

Analisis Pengaruh Fasilitas Kesehatan terhadap Kematian COVID-19 dengan Integrasi Data, Prediksi, dan MongoDB

**Oppir Hutapea¹, Laura Pasaribu², Yenita Sidabutar³, Kevin Hutajulu⁴, Syahril Sinaga^{*5},
Emalia Telaumbanua⁶**

^{1,2,3,4,5,6}Teknologi Rekayasa Perangkat Lunak, Fakultas Vokasi, Institut Teknologi Del, Sumatera Utara

e-mail: ¹oppir.hutapea@del.ac.id, ²if422009@students.del.ac.id, ³if422022@students.del.ac.id, ⁴if422034@students.del.ac.id, ^{*5}if422045@students.del.ac.id, ⁶if422061@students.del.ac.id

Abstrak

Pandemi COVID-19 menimbulkan tantangan besar bagi sistem kesehatan di Indonesia, dengan lebih dari 155.000 kematian hingga tahun 2022. Penelitian ini bertujuan untuk menganalisis pengaruh ketersediaan fasilitas kesehatan terhadap tingkat kematian akibat COVID-19 di Indonesia dengan pendekatan data science. Data yang digunakan mencakup informasi fasilitas kesehatan dari Dinas Kesehatan dan Kementerian Kesehatan, serta data epidemiologi COVID-19, yang diintegrasikan berdasarkan wilayah administratif. Proses analisis mencakup tahapan pra-pemrosesan data, penyimpanan menggunakan MongoDB, serta pemodelan prediktif menggunakan algoritma regresi linier. Hasil penelitian menunjukkan bahwa variabel seperti jumlah kasus baru, kepadatan penduduk, dan jumlah fasilitas kesehatan memiliki pengaruh signifikan terhadap tingkat kematian. Visualisasi data mendukung temuan ini, di mana wilayah dengan ketersediaan fasilitas yang lebih baik cenderung memiliki angka kematian yang lebih rendah. Penelitian ini menegaskan pentingnya penguatan sistem informasi kesehatan berbasis big data sebagai strategi untuk meningkatkan ketahanan sistem kesehatan nasional dalam menghadapi krisis di masa mendatang.

Kata kunci—COVID-19, fasilitas kesehatan, MongoDB, regresi linier, data science

1. PENDAHULUAN

Pandemi COVID-19 yang mulai merebak pada akhir tahun 2019 telah membawa dampak luas secara global, khususnya terhadap sistem pelayanan kesehatan [1]. Di Indonesia, hingga tahun 2022 tercatat lebih dari 155.000 kematian akibat virus ini, dengan tingkat kematian tertinggi terjadi di wilayah berpenduduk padat seperti DKI Jakarta dan Jawa Tengah. Tingginya angka kematian tidak hanya dipengaruhi oleh kondisi klinis pasien, tetapi juga oleh kesiapan dan kapasitas infrastruktur kesehatan di masing-masing daerah [2][3].

Sejumlah penelitian menunjukkan bahwa ketidakseimbangan dalam distribusi fasilitas kesehatan, seperti rumah sakit, ICU, dan jumlah tenaga medis, menjadi faktor penting yang menentukan efektivitas penanganan pandemi. Misalnya, studi yang dilakukan oleh Maulana dan Arcana [11] menekankan bahwa faktor klinis seperti gangguan pernapasan serta karakteristik demografis seperti jenis kelamin dapat meningkatkan risiko kematian. Namun, sebagian besar kajian di Indonesia masih bersifat deskriptif dan belum memanfaatkan pendekatan prediktif maupun integrasi data spasial secara optimal [4].

Seiring berkembangnya teknologi informasi, transformasi digital telah memungkinkan pengelolaan dan analisis data menjadi lebih cepat, kompleks, dan presisi. Menurut laporan IDC

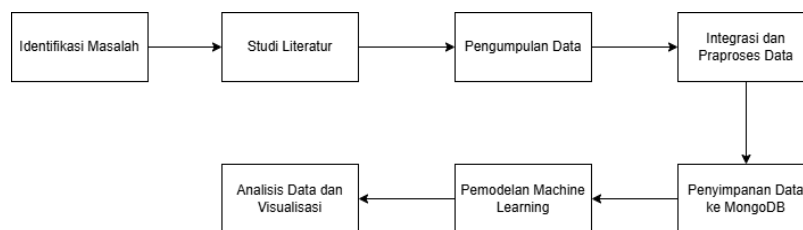
tahun 2023, volume data global diperkirakan mencapai 175 zettabyte pada 2025, mendorong kebutuhan akan sistem penyimpanan data yang fleksibel dan mampu menangani skala besar. Sistem manajemen basis data relasional seperti MySQL dan PostgreSQL selama ini menjadi standar untuk data terstruktur, namun memiliki keterbatasan dalam menangani data semi-terstruktur dan tidak terstruktur [5].

Untuk mengatasi hal tersebut, banyak institusi mulai mengadopsi basis data NoSQL seperti MongoDB yang menyediakan fleksibilitas skema, kemampuan penskalaan horizontal, serta performa tinggi dalam pengambilan data [6]. Teknologi ini telah diterapkan di berbagai bidang termasuk pendidikan, kesehatan, dan pengembangan sistem informasi berbasis REST API [7]. Di sisi lain, kemajuan teknologi kecerdasan buatan dan pembelajaran mesin juga mendorong penggunaan basis data vektor seperti FAISS dan Milvus untuk keperluan pencarian kemiripan data, mendukung aplikasi seperti NLP dan sistem rekomendasi. Kombinasi antara MongoDB, SQL, dan alat analitik seperti Python terbukti mampu menghasilkan wawasan yang relevan dari data besar [8].

Berangkat dari permasalahan tersebut, penelitian ini bertujuan untuk mengevaluasi sejauh mana ketersediaan fasilitas kesehatan mempengaruhi tingkat kematian akibat COVID-19 di Indonesia [9]. Pendekatan yang digunakan berbasis data science, melalui integrasi multi-sumber data, pemodelan prediktif menggunakan algoritma *machine learning*, dan pemanfaatan MongoDB sebagai penyimpanan utama. Hasil dari penelitian ini diharapkan dapat memberikan masukan strategis bagi pengambil kebijakan serta memperkuat sistem informasi kesehatan nasional agar lebih siap dalam menghadapi krisis serupa di masa mendatang [10].

2. METODE PENELITIAN

Penelitian ini dilaksanakan melalui beberapa tahapan utama yang disesuaikan dengan pendekatan data science, yaitu pengumpulan data, integrasi dan pra-pemrosesan data, implementasi sistem penyimpanan big data menggunakan MongoDB, dan pemodelan prediktif menggunakan machine learning. Tahapan-tahapan ini dijelaskan secara detail pada Gambar 1.



Gambar 1. Tahapan proses Penelitian

2.1. Identifikasi Masalah

Pada tahap ini, menentukan dan merumuskan masalah yang akan diteliti. Identifikasi masalah penting untuk memastikan fokus penelitian yang jelas.

2.2. Studi Literatur

Pada tahap ini, melakukan telaah pustaka untuk mengumpulkan informasi dan kajian sebelumnya yang relevan dengan masalah yang diidentifikasi. Ini membantu memahami konteks dan dasar teori yang ada

2.3. Pengumpulan Data

Data yang digunakan dalam penelitian ini terdiri dari dua sumber utama: (1) data Fasilitas Kesehatan dari Dinas Kesehatan dan Kementerian Kesehatan RI, yang mencakup nama fasilitas, jenis fasilitas (rumah sakit, puskesmas, klinik), lokasi, serta koordinat spasial;

dan (2) data COVID-19 dari sumber resmi pemerintah seperti Kementerian Kesehatan dan BPS yang mencakup jumlah kasus, jumlah kematian baru (New Deaths), jumlah kesembuhan, populasi, dan kepadatan penduduk.

2.4. Integrasi dan Pra-proses Data

Data yang telah dikumpul kemudian diproses yang mencakup EDA awal (statistik deskriptif dan identifikasi masalah pra-proses data), data selection, data preparation, data transformation, dimensionality reduction, data integration, dan discretization

2.5. Penyimpanan Data ke MongoDB

Penyimpanan data ke dalam MongoDB, yang memungkinkan penyimpanan dalam format dokumen (NoSQL). Setiap dokumen merepresentasikan satu fasilitas kesehatan dan data COVID terkait wilayahnya.

2.6. Pemodelan Machine Learning

Pada tahap ini, mengembangkan model analisis berdasarkan data yang telah dibagi dengan menggunakan linear Regression karena kesederhanaannya dan interpretabilitasnya dalam melihat kontribusi masing-masing fitur terhadap variabel target. Dataset dibagi menjadi data pelatihan (80%) dan data pengujian (20%). Model machine learning ini dibangun untuk menganalisis hubungan antara fasilitas kesehatan dan kematian akibat COVID-19 dan memprediksi jumlah kematian baru (New Deaths) berdasarkan fitur numerik seperti New Cases, Population, dan Population Density.

2.7. Analisis dan Visualisasi Hasil

Pada tahap ini, mencakup proses analisis data yang telah diprediksi dan visualisasi untuk mempermudah interpretasi hasil model. Beberapa visualisasi yang dilakukan antara lain Heatmap Korelasi Antar Variabel, Scatter Plot Prediksi vs Aktual, Bar Chart Komposisi Fasilitas Kesehatan, Bar Chart Tingkat Kematian dan Kesembuhan.

3. HASIL DAN PEMBAHASAN

3.1. Integrasi dan Pra-proses Data

Integrasi dan pra-pemrosesan data dilakukan pada dua dataset, yaitu Fasilitas Kesehatan BPJS 2019 (Gambar 2) yang terdiri dari 29.157 baris dan 11 kolom berisi informasi seperti tipe, kode, nama, dan alamat faskes, serta Data COVID-19 Indonesia. Gambar 3 menampilkan statistik deskriptif dari hasil integrasi kedua data tersebut.

NoLink	Provinsi	KotaKab	Link	TipeFaskes	No	KodeFaskes	NamaFaskes	LatLongFaskes	AlamatFaskes	TelpF
0	Nangroe Aceh Darussalam	Kode Faskes dan Alamat Rumah Sakit BPJS di Kota...	https://kovidia.life/id/health/bpsj/area/kota...	Rumah Sakit	1	0001R001	RSU Cut Nyak Dhiem	http://maps.google.co.id/?q=4.488058,97.947963	Jl. Tim Bahrum No. 1 Langkat	062101
1	Nangroe Aceh Darussalam	Kode Faskes dan Alamat Rumah Sakit BPJS di Kota...	https://kovidia.life/id/health/bpsj/area/kota...	Rumah Sakit	2	0001R004	RSU Cut Meutia Langkat	http://maps.google.co.id/?q=4.488058,97.947781	Jl. Ganata Kusun Baru Langkat	064007
2	Nangroe Aceh Darussalam	Kode Faskes dan Alamat Rumah Sakit BPJS di Kota...	https://kovidia.life/id/health/bpsj/area/kota...	Rumah Sakit	3	0105R001	RSUD Langkat	http://maps.google.co.id/?q=4.472208,97.975533	Jln. A. Yani No. 1 Langkat	062101
3	Nangroe Aceh Darussalam	Kode Faskes dan Alamat Rumah Sakit BPJS di Kota...	https://kovidia.life/id/health/bpsj/area/kota...	Rumah Sakit	4	0105R013	RSU Ummi	http://maps.google.co.id/?q=4.470376,97.991915	Jln. Prof. A. Mardjito Ibrahim	062101
4	Nangroe Aceh Darussalam	Kode Faskes dan Alamat Rumah Sakit BPJS di Kota...	https://kovidia.life/id/health/bpsj/area/kota...	Puskesmas	1	00010001	Langkat	http://maps.google.co.id/?q=4.476172,97.949988	Gempong Meurandeh Dayah	06412121

Gambar 2. Data Fasilitas Kesehatan BPJS 2019

```
datafaskes.describe(include='all')
```

	NoLink	Provinsi	KotaKab	Link	TipeFaskes	No	KodeFaskes	NamaFaskes	LatLongFaskes	AlamatFaskes	TelpFaskes
count	29157.000000	29157	29157	29157	29157	29157	29157	29157	29157	29157	29157
unique	NaN	34	512	512	7	150	28322	27421	22777	27204	25325
top	NaN	Jawa Tengah	Kode Faskes dan Alamat Rumah Sakit BPJS diY/N...	https://lovia.life/id/health/bpjs/area/kota-sa...	Puskesmas	1	-	-	-	-	V/VN
freq	NaN	3688	364	364	9913	2733	836	836	5408	843	1245
mean	227.003978	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
std	122.870434	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
min	0.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
25%	149.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
50%	211.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
75%	306.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
max	513.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Gambar 3. Statistik Deskriptif Data Fasilitas Kesehatan BPJS 2019

Pada Gambar 4 dapat dilihat bahwa tidak ada *missing value* yang terdeteksi pada kolom Data Faskes BPJS 2019, sehingga tidak perlu dilakukan penanganan *missing value*.

```
# Jumlah missing per kolom
missing_per_column = datafaskes.isnull().sum()
print("Missing per kolom:\n", missing_per_column)

Missing per kolom:
NoLink      0
Provinsi    0
KotaKab     0
Link        0
TipeFaskes  0
No          0
KodeFaskes  0
NamaFaskes  0
LatLongFaskes 0
AlamatFaskes 0
TelpFaskes  0
dtype: int64
```

Gambar 4. Missing Value Data Fasilitas Kesehatan BPJS 2019

Pada Gambar 5 ditunjukkan proses standarisasi format huruf pada kolom Provinsi, KotaKab, dan TipeFaskes ke dalam format *title case*. Selanjutnya, kolom TipeFaskes diubah menjadi tipe data kategori dan ditambahkan kolom baru TipeFaskes_code sebagai representasi numeriknya.

```
# Standarisasi format huruf
datafaskes['Provinsi'] = datafaskes['Provinsi'].str.title()
datafaskes['KotaKab'] = datafaskes['KotaKab'].str.title()
datafaskes['TipeFaskes'] = datafaskes['TipeFaskes'].str.title()

# Buat encoding TipeFaskes
datafaskes['TipeFaskes_code'] = datafaskes['TipeFaskes'].astype('category').cat.codes
```

Gambar 5. Standarisasi Data Fasilitas Kesehatan BPJS 2019

Pada Gambar 6 berisikan data COVID-2019 dengan 21759 baris dan 38 kolom yaitu date, location iso code, location, new cases, new deaths, dll dari Data_COVID19_Indonesia.

```
dataCovid.head()
```

	Date	Location ISO Code	Location	New Cases	New Deaths	New Recovered	New Active Cases	Total Cases	Total Deaths	Total Recovered	Latitude	New Cases per Million	Total Cases per Million	New Deaths per Million	Total Deaths per Million	Total Recovered per 1000b	Case Fatality Rate	Recon
0	3/1/2020	ID-JK	DKI Jakarta	2	0	0	2	39	20	41	-6.204699	0.18	3.60	0.0	1.84	0.18	51.28%	100
1	3/2/2020	ID-JK	DKI Jakarta	2	0	0	2	41	20	41	-6.204699	0.18	3.78	0.0	1.84	0.18	48.78%	100
2	3/2/2020	IDN	Indonesia	2	0	0	2	2	0	0	-0.789275	0.01	0.01	0.0	0.00	0.00	0.00%	C
3	3/2/2020	ID-RI	Riau	1	0	0	1	2	0	3	-0.511648	0.16	0.33	0.0	0.00	0.00	0.00%	15C
4	3/3/2020	ID-JK	DKI Jakarta	2	0	0	2	43	20	41	-6.204699	0.18	3.96	0.0	1.84	0.18	46.51%	95

5 rows x 38 columns

Gambar 6. Data COVID-2019

Gambar 7 menunjukkan adanya *missing value* pada beberapa kolom, seperti City or Regency (21.759), Province, Island, dan Time Zone (masing-masing 642), Special Status (18.636), serta beberapa kolom lain seperti Total Cities, Urban Villages, Rural Villages, dan *growth factor* kasus maupun kematian baru.

```
dataCovid.isnull().sum()
```

Date	0
Location ISO Code	0
Location	0
New Cases	0
New Deaths	0
New Recovered	0
New Active Cases	0
Total Cases	0
Total Deaths	0
Total Recovered	0
Total Active Cases	0
Location Level	0
City or Regency	21750
Province	642
Country	0
Continent	0
Island	642
Time Zone	642
Special Status	18636
Total Regencies	0
Total Cities	614
Total Districts	0
Total Urban Villages	617
Total Rural Villages	642
Area (km2)	0
Population	0
Population Density	0
Longitude	0
Latitude	0
New Cases per Million	0
Total Cases per Million	0
New Deaths per Million	0
Total Deaths per Million	0
Total Deaths per 100rb	0
Case Fatality Rate	0
Case Recovered Rate	0
Growth Factor of New Cases	1187
Growth Factor of New Deaths	2467
dtype: int64	

Gambar 7. Missing Value Data COVID-2019

Penanganan missing value dilakukan dengan menghapus kolom City or Regency karena seluruh datanya kosong, serta mengisi kolom Special Status dengan 'bukan daerah khusus'. Kolom Province, Island, dan Time Zone diisi dengan 'belum terdaftar dan ditemukan'. Kolom Total Cities, Urban Villages, dan Rural Villages diubah ke tipe string dan missing value-nya diisi dengan keterangan serupa. Sementara itu, missing value pada kolom Growth Factor kasus dan kematian baru diisi dengan nilai 0 yang menandakan tidak ada pertumbuhan. Proses Penanganan missing value ditunjukkan pada Gambar 8.

```
# Drop kolom yang 100% missing
dataCovid.drop(columns=['City or Regency'], inplace=True)

# Isi missing value pada kolom 'Special Status' dengan "bukanDaerahKhusus"
dataCovid['Special Status'] = dataCovid['Special Status'].fillna("bukanDaerahKhusus")

# Daftar kolom wilayah administratif dengan missing value yang ingin diisi
wilayah_kolom = ['Province', 'Island', 'Time Zone']

# Isi semua nilai NaN dengan string "belum terdaftar dan ditemukan"
dataCovid[wilayah_kolom] = dataCovid[wilayah_kolom].fillna("belum terdaftar dan ditemukan")

# Ubah kolom numerik ke string (atau isi NaN)
cols_numerik_admin = ['Total Cities', 'Total Urban Villages', 'Total Rural Villages']

for col in cols_numerik_admin:
    dataCovid[col] = dataCovid[col].astype('object').fillna("belum terdaftar dan ditemukan")

# Imputasi growth factor dengan 0 (tidak ada pertumbuhan)
dataCovid['Growth Factor of New Cases'] = dataCovid['Growth Factor of New Cases'].fillna(0)
dataCovid['Growth Factor of New Deaths'] = dataCovid['Growth Factor of New Deaths'].fillna(0)
```

Gambar 8. Penanganan Missing Value

Gambar 9 menampilkan bahwa tidak ada lagi kolom yang memiliki *missing value* karena sudah ditangani menggunakan Imputasi, spesial status dan drop column.

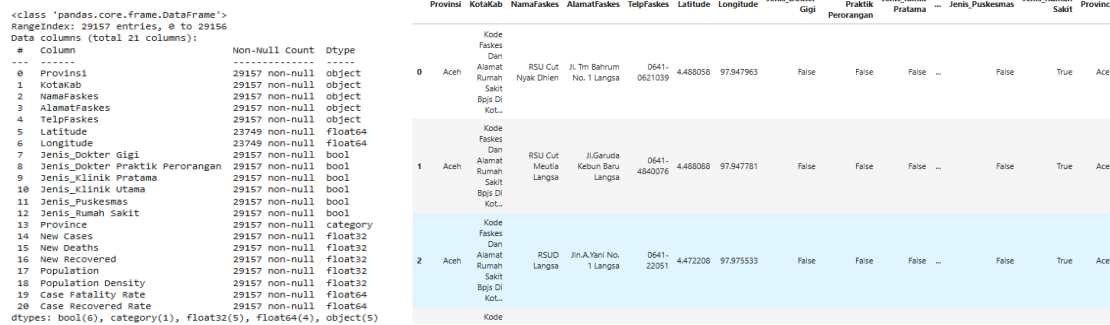
```
dataCovid.isnull().sum()
```

Date	0
Location ISO Code	0
Location	0
New Cases	0
New Deaths	0
New Recovered	0
New Active Cases	0
Total Cases	0
Total Deaths	0
Total Recovered	0
Total Active Cases	0
Location Level	0
Province	0
Country	0
Continent	0
Island	0
Time Zone	0
Special Status	0
Total Regencies	0
Total Cities	0
Total Districts	0
Total Urban Villages	0
Total Rural Villages	0
Area (km2)	0
Population	0
Population Density	0
Longitude	0
Latitude	0
New Cases per Million	0
Total Cases per Million	0
New Deaths per Million	0
Total Deaths per Million	0
Total Deaths per 100rb	0
Case Fatality Rate	0
Case Recovered Rate	0
Growth Factor of New Cases	0
Growth Factor of New Deaths	0
dtype: int64	

Gambar 9. Hasil Penangan Missing Value Data COVID-2019

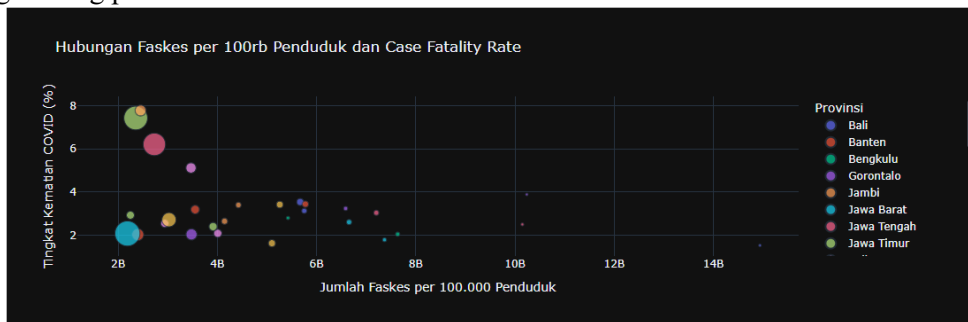
Gambar 10 setelah eksplorasi dan pembersihan data (seperti penanganan *missing value*, standarisasi, encoding, dan seleksi fitur), dilakukan integrasi data dengan menyiapkan data COVID-19. Dataset COVID-19 diurutkan berdasarkan tanggal, kemudian baris dengan nilai *Case*

Fatality Rate (CFR) yang kosong dihapus. Untuk menghindari duplikasi per provinsi, hanya baris terakhir (paling mutakhir) dari masing-masing provinsi yang diambil. Kolom CFR dikonversi ke tipe float untuk memudahkan perhitungan statistik. Setelah itu, data COVID-19 digabungkan (*merged*) dengan data fasilitas kesehatan menggunakan fungsi `merge()` berdasarkan nama provinsi.



Gambar 10. Integrasi Data Fasilitas Kesehatan BPJS 2019 dan Data COVID-2019

Gambar 11 merupakan visualisasi yang menampilkan scatter plot yang menggambarkan hubungan antara jumlah faskes per 100.000 penduduk dengan tingkat kematian COVID-19 untuk masing-masing provinsi di Indonesia.



Gambar 11. Visualisasi Hubungan Antara Jumlah Fasilitas Kesehatan Dengan Tingkat Kematian COVID-19

3.2. Penyimpanan Data ke MongoDB Menjalankan dan membuka MongoDB

```

C:\Users\User\mongodb>
{"t":{"date":"2025-05-30T17:32:24.907+07:00","s":"I","c":"NETWORK","id":"4915701","ctx":{"thread":"msg"},"msg":"Initialized wire specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":21},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":21},"outgoing":{"minWireVersion":6,"maxWireVersion":21},"isInternalClient":true}}},"t":{"date":"2025-05-30T17:32:24.913+07:00","s":"I","c":"CONTROL","id":"23285","ctx":{"thread":"msg"},"msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"},"t":{"date":"2025-05-30T17:32:28.646+07:00","s":"I","c":"NETWORK","id":"4648602","ctx":{"thread":"msg"},"msg":"Implicit TCP FastOpen in use."},"t":{"date":"2025-05-30T17:32:28.655+07:00","s":"I","c":"REPL","id":"5123008","ctx":{"thread":"msg"},"msg":"Successfully registered PrimaryOnlyService","attr":{"service":"tenantMigrationDonorService","namespace":"config.tenantMigrationDonors"},"t":{"date":"2025-05-30T17:32:28.656+07:00","s":"I","c":"REPL","id":"5123008","ctx":{"thread":"msg"},"msg":"Successfully registered PrimaryOnlyService","attr":{"service":"tenantMigrationRecipientService","namespace":"config.tenantMigrationRecipients"},"t":{"date":"2025-05-30T17:32:28.657+07:00","s":"I","c":"CONTROL","id":"5845609","ctx":{"thread":"msg"},"msg":"Multi threading initialized"},"t":{"date":"2025-05-30T17:32:28.658+07:00","s":"I","c":"TENANT_M","id":"7091500","ctx":{"thread":"msg"},"msg":"Starting TenantMigrationAccessBlockerRegistry"},"t":{"date":"2025-05-30T17:32:28.664+07:00","s":"I","c":"CONTROL","id":"4615611","ctx":{"initandlisten":"msg"},"msg":"MongoDB starting","attr":{"pid":11396,"port":27017,"dbPath":"C:/data/db","architecture":"64-bit","host":"LAPTOP-VEGAWANI"},"t":{"date":"2025-05-30T17:32:28.664+07:00","s":"I","c":"CONTROL","id":"23398","ctx":{"initandlisten":"msg"},"msg":"Target operating system minimum version","attr":{"targetVersion":"Windows 7/Windows Server 2008 R2"},"t":{"date":"2025-05-30T17:32:28.664+07:00","s":"I","c":"CONTROL","id":"23403","ctx":{"initandlisten":"msg"},"msg":"Build info","attr":{"buildInfo":{"version":"7.0.4","gitVersion":"30f3a7057a3d2e9f4a39142681a706c2d582e","modules":[],"allocator":"tcmalloc","environment":{"distmod":"windows","distarch":"x86_64","target_arch":"x86_64"}}},"t":{"date":"2025-05-30T17:32:28.665+07:00","s":"I","c":"CONTROL","id":"51705","ctx":{"initandlisten":"msg"},"msg":"Operating system","attr":{"os":{"name":"Microsoft Windows","version":"10.0 (build 22000)}}},"t":{"date":"2025-05-30T17:32:28.666+07:00","s":"I","c":"CONTROL","id":"21951","ctx":{"initandlisten":"msg"},"msg":"Options set by command line","attr":{"options":{}}},"t":{"date":"2025-05-30T17:32:28.671+07:00","s":"E","c":"CONTROL","id":"20957","ctx":{"initandlisten":"msg"},"msg":"DBException in initandlisten, terminating","attr":{"error":{"message":"Data directory C:\\data\\db\\ not found. Create the missing directory or specify another path using (1) the --dbpath command line option, or (2) by adding the 'storage.dbPath' option in the configuration file."},"t":{"date":"2025-05-30T17:32:28.671+07:00","s":"I","c":"REPL","id":"4784900","ctx":{"initandlisten":"msg"},"msg":"Stepping down the ReplicationCoordinator for shutdown","attr":{"waitTimeMillis":15000}}
  
```

Gambar 12. Menjalankan Mongod

```

C:\Users\User>mongosh
Current Mongosh Log ID: 68398943d308a1f97fe9e1
Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.0
Using MongoDB: 6.0.6
Using Mongosh: 2.1.0
Mongosh 2.5.1 is available for download: https://www.mongodb.com/try/download/shell
For mongosh info see: https://docs.mongodb.com/mongodb-shell/

-----
The server generated these startup warnings when booting
2025-05-30T17:03:42.693+07:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----
test>

```

Gambar 13. Menjalankan Mongosh

Membuat Database dan Collection

```

test> use Proyek_db
switched to db Proyek_db
Proyek_db> db.createCollection("tweets")
{ ok: 1 }
Proyek_db>

```

Gambar 14. Membuat Database dan Collection

Import Dataset

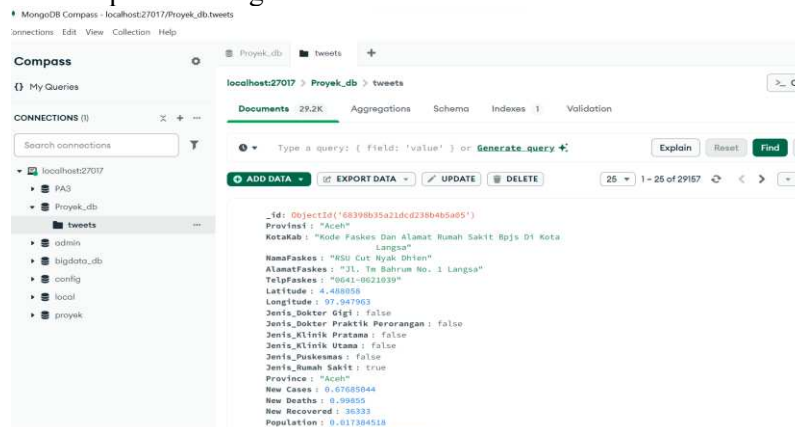
```

C:\Users\User\Big Data\dataSciencePaper>mongoimport --db Proyek_db --collection tweets --file df_finalgabungandatafaskesdandatacovid.json
2025-05-30T17:40:52.898+0700 connected to: mongodb://localhost/
2025-05-30T17:40:55.904+0700 [#####] Proyek_db.tweets 12.4MB/18.0MB (68.6%)
2025-05-30T17:40:57.106+0700 [#####] Proyek_db.tweets 18.0MB/18.0MB (100.0%)
2025-05-30T17:40:57.107+0700 29157 document(s) imported successfully. 0 document(s) failed to import.
C:\Users\User\Big Data\dataSciencePaper>

```

Gambar 15. Import Dataset

Berhasil disimpan ke MongoDB



Gambar 16. Berhasil Menyimpan Data ke MongoDB

3.3. Pemodelan Machine Learning

Untuk membangun model *machine learning*, langkah awal yang dilakukan adalah membagi data hasil integrasi antara data fasilitas kesehatan dan data COVID-19. Pembagian ini bertujuan untuk memisahkan data menjadi dua bagian, yakni data pelatihan (X_{train} , y_{train}) dan data pengujian (X_{test} , y_{test}), dengan rasio 80:20. Artinya, 80% dari total data digunakan untuk melatih model, sementara 20% sisanya digunakan untuk menguji performa model. Model yang digunakan adalah regresi linier, yang diimplementasikan menggunakan kelas `LinearRegression()` dari pustaka `scikit-learn`. Model ini dilatih menggunakan data pelatihan melalui metode `fit()`, yang bertugas mempelajari hubungan antara variabel independen (X_{train}) dan variabel target (y_{train}). Dalam konteks penelitian ini, variabel target yang diprediksi adalah jumlah kematian

baru akibat COVID-19 (New Deaths), sedangkan variabel input (independen) mencakup jumlah kasus baru (New Cases), jenis rumah sakit (Jenis_Rumah Sakit), jumlah pasien sembuh (New Recovered), jumlah penduduk (Population), kepadatan penduduk (Population Density), tingkat kematian kasus (Case Fatality Rate), dan tingkat kesembuhan kasus (Case Recovered Rate). Tujuannya adalah agar model mampu memprediksi nilai target secara akurat berdasarkan input data. Setelah proses pelatihan selesai, dilakukan prediksi terhadap data pengujian (X_{test}) menggunakan model yang telah dilatih. Hasil prediksi tersebut kemudian dievaluasi menggunakan dua metrik: Mean Absolute Error (MAE) dan R-squared (R^2 Score). MAE memberikan gambaran tentang rata-rata selisih absolut antara nilai aktual (y_{test}) dan hasil prediksi (y_{pred}), sedangkan R^2 Score menunjukkan seberapa besar proporsi variasi data target yang berhasil dijelaskan oleh model.

MAE: 0.0765156765956073

R2 Score: 0.9994345059718258

Gambar 17. Hasil Perhitungan Dua Metrik Evaluasi Dengan MAE dan R2 Score

```
# Contoh prediksi data baru (ingat harus diskalakan juga)
data_baru = pd.DataFrame([
    'New_Cases': 1000,
    'Jenis_Rumah_Sakit': 62,
    'New_Recovered': 800,
    'Population': 200000,
    'Population_Density': 1500,
    'Case_Fatality_Rate': 0.02,
    'Case_Recovered_Rate': 0.95
])

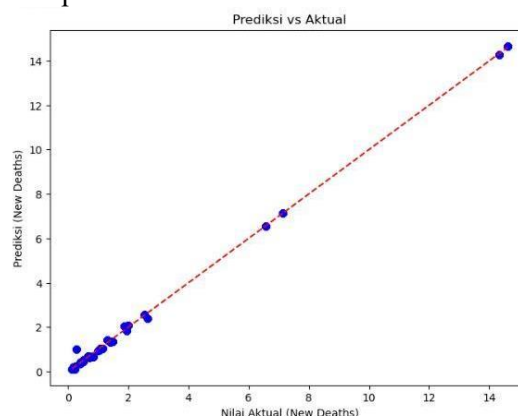
# Scaling data baru (numerik)
data_baru_scaled_num = scaler.transform(data_baru[num_features])

# Gabungkan dengan fitur kategori
data_baru_final = np.hstack([data_baru_scaled_num, data_baru[['Jenis_Rumah_Sakit']].values])

# Prediksi
prediksi = model.predict(data_baru_final)
print(f"Prediksi jumlah kematian baru untuk data input: {prediksi[0]:.2f}")
Prediksi jumlah kematian baru untuk data input: 972834.44
```

Gambar 18. Prediksi dengan Data Baru

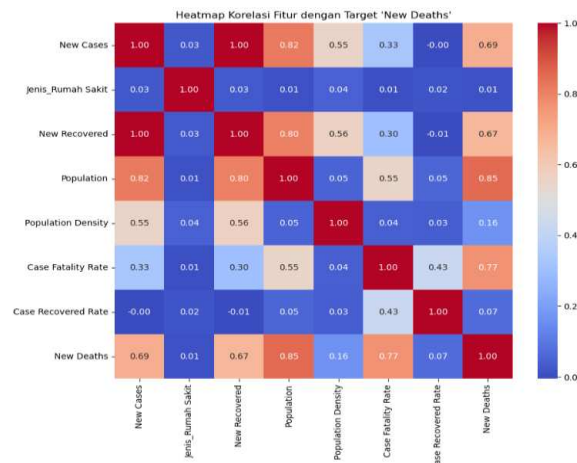
Gambar 19 ini menampilkan plot scatter antara nilai prediksi (Prediksi (New Deaths)) dan nilai aktual (Nilai Aktual (New Deaths)) dari suatu data. Garis diagonal merah menunjukkan kondisi ideal di mana nilai prediksi sama persis dengan nilai aktual. Sebagian besar titik-titik scatter berada di sekitar garis diagonal, mengindikasikan bahwa nilai prediksi cukup dekat dengan nilai aktual. Namun, terdapat beberapa titik yang menyimpang dari garis diagonal, menunjukkan adanya perbedaan antara nilai prediksi dan nilai aktual.



Gambar 19. Visualisasi Nilai Prediksi dan Nilai Aktual

Gambar 20 menunjukkan *heatmap* korelasi antar fitur seperti New Cases, Population, Case Fatality Rate, hingga New Deaths. Warna merah menandakan korelasi positif, biru negatif,

dengan intensitas warna menunjukkan kekuatan korelasi. Fitur New Cases dan Population memiliki korelasi positif terkuat terhadap New Deaths, sedangkan Case Recovered Rate menunjukkan korelasi negatif, menandakan semakin tinggi tingkat kesembuhan, semakin rendah angka kematian baru.



Gambar 20. Visualisasi Nilai Korelasi Antara Fitur

4. KESIMPULAN

Penelitian ini bertujuan untuk menganalisis pengaruh ketersediaan fasilitas kesehatan terhadap tingkat kematian akibat COVID-19 di Indonesia menggunakan pendekatan data science dengan MongoDB dan model machine learning. Berdasarkan tahapan yang telah dilakukan, berikut kesimpulan utama yang dapat diambil:

1. Terdapat hubungan negatif yang signifikan antara ketersediaan fasilitas kesehatan per 100.000 penduduk dengan angka kematian baru akibat COVID-19. Hasil analisis korelasi Pearson menunjukkan nilai korelasi sebesar -0.52 ($p\text{-value} = 0.004$), mengindikasikan bahwa provinsi dengan ketersediaan fasilitas kesehatan yang lebih baik cenderung memiliki angka kematian yang lebih rendah. Hal ini diperkuat oleh hasil regresi linier pada gambar 20, di mana variabel ini memberikan kontribusi negatif terhadap prediksi variabel target New Deaths.
2. Fitur "New Cases" dan "Population" merupakan prediktor paling kuat terhadap jumlah kematian baru. Berdasarkan heatmap korelasi, kedua fitur ini memiliki korelasi positif yang signifikan terhadap "New Deaths", menandakan bahwa lonjakan kasus dan populasi tinggi meningkatkan beban sistem kesehatan yang pada akhirnya berdampak pada angka kematian.
3. Model regresi linear memberikan interpretasi yang cukup baik. Meskipun model ini sederhana, hasil evaluasi dengan MAE dan R-squared menunjukkan performa yang cukup akurat dalam memprediksi jumlah kematian baru berdasarkan data numerik.
4. MongoDB terbukti efektif dalam menyimpan dan mengelola data besar dari berbagai sumber. Keunggulan dalam fleksibilitas skema dan performa real-time menjadikan MongoDB pilihan tepat untuk menyimpan data fasilitas kesehatan dan data COVID-19 yang heterogen.
5. Integrasi data dari berbagai sumber menghasilkan dataset yang bersih dan terstruktur. Proses preprocessing dan penyatuan berdasarkan wilayah administratif berhasil menciptakan dataset yang siap untuk dianalisis lebih lanjut, serta memperkuat kualitas insight dan prediksi dalam memahami penyebaran dan dampak pandemi.

5. SARAN

Penggunaan MongoDB dalam penelitian ini terbukti efektif dalam penyimpanan data yang fleksibel dan efisien, namun masih memiliki tantangan tersendiri bagi pengguna yang belum familiar dengan basis data NoSQL. Untuk itu, pengembangan ke depan dapat diarahkan pada pemanfaatan layanan cloud seperti MongoDB Atlas atau Google Cloud Firestore, yang menawarkan kemudahan dalam pengelolaan dan skalabilitas data secara otomatis, sehingga mendukung integrasi data berskala nasional dengan lebih praktis. Di sisi lain, guna meningkatkan ketepatan prediksi serta pemahaman terhadap hubungan antar variabel yang kompleks, disarankan untuk mengintegrasikan model pembelajaran mesin yang lebih canggih seperti Random Forest, XGBoost, atau Deep Neural Networks. Pendekatan ini berpotensi memberikan analisis yang lebih mendalam terhadap data fasilitas kesehatan dan epidemiologi, sekaligus mendukung pengambilan keputusan berbasis data yang lebih akurat di bidang kesehatan.

UCAPAN TERIMA KASIH

Kami mengucapkan terima kasih yang sebesar-besarnya kepada Institut Teknologi Del sebagai institusi tempat kami menempuh pendidikan, yang telah memberikan dukungan berupa fasilitas, akses literatur, serta lingkungan belajar yang kondusif selama proses pelaksanaan penelitian ini. Kami juga menghargai semua pihak yang secara tidak langsung telah memberikan inspirasi, motivasi, dan semangat, sehingga penelitian ini dapat diselesaikan dengan baik.

DAFTAR PUSTAKA

- [1] M. J. Adebowale, "International Journal of Research Publication and Reviews Integrating Database Management Systems with Predictive Analytics for Enhanced Business Decision-Making," vol. 5, no. 12, pp. 1415–1429, 2024.
- [2] B. A. B. Iv dan D. Kematian, "Bab IV: Determinan Kematian COVID-19," *Jurnal Kesehatan Masyarakat*, vol. 2, no. 1, hlm. 73–99, 2022.
- [3] S. Kasus dan K. Kota, "Determinants of COVID-19 Mortality Risk," *Jurnal Epidemiologi Indonesia*, vol. 19, no. 2, hlm. 977–984, 2021.
- [4] S. Wibisono, W. Hadikurniawati, I. Husni, and A. Almin, "AHP-COPRAS untuk Pemingkatan Ketersediaan Fasilitas Kesehatan di Indonesia," vol. 08, 2023.
- [5] H. Abbes and F. Gargouri, "Big Data Integration : a MongoDB Database and Modular Ontologies based Approach," *Procedia - Procedia Comput. Sci.*, vol. 96, no. September, pp. 446–455, 2016, doi: 10.1016/j.procs.2016.08.099.
- [6] A. J. Syahwali, "Pemanfaatan MongoDB dalam Sistem Informasi Akademik untuk Pengelolaan Data Mahasiswa Pada Universitas Bina Darma," vol. 3, no. 2, pp. 466–469, 2025.
- [7] R. Ramakrishnan, K. Krishnakumar, dan B. Keerthika, "Enhancing Data Management with MongoDB and its REST API," *IJRASET Journal for Research in Applied Science & Engineering Technology*, vol. 10, no. 7, Jul. 2023, doi:10.22214/ijraset.2023.54593.
- [8] S. Anurag and A. Saxena, "Leveraging MongoDB for Efficient Storage of MIMIC-IV CXR X-ray Images : A Research Perspective," vol. 10, 2025.
- [9] R. N. Putri, "Indonesia dalam Menghadapi Pandemi COVID-19," *Jurnal Ilmiah Universitas Batanghari Jambi*, vol. 20, no. 2, hlm. 705–709, 2020, doi:10.33087/jiubj.v20i2.1010.
- [10] H. Harlan, I. Sri, and S. Wiji, "Analisis Korelasi Faktor Resiko Kematian Di Ruang Isolasi Covid-19," vol. 3, no. 1, pp. 51–60, 2021.

- [11] A. Maulana and K. Arcana, "Prediksi Tingkat Kematian COVID-19 Berdasarkan Faktor Klinis dan Demografis di Indonesia," *Jurnal Kesehatan Masyarakat Nasional*, vol. 17, no. 2, pp. 98–106, 2022.
- [12] F. Ramadhani, T. Setiawan, and D. Wulandari, "Distribusi Spasial Fasilitas Kesehatan dan Mortalitas COVID-19: Studi Kasus di Jawa Tengah," *Jurnal Geospasial dan Kesehatan*, vol. 5, no. 3, pp. 205–213, 2022.
- [13] R. A. Putri and B. Hidayat, "Pemanfaatan MongoDB dalam Sistem Prediksi Ketersediaan Tempat Tidur Rumah Sakit saat Pandemi COVID-19," *Jurnal Teknologi Informasi dan Sistem Komputer*, vol. 10, no. 1, pp. 33–40, 2021.
- [14] S. Nugroho and L. Amalia, "Pemodelan Statistik Faktor-Faktor yang Mempengaruhi Case Fatality Rate COVID-19 di Indonesia," *Jurnal Statistika Terapan Indonesia*, vol. 8, no. 2, pp. 45–54, 2023.
- [15] M. L. Dewi and R. Kurniawan, "Integrasi Data Spasial dan Machine Learning untuk Prediksi Risiko Kematian COVID-19," *Jurnal Sains Data dan Analitika*, vol. 2, no. 4, pp. 180–190, 2023.
- [16] M. F. Rahmi, P. S. Prasetyo, R. N. Madjida, et al., "Pengelompokan Provinsi di Indonesia berdasarkan Jumlah Kasus COVID-19 dan Fasilitas Kesehatan," *Jurnal Aplikasi Statistika & Komputasi Statistik*, vol. 13, no. 1, 2023