

Tugas Akhir Data Management:

Analisa *COVID-19: Continents In Relation to Time*

Dengan Menggunakan PyMongoDB

Asal Data:

<https://www.kaggle.com/code/michau96/covid-19-continents-in-relation-to-time/input>

Nama : Faiz Noor Adhytia
NIM : 22K30006
Language : PyMongo
Source : [Dataset Covid-19](#)

I. Importing Module

```
# Dataframe Operation
import pandas as pd

# Database Operation
import pymongo
import json
```

Modul yang digunakan adalah pandas, pymongo, dan json.

Pandas berfungsi untuk pembentukan dataframe untuk pengolahan data dan salah satu kegunaan pada kasus ini adalah untuk importing dataset csv.

Pymongo berfungsi sebagai penghubung antara python dengan mongodb, dalam kasus ini dapat terlihat saat connecting pada server

Json tidak berfungsi secara langsung melainkan berfungsi Ketika melakukan record ke dalam database. Data yang diberikan adalah bertipe csv yang kemudian *diinput* dengan membentuk sebuah dataframe oleh pandas, setelah itu baru json bekerja untuk merecord dataframe ke dalam database.

II. Connecting to MongoDB and Dataset

```
client = pymongo.MongoClient("mongodb://localhost:27017")
Report = pd.read_csv("covid19_data.csv")
Report.head()
data = Report.to_dict(orient="records")
data
Report.dtypes
Report.dtypes
```

Variable client berfungsi untuk menyambungkan antara python dengan mongodb dengan menggunakan localhost mongodb

Variable report berfungsi untuk menampung value dari function pd.read_csv() untuk pembacaan file csv dan diubah ke dalam dataframe.

III. Create Data

```
db = client["dbCovid19"]
collection = db["Covid19_report"]
collection.insert_many(data)
```

db untuk variable penampung pembentukan database

collection untuk variable menampung pembentukan collection (table)

IV. Read Data

```
print("Reading data sorted by Confirmed cases in ascending order:")
for doc in collection.find(
    {},
    {
        "_id": 0,
        "Province": 1,
        "Last Update": 1,
        "Confirmed": 1,
        "Deaths": 1,
        "Recovered": 1,
    },
).sort("Confirmed", pymongo.ASCENDING):
    print(doc)
```

Pembacaan data dilakukan secara ASCENDING. Pembacaan data dilakukan secara menaik atau dibaca dari terendah hingga tertinggi, hal ini untuk mengurutkan kasus terkonfirmasi dari terendah hingga terparah pada suatu kasus. Dalam *code* di atas **1 = true | 0 = false**, hal ini berarti untuk setiap kolom bernilai **1 maka akan tampil dan untuk bernilai 0 maka tidak tampil**. Doc dalam hal ini adalah mengembalikan *single document* dari database atau doc merupakan document yang tercantum dalam record. Dalam *collection.find()* nilai baliknya adalah sebuah *cursor* untuk mengakses dokumen yang sesuai.

V. Update Data

```
print("Documents with Confirmed value of 20:")
for doc in collection.find(
    {"Province": "Shanghai", "Confirmed": 20.0},
    {
        "_id": 0,
        "Province": 1,
        "Last Update": 1,
        "Confirmed": 1,
        "Deaths": 1,
        "Recovered": 1,
    },
).sort("Confirmed", pymongo.ASCENDING):
    print(doc)

collection.update_one(
    {"Province": "Shanghai", "Confirmed": 20.0}, {"$set": {"Confirmed": 22.0}}
)

collection.update_one(
    {"Province": "Shanghai", "Last Update": "1/24/20 17:00", "Confirmed": 22.0},
    {"$set": {"Last Update": "1/24/2020 17:00"}},
)
```

Update data untuk Provinsi: Shanghai | Confirmed: 20. Update yang akan dilakukan adalah memperbaharui data terkonfirmasi berjumlah 20 pada provinsi Shanghai menjadi 22 karena adanya kesalahan input.

Selain itu juga dilakukan pembaharuan pada *Last Update* pada tanggal 1/24/20 menjadi 1/24/2020,

VI. Delete Data

```
print("Documents with Confirmed value of 22:")
for doc in collection.find(
    {"Province": "Shanghai", "Confirmed": 22.0},
    {
        "_id": 0,
        "Province": 1,
        "Last Update": 1,
        "Confirmed": 1,
        "Deaths": 1,
        "Recovered": 1,
    },
).sort("Confirmed", pymongo.ASCENDING):
    print(doc)
result = collection.delete_one(
    {"Province": "Shanghai", "Last Update": "1/22/2020 17:00", "Confirmed": 22.0}
)

# Check if a document was deleted and print a success message
if result.deleted_count > 0:
    print("Deleted successfully!")
else:
    print("No document found to delete.")
```

Agar tidak terjadi penggandaan data terkonfirmasi, maka data yang tidak tepat harus dihapus, dalam hal ini data pada tanggal 1/22/2020 17:00, dihapus.