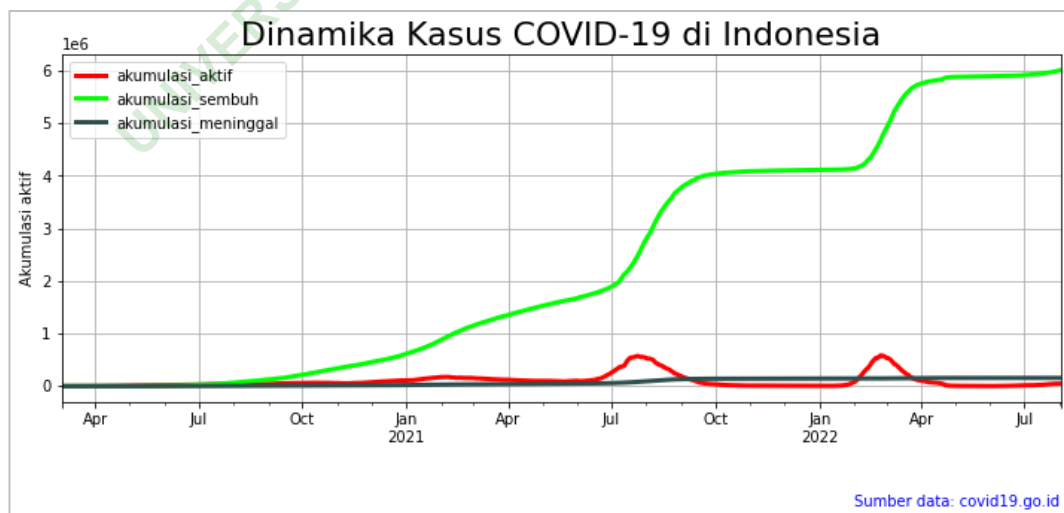
...	tanggal	akumulasi_aktif	akumulasi_sembuh	akumulasi_meninggal
	879 2022-07-29	47988	5992537	156970
	880 2022-07-30	49105	5996805	156983
	881 2022-07-31	48703	6001402	156993
	882 2022-08-01	47809	6005981	157004
	883 2022-08-02	49048	6010545	157028

Gambar 4.4-13 Akumulasi kasus aktif

Grafik akumulasi aktif COVID-19 hingga Agustus 2022 ditunjukkan pada gambar 4.4-14, dan grafik dinamika kasus COVID-19 ditunjukkan pada gambar 4.4-15.



Gambar 4.4-14 Grafik akumulasi kasus aktif



Gambar 4.4-15 Grafik dinamika kasus COVID-19 di Indonesia

4.4.3 Preprocessing Data

Setelah melakukan EDA dan menemukan beberapa *insight* baru mengenai perkembangan COVID-19, maka dilanjutkan dengan proses *Preprocessing Data* agar kasus bisa di prediksi.

Setelah mengunduh data kasus COVID-19 harian per provinsi dan nasional melalui sumber data covid19.go.id. Kemudian file tersebut kemudian di gabung (*merge*) menjadi satu file agar pengolahan data lebih mudah. Kode untuk menggabungkan *multiple file CSV*:

```
# 1. mendefinisikan path ke file csv
path = "/Users/yusri/Documents/TA Skipsi/Code/provinsi/"

# 2. membuat list dengan file untuk digabungkan berdasarkan
konvensi nama
file_list = [path + f for f in os.listdir(path) if
f.startswith('PROVINSI_')]

# 3. membuat list kosong untuk memasukkan file-file yang
dikonversi ke pandas DF
csv_list = []

# 4. membaca setiap file (sorted) di file_list, mengonversi
menjadi pandas DF dan menambahkannya ke csv_list
for file in sorted(file_list):
    csv_list.append(pd.read_csv(file).assign(File_Name
    = os.path.basename(file)))

# 5. menggabungkan single pandas DFs
csv_merged = pd.concat(csv_list, ignore_index=True)

# 6. DF tunggal disimpan ke path dalam format CSV, tanpa kolom
indeks
csv_merged.to_csv(path + 'COVID19_DAILY_FULL.csv', index=False)
```

Data yang sudah di *merge* (gabung) tersimpan dalam file 'COVID19_DAILY_FULL.csv'. kemudian melakukan *processing* manual dengan *tools excel* dengan mengambil data kasus terkonfirmasi positif dan disimpan dengan nama 'DAILY_CONFIRMED.xlsx'. Gambar 4.4-16 menunjukkan info data final setelah proses *preprocessing*.

```

df = pd.read_excel('DAILY_CONFIRMED.xlsx')
df.info()

```

[20] ✓ 0.6s

*** Output exceeds the [size limit](#). Open the full output data [in a text editor](#)

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 883 entries, 0 to 882
Data columns (total 35 columns):
 #   Column                                     Non-Null Count  Dtype
---  -
 0   tanggal                                   883 non-null    datetime64[ns]
 1   PROVINSI_ACEH                             883 non-null    int64
 2   PROVINSI_BALI                             883 non-null    int64
 3   PROVINSI_BANTEN                           883 non-null    int64
 4   PROVINSI_BENGKULU                         883 non-null    int64
 5   PROVINSI_DAERAH_ISTIMEWA_YOGYAKARTA      883 non-null    int64
 6   PROVINSI_DKI_JAKARTA                      883 non-null    int64
 7   PROVINSI_JAMBI                            883 non-null    int64
 8   PROVINSI_JAWA_BARAT                       883 non-null    int64
 9   PROVINSI_JAWA_TENGAH                     883 non-null    int64
10  PROVINSI_JAWA_TIMUR                       883 non-null    int64
11  PROVINSI_KALIMANTAN_BARAT                 883 non-null    int64
12  PROVINSI_KALIMANTAN_TIMUR                 883 non-null    int64
13  PROVINSI_KALIMANTAN_TENGAH               883 non-null    int64
14  PROVINSI_KALIMANTAN_SELATAN              883 non-null    int64
15  PROVINSI_KALIMANTAN_UTARA                883 non-null    int64
16  PROVINSI_KEPULAUAN_RIAU                  883 non-null    int64
17  PROVINSI_KEPULAUAN_BANGKA_BELITUNG      883 non-null    int64
18  PROVINSI_LAMPUNG                          883 non-null    int64
19  PROVINSI_MALUKU                          883 non-null    int64
...
33  PROVINSI_SUMATERA_UTARA                   883 non-null    int64
34  PROVINSI_GORONTALO                       883 non-null    int64
dtypes: datetime64[ns](1), int64(34)
memory usage: 241.6 KB

```

Gambar 4.4-16 Info data yang sudah di Cleaning

Setelah itu, menambah kolom 'Total_Kasus_Harian' untuk menjadi acuan data pada prediksi nasional, hasil keluaran ditunjukkan pada gambar 4.4-17. Data pada kolom tersebut diperoleh dari penjumlahan semua kolom per provinsi dengan menggunakan fungsi `sum()`.

```

# Menambahkan kolom Total_Kasus_Kumulatif
df.insert(loc=1, column='Total_Kasus_Harian', value=df.sum(axis = 1, skipna = True))
df.tail()

```

Python

... C:\Users\yusr1\AppData\Local\Temp\ipykernel_71896\3222731717.py:2: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.
df.insert(loc=1, column='Total_Kasus_Harian', value=df.sum(axis = 1, skipna = True))

	tanggal	Total_Kasus_Harian	PROVINSI_ACEH	PROVINSI_BALI	PROVINSI_BANTEN	PROVINSI KEPULAUAN_BANGKA_BELITUNG	PROVINSI_BENGKULU	PROVINSI_DAERAH_ISTIMEWA_YOGYAKARTA
878	2022-07-27	6430	7	172	757	9	0	121
879	2022-07-28	6354	0	222	745	6	0	72
880	2022-07-29	5834	6	140	717	5	0	89
881	2022-07-30	5400	0	161	626	1	0	85
882	2022-07-31	4205	0	111	543	0	2	49

5 rows x 36 columns

Gambar 4.4-17 Data kolom 'Total_Kasus_Harian'

Setelah itu, me-*transpose* data agar kolom tanggal berubah menjadi *header* dan menyimpan hasil *transpose* data ke dalam file csv menggunakan kode:

```
df = df.T.to_csv('DAILY_CONFIRMED_CLEAN.csv', header=False)
```

Data 5 teratas dari hasil akhir *cleaning* data tersebut ditunjukkan pada gambar 4.4-18, dan info data final ditunjukkan pada gambar 4.4-19.

```

data = pd.read_csv('DAILY_CONFIRMED_CLEAN.csv')
data.head()

```

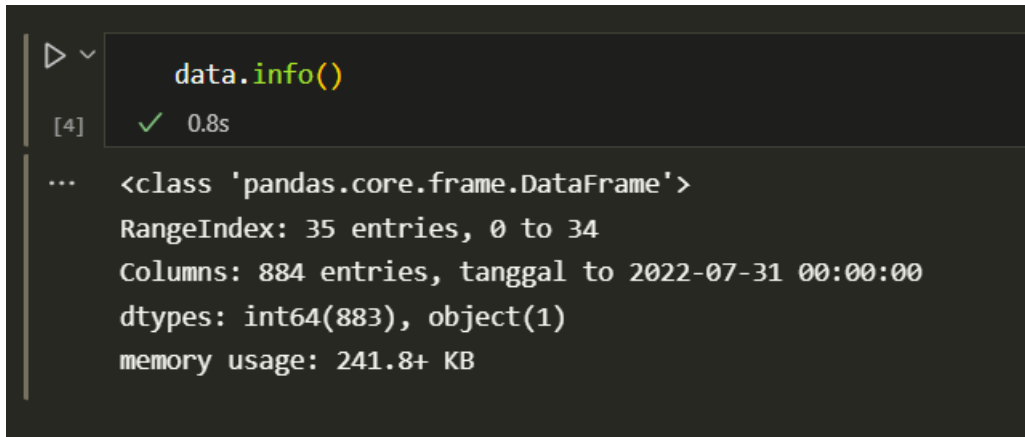
[25] ✓ 0.2s

	tanggal	2020-03-01 00:00:00	2020-03-02 00:00:00	2020-03-03 00:00:00	2020-03-04 00:00:00	2020-03-05 00:00:00	2020-03-06 00:00:00	2020-03-07 00:00:00
0	Total_Kasus_Harian	2	3	3	3	1	2	0
1	PROVINSI_ACEH	0	0	0	0	0	0	0
2	PROVINSI_BALI	0	0	0	0	0	0	0
3	PROVINSI_BANTEN	0	0	0	0	0	1	0
4	PROVINSI_BENGKULU	0	0	0	0	0	0	0

5 rows x 884 columns

+ Code

Gambar 4.4-18 Data Transpose



```

data.info()
[4] ✓ 0.8s
... <class 'pandas.core.frame.DataFrame'>
    RangeIndex: 35 entries, 0 to 34
    Columns: 884 entries, tanggal to 2022-07-31 00:00:00
    dtypes: int64(883), object(1)
    memory usage: 241.8+ KB

```

Gambar 4.4-19 Info Data Final

4.4.4 Splitting Data Train dan Testing

Kode dibawah ini adalah untuk menghitung prakiraan prediksi kasus dalam 30 hari ke depan, mengubah semua tipe bilangan *integers* menjadi *datetime*, membagi serta visualisasi data *training*.

```

import numpy as np
import datetime

dates = data.columns.values[1:]
days_since = np.array([i for i in range(len(dates))]).reshape(-1,1)
daysinfuture = 30
future_forecast = np.array([i for i in range(len(dates)+
daysinfuture)]).reshape(-1,1)
adjusted_dates = future_forecast[:-120]

start = '2020-03-01'
start_date = datetime.datetime.strptime(start, '%Y-%M-%D')
future_forecast_dates = []
for i in range(len(future_forecast)):
    future_forecast_dates.append((start_date + datetime.timedelta
(days = i)).strftime('%Y-%m-%d'))
cases = data.iloc[0,:].values[1:]
total_cases = np.array(cases).reshape(-1,1)

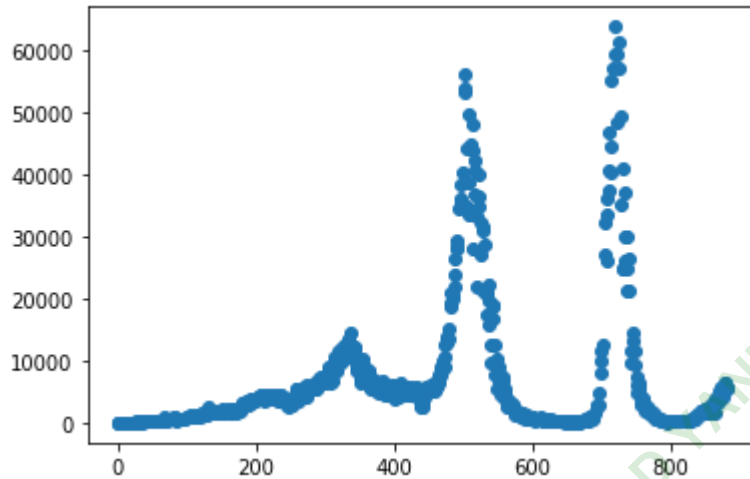
# Applying linear regression
from sklearn.model_selection import train_test_split

xtrain,xtest,ytrain,ytest = train_test_split(days_since,
total_cases,train_size=0.8,test_size = 0.2, random_state=100)

# Plotting graphs for the training set
plt.scatter(xtrain, ytrain)
plt.show()

```

Hasil *output* dari kode di atas ditunjukkan pada gambar 4.4-20.



Gambar 4.4-20 Grafik data training

4.4.5 Training Model

Kode dibawah ini untuk melatih model berdasarkan data *training*.

```
from sklearn.preprocessing import PolynomialFeatures

poly = PolynomialFeatures(degree = 3)
polyx = poly.fit_transform(days_since)

polyforecast = poly.transform(future_forecast)
polyxtest= poly.transform(xtest)
```

4.4.6 Prediksi Model

Dengan menggunakan model yang terlatih, dapat memprediksi variabel dari sample data test.

```
test1 = [[808]]
test2 = poly.transform(test1)
print(test2)
pred11 = lr.predict(test2)
print(pred11)
```

Hasil output prediksi :

```
[[1.00000000e+00 8.08000000e+02 6.52864000e+05 5.27514112e+08]]
[[5480.98688152]]
```

4.4.7 Evaluation Model

Setelah proses *training* dan testing prediksi selesai, selanjutnya melakukan evaluasi kinerja model menggunakan matriks yang berbeda yang diimpor dari metrik sklearn.

```
# Parameter Evaluasi R2 Score & RMSE
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

lr = LinearRegression(fit_intercept = False)
lr.fit(polyx, total_cases)

ypred = lr.predict(polyxtest)
ypredforecast = lr.predict(polyforecast)

rmse = np.sqrt(mean_squared_error(ytest,ypred))
r2 = r2_score(ytest,ypred)

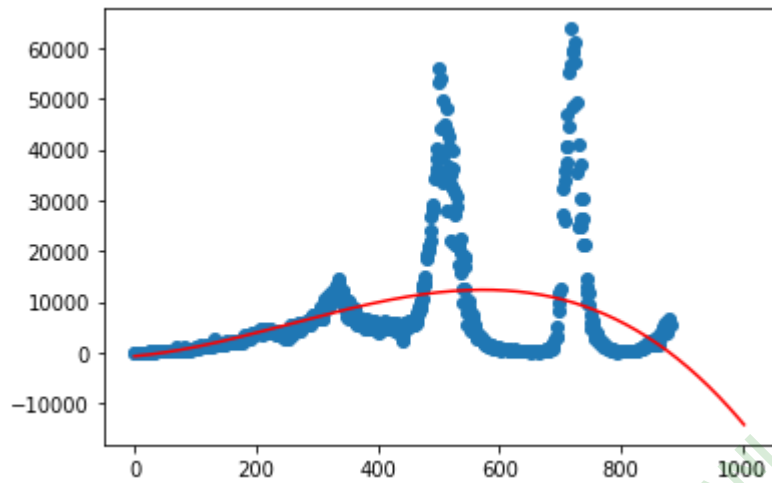
print(rmse, 'RMSE Done')
print(r2, 'R2 Done')
```

Output:

```
10681.75338518833 RMSE Done
0.10484674603408806 R2 Done
```

Dari output evaluasi model diatas menunjukkan bahwa nilai R^2 adalah 0.10484674603408806 tidak akan menjadi prediktor yang baik karena memiliki nilai R^2 yang lebih kecil. Grafik evaluasi model ditunjukan pada gambar 4.4-21:

```
plt.scatter(xtrain, ytrain)
plt.plot(ypredforecast, 'red')
plt.show()
```

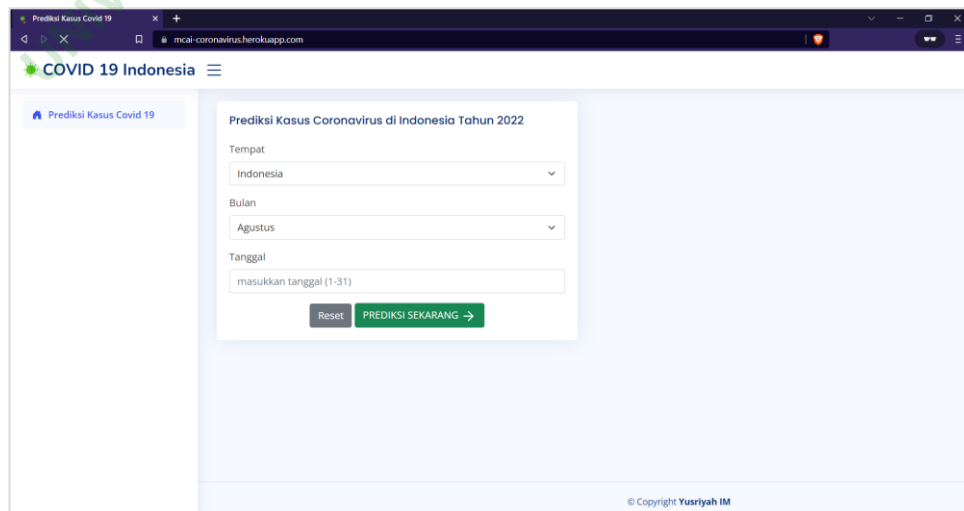



Gambar 4.4-21 Grafik evaluasi model

Hasil evaluasi berdasarkan bagan model yang ditunjukkan pada Gambar 4.4-21 menunjukkan bahwa evaluasi dari hari ke 0 hingga hari ke 300 memprediksi jumlah kasus positif akibat virus COVID-19 di Indonesia cukup baik, namun semenjak ada lonjakan yang cukup tinggi dari hari 350 hingga 900, *score* evaluasi menjadi tidak linier lagi.

4.4.8 Deployment Model

Deployment aplikasi dapat diakses dalam laman <https://mcai-coronavirus.herokuapp.com/>. Gambar 4.4-22 merupakan hasil tampilan awal website yang telah di *deploy*.



Gambar 4.4-22 Tampilan website setelah proses *deployment*

Hasil Prediksi VS Hasil Aktual

Hasil prediksi kasus nasional ditunjukkan pada gambar 4.4-23 dan hasil aktual data ditunjukkan pada gambar 4.4-24.

Gambar 4.4-23 Hasil prediksi kasus nasional



Gambar 4.4-24 Hasil aktual nasional

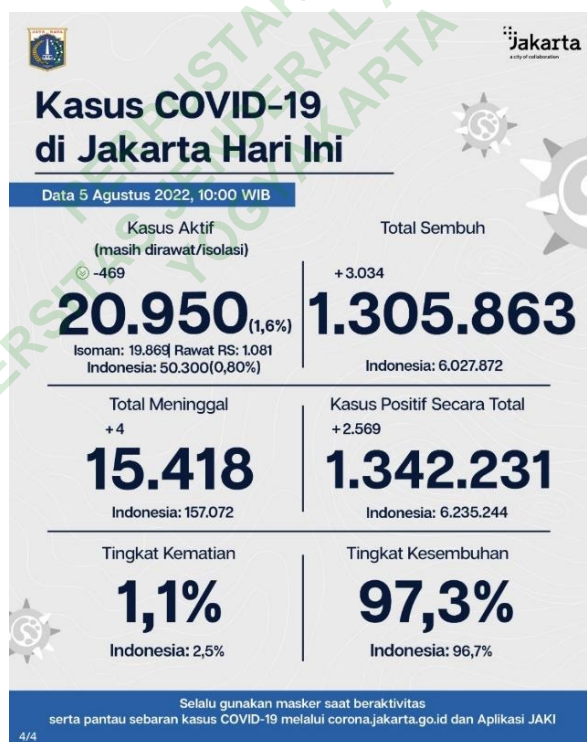
Sumber: <https://twitter.com/KemendesRI/status/1555541942091493376/photo/1>

Hasil *input* prediksi pada 5 Agustus lalu, jumlah kasus terkonfirmasi positif COVID-19 di Indonesia mencapai 4.884, dan hasil aktual yang diperoleh dari Kementerian Kesehatan RI mencapai 5.929 kasus. Sehingga dapat diketahui

bahwa selisih antara hasil prediksi COVID-19 menggunakan algoritma *linear regression* dan hasil aktual sebanyak 1045 kasus.

Hasil prediksi lain provinsi di Indonesia, salah satunya di Ibukota DKI Jakarta ditunjukkan pada gambar 4.4-25, dan hasil aktual kasus positif ditunjukkan pada gambar 4.4-26.

Gambar 4.4-25 Hasil prediksi kasus di DKI Jakarta



Gambar 4.4-26 Hasil aktual di DKI Jakarta.

Sumber: <https://twitter.com/DKIJakarta/status/1555550416867246080/photo/2>